Semantically Driven Soft-clustering of Documents using Lexical Chains

Dipti Deodhare[*], **Govind Sharma**[†], **Ashish Srivastava**[*], **Alind Sharma**[*]
[*](Centre for Artificial Intelligence and Robotics,
Defence R & D Organisation, Bangalore, INDIA),
[†](B-Tech, Manipal Institute of Technology)
dipti@cair.drdo.in,
govindsharmajsk@gmail.com,
ashishsrivastava@cair.drdo.in,
alindsharma@cair.drdo.in

Abstract

Automatic document clustering is one of the important operations performed on text documents. Most clustering algorithms put each data point (here, document) into one cluster. In the real world, each document contains multiple themes which cannot be detected by hard clustering algorithms. Thus, we provide a soft clustering algorithm, wherein each document can be associated to multiple clusters. We transform the dataset into a feature space of lexical chains using the WordNet. Lexical chains as document features are advantageous over the Bag of Words approach, both in terms of speed and quality of results obtained. Our algorithm uses the global lexical chain set, which is nothing but the union of the lexical chains from individual documents. A semantic similarity matrix is calculated on this global set of lexical chains. This matrix is further used to create a graph, each node of which represents a lexical chain. The algorithm we propose connects each node (lexical chain) to another node only if the semantic similarity between them is greater than a given threshold. Once this is done, we find maximal cliques of length one from the graph. Lexical chains belonging to a clique are a cluster of semantically related topics. After the formation of lexical-chain clusters, we associate each of them to each document, which is qualified by the lexical chains extracted from it. One document can be associated to more than one cluster of lexical chains giving soft clusters of documents. Documents associated with the same cluster would have semantically similar lexical chains. Further if each cluster represents a distinct topic, this soft clustering algorithm can facilitate topic detection. Empirically we compare the performance of our algorithm with some recently reported soft clustering algorithms in the literature. The algorithm is demonstrated on three popular benchmarks, namely, Brown corpus, Reuters corpus and 20 Newsgroups dataset.

Keywords: Lexical chains, wordnet, semantic distance matrix, clique, soft clustering.

1 Introduction

Document clustering is an unsupervised categorisation of documents based on its contents. Document clusters are useful for various tasks such as text mining, topic detection and tracking, *etc.* Text data usually contains complex semantic information which is communicated using a combination of words. Ideally, the representation used should capture and reflect this fact in order to semantically drive the clustering algorithm and obtain better results. In this paper we use lexical chains to represent the semantic information contained in the document. As shown in (Jayarajan et al., 2008), this representation results in a drastic reduction in the size of the feature space. We use this representation to establish a semantically driven method for the graphical representation of clusters of lexical chains. This graphical representation is further used to establish an algorithm for soft clustering of documents.

Proceedings of ICON-2010: 8th International Conference on Natural Language Processing
Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2010

Section 2 describes the technique to generate the lexical chains. Lexical Chaining is a technique which seeks to identify and exploit the semantic relatedness of words in a document. A semantic measure to select important lexical chains is then described in Section 3. An information-based approach is used to find the utility of the lexical chains to the clustering process. Consequent to this, a global set of lexical chains with good information content is obtained. Section 4 deals with the calculation of a semantic similarity based matrix on this global set of lexical chains. Each matrix element (a_{ij}) represents the similarity between the i^{th} and the j^{th} lexical chains. The method combines a lexical taxonomy structure with corpus statistical information so that the semantic distance between nodes in the semantic space constructed by the taxonomy can be better quantified with the computational evidence derived from a distributed analysis of corpus data. Using the *information content* and *similarity matrix* a graph is generated in Section 5. Lexical chains are represented as the nodes of the graph. If the similarity between the lexical chains is above a defined threshold the chains are connected by an edge. Section 6 describes the technique to form clusters of documents using the graph generated. The complete algorithm for soft clustering of documents is given in this section along with the results. Section 7 includes the results and Section 8 the conclusions.

2 Obtaining Lexical Chains based Features from Documents

Lexical Chains are groups of words which exhibit lexical cohesion. Given a corpus of text documents, the following steps need to be carried out to extract useful information from them in the form of candidate words:

(*i*) **Tokenization**: Using a standard lexical analyser the input documents are tokenized to extract words, numbers and punctuations present in them.

(*ii*) **Stop Word Removal**: Frequently occuring words such as *this, the, what, is, end, etc.*, which carry no (or very less) semantic information are removed in this step.

(*iii*) **Morphological Analysis**: An analysis that involves detaching inflectional endings from words and checking for exceptions using WordNet (Miller et al., 1990) is then performed.

(*iv*) **Word Sense Disambiguation**: It is done to

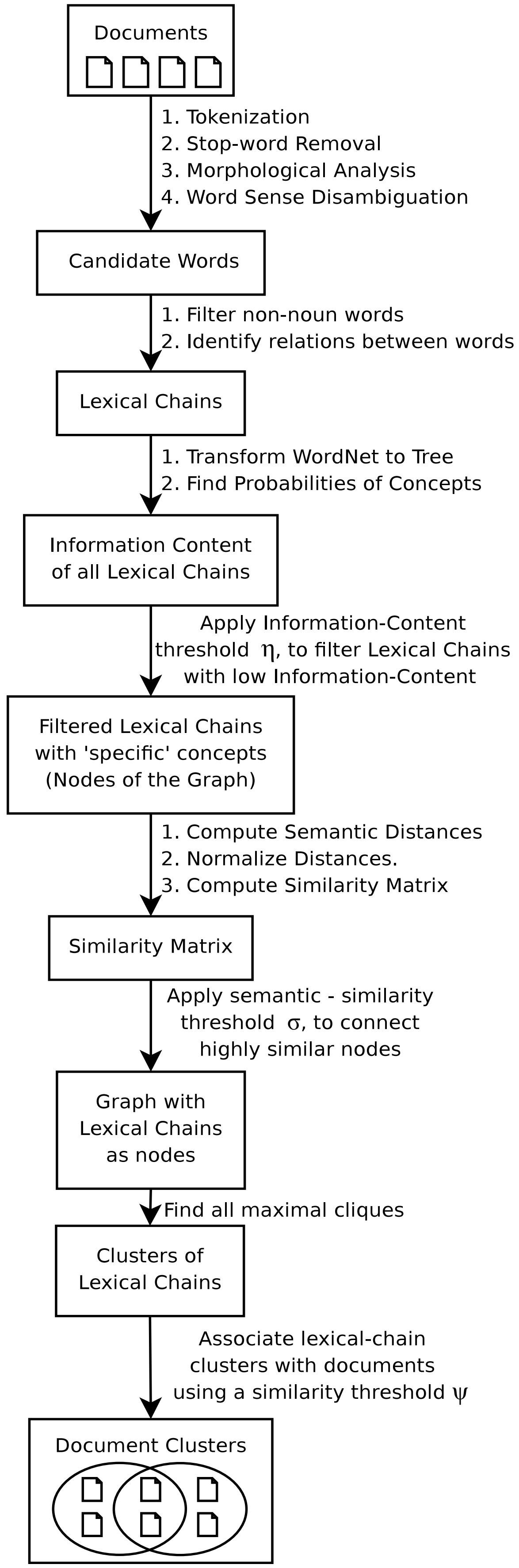
Documents
1. Tokenization
2. Stop-word Removal
3. Morphological Analysis
4. Word Sense Disambiguation
Candidate Words
1. Filter non-noun words
2. Identify relations between words
Lexical Chains
1. Transform WordNet to Tree
2. Find Probabilities of Concepts
Information Content of all Lexical Chains
Apply Information-Content threshold η, to filter Lexical Chains with low Information-Content
Filtered Lexical Chains with 'specific' concepts (Nodes of the Graph)
1. Compute Semantic Distances
2. Normalize Distances.
3. Compute Similarity Matrix
Similarity Matrix
Apply semantic - similarity threshold σ, to connect highly similar nodes
Graph with Lexical Chains as nodes
Find all maximal cliques
Clusters of Lexical Chains
Associate lexical-chain clusters with documents using a similarity threshold ψ
Document Clusters

Figure 1: Step by step procedure for the formation of soft document clusters.

automatically disambiguate the meaning of a word from its context. The algorithm (Patwardhan et al., 2003) that does so in reference to WordNet has been used.

We then filter out all the non-noun words, based on the assumption that nouns are sufficient at reflecting the topics contained in the document. This has been empirically demonstrated in (Jayarajan et al., 2008). To form lexical chains, we use the WordNet to identify the relations between words. Only the identity and synonymy relations (treated as a single 'IS' relation) are used to compute the chains. It has been shown (Jayarajan et al., 2008) that empirically, the usage of these two relations resulted in chains representing crisp topics. A lexical chain contains a list of words which are related to each other and each word is represented as a 4-tuple <term, pos, sense, rel>, where 'pos', 'sense' and 'rel' are part-of-speech, WordNet sense number and relation of this word to the lexical chain respectively.

The algorithm presented in Algorithm 1 (reproduced from (Jayarajan et al., 2008)) maintains a global set of lexical chains, each of which represents a topic. It then identifies all possible lexical chains for a document by comparing the candidate words of each document with the global list to identify those chains with which it has an *identity* or *synonymy* relation. If no chains are identified, then a new chain is created and put in the global list. At the end, a global set is obtained which lists all the chains contained in all the documents.

Algorithm 1 Generate Lexical Chains

```
Maintain a global set of lexical chains, initialised to a Null set.
for each document do
    for each candidate word in document do
        Identify lexical chains in global set with which the word has a identity/synonymy relation
        if No chain is identified then
            Create a new chain for this word and insert in global set
        end if
        Add word to the identified/created chains in Global set
    end for
end for
```

3 Sorting Lexical Chains based on Information Content

Lexical chains reflect the discourse structure of the documents. But not all lexical chains are important. A mathematical measure has been proposed (Jayarajan et al., 2008) to select and use a subset of 'good' chains from the set of chains assigned to each document to represent it, by calculating the utility of a lexical chain. A more semantic measure to select 'good' chains would be Resnik's (Resnik, 1995) information-based approach. This method has been proposed to assess semantic similarity between words using WordNet. We extend this method to lexical chains. Since we only use identity and synonymy relations to compute a lexical chain, *each lexical chain is adequately represented by the first word in it.* Hence Resnik's method can be applied to lexical chains as well. According to this method, for any concept c in a taxonomy, let $p(c)$ be the probability of encountering an instance of the concept. Following the standard definition from information theory, the information content of c, $IC(c)$, is $-\log p(c)$.

To realize this, the WordNet is transformed into a tree, with each concept (word+sense) acting as a node and the IS-A (hypernymy-hyponymy) relation forming the links between the nodes. In other words, each parent is a direct hypernym (generalization) of its immediate child (Refer Figure 2). For example, using the 20 Newsgroup dataset (Rennie, 1995) we establish the probability of each lexical chain (a concept). We first calculate lexical chains across all the documents, using the algorithm mentioned in Section 2. Then, the global set of lexical chains, (GLC), is taken and probabilities of encountering the concepts (here lexical chains) are calculated using the following formula:

$$p(lc) = \frac{\sum\limits_{l \in W(lc)} length(l)}{\sum\limits_{l \in GLC} length(l)} \quad (1)$$

where $W(lc)$ is the set of all concepts that are subsumed by the lexical chain, lc, *i. e.,* all children (in the WordNet tree) of lc, starting from immediate node to leaf nodes, that are also present in the document.

This can be converted to a more understandable form, *i. e.*, the information content. A table relating lexical chains to their information contents is formed for all the lexical chains, using the Algorithm 2.

Algorithm 2 Calculate Information Content for all Lexical Chains

```
for each $lc \in GLC$ do
    $IC[lc] = -\log p(lc)$
end for
```

This can be further used to find the semantic similarity between two lexical chains and to improve the quality of our clusters by omitting the lexical chains having less information content.

4 Computing a Semantically Driven Similarity Matrix

To cluster the lexical chains present in the documents, the semantic similarity needs to be calculated between them. Many approaches (Budanitsky and Hirst, 2006) for finding the similarity between words have been proposed. They are widely categorized into edge-based and node-based methods. (Jayarajan, 2009) uses the Dice Coefficient as the similarity measure for soft clustering. But that gives similarity based on the number of lexical chains in two documents, which is a statistical approach.

In this paper, Jiang and Conrath's semantic similarity approach (Jiang and Conrath, 1997), that is both edge-based and node-based has been chosen. According to their postulation, the distance between two concepts is given in terms of their information content. This approach has been extended to lexical chains by applying the following formula:

$$dist_{JC}(lc_1, lc_2) = IC(lc_1) + IC(lc_2) - 2 \times IC(lso(lc_1, lc_2)) \quad (2)$$

This gives the semantic distance between two lexical chains, lc_1 and lc_2. Here, $lso(lc_1, lc_2)$ represents the lowest super-ordinate (or most specific subsumer) of lc_1 and lc_2. This can be clear from Figure 2, which shows a part of the WordNet tree, containing words/concepts as nodes. The lowest super-ordinate (lso) for nickel and dime is the lowest concept in the tree subsuming both nickel and dime, which is coin. Similarly, lso(dime, credit card)=medium of exchange.

The semantic distance given by equation 2 can be converted to semantic similarity by first normalizing the distance, $dist_{JC}$ between 0 and 1, and then subtracting it from 1 (Budanitsky and Hirst, 2006). Thus,

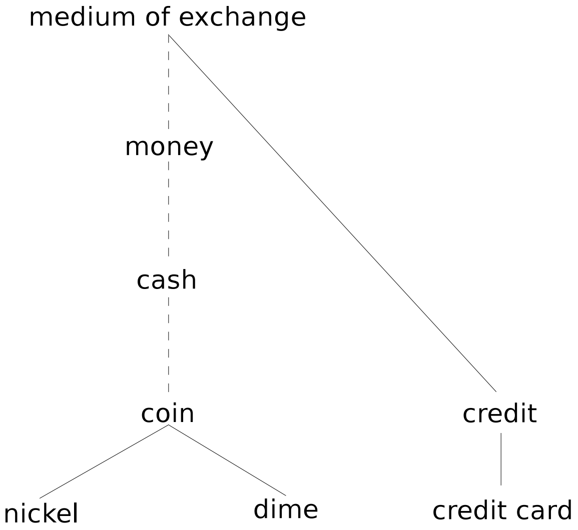

Figure 2: Fragment of the WordNet tree; dashed lines indicate that some intervening nodes have been omitted. Adapted from (Resnik, 1995)

$$sim_{JC} = 1 - dist_{JC} \quad (3)$$

provided $dist_{JC} \in [0, 1]$.

Similarities computed above are used to form a similarity matrix, SM, which is a symmetric matrix and has all diagonal elements as 1. The formation of such a matrix is given in Algorithm 3.

Algorithm 3 Calculate Similarity Matrix

```
Let $N$ be the total number of lexical chains in $GLC$
for $i \leftarrow 1$ to $N$ do
    $SM[i][i] = 1$
    for $j \leftarrow 1$ to $i - 1$ do
        $SM[i][j] = sim_{JC}(lc_i, lc_j)$
        $SM[j][i] = SM[i][j]$
    end for
end for
```

5 Evolving the Similarity Graph

Using the information content and the similarity matrix, obtained from Section 3 and 4 respectively, we transform our dataset into a graph in which nodes are formed by lexical chains. As stated earlier, not all lexical chains are significant when the clustering of documents is considered.

To decide which lexical chains to consider for clustering, the information content table formed in Section 3, which categorizes them into those showing 'general' and 'specific' concepts is required. The former gives unnecessary clusters that have no (or very less) significance and the latter gives highly significant clusters containing lexical chains that are more informative. To realize this, a lower limit, η is imposed on the information content of a lexical chain. Thus, only those lexical chains which satisfy the threshold (*i. e.,* $IC(lc) \geq \eta$) qualify to act as nodes of the to-be-formed graph, G.

The edges of the graph are decided on the basis of the similarity between the nodes (lexical chains). Using the similarity matrix formed in Section 4, it is decided whether or not two nodes have a connection. If the concepts shown by two nodes are dissimilar, they should not be connected. To decide what 'similar' means, a lower limit, σ, is imposed on the similarities present in the similarity matrix.

Algorithm 4 Transform the input to a graph, G

```
Let $GLC = \{lc_1,\ lc_2,\ lc_3,\ ...\ lc_N\}$
$V = \phi$   // the set of vertices
$E = \phi$   // the set of edges
for $i \leftarrow 1\ to\ N$ do
    if $IC[i] \geq \eta$ then
        $V = V \cup \{lc_i\}$
    end if
    for $j \leftarrow 1\ to\ i-1$ do
        if $SM[i][j] \geq \sigma$ and $IC[j] \geq \eta$ then
            $E = E \cup \{lc_i,\ lc_j\}$
        end if
    end for
end for
$G = (V,\ E)$
```

If $\sigma \approx 1$, massive and highly overlapping clusters are formed. These clusters will not be very distinct from each other, as two clusters will have a large portion of common lexical chains and very few distinct ones. On the other hand, if $\sigma \approx 0$, very few lexical chains will be involved in the clustering process, and distinct clusters will be formed with two nodes only. Only the 'highly similar' lexical chains will form clusters. All the pairs of lexical chains, (lc_i, lc_j) represented by node pairs satisfying the threshold ($SM[i][j] \geq \sigma$) have edges between them (See Algorithm 4). Finally, we have a graph, G with nodes as lexical chains and edges between 'similar' nodes.

6 The Soft Clustering Algorithm

Forming a graph in the previous section has already clustered the lexical chains on a semantic basis. All we need to do is to identify the clusters out of it. There may be clusters of size 1 (distinct points) to size i (a completely connected graph having i nodes). The aim is to find all maximal cliques of length one from the graph, G. A clique in a graph is a subset of its vertices such that every two vertices in the subset are connected by an edge. A maximal clique is a clique that is not included in a larger clique. Maximal cliques of cardinality 1 are the points which are not connected to any other points (*i. e.*, outliers) and those of cardinality 2 are 2-point clusters. Similarly, maximal cliques of cardinality n form n-point clusters.

Bron-Kerbosch algorithm (Bron and Kerbosch, 1973) is a popular and effective algorithm for finding cliques and its second version is shown in Algorithm 5. The algorithm maintains three sets of nodes R, P, X to calculate maximal cliques that include all the vertices in R, some of the vertices in P and none of the vertices in X. The recursion is initiated by calling the function using **BronKerbosch**(ϕ, $V(G)$, ϕ), where V is the set of all vertices of graph, G. This version of the algorithm involves a pivot vertex, chosen from $P \cup X$ (Cazals and Karande, 2008). Also note that $N(u)$ is the set of neighbours of u, *i. e.,* $N(u) = \{v \mid (u,\ v)\ \in\ E(G)\}$.

Algorithm 5 Bron-Kerbosch recursive algorithm to find maximal cliques

```
Let $P$ be the vertex set of the graph.
Sets $R,\ X\ =\ \phi$ initially
BronKerbosch($R$, $P$, $X$)
if $P$ is empty and $X$ is empty then
    return $R$
end if
choose a pivot vertex, $u\ \in\ (P \cup X)$
for all $v$ such that $v\ \in\ (P \setminus \mathrm{N}(u))$ do
    BronKerbosch($R \cup \{v\}$, $P \cap \mathrm{N}(v)$, $X \cap \mathrm{N}(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end for
```

All maximal cliques obtained from the graph represent clusters of similar lexical chains, across the documents. They even represent a particular topic. Each cluster reflects a distinct theme, which needs to be associated with documents, which themselves are represented as a collection of lexical chians.

Now a correspondence matrix that maps documents to clusters need to be computed. This is a $N_c \times N_d$ binary matrix (N_c = number of lexical chain clusters, N_d = number of documents) showing the association of documents to clusters. To obtain the semantic similarity between a document 'd' and a cluster 'c' of lexical chains the following measure is proposed:

$$sim(c,d) = \frac{\log(1 + |c \cap d|)}{\log(1 + |c| + |d|)} \times 100 \quad (4)$$

where $|x|$ is the cardinality of the cluster x, *i.e.*, it represents the number of lexical chains in the document or the cluster and $|c \cap d|$ represents the number of common lexical chains in the cluster and the document.

To decide whether or not a document is associated to a cluster, we need to impose another lower-limit, ψ on the values of this matrix. Ultimately, we get a correspondence matrix, $CM_{|c|\times|d|}$ as follows:

$\forall\, i \in c,\ j \in d,$

$$\begin{aligned} CM[i][j] \;\; &= \;\; 1,\ \text{if } sim(c,d) \geq \psi \\ & \quad\;\; 0,\ \text{otherwise.} \end{aligned} \quad (5)$$

If $\psi \approx 0$, then there are chances that even the dissimilar or less similar documents get clustered into one single cluster. Imposing this threshold on the similarity matrix formed above, a correspondence matrix is obtained, showing the association of a document to a cluster.

7 Results and Discussions

A pictorial representation of the complete soft clustering algorithm has been given in Figure 1. The results have been analysed in two ways *viz* lexical chain clustering and soft clustering of documents. We first present the results with reference to the lexical chain clustering. For this we use the Browns Corpus (Francis and Kučera, 1982). To discuss our results for soft clustering we use a small subset of documents from 20 Newsgroup (Rennie, 1995) and the entire training set of the Reuters Corpus (Lewis, 1987).

(*i*) **Results of lexical chain clustering on Browns corpus**: We use the Brown corpus to discuss the output of the lexical chain clustering step in the algorithm. Randomly selected eight clusters have been reproduced in Table 1. The first word of each lexical chain has been included. This is valid since only the identity and the synonymy relations are being used. The words are accompanied by a sense-number generated by the word sense disambiguation (WSD) algorithm discussed in section 2, which is basically the index of the synset containing the word from the *data.pos* file in WordNet (Miller et al., 1990). For instance, the word, 'red' in the second cluster has sense number 4, which, according to WordNet, means, "the amount by which the cost of a business exceeds its revenue", which shows that it is relevant to the cluster.

It can be noted that these clusters, though small in size, are very accurate. Furthermore, the accuracy can be controlled by changing the parameters, η and σ. Decreasing the similarity threshold, σ, will decrease the accuracy of the clusters, as it would cluster not-so-similar lexical chains into one. Also, the lexical chains shown in Table 1 have high information-content values because the information-content threshold, η, is high. Decreasing it will lead to clustering of lexical chains with too general concepts (low information-content), ultimately decreasing the quality of the clusters.

(*ii*) **Results on a subset of 20 Newsgroups documents**: (Jayarajan, 2009) provides a qualitative evaluation based on 31 documents selected from the 20 Newsgroup dataset (Rennie, 1995). The number of documents are kept small in order to permit a manual analysis. The same dataset has been used to facilitate a comparison. The 31 documents are as follows, 4 from *comp.graphics*, 9 *talk.politics.guns*, 9 from *talk.politics.mideast*, 5 from *talk.religion.misc* and 4 from *rec.autos* whose names start with '3', '5', '7', '8' and '10' respectively are taken. The threshold values are set as follows: $\eta = 0.6$, $\sigma = 0.6$ and $\psi = 18\%$. The clusters obtained using our proposed algorithm are given in Table 2 . Bold-faced numerals signify different directories according to the 20 Newsgroups corpus. Comparing our results with that of (Jayarajan, 2009) we find that our algorithm gives better soft clusters. Even if the documents are from different mailing lists, they get clustered into the same cluster if any of the theme among these documents is common.

In all 23 clusters were generated by the algo-

Table 1: Clusters of similar lexical chains represented as word$_{sense\#}$ taking $\eta = 0.6$ and $\sigma = 0.6$

(*i*)	stock$_1$ capital$_1$ property$_2$ payment$_1$ loss$_3$ livelihood$_1$ profit$_1$ income$_1$ net$_3$ living$_4$ red$_4$ resource$_1$ belongings$_1$ support$_6$ gain$_4$
(*ii*)	fund$_2$ capital$_1$ property$_2$ payment$_1$ loss$_3$ profit$_1$ income$_1$ net$_3$ stock$_4$ red$_4$ resource$_1$ belongings$_1$ gain$_4$
(*iii*)	management$_1$ duty$_2$ battle$_1$ fight$_1$ obligation$_1$ conflict$_3$ proceeding$_1$ fight$_4$ assembly$_5$ struggle$_2$ combat$_2$ direction$_5$ gathering$_2$
(*iv*)	race$_3$ affair$_3$ final$_1$ meet$_1$ contest$_1$ athletics$_2$ match$_2$ occasion$_2$ picture$_6$ competition$_2$ flick$_2$ function$_6$
(*v*)	charge$_2$ tax$_1$ gift$_1$ capital$_1$ property$_2$ levy$_1$ payment$_1$ profit$_1$ income$_1$ net$_3$ belongings$_1$ gain$_4$
(*vi*)	jr$_1$ child$_2$ daughter$_1$ mother$_1$ baby$_1$ kid$_4$ issue$_6$ progeny$_1$ boy$_3$ girl$_3$
(*vii*)	estate$_2$ property$_2$ payment$_1$ income$_1$ acres$_1$ land$_7$ realty$_1$ belongings$_1$ gain$_4$
(*viii*)	horse$_1$ dog$_1$ bird$_1$ cows$_1$ mount$_1$ human$_1$ primate$_2$ canine$_2$ equine$_1$ ruminant$_1$ bovine$_1$ insect$_1$ fish$_1$ feline$_1$

rithm. We have chosen 3 different cases for analysis of our result. In case (1), the cluster number (*xx*) has all the documents from the same mailing list, *talk.politics.guns*. In case (2), the cluster number (*xxii*) has one document from the *talk.politics.guns* mailing list and rest from the same mailing list namely *talk religion.misc*. In case (3), cluster number (*viii*) has all the documents from different mailing lists: document 101677 from the *rec.autos*, document 76486 from *talk.politics.mideast* and document 82785 from *talk.religion.misc*.

Case (1): Cluster number (*xx*) has documents from the same mailing list. These documents contain lexical chains with words such as *person, government, law, politics, constitution and court*.

Case (2): Cluster number (*xxii*) has one document, 54269 from *talk.politics.guns* containing lexical chains with words such as *rule, duty, government and person*. Document 82784 and document 82785 include *religion, god, government, morality and rule*.

Case (3): Cluster number (*viii*) consist of 3 documents viz 101677, 76486 and 82785, from 3 different mailing lists. Let us look at some semantically similar lexical chains in these documents pairwise. The document 101677 contains lexical chains with words such as *money, amount, technology, market, hp, price, cost, trend*. Document 76486 contains lexical chains with words such as *money, tax, income, industry, investing, profit, expenses*. Therefore these documents do show some semantic relatedness. Now let us consider 76486 and 82785. The former document talks about words such as *government, religious, city, employer, resident, secular, community*. The latter includes the words such as *government, person, human, god, morality*. Again the semantic relatedness of these documents is evident. Finally let us consider documents 101677 and 82785. The former contains lexical chains with words such as *money, sports, competition, warrant, power, trend, fashion*. The latter contains lexical chains with words such as *game, won, race, playing, morality, force, choice*. It can be easily agreed that pairwise these documents have similar concepts. However, looking at the documents altogether we find that no common theme is evident from this cluster of 3 documents. One could change the thresholds, η, σ and ψ, perhaps make them more stringent and see if such clusters are suppressed. Another approach could be more difficult and formal, wherein some discourse analysis could be attempted on each of the documents to filter out such clusters.

(*iii*) **Results on Reuters corpus**: The Reuters Corpus' (Lewis, 1987) well known version Reuters-21578 "ApteMod" is generally used for text categorization. It is a collection of 10788 documents from the Reuters financial newswire service, partitioned into a training set with 7769 documents and a test set with 3019 documents. In this corpus, each document belongs to one or more categories. There are 90 categories in the corpus. The average number of categories per document is 1.235 and the average number of documents per category is about 148 or 1.37% of the corpus. The soft clustering algorithm was run on the training dataset of the Reuters Corpus. The threshold values were set as, $\eta = 0.7$, $\sigma = 0.75$ and $\psi = 18\%$. 96 soft clusters were obtained. Manual inspection of some of the clusters formed indicated that the soft clusters so formed were of good quality. Table 3 gives the documents in cluster 90 and cluster 96 and the set of categories asso-

Table 2: Results on a subset of 20 Newsgroups documents. **3**, **5**, **7**, **8**, **10** signify different directories as per the corpus.

Cluster No.	Documents
(*i*)	**10**1557, **10**1677, **5**4206, **5**4358
(*ii*)	**10**1557, **3**8406
(*iii*)	**10**1557, **5**3294, **5**4206
(*iv*)	**10**1574, **10**1597
(*v*)	**10**1677, **3**7261, **3**8406, **5**4206, **8**2782
(*vi*)	**10**1677, **5**3294, **5**4358, **7**5414, **7**6289, **8**2782, **8**2783, **8**2784, **8**2785
(*vii*)	**10**1677, **5**4206, **8**2785
(*viii*)	**10**1677, **7**6486, **8**2785
(*ix*)	**3**7261, **5**3294
(*x*)	**3**7261, **7**5414, **8**2782
(*xi*)	**3**7261, **7**6184, **7**6506
(*xii*)	**3**8400, **5**4152, **5**4819, **7**5933, **8**2785
(*xiii*)	**3**8400, **5**4269, **8**2782
(*xiv*)	**3**8406, **5**3354
(*xv*)	**3**8406, **5**4206, **7**6289, **8**2782
(*xvi*)	**5**3354, **5**4152, **5**4358, **5**4455
(*xvii*)	**5**3354, **5**4206, **5**4269, **5**4819
(*xviii*)	**5**4152, **5**4358, **5**4455, **8**2782
(*xix*)	**5**4206, **5**4269, **5**4358, **8**2785
(*xx*)	**5**4253, **5**4269, **5**4358, **5**4819
(*xxi*)	**5**4253, **7**6099, **7**6227, **7**6306, **7**6486
(*xxii*)	**5**4269, **8**2784, **8**2785
(*xxiii*)	**7**5933, **7**6227, **7**6506, **8**2781

Table 3: Results on Reuters Corpus.

Cluster No.	Documents	Document categories
(*90*)	11385, 12044, 12209, 13118, 14719, 2039, 3110, 356, 3864, 4345, 5053, 5525, 6217, 6592, 6603, 6914, 7968, 8009, 8097, 8189, 8780, 9139, 9371, 9372, 9906	ship, earn, gas, strategic metal, acq, grain ship, money-fx dlr, trade dlr money-fx
(*96*)	12044, 12209, 14719, 2039, 356, 3864, 3895, 5525, 6217, 6592, 6914, 6934, 7968, 9139	earn, gas, strategic metal, acq, tin

ciated with these documents. Cluster 90 includes lexical chains with words such as *refining, oil, gas, crude energy, uranium, reactor and mining*. It can be inferred that *mining and refining business* is the topical content of this cluster of documents. Cluster 96 includes lexical chains with words *transport, shipping, canal, coast, seaway, import, trade, adress, sales and profit*. This shows that this cluster of documents is about the *shipping industry and transport business*.

8 Conclusion

In this paper, a new approach for soft clustering of documents has been presented. This algorithm seeks to group the documents based on their semantic content. Lexical chains have been used as the document feature. The algorithm generates a graph using these lexical chains as nodes whose edge indicate semantic relatedness between the chains. Soft clusters of lexical chains are formed by finding maximal cliques of length one in this graph. Each maximal clique represents a particular theme (topic) in the corpus of documents. Clustering of documents is done by associating the documents to each maximal clique. This in turn results in soft clustering of documents as depending upon the topical variety in the document more than one cluster of lexical chains can get associated with the document. The efficacy of the algorithm has been demonstrated on three popular benchmarks.

Acknowledgement

The authors would like to thank Director, CAIR, for the encouragement and support given for this work.

References

Coen Bron and Joep Kerbosch. 1973. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM (ACM)*, 16(9):575–577, September.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32:13–47, March.

Frédéric Cazals and Chinmay Karande. 2008. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1):564–568, November.

Nelson Winthrop Francis and Henry Kučera. 1982. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin, Boston.

Dinakar Jayarajan, Dipti Deodhare, and Balaraman Ravindran. 2008. Lexical chains as document features. In *Proceedings of the Third International Joint Confernce on Natural Language Processing (IJCNLP 2008)*, volume 1, pages 111–117, Hyderabad, India, January.

Dinakar Jayarajan. 2009. Using semantics in document representation: A lexical chain approach. Master's thesis, Indian Institute of Technology Madras, Chennai, India, September.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics (ROCLING X)*, pages 19–33, Taiwan.

David D. Lewis. 1987. The reuters-21578 benchmark corpus, aptemod version (http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html).

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography*, 3(1):235–244.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City, Mexico, February.

Jason Rennie. 1995. 20 newsgroups dataset (http://people.csail.mit.edu/jrennie/20newsgroups/).

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal, Canada, August.