Exploiting Document Level Semantics in Document Clustering

Muhammad Rafi*, Muhammad Naveed Sharif†, Waleed Arshad‡,
Habibullah Rafay§, Sheharyar Mohsin¶ and Dr. Mohammad Shahid Shaikh‖

*Department of Computer Science
FAST NUCES, Karachi, Pakistan

‖Faculty of Electrical Engineering
Habib University, Karachi, Pakistan

***Abstract*—Document clustering is an unsupervised machine learning method that separates a large subject heterogeneous collection (Corpus) into smaller, more manageable, subject homogeneous collections (clusters). Traditional method of document clustering works around extracting textual features like: terms, sequences, and phrases from documents. These features are independent of each other and do not cater meaning behind these word in the clustering process. In order to perform semantic viable clustering, we believe that the problem of document clustering has two main components: (1) to represent the document in such a form that it inherently captures semantics of the text. This may also help to reduce dimensionality of the document and (2) to define a similarity measure based on the lexical, syntactic and semantic features such that it assigns higher numerical values to document pairs which have higher syntactic and semantic relationship. In this paper, we propose a representation of document by extracting three different types of features from a given document. These are lexical α, syntactic β and semantic γ features. A meta-descriptor for each document is proposed using these three features: first lexical, then syntactic and in the last semantic. A document to document similarity matrix is produced where each entry of this matrix contains a three value vector for each lexical α, syntactic β and semantic γ. The main contributions from this research are (i) A document level descriptor using three different features for text like: lexical, syntactic and semantics. (ii) we propose a similarity function using these three, and (iii) we define a new candidate clustering algorithm using three component of similarity measure to guide the clustering process in a direction that produce more semantic rich clusters. We performed an extensive series of experiments on standard text mining data sets with external clustering evaluations like: F-Measure and Purity, and have obtained encouraging results.**

***Keywords*—Document Clustering; Text Mining; Similarity Measure; Semantics**

I. Introduction

Document clustering [9] can be defined as an unsupervised learning approach, which clusters the document repository into meaningful smaller and manageable sub-collections. These resultant sub-collections contain high intra-cluster similarity (that is, the documents in a single cluster are mainly similar in some sense), and low inter-cluster similarity (documents in two sub-collections are largely dissimilar). It has found its niche in management of large document repositories. Learning the common features for grouping implicitly is the main spirit of document clustering. Traditionally, document clustering algorithms utilized simple features present in the documents like (word, phrases and sequence of words) to cluster the documents. These simple features are independent of document context and thus the semantic of the document cannot be incorporated into the clustering process. In order to perform semantic viable clustering, we believe that the problem of document clustering has two main aspects: (1) to represent the document in such a form that it inherently captures semantics of the text. This may also help to reduce dimensionality of the document . Other, (2) to define a similarity measure based on the semantic representation such that it assigns higher numerical values to document pairs which have higher semantic relationship. In general, the degree of similarity between documents is measured by the words and sentences or meaning of the document. A similarity measure should also address the problem of partial matching of documents. Several efforts have been made to address the problem of document partial matching, using the lexical features from a document like: keywords, concepts, nouns, verb etc. Documents are modeled in such a way that it allows the similarity methods to compute partial contribution of these individual units in the similarity values. The overall document similarity is obtained as a function of those partial measures. We have observed that while doing this partial document matching towards similarity function, there are two problems that have not been handled in previous works (i) Word Order problem i.e, In Human spoken languages, the selection of words is mainly base on the contextual information being contained in the document. Similarly, the order of the words appearing in the text influences the meaning of the text. For example, the sentences “A hires B” and “B hires A” are composed by selecting the same words, but the order completely changes their meaning and (ii) Semantic matching i.e, Sentences with the same meaning but different words. For example, the sentences ”Joe is an intelligent boy” and ”Joe

is a smart lad" have similar meaning, if the context in which they appear does not change much.

According to the work in [5], the paper proposed a three-layer representation of documents. Unlike sentences, we apply these three layer transformation on an entire document. These layers are lexical, syntactic and semantic layers. Each layer extracts specific features from the same document. The lexical analysis is performed in the first layer in which we extract the bag of word vectors from the documents. Documents are preprocessed by removing stop words and stemming. The syntactic layer uses relations (predicates) extracted from the RDFs of the documents to handle word order problem. The RDFs of the documents are generated using an online tool Alchemy API [13]. The semantic layer employs the Semantic Role Annotation (SRA) to handle the semantics problem. The SRA analysis is done using Fred API [11] which returns the meaning of the actions, the actor who performs the action, and the object/actor on which the action is being performed. FRED is a tool for automatically producing RDF/OWL ontology's and linked data from natural language sentences. The method is based on combinatorial Categorical Grammar, Discourse Representation Theory, Linguistic Frames, and Ontology Design Patterns. Results are enriched with Named Entity Resolution (NER) and Word-Sense Disambiguation (WSD). Below is an example based on a sample document; the three representations are presented next to each other.
Below is an example of a simple sentence;

D1 = "Pakistani boys love to play cricket and hockey"

Lexical Feature:
{Pakistani, Boy, Love, Cricket, Hockey}

Syntactic Feature:

```
<relation>
    <sentence>Pakistani boys love cricket and hockey.</sentence>
    <subject>
      <text>Pakistani boys</text>
    </subject>
    <action>
      <text>love</text>
      <lemmatized>love</lemmatized>
      <verb>
        <text>love</text>
        <tense>present</tense>
      </verb>
    </action>
    <object>
      <text>cricket and hockey</text>
    </object>
</relation>
```

Semantic Feature:

Activity	{Hockey, Cricket}
Boy	{PakistaniBoy}
Event	{Love}
Experiencer_focus	{Love}
supersense-noun_act	{Hockey, Cricket}

Fig. 1. Three layer representation of a sample document

Below is an example document from NEWS20 data set, the meta-descriptor clearly contains three types of features just after another. Here is an example from document 103122;

D2 = "Most people who go fast wear goggles. So do most of helmetless motorcyclists"

Lexical Feature:
{Most, People, Who, Go, Fast, Wear, Goggles, So, Do, Helmetless, Motorcyclist}

Syntactic Feature:

```
<relation>
    <sentence> Most people who go fast wear goggles.</sentence>
    <subject>
      <text>Most people</text>
    </subject>
    <action>
      <text>go</text>
      <verb>
        <text>go</text>
        <tense>present</tense>
      </verb>
    </action>
    <object>
      <text>fast wear goggles</text>
    </object>
</relation>
```

```
<relation>
    <sentence> So do most of helmetless motorcyclists.</sentence>
    <subject>
      <text>So</text>
    </subject>
    <action>
      <text>do</text>
      <verb>
        <text>do</text>
        <tense>present</tense>
      </verb>
    </action>
    <object>
      <text>most helmetless motorcyclists.</text>
    </object>
</relation>
```

Semantic Feature:

Event	{Do, Go}	Quality	{Most}
PhysicalObject	{Goggles, Wear}	Motorcyclist	{HelmetlessMotorcyclist}

Fig. 2. Three layer representation of an example document 103122

We believe that the major confusion in clustering process is just because we consider lexical features alone. The syntactic and semantic features may guide us on the right decision to merge two documents or not, hence a good clustering arrangement is learned. We have carried out an extensive set of experiments with standard text mining data sets. Our proposed approach clearly surpasses the traditional document clustering methods on evaluation like: F-Score and Purity. The paper organized as follows: Section 2 discusses the related work in area of text document clustering, specifically in the realm of semantic based document clustering. Section 3 describes our proposed approach along with some examples. Section 4 presents the experimental setup, data sets, comparative algorithms, and evaluation measures. Section 5 discusses the experimental results. Conclusion is presented in section 6.

II. The Literature Review

Data clustering [9] is an unsupervised technique which creates succinct sub-groups from the data for discovering valuable knowledge. Document clustering is a specialized data clustering problem where the objects are in the form of documents. The objective of the clustering process is to group the similar documents and separate different ones. The difficult part of this unsupervised task is to learn how many clusters of such groups exist in a given data set. Document Clustering aims to discover natural grouping among documents in such a way that documents within a cluster are similar (high intra-cluster similarity) to one another and are dissimilar to documents in other clusters (low inter cluster similarity). Exploring, analyzing, and correctly classifying the unknown natures of data in a document without supervision is the major requirement of document clustering method. Clustering is an effective method for search computing [1]. It offers the possibilities like:

grouping similar results [3], comprehending the links between the results [8] and creating the succinct representation and display of search results[3,4]. Document clustering has three main steps: (i) document representation model, (ii) similarity measure between a pair of documents in selected form of representation and (iii) clustering algorithm that produce the final clustering arrangement. Document representation is very sensitive for the task of document clustering. Traditionally, document clustering algorithms mainly use features like: words [7], phrases [2], and sequences [6, 10] from the documents to perform clustering. These algorithms generally apply simple feature extraction techniques that are mainly based on feature counting frequency distribution of the features. The approach in [6] proposed a frequent itemset-based representation of documents for clustering (FIHC). Motivated from the idea of market basket analysis, the authors considered a document as basket and the terms used in the document are considered as itemsets present in it. A representation based on frequent items (frequent phrases) is proposed. The work in [10] proposed two solutions to document representations (i) frequent word sequences (CFWS) and (ii) frequent word meaning sequences (CFWMS). The two approaches first parses a given document to get the frequent word sequences of some arbitrary length (2-word set for their experiment). The first uses the frequent word sequences and the second uses an external lexical database WordNet [4] to annotate the word with their meaning to cover the word meaning problem, such as synonymy, polysemy, and hyponymy/hypernymy. Their experimental studies have shown an improvement in F-Measure for both CFWS and CFWMS over FIHC. Although these approaches use phrases or order of words in representation of documents, their results are still fallible on semantics of the clusters produced. These techniques simply perform clustering independent of the context. Document written in human language contains a context and words that are largely depending on it. A more recent approach to represent a document is based on dependency graph (DGDC), proposed in [14]; each document is parsed to form a dependency graph. This dependency graph captures the semantic representation of documents; thus, it offers more semantic rich clustering. It also introduced a novel similarity measure based on common features of the two corresponding graphs of the documents. One more recent approach to capture semantic representation of documents in document representation model is introduced in [12] in which the authors proposed a topic maps based representation by using an online tool Wandora for extracting topics from a document. They also reported encouraging results for document clustering based on semantic notions. We conclude that there are features like frequent item sets, common frequent sequences or word meaning sequences, dependency graphs, and topic maps that can be used to reduce the dimensionality of document space and at the same time offer more semantics in representations over simple Bag-of-words (BOW); These approaches still fail to incorporate semantics on larger scale. The phrases or sequence are a good measure for identifying semantics of the text. We believe that a sentence specific measure will be more semantic rich, and extending a sentence level similarity to a complete document is a challenging aspect of semantic oriented document clustering. A sentence similarity can easily capture similarity between phrases and sequences, but this similarity should also address the issue of partial information like: when one sentence splits into two or more short texts and phrases that contain two or more sentences, it should assign partial score to matched phrases or sequences. The score should directly proportionate to a number of such units found in the two sentences. In [5] authors describe a sentence similarity measure that uses three-layer of sentences meta descriptor to capture the semantic in the similarity measure. We have been motivated by this idea and extended it to a full document for eventually performing the task of clustering. A document is transformed into three meta-representations based on lexical, syntactic and semantic layers. A similarity measure for each representation is defined based on features extracted from each layer. Cosine similarity is used for each pair of documents in all the three layers; hence, we get three similarity values in each of the three layers, that is lexical, syntactic and semantic meta-descriptor. Final document clustering is performed on N x N matrix(containing the vectors$< \alpha, \beta, \gamma >$ from three layers) by candidate based document clustering algorithm. We have conducted an extensive set of experiments with standard text mining data sets. Our proposed approach clearly surpasses the traditional document clustering methods on evaluation like: F-Score and Purity.

III. EXPLOITING DOCUMENT LEVEL SEMANTICS IN DOCUMENT CLUSTERING

A. Document Representation

In this paper, we propose a representation of each document based on three levels, namely lexical, syntactic and semantic levels. Each level produces a separate document-to-document similarity score. We generate a vector of similarity scores based on these three levels i.e.

$$Vector(D_a \leftarrow D_b) = Sim < \alpha, \beta, \gamma > \quad (1)$$

Where $\alpha,\ \beta\ and\ \gamma$ are the similarity scores of lexical, syntactic and semantic level from Document a to Document b.

1) Extraction of lexical features: Bag of Words are extracted as the features of the document. From all the documents in the set, a dictionary of words (vector) is built. For each document a vector is built which contains the common tokens between the document and the dictionary. The vector contains values formed by calculating $TF * IDF$ for each token. $TF * IDF$ is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
Stop word removal rules out words with little representative value to the document, e.g. articles and pronouns, and the punctuation.
Stemming is a pre-processing service, which translates the tokens in its basic form. For instance, plural words are made singular and all verb tenses and persons are exchanged by the verb infinitive.

2) Extraction of syntactic features: Alchemy API has been used for all the documents to extract the Subject, Action and Object of each sentence in every document. The syntactic analysis represents an order relation among the extracted features of the documents. It describes the syntactic structures of the language; and decomposes the text into syntactic units in order to know the arrangements of syntactic elements. Such kind of relations could be used in applications, as for instance,

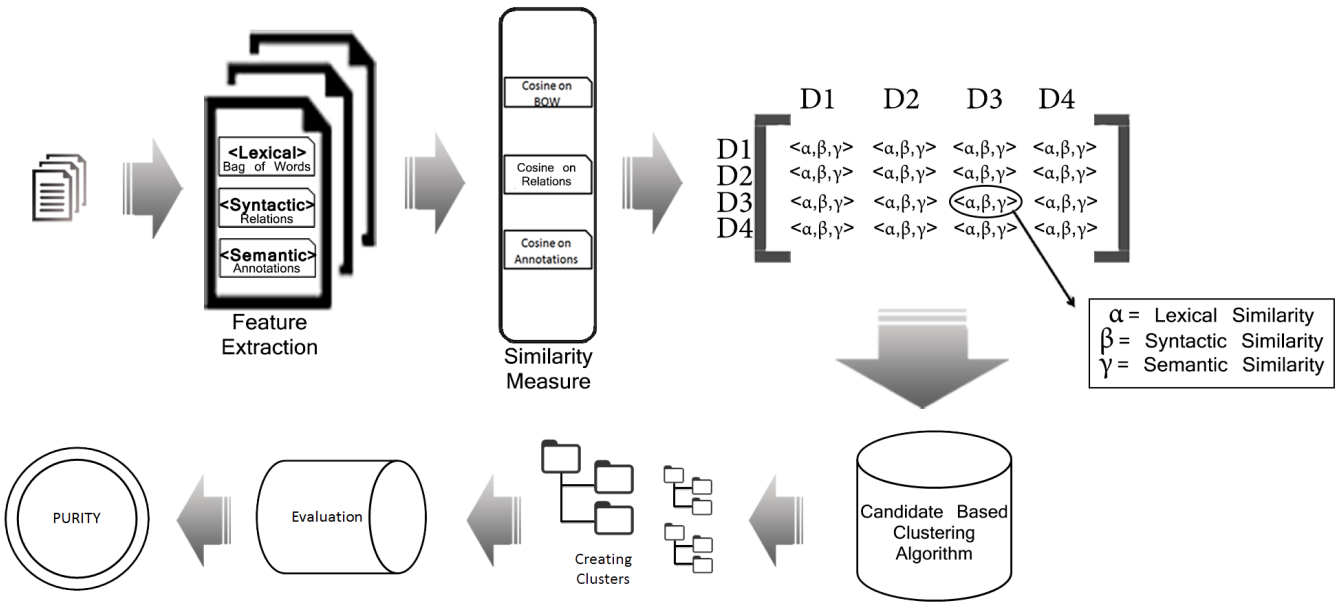

Fig. 3. DLS-DC approach

automatic text summarization, text categorization, information retrieval, etc.

3) Extraction of semantic features: Semantic representation is built using Fred API. The API gives a semantic rich representation of a document in terms of RDFs. FRED is a tool to automatically transform knowledge extracted from text into RDF and OWL, i.e. it is a machine reader for the Semantic Web. It is event-centric; therefore it natively supports event extraction. The API uses word sense disambiguation. FRED has got precision, recall, and accuracy largely better than the other tools attempting event extraction.
Semantic annotating features are then extracted using SPARQL queries and saved as triples. For example, in a certain document, the object Event is the annotating feature for its subjects call, do, see and think, which means that all these words graphically point to the object Event.

B. Similarity Measures

1) Lexical similarity: The similarity is calculated using Cosine Similarity measure which takes two document vectors and returns a similarity score between the documents. The vector size of each document is the size same as of the dictionary vector. The similarity is calculated using the formula below;

$$Sim(D_a \leftarrow D_b) = Cosine(< tf.idf\ V > D_a, < tf.idf\ V > D_b) \tag{2}$$

Where 'V' is $TF * IDF$ Vector.

2) Syntactic similarity: Word Order Problem has been handled by assigning equal weights to each of the three features given by Alchemy. Predicate is checked before the Subject and Object. For example, in the sentences "Joe killed Mary" and "Mary killed Joe", the predicate 'killed' is similar. As a result, it is assigned a weight of $1 * 0.33$, whereas 'Joe' as a subject in the first sentence does not match with 'Mary' as a subject in the second sentence and same goes for 'Mary' as an object in the first sentence with 'Joe' as an object in the second. Consequently, they are assigned a weight of $0 * 0.33$ each. This gives us $0.33 + 0 + 0 = 0.33$ as a similarity score between the two sentences.

This was for sentence to sentence measure. On document to document, the similarity is calculated through the following formula:

$$Sim(D_a \leftarrow D_b) = \frac{\Sigma_{i=1}^{n} max(Sim(S_{a_i} \leftarrow S_{b_i} \dots S_{b_n}))}{max(D_a.size, D_b.size)} \tag{3}$$

Where S_{a_i} is the i^{th} sentence of document 'a' and $S_{b_i} ... S_{b_n}$ are all the sentences in document 'b'. Every sentence of document 'a' is matched to every sentence in document 'b' and the maximum similarity scores of all sentences are averaged by the number of sentences of the document which has larger number of sentences in it.

3) Semantic similarity: Event {Call, Do, See, Think}, where Event is the annotating feature and the words in the braces are its subjects, i.e. all the incoming nodes to the object.

Similarly, there are objects (annotating features), generated by the FRED, as Activity, For and others, depending upon the document. To build semantic representation of a document, all the annotating features along with their subjects are extracted using SPARQL.

While comparing two documents for similarity, the algorithm first checks if the object (annotating feature) of one document matches with the other. If it matches, the similarity

is computed on the subjects of both the objects using Cosine Similarity measure. All the subjects in all annotating features are matched based on the condition above and the similarities of all the features are summed up. The document to document similarity score is calculated by dividing the final similarity sum with the average of the total number of objects in both the documents. The above representation also captures word sense disambiguation. For Example, **D3: "I am doing research on Semantics. D4: "My research is on Semantics**

The semantic score for the above documents will be 1 by using this representation. Below is an algorithm for document to document similarity calculation.

Algorithm 1 Document to Document Similarity

Require: Documents N
Ensure: $Similarity\ Matrix\ M\ containing\ Vector\ V < \alpha, \beta, \gamma >$
$Documents\ D[] \leftarrow N$
$X \leftarrow SELECT\ ?subject\ ?predicate\ ?object\ WHERE\{?subject\ ?predicate\ ?object\ .\}$
for $i\ \ldots\ N.length$ **do**
  $A_{lex} \leftarrow Terms(D[i])$
  $A_{syn} \leftarrow Alchemy(D[i])$
  $A_{sem} \leftarrow getAnnotatingFeatures(Q(D[i]))$
  for $j\ \ldots\ N.length$ **do**
    $B_{lex} \leftarrow Terms(D[j])$
    $B_{syn} \leftarrow Alchemy(D[j])$
    $B_{sem} \leftarrow getAnnotatingFeatures(Q(D[j]))$
    $\alpha \leftarrow CosineSim(A_{lex},\ B_{lex})$
    $\beta \leftarrow CosineSim(A_{syn},\ B_{syn})$
    for all $s_1 \in A_{sem}.keySet()$ **do**
      for all $s_2 \in B_{sem}.keySet()$ **do**
        if $s_1 == s_2$ **then**
          $\gamma += Cosine(A_{sem}.get(s_1),\ B_{sem}.get(s_2))$
        end if
      end for
    end for
    $\gamma = \gamma\ /\ Avg(A_{sem}.keySet().size(), B_{sem}.keySet().size())$
    $V_{i,j} \leftarrow V < \alpha, \beta, \gamma >$
    $M_{i,j} \leftarrow V_{i,j}$
  end for
end for

C. Candidate Based Clustering Algorithm

Here is an algorithm for candidate based document clustering.

The algorithm takes a similarity matrix of size $D \times D$ containing a vector of size three on each value. M_{prev} and M_{curr} are the matrices taken for keeping the record of updated matrix each time after the matrix is merged in each iteration. DocPair is a pair of two documents that is extracted through the matrix containing the vectors. Example of a DocPair 103124 - 102616. C_{pair} is the final Cluster Pair decided after making the candidate based decisions. The C_{pair} is sent to the MERGE and UPDATE function which reduces the matrix by one column and one row and updates the values of the matrix using Average Linkage Strategy. Clusters is a list of Clusters that initially contains D clusters and the two Clusters based upon the decision are merged using ClustersUpdate function in order to get the final K level clusters.

Algorithm 2 Candidate Clustering Algorithm

Require: $Similarity\ Matrix\ M\ containing\ Vector\ V < \alpha, \beta, \gamma >, ClusterLevels\ K\ ,\ DocumentIDs\ D$
Ensure: $Partition\ of\ D\ with\ K\ classes$
$M_{prev} \leftarrow M$
$M_{curr} \leftarrow M$
$Clusters \leftarrow D$
for $i\ \ldots\ D.length\ -\ K$ **do**
  $DocPair_{lex} \leftarrow M.extractMax(V(\alpha))$
  $DocPair_{syn} \leftarrow M.extractMax(V(\beta))$
  $DocPair_{sem} \leftarrow M.extractMax(V(\gamma))$
  $ClusterPair\ C_{pair} \leftarrow DocPair_{lex}$
  if $DocPair_{lex} == DocPair_{syn}$ **then**
    $C_{pair} \leftarrow DocPair_{lex}\ OR\ DocPair_{syn}$
  end if
  if $DocPair_{syn} == DocPair_{sem}$ **then**
    $C_{pair} \leftarrow DocPair_{syn}\ OR\ DocPair_{sem}$
  end if
  if $DocPair_{lex} == DocPair_{sem}$ **then**
    $C_{pair} \leftarrow DocPair_{lex}\ OR\ DocPair_{sem}$
  end if
  $M_{prev} \leftarrow M_{curr}$
  $M_{curr} \leftarrow MERGE(M_{prev}\ ,\ C_{pair}\ )$
  $M_{curr} \leftarrow UPDATE(M_{prev}\ , C_{pair}\ ,\ AveregeLinkage)$
  $ClustersUpdate(C_{pair}.leftChild\ ,\ C_{pair}.rightChild)$
end for

IV. EXPERIMENTAL STUDIES

A. Implementation of Algorithm

The proposed algorithmic approach has been compared with a number of recently proposed document clustering algorithms on the popular standard dataset of NEWS20 and Reuters21578 for the problem of document clustering. The DLS-DC is implemented in Java programming language. The experiment is executed on a Dell 5547 Notebook with Intel Core i7 processor and 8GB of RAM with 1TB of Hard Disk Storage.

B. Datasets

We have used the popular text data sets NEWS20 and Reuters21578 for our experiments.

Data Set	Data Sources	No. of Docs	No. of Classes
D1	NEWS20	50	5
D2	NEWS20	100	10
D3	NEWS20	200	15
D4	NEWS20	400	15
D5	NEWS20	813	20
D6	Reuters21578	797	15

TABLE I. SAMPLE DATA SETS FROM NEWS20 AND REUTERS

C. Evaluation

We justify the effectiveness of our proposed method by using standard cluster quality measures like

1) F-Measure: The F-measure uses a combination of precision and recall values of clusters. The F-measure, $F(i, j)$, of a class

$$F(i,j) = \frac{2 * prec(i,j) * rec(i,j)}{prec(i,j) * rec(i,j)} \quad (4)$$

The F-measure for entire clustering result is defined as

$$F = \sum_{i}^{n} \frac{n_i}{n} max(F(i,j)) \quad (5)$$

2) Purity: Purity can be defined as the maximal precision value for each class j. We compute the purity for a cluster j as:

$$Purity(j) = \frac{1}{c_j} max(c_{ij}) \quad (6)$$

We then define purity of the entire clustering result as:

$$Purity = \sum_{j}^{n} \frac{c_j}{N} purity(j) \quad (7)$$

3) Baseline: The baseline for this experiment is set using the bag-of-words representation for documents. We are using $TF * IDF$ based representation for document vectors and cosine measure to create a clustering arrangement for our baseline.

D. Comparative Work

We would like to compare our proposed approach to the three recent approaches that claim that they produce semantic rich clustering. The approach in [6] proposed a frequent item set-based representation of documents for clustering (FIHC), the second is from [10] from where we only compare with frequent word sequences (CFWS), and third and final is from [12] where authors used topic maps based representation of documents. We have implemented the proposed approaches as described in [6, 10, 12].

V. RESULT & DISCUSSION

In this paper, we present a new approach to cluster the documents based on semantic rich features using a vector representation. The inferred knowledge from the three representations is used to define the similarity measures between the pair of documents. Each of these measures is used in a matrix with each value containing a vector of size three to cluster the set of documents by using Candidate Based Clustering Algorithm CBCA. First, we would like to discuss F-measure of Hierarchical Clustering on individual levels with Candidate Based Clustering Algorithm CBCA.

DataSets	Lex	Syn	Sem	CBCA
D1	0.55	0.29	0.35	0.67
D2	0.34	0.18	0.22	0.42
D3	0.41	0.22	0.27	0.5
D4	0.49	0.26	0.32	0.6
D5	0.53	0.28	0.34	0.65
D6	0.75	0.49	0.61	0.89

TABLE II. F-MEASURE FROM THE EXPERIMENTS

Results of F-Measure on different layers for Hierarchical Clustering on individual levels with Candidate Based Clustering Algorithm CBCA.

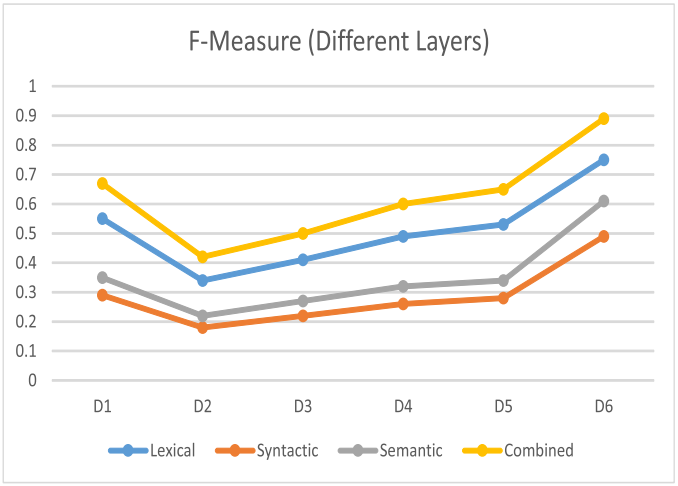

Fig. 4. F-measure from the experiments

We next want to discuss the purity of Hierarchical Clustering on individual levels with Candidate Based Clustering Algorithm CBCA.

DataSets	Lex	Syn	Sem	CBCA
D1	0.58	0.33	0.39	0.71
D2	0.4	0.23	0.27	0.49
D3	0.46	0.26	0.31	0.56
D4	0.52	0.3	0.35	0.63
D5	0.55	0.31	0.37	0.67
D6	0.72	0.41	0.42	0.88

TABLE III. PURITY FROM THE EXPERIMENTS

Results of Purity on different layers for Hierarchical Clustering on individual levels with Candidate Based Clustering Algorithm CBCA.

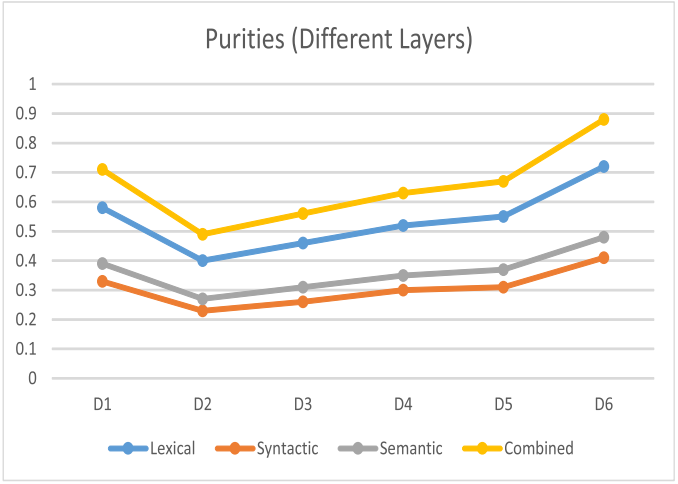

Fig. 5. Purity from the experiments

The higher purity values by candidate based clustering algorithm is an indication of producing high-quality clusters which is again due to the fact that a combined representation scheme is used for clustering. The experimental results show that DLS-DC performs better than comparative algorithms of this study in terms of quality of the clusters produced. Increased cluster purity clearly establishes the fact that the features extracted from the three representations capture the semantics of the documents. The three approaches FIHC [6], CFWS [10] and TMHC [12] produced F-measure for the data sets (See Table IV).

DataSet	FIHC	CFWS	TMHC	CBCA
D1	0.55	0.62	0.63	0.67
D2	0.54	0.58	0.66	0.42
D3	0.48	0.44	0.58	0.5
D4	0.43	0.44	0.58	0.6
D5	0.41	0.39	0.53	0.65
D6	0.88	0.68	0.89	0.89

TABLE IV. F-MEASURE FROM DIFFERENT APPROACHES ON TEST DATASETS

The proposed approach clearly had shown improvement in most of test cases. This is due to the fact that the multiple representations of documents in the collection capture the semantics in a better way, and are able to produce high F-Measure which is an indication of balance precision and recall (See Figure 6).

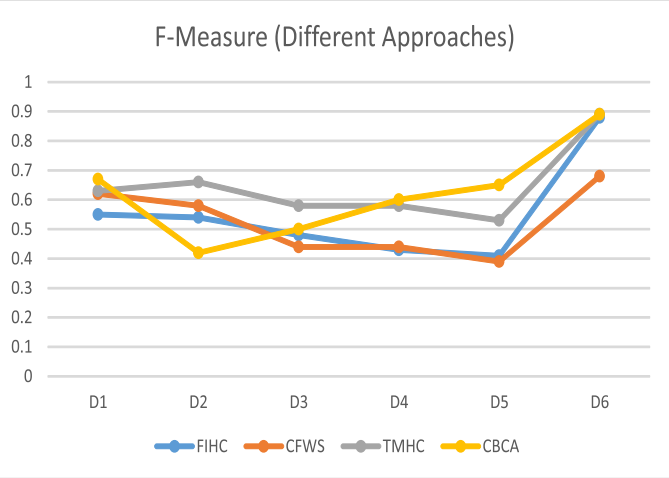

Fig. 6. F-measure from different approaches on test datasets

Similarly, another evaluation that is very instrumental in identifying the better clustering is purity. The proposed approach produces better purity values when compared to the comparative algorithms (See Table V).

DataSet	FIHC	CFWS	TMHC	CBCA
D1	0.6	0.62	0.66	0.71
D2	0.6	0.62	0.64	0.49
D3	0.58	0.6	0.68	0.56
D4	0.57	0.6	0.64	0.63
D5	0.56	0.58	0.6	0.67
D6	0.78	0.72	0.81	0.88

TABLE V. PURITY FROM DIFFERENT APPROACHES ON TEST DATASETS

The purity with the proposed approach DLS-DC indicates that our idea of different representation of the same document (with different focus) has produced better understanding at representation level. Hence, the automatic clustering process implicitly identifies the common attributes to produce better purity values (See Figure 7). In most of the approaches DLS-DC is performing well as evident in results of purity and F-Measure. The dataset classes D1 to D5 are created manually from the complete dataset of news20.Due to confusion between documents in dataset classes D2 and D3, the graph shows slightly lower purity and F-measure values.

VI. CONCLUSION

We propose an approach that exploits document level semantics in document clustering. The representation of document comprises of three levels namely: lexical, syntactic and semantic, that are defined for the document. In lexical representation, we only use lexical features. The syntactic representation comprises of syntactical features through transformation using Alchemy API. We also cater Word Order Problem in Syntactic analysis. The Semantic representation is defined by FRED API and RDF based annotated structures that are extracted from each document by using SPARQL queries and the similarity is calculated by using the algorithm defined. Clustering is performed by using a candidate based clustering approach. The proposed approach clearly surpasses purity and F-measure in comparison to recently proposed approaches like (FIHC, CFWS and TMHC), which is an indication of better clustering results. We like to extend this research in a number of ways. First, we would like to introduce document constraints in the clustering approach. Secondly, we would like to introduce some methods to increase the weight given to the semantic representation, while making decisions in the clustering algorithm. Moreover, we would like to further investigate the combined representation of document using the three representations because it seems a more challenging aspect for good clustering.

Fig. 7. Purity from different approaches on test datasets

REFERENCES

[1] Alessandro Campi and Stefania Ronchi. “The Role of Clustering in Search Computing”. In: *Database and Expert Systems Application, 2009. DEXA’09. 20th International Workshop on*. IEEE. 2009, pp. 432–436.

[2] Hung Chim and Xiaotie Deng. “Efficient phrase-based document similarity for clustering”. In: *Knowledge and Data Engineering, IEEE Transactions on* 20.9 (2008), pp. 1217–1229.

[3] Douglass R Cutting et al. “Scatter/gather: A cluster-based approach to browsing large document collections”. In: *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 1992, pp. 318–329.

[4] Christiane Fellbaum. “ed. WordNet: an electronic lexical database”. In: *MIT Press, Cambridge MA* 1 (1998), p. 998.

[5] Rafael Ferreira et al. “A new sentence similarity assessment measure based on a three-layer sentence representation”. In: *Proceedings of the 2014 ACM symposium on Document engineering*. ACM. 2014, pp. 25–34.

[6] Benjamin CM Fung, Ke Wang, and Martin Ester. "Hierarchical Document Clustering using Frequent Itemsets." In: *SDM*. Vol. 3. SIAM. 2003, pp. 59–70.

[7] Khaled M Hammouda and Mohamed S Kamel. "Efficient phrase-based document indexing for web document clustering". In: *Knowledge and Data Engineering, IEEE Transactions on* 16.10 (2004), pp. 1279–1296.

[8] Marti A Hearst and Jan O Pedersen. "Reexamining the cluster hypothesis: scatter/gather on retrieval results". In: *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 1996, pp. 76–84.

[9] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. "Data clustering: a review". In: *ACM computing surveys (CSUR)* 31.3 (1999), pp. 264–323.

[10] Yanjun Li, Soon M Chung, and John D Holt. "Text document clustering based on frequent word meaning sequences". In: *Data & Knowledge Engineering* 64.1 (2008), pp. 381–404.

[11] Valentina Presutti, Francesco Draicchio, and Aldo Gangemi. "Knowledge extraction based on discourse representation theory and linguistic frames". In: *Knowledge Engineering and Knowledge Management*. Springer, 2012, pp. 114–129.

[12] Muhammad Rafi, M Shahid Shaikh, and Amir Farooq. "Document clustering based on topic maps". In: *arXiv preprint arXiv:1112.6219* (2011).

[13] Joseph Turian. *Using AlchemyAPI for Enterprise-Grade Text Analysis*. Tech. rep. Technical report, AlchemyAPI (August 2013), 2013.

[14] Yujing Wang et al. "Representing document as dependency graph for document clustering". In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM. 2011, pp. 2177–2180.