Document Clustering based on Semantic Notions

by

Muhammad Rafi

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

to

National University of Computer & Emerging Sciences

Department of Computer Science
National University of Computer and Emerging Sciences
Karachi, Pakistan
(December 2016)
It is hereby declared that the work presented in this thesis is my original work and does not contain any material that is already published by another person except where properly cited and referenced. The submitted text contains a description of my research work that I have carried out during my PhD studies and does not include any material that has been submitted to qualify for the award of any other degree or diploma in any university or institution.

Signature _____________________

Name_________________________
Plagiarism Undertaking

It is hereby declared that the work presented in this thesis is not copied or plagiarized from any other source or existing literature, except where due references and citations are made.

Signature _____________________

Name ________________________
Abstract

The exponential growth of electronic documents, in both proprietary and public information systems, pose new challenges in finding relevant information from these large repositories. Document clustering is a specialized technique that has found its niche in effectively browsing, filtering, managing and summarizing these collections. Document clustering process has three distinct steps: (i) document representation, (ii) computation of pair-wise document similarity, and (iii) application of clustering algorithm. Document clustering methods are very sensitive to document representation schemes. Conventionally, document representations are based on extracting simple features such as terms/n-grams/frequent words/sequences from the documents that can be used as meta-descriptors for documents. These features reduce the dimensionality of the problem but simply fail to capture the semantics of the text in a transformed compact representation. These representations completely ignore the order and relationships among words/features. Documents written in human languages generally contain a context and use of words are mainly dependent on the same context. Motivated by this a novel document representation scheme that first extracts lexical chains from the documents and exploits topic maps structure for the lexical chains is proposed. The scheme takes advantage of lexical cohesion structure along with topic map relationships to get a semantic based representation of document. Topic Maps (TM) is an international standard for codification of knowledge. Moreover, a good similarity measure is essential for the clustering task. The similarity function should make use of semantic relationship among features (lexical topics) to provide a viable clue for relatedness between any pair of documents. A similarity function based on lexical chain similarity and frequent common tree patterns extracted from the topic maps of documents is defined. Hence these patterns (hierarchical lexical topics with different granularity) also inherently capture semantics in similarity calculation. An extensive set of experiments on four publicly available document datasets is performed. The evaluation measures like F-score, purity and entropy clearly established that the proposed approach is better than traditional document clustering approaches.
## Contents

Abstract .................................................................................................................................................. iv  
Table of Figures .................................................................................................................................... viii  
List of Table ............................................................................................................................................ ix  
List of Publications .................................................................................................................................. x  
Peer Reviewed - Journal Papers ........................................................................................................... x  
Peer Reviewed – Conference Papers .................................................................................................. x  
Chapter 1 ............................................................................................................................................... 11  
1.1 Formal Description of the Problem ............................................................................................. 15  
1.2 Application of Document Clustering ........................................................................................... 16  
1.3 Issues & Challenging ................................................................................................................... 19  
1.4 Background and Motivation ....................................................................................................... 21  
1.5 Contribution of the Thesis .......................................................................................................... 24  
1.6 Thesis Organization ..................................................................................................................... 27  
Chapter 2 ............................................................................................................................................... 29  
Background and Related Works ............................................................................................................ 29  
2.1 Introduction .................................................................................................................................... 29  
2.2 Document clustering ....................................................................................................................... 30  
2.2.1 Document clustering using simple features ............................................................................ 31  
2.2.2 Document clustering using semantic features ........................................................................ 34  
2.2.2.1 Semantics through WordNet ............................................................................................ 34  
2.2.2.2 Lexical Chains .................................................................................................................... 36  
2.2.2.2 Topics and Topic maps ...................................................................................................... 37  
2.2.2.3 Semantic Similarity ............................................................................................................ 38  
2.2.3 Document clustering –Evaluation ............................................................................................ 39  
2.2.4 Research Gap – Document clustering ...................................................................................... 39  
2.2.5 Research Contribution- Document clustering ......................................................................... 40  
Chapter 3 ............................................................................................................................................... 42  
Document Representation .................................................................................................................... 42  
3.1 Document Pre-processing ........................................................................................................... 43  
3.2 Simple Boolean Model for Documents ....................................................................................... 44  
3.3 Vector Space Model .................................................................................................................... 45  
3.4 Suffix Tree Model ........................................................................................................................ 46
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGURE 1</td>
<td>SUFFIX TREE MODEL FOR DOCUMENTS</td>
<td>47</td>
</tr>
<tr>
<td>FIGURE 2</td>
<td>GRAPH MODEL FOR EXAMPLE DOCUMENTS</td>
<td>49</td>
</tr>
<tr>
<td>FIGURE 3</td>
<td>TOPIC MAPS MODEL FOR EXAMPLE DOCUMENT D1</td>
<td>52</td>
</tr>
<tr>
<td>FIGURE 4</td>
<td>LEXICAL TERMS BASED TOPIC MAPS MODEL FOR DOCUMENT D1</td>
<td>53</td>
</tr>
<tr>
<td>FIGURE 5</td>
<td>DIAGRAM FOR MTS-DC APPROACH</td>
<td>71</td>
</tr>
<tr>
<td>FIGURE 6</td>
<td>F-MEASURE FROM THE EXPERIMENTS</td>
<td>96</td>
</tr>
<tr>
<td>FIGURE 7</td>
<td>PURITY FROM THE EXPERIMENTS</td>
<td>98</td>
</tr>
<tr>
<td>FIGURE 8</td>
<td>ENTROPY FROM THE EXPERIMENTS</td>
<td>101</td>
</tr>
</tbody>
</table>
List of Table

TABLE 1-LIST OF SYMBOL .............................................................................................................................. 16
TABLE 2- COMPARISON OF DIFFERENT DOCUMENT REPRESENTATION MODELS ............................................ 54
TABLE 3- EXAMPLE DOCUMENT FROM 20 NEWSGROUP DOCID- 20361 .......................................................... 79
TABLE 4- KEY CHARACTERISTICS OF DATASET .......................................................................................... 86
TABLE 5- CHARACTERISTICS OF SELECTED DATASETS ............................................................................. 87
TABLE 6- F-MEASURE FROM THE EXPERIMENTS ...................................................................................... 97
TABLE 7- PURITY FROM THE EXPERIMENTS .............................................................................................. 99
TABLE 8- ENTROPY VALUES FROM THE EXPERIMENTS ............................................................................ 100
List of Publications

Peer Reviewed - Journal Papers


Peer Reviewed – Conference Papers

Chapter 1

Introduction

Text is ubiquitous and textual data in electronic form is being generated in huge quantities everywhere. It is common to have hundreds of terabytes of textual data in public and proprietary systems. Information retrieval from these textual repositories is an arduous, and challenging, but unavoidable activity. At the same time, these document collections contain knowledge that can be very useful for the organization. Clustering is a useful data mining technique that is unsupervised and can automatically arrange the given collection of objects into meaningful sub-collections called clusters. The use of clustering for document repositories enables efficient search, effective management, smart navigation, proficient filtering, and concise summarization. Clustering in this perspective is specifically termed as "Document Clustering".

Document clustering is a specialized data clustering problem where the objects are in the form of documents. These documents are generally written in human natural languages, and can appear in various forms and formats. The objective of the clustering process in this setting is to group the documents which are similar in nature into separate groups i.e. clusters. The difficult part is to learn how many groups (classes or clusters) of such documents exist in the dataset (collection). Exploring, analyzing, and correctly classifying the unknown nature of
textual data in a document without supervision is the major requirement of document clustering method.

Problem Statement

"Given a large collection of heterogeneous documents, the task of document clustering is to segregate the collection automatically into different homogenous groups called clusters."

Document clustering is a multifaceted problem. The task of clustering encompasses many tasks such as selecting a suitable abstract model for document, defining an effective computational function for computing pair-wise document similarity, identifying number of distinct groups of documents in the collection (finding number of clusters) and meritoriously assigning documents to identified clusters.

Document clustering generally comprises of three distinct steps:

(a) **Document Representation**- creating a succinct meta-representation for documents. There are two main advantages of doing this (i) dimension reduction and (ii) efficiency.

(b) **Similarity Measure or Function**- defining a similarity measure that assigns a numerical value to every pair of document. It is a major computational burden for the challenging task of clustering. This function assigns a score between (0-1); a score of 1 means the pair of documents is identical whereas a score closes to 0 means the pair of documents is completely different. Consequently, the pair of documents with a 0 score would be part of different groups in the final clustering arrangement.

(c) **Clustering Algorithm** that exploits the document representation as well as the similarity values produced for the entire datasets of documents in terms of pair-wise values.
The document representation model is a scheme through which one only retains the favorable features from documents for representation. The clustering process is highly sensitive to this representational scheme. A document written in human language may be thought of in terms of a large number of diverse features like words, phrases, sequences, sentences, paragraphs, chapters, etc. The traditional document representation method generally uses these features independently of each other. Documents written in natural languages generally have a context behind the text. The writer of the text usually selects these words, sequences of words, pair of words, pair of sentences, etc based on the idea of the very same context. We believe that the use of these features is not at all independent. A document representation that can exploit this dependency will be a strong prerequisite for better document clustering. Moreover, a major computational burden, while performing document clustering. It should assign relative scores like high score to relatively higher similar document pairs and low score to distinctly different pair of documents. It should map this score depending upon the degree of semantic similarity (similarity between the meanings behind text) for any pair of documents. Customarily, a vector based representation is used for documents. In this representation, each document is placed in a high-dimensional features (term) space. Every document is a vector of terms appears in that document, terms can be used with some weighting criteria such as term frequency-inverse document frequency (tf*idf). A similarity measure in vector space model (Aggarwal & Reddy, 2013) that is generally very effective is cosine similarity. It is defined as the angle between the vector representations of the two documents in feature space. If the angle is small the documents are close to each other that is they share more common features. It is also a symmetric measure and a metric. The values of a cosine similarity are between 0 and 1. Another challenge is to incorporate the semantic features of the text into the similarity calculation. A similarity measure that exploits the semantic features of the text in similarity calculation will definitely account for the better semantic scores.
There are two major clustering approaches (i) Partitioned clustering (ii) Hierarchical clustering. These two are very effectively utilized in performing document clustering. The key differences between these two are in terms of time complexity, input parameters, and the output cluster arrangements. Typical partitioned clustering divides the data into several subsets. It is an NP-complete problem and checking all possible subsets is computationally hard. Hence, greedy heuristics are used (Chim & Deng, 2007), (Farahat & Kamel, 2011; Kalogeratos & Likas, 2011, 2012; Li, Chung, & Holt, 2008; Rafi, Shaikh, & Farooq, 2011). The running time of greedy heuristics is proportional to the number of documents in the collection. One disadvantage of partitioned clustering is the parameter requirement; it has to provide an initial number of clusters and seeds for clustering. Partition based clustering algorithms are also called centroid-based clustering. One popular algorithm in this class is K-Means clustering. There are various variations in K-Means algorithm, such as, k-medoids, k-medians, k-means++ and Fuzzy c-means. The hierarchical clustering algorithms need a document to document similarity measure; hence, it is computationally complex. It can produce a hierarchical output which can be cut to the level where the actual number of clusters is required. These algorithms are also termed as "connectivity based clustering". One of the most popular algorithm in this class is "Hierarchical Agglomerative clustering-(HAC)". There are several variations of this algorithm available in literature mainly based on different criteria of distances and links among the objects to cluster. The details on HAC is presented in chapter on clustering algorithms (Chapter 5).

Hierarchical document clustering is more popular in offline settings. A density based clustering tries to cluster objects based on higher density of their occurrences. This class of algorithms can learn a cluster arrangement of arbitrary shape.

Cluster validation is the process of evaluation of the result of a clustering algorithm. Clustering process is unsupervised; hence its evaluation is a complex phenomenon. There are three main kinds of cluster evaluation (cluster analysis):
(i) Internal evaluation is a process in which a clustering algorithm is evaluated on the data/objects on which it is performing clustering.

(ii) External evaluation is a process in which a clustering algorithm is evaluated on the data/objects that is not used for clustering. These are the benchmark datasets, often human created, which consist of a set of pre-classified items.

(iii) Relative evaluation is validation criteria for two different clustering algorithms.

In document clustering, generally external evaluation is used to perform cluster analysis (Lee, Pincombe, & Welsh, 2005; Tonella, Ricca, Pianta, & Girardi, 2003; Zhao & Karypis, 2002).

### 1.1 Formal Description of the Problem

Consider, a large collection $D$ of $N$ documents in which each distinct document is denoted by $d_i, 1 \leq i \leq N$. Let the clustering algorithm produces $C_p = (c_1, c_2, \ldots, c_p)$, where $p$ is the number of generated clusters using the document representation model along with the similarity measure defined on respective features. Each document is represented as a feature vector of features extracted from the document. Hence $f_{vi} = (f_{v1}^i, f_{v2}^i, \ldots, f_{vn_i}^i)$ is a $n$-dimensional feature vector in $\mathbb{R}^n$.

A similarity function $\psi(d_i, d_j)$ is defined for representing a value between (0-1) for every pair of documents $i$ and $j$. Let the ground truth for the given dataset comprise of $K_q = (k_1, k_2, \ldots, k_q)$ number of clusters. The performance of clustering algorithm generally measures how close the two clustering arrangements, $C_p$ and $K_q$ are. Lexical chain is a sequence of words that collectively used to convey the main idea of a document. A description of symbols used in this manuscript is describe in Table 1.
1.2 Application of Document Clustering

The ever increasing textual documents in both public and proprietary systems are rich in actionable knowledge contents. The challenge is to have a fast and efficient way to sift through this collection and to come up with the content based on the user's information needs. Document clustering is an unsupervised machine learning technique that offers many benefits to the users in the same problem's perspective. According to the cluster hypothesis the documents that are clustered together through the document clustering process are relevant for the information need of the user. Most of the applications of clustering exploit the very assumption of this clustering hypothesis. Following are some of the widely used applications of document clustering:
• **Information filtering** - document clustering creates succinct document clusters (Ungar & Foster, 1998; Xue et al., 2005). Hence it masks out the irrelevant documents. Modern search engines return a ranked list of web pages represented by page excerpts called the web snippets, these are generally key sentences from the documents. The ranking is computed according to some relevance criterion (Käki & Aula, 2005).

• **Search Results** - document clustering can be used to cluster search results (Swen, 2005) in a better visual presentation. **Vivisimo** - a meta-search engine is an example of document clustering for effective web search (Koshman, Spink, & Jansen, 2006). In modern search engine, a brief summary of the document is presented in hierarchical manner for better comprehension of the results (Ferragina & Gulli, 2004).

• **Finding similar documents** - document clustering can be used to find similar documents (Zamir & Etzioni, 1999). There is a growing number of applications that use document features along with synonyms, etc. to relax and find similar documents from the web (Saraçoğlu, Tütüncü, & Allahverdi, 2007).

• **Automatic organization of document repositories** - a better way to explore the content of large document repositories is to perform clustering repeatedly based on the phenomenon of scatter–gather (Cutting, Karger, Pedersen, & Tukey, 1992).

• **Duplicate Detection** - one of the challenging problems is to detect duplicate or nearly duplicate documents from the web. Document clustering can be very helpful in such cases (Chowdhury, Frieder, Grossman, & McCabe, 2002).

• **Summarization** - document clustering can be used to create summaries of a document collection. After clustering, one can look for clusters labels, and these terms better describe documents in a clusters (ogly Alguliev & ogly Alyguliev,
A summarization module can work on top of these labels to produce a summary for the cluster.

- **Visual insight into documents** - document clustering can also be used as a visual aid to see how features (Wise et al., 1995) among documents are related. This is an emerging application of document clustering.

- **Exploratory analysis** - document clustering can be very useful in exploratory analysis of large document repositories (Dubes & Jain, 1980). Through these one can learn different topics and their distribution. An interactive and hierarchical clustering approach is very beneficial in this analysis (Ferragina & Gulli, 2004).

- **Text compression** - document clustering can be used to perform compression of textual collection. Features of the textual documents can be used to perform compression (Zobel & Moffat, 1995).
1.3 Issues & Challenging

Document clustering is a very challenging problem as it has many dimensions and many factors.

- Documents are generally written in human language and they usually have some context. The use of words is mainly based on this very context. It is observed that on average 600-1000 unique words or terms are used in a reasonably sized document. Using term (word) as a feature is quite challenging in itself. A vector based representation of these documents introduces very large matrix which is highly sparse. It is extremely desirable that this should be reduced to some tractable limit. Hence, dimension reduction is usually applied to compactly represent the document.

- Text is ubiquitous and is highly unstructured. There are numerous possible features. Both feature extraction and feature generation are possible for text. Some of the widely used text features that are extracted from a document are words, phrases, words sequences, and concepts. These features are independent of each other and they do not express the relationships among the words. Thus, the semantics is missing in both document representation and similarity calculation. The feature generation for a document is also computationally expensive. The features like dependency graph, graph-of-words, and association of terms are hard to generate and the little semantics that they offer does not justify the effort required to generate them. The answer to this problem is to have a feature which is easy to generate or extract and at the same time offers better semantics.

- A major computational step in document clustering is the computation of a similarity function. The clustering task would not be possible without this. Defining an appropriate similarity measure for the selected representation is a difficult task. A
similarity function should be easy to compute, and it should assign a score between 0 and 1 where a value of one means practically identical documents, whereas the value zero means totally different documents. If a document representation is semantically viable, the similarity function should also exploit the semantics in similarity calculations.

- Another challenging aspect of this problem is to identify exactly how many distinct groups of documents exist in a collection. It is unclear from the document dataset that how many clusters would be formed from the clustering process. A \textit{cluster tendency} test is generally applied to see if the data is actually favorable for clustering or not (Anastasiu, Tagarelli, & Karypis, 2013).

- Document clustering can be crisp, where the process of clustering assigns each document to exactly one cluster (group). It can be fuzzy, where a document may belong to more than one cluster (Yang, 1993). A cluster membership value describes how close a document is related to a cluster (Jawahar, Biswas, & Ray, 1997).

- Document clustering is an unsupervised approach, and it is always challenging to know if a solution to clustering is really a valid one. Cluster validation is a process through which a clustering algorithm is validated (Halkidi, Batistakis, & Vazirgiannis, 2001). The general approach is to first perform document clustering on the document dataset without seeing the ground truth of the dataset (actual number of clusters and their memberships). When a result of a clustering algorithm is produced, it is then compared with the ground truth. If the cluster arrangement produced by the algorithm is close to the arrangement of the ground truth, the algorithm is considered good. There are three main types of cluster validation (i) Internal (ii) External and (iii) Relative.
1.4 Background and Motivation

Clustering is an old problem and has a long history (Hansen & Jaumard, 1997). It is an unsupervised machine learning method that tries to segregate a given object collection into distinctive sub collections with high intra-cluster similarity and low inter-cluster similarity. Clustering phenomenon is extensively studied [For details see (A. K. Jain, Murty, & Flynn, 1999) and (Anil K Jain, 2010)]. Document clustering is specialized data clustering where objects are textual documents. Document clustering as an unsupervised approach, clusters the document repository into meaningful smaller and more manageable sub-collections. These resultant sub-collections contain high intra-cluster similarity (the documents in a single cluster are mainly similar), and low inter-cluster similarity (documents in two sub-collections are largely dissimilar). It has found its niche in management of large document repositories (Cutting et al., 1992). These document collections are now a common phenomenon for every organization and clustering(Vdorhees, 2017) so help in finding relevant information quickly. There are three distinct steps in the document clustering process (i) Document representation (ii) Similarity function and (iii) clustering algorithm. Textual documents offer big challenges (Tan, 1999) as text mining is more complex as compared to data mining (Hotho, Nürnberger, & Paaß, 2005). Document representation is a text mining problem (Anastasiu et al., 2013) which requires a compact and semantically rich document representation. Traditionally, documents are represented by extracting features from documents like terms, phrases, sequences, concepts, ontology and dependency graphs of the text (N. O. Andrews & Fox, 2007; Fung, Wang, & Ester, 2003; Hotho, Staab, & Stumme, 2003; Li et al., 2008; Steinbach, Karypis, & Kumar, 2000; Y. Wang, Ni, Sun, Tong, & Chen, 2011). These representations,
except for dependency graphs of text, do not express the meaning and order of the words used to write the content of the text. Documents written in human languages generally have a context and the use of words is mainly dependent on the context.

A representation that captures relationships between the words used in documents will be better. Lexical chain is a sequence of related words generally used to convey the main idea in written/oral communication. These lexical chains may span short or long distances in the text. Lexical Chains are generally exploited in the semantic relatedness of words in a document and are also independent of the grammatical structure of the text. We strongly believe that in order to take advantage of semantics of textual documents in a representation scheme we need to collect these chains from the document first. We propose a simple yet semantically rich representation for documents based on extracting lexical chains and linking these chain words with topic maps relationships. Topic Maps is becoming an international standard for knowledge representation (Pepper & Moore, 2010). It is based on formal model and modern information management. The use of topic maps is escalating in the projects that perform enterprise information integration, knowledge management, digital libraries, and web-based information integration and management (Steiner, Essmayr, & Wagner, 2001). The subject-centric nature of topic-maps paradigm is what we perceive as the very nature of human-centered clustering process. We wish to exploit the very nature of this subject centric information in topic maps for document representation. Topic maps offer an out of the box meta-model for enabling subject based information management. We propose a document representation model by inferring the topic maps based data-structures from the documents. It provides us two benefits (i) reducing the document to a compact form (dimension reduction) and (ii) semantic based representation of the topic terms from the documents. The concept has already proven effective (Rafi & Shaikh, 2013). One of the major computational burdens in performing the task of document clustering is similarity measure/ function. We believe that a similarity measure that
exploits the semantic structures of the document representation for inferring a similarity score can produce clusters of high quality. A similarity function assigns a score between (0-1) for a pair of documents, and a similarity function should be mapped to metric space (Zezula, Amato, Dohnal, & Batko, 2006). In previous works, researchers proposed similarity measure based on different distance in vector space of features extracted from documents (Huang, 2008; Rafi & Shaikh, 2013; Strehl, Ghosh, & Mooney, 2000). We have identified that there is a serious limitation of document representation and similarity function for the task of document clustering (Rafi & Shaikh, 2013; Rafi et al., 2011). There are two main challenges for performing semantically viable clustering: (i) a representation of textual document that inherently captures the semantics, and (ii) a similarity function that also exploits these semantically rich representational features for similarity measure.

This thesis proposes a document representation based on topic maps. A document is transformed by only taking features from the inferred data structures of topic maps from the text of the document. The thesis also proposes a novel similarity measure that is based on lexical chains and topic maps based common tree patterns extracted from documents. Hence, these patterns, (sub-trees) also inherently capture semantics in similarity calculation.

There are two major categories of clustering algorithms for performing document clustering (i) partition, and (ii) hierarchical clustering. There are many subtypes of algorithms exist in each of this category. Hierarchical clustering works repeatedly by either merging smaller clusters into larger ones, or by splitting larger clusters into smaller ones (Reynolds, Richards, de la Iglesia, & Rayward-Smith, 2006). The idea on which this splitting and merging is performed produce different types of algorithm in this category. The hierarchical algorithms at the end produce a dendrogram or tree like structure. On the other side, partition clustering algorithms directly consider the data object to separate the dataset into disjoint sets (clusters). The algorithm tries to optimize criterion functions so that the disjoint sets should have high intra-
cluster similarity and low inter-cluster similarity. The k-Mean algorithm belongs to partition clustering type. k-Mean clustering algorithm is a simple and fast algorithm and is considered as an old and widely applicable algorithm for the last 50 years (Anil K Jain, 2010).

The key differences between these two are in term of time complexity, input parameters, and the output cluster arrangements (Aggarwal & Reddy, 2013).

1.5 Contribution of the Thesis

There are several contributions the thesis under study makes to the domain of document clustering. Document clustering is no-doubt a new-fangled, growing-big, complex and emerging problem with the advent of large document repositories in both public and proprietary information systems. An efficient, scalable, robust, and semantically viable document clustering algorithm is a dire need of organizations for better management, smart navigation, efficient filtering and concise summarization of information needs from these hefty document collections.

Following are the contributions that the thesis under study makes to the field:

- A new document representation model is proposed based on lexical chains and linking the words in a chain through topic maps related structures. It is a novel scheme that takes advantage of lexical cohesion along with topic maps structures. Lexical chain is a sequence of words generally used in combination to convey the central idea of a written document. Topic map is an emerging standard for information management. The use of topic maps is escalating in the projects that perform enterprise information integration, knowledge management, digital libraries and web-based information integration and management. The subject-centric nature of topic-maps paradigm is what
we perceive to be the very nature of human central document clustering process. We believe that a topic map based representation of documents first reduce the dimensionality of text representation problem and at the same time the structures of the topic map offers better semantic in the representation of document. Hence, the subject centric nature of topic maps can be better utilized for the document clustering task.

- A new document-to-document similarity measure is proposed. It is an essential requirement for document clustering. An earlier work proposed a simple similarity function that exploits common topics, topic tags, and tag-values for numerically associating a score between 0 and 1 to a pair of documents. A novel similarity function based on lexical chains and topic maps construct that contains lexical terms from the documents, is proposed. The common tree patterns (sub-trees) also inherently captures semantics in similarity calculation via common links and hierarchy of topic maps. Hence, this similarity function offers better semantic value for a pair of documents.

- The algorithms are designed, developed, and practically implemented for lexical chain, topic extraction from the documents. A similarity measure function to measure similarity between a pair of documents is proposed that exploits the proposed representation of document. Hierarchical Agglomerative Clustering (HAC) is used for final clustering. A complete object oriented implementation for specific problem of document clustering is implemented in Java programming language. The implementation is scalable, extensible and easy to understand.

- An extensive series of experimentation is performed to demonstrate the effectiveness of the proposed method named as Multi-Layer Topic Maps based Semantic-rich hierarchical Document Clustering (MTS-DC). These experimental studies offer a deep insight into the problem of document clustering.
- The proposed MTS-DC is compared with a number of baselines which are specifically designed to evaluate the proposed approach. Also, some of the recently proposed methods of document clustering are compared with our proposed method. The algorithms such as Frequent Itemset-based Hierarchical Clustering (FIHC) (Fung et al., 2003), text document Clustering based on Frequent Word Sequences (CFWS) (Li et al., 2008), and Document clustering based on Topic Maps (TMHC) (Rafi et al., 2011) are implemented and compared with MTS-DC. It is experimentally shown that the proposed method clearly out-performs these comparative approaches to document clustering. The proposed MTS-DC, produced better purity, entropy and F-measure on standard text mining datasets, when compared with these methods. Thus, our approach offers a better way to manage large documents collection which is now a common problem in organizations.

- There are several directions of future work in the realm of proposed approach which offers a great opportunity for new researchers.
1.6 Thesis Organization

The thesis is organized into nine chapters. Chapter 1, which is the current chapter, is an introduction.

Chapter 2: Background and Related Works - it presents a comprehensive review of classical and state of the art methods, algorithms and approaches to the task of document clustering. The survey in the chapter is divided into three sections (i) document representation (ii) similarity functions and (iii) clustering algorithms. In addition, techniques for preprocessing of natural language based document collections, extraction of meta-descriptors (features) from these documents and effective computation of similarity functions for every pair of documents are discussed in detail. The comprehensive spectrum of various suggested clustering algorithms and their usage for document clustering are discussed. This chapter also presents the research gap that is resolved by the research proposed in this thesis.

Chapter 3: Document Representation - the chapter discusses many proposed document representation and their merits and demerits. It also explains how a topic maps based representation of documents is different from these traditional document representations.

Chapter 4: Documents Similarity - a similarity measure is a mathematical function that assigns a score between (0-1) to a pair of documents. It is a major computational step of document clustering process. The chapter further discusses the pros and cons of different similarity measures proposed in literature. A topic maps based similarity measure proposed in this thesis is discussed, a running example is presented. Our claims that this similarity measure implicitly use textual semantics is established.
Chapter 5: Document Clustering Algorithms - this chapter presents various document clustering algorithms proposed in literature. It outlines the various requirements of a document clustering algorithm.

Chapter 6: Document Clustering on Semantic Notions – Our MTS-DC is proposed and a detail running example is used to demonstrate the actual approach. The proposed algorithms are presented.

Chapter 7: Experimental Studies - this chapter talks about the experimental setting, datasets used, evaluations performed in order to demonstrate the effectiveness and performance of proposed document representation, similarity measure and document clustering algorithm. The evaluation also provides two baselines for the assessment of different approaches used in the experiments.

Chapter 8: Results and Findings - it presents a comprehensive review on experimental results of state of the art algorithms for document clustering, it discusses on how baseline is set forth, how improvements on evaluation are achieved and insights of the results obtained.

Chapter 9: Conclusion and Future works - this chapter summarizes the finding of this thesis and discuss the potential directions for the future research.
Chapter 2

Background and Related Works

2.1 Introduction

Data clustering is an unsupervised approach for discovering valuable knowledge from the data. It is quite an old techniques (Hansen & Jaumard, 1997) and very vast, intensive and ever growing literature on the subject can be found across a wide variety of discipline (Anil K Jain, 2010). A detailed survey on data clustering can be found in (A. K. Jain et al., 1999). The data clustering process autonomously group similar data into one group (cluster), while simultaneously grouping dissimilar one into different groups. The core of the process is to implicitly identifying the hidden patterns in the data and to separate them out into distinguishable groups (clusters). Clustering (Xu & Wunsch, 2008) is useful in a lot of different setting and a variety of application areas. Clustering is a preferred and known technique in machine learning, information retrieval (Cutting et al., 1992), grouping and segmentation, pattern analysis, data mining and data analysis (Aggarwal & Reddy, 2013). It is challenging as it is a combinatorial problem, the complexity increases as we increase the size of the clustering task, the number of objects to cluster. It can also be seen as an optimization problem.
2.2 Document clustering

Document clustering is a specialized clustering problem in which the data objects to be clustered are documents. Document clustering (Anastasiu et al., 2013) as an unsupervised approach, clusters the document repository into meaningful smaller and more manageable sub-collections. These resultant sub-collections contain high intra-cluster similarity (means the documents in a single cluster are mainly similar), and low inter-cluster similarity (documents in different sub-collections are largely dissimilar). It has found its niche in management of large document repositories. These document collections are now a common phenomenon for every organization. Exploring, analyzing, and correctly classifying the unknown natures of data in a document without supervision is the major requirement of document clustering method.

Document clustering algorithm 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 (Shafiei et al., 2007) is very sensitive for the task of document clustering. We first discuss a traditional way of performing document clustering by extracting simple features from the documents. These document clustering algorithms mainly use simple document features like: words, phrases, and sequences from the documents to perform cluster (Fung et al., 2003; Li et al., 2008). These algorithms generally apply simple feature extraction techniques that are mainly based on feature counting and frequency distribution of the features to decide about the relatedness among documents.

All these approaches thus, cannot cater to the meaning behind the text (words). These techniques simply perform clustering independently of the context. We next categorize the
document clustering algorithms that try to utilize semantics (meaning) from the documents into the clustering process. These clustering approaches generally use some external knowledge repositories to infer the implicit relationships from the terms (simple features) to extract the semantic information. There are many general clustering algorithms that are extended for performing document clustering. There are two main categories of document clustering algorithms (i) partition based- in which the document collection is separated into a distinct groups (Anil K Jain, 2010) and each document fall into a single cluster, (ii) hierarchical- in which documents are arranged into an overlapping hierarchy (Fung et al., 2003; Reynolds et al., 2006). The work by (Steinbach et al., 2000) compares and suggests a new efficient algorithm in between these two categories.

2.2.1 Document clustering using simple features

Document clustering is very sensitive to documents representation. Traditionally, document clustering algorithms mainly uses features like: words, phrases, and sequences 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, to build the document representations (Shafiei et al., 2007). The very first work in this category was scatter/gather (Cutting et al., 1992), which demonstrated the effectiveness of clustering for a document collection search and exploration(Vdorhees, 2017).

Another seminal work in this type of clustering is from (Steinbach et al., 2000). An approach called Frequent Itemset-based Hierarchical Clustering (FIHC) (Fung et al., 2003) is proposed for document clustering based on the idea of frequent item-sets mining technique that are very useful in market basket analysis. The approach considers each term of the document as an item for shopping and different sets of documents (clusters) share some frequent terms. A frequent itemset is a set of words that occur together in some minimum fraction of documents that can
be grouped as a cluster, this is the main idea of this approach. The approach has some clear advantages over previous proposed techniques for document clustering. First of all, it helps to reduce the dimensionality of the problem and the clusters can have a good description from these frequent terms. It was highly competitive approach at the time it was proposed. There are numerous variations of this itemset mining methods that have been proposed for clustering (C.-L. Chen, Tseng, & Liang, 2010), and this approach is also very effective for web clustering (Zhuang & Dai, 2004). The introduction of fuzzy frequent item sets for clustering (C.-L. Chen et al., 2010) was also an extension of this work. Another work that uses frequent words for document clustering is from (Li et al., 2008).

Clustering based on Frequent Word Sequences (CFWS) treats the text document as a sequence of words, and the frequent word sequences among documents are used as an implicit feature to perform clustering. The approach reduces each document to a compact frequent sequence of words (sequence length=2), it caters the sequential occurrences of words in the document to implied the semantics richness in the representation. It uses a generalized suffix tree from all the compact documents to get the frequent word sequences that are shared among different set of documents. The set of documents that contains more common such sequences are grouped to form a candidate cluster. The researchers proposed two variations of this algorithm: one that uses common frequent sequences of words and other that uses common frequent word meaning sequences. The later uses an external lexical database WordNet (Miller, 1995) to annotate the word with their meaning to cover the word meanings, such as synonymy, polysemy, and hyponymy/hypernymy. Their experimental studies have shown an improvement in F-measure for both CFWS and CFWMS over FIHC. A very similar approach (Chim & Deng, 2008) uses phrases as a feature for document clustering, by defining a similarity measure over common phrases. Although, these approaches use phrases or order of words in representation of documents but still their results are fallible on semantics of the clusters produced. These
techniques simply perform clustering independent of the larger context. Document written in
human language contains a context and the usage of words are largely dependent on this
context. The feature selection (Alelyani, Tang, & Liu, 2013; Dhillon, Kogan, & Nicholas,
2004) methods are also produce some good results for the document clustering task.

Document clustering based on Topic maps (TMHC) is another approach that exploits the
lexical semantics for the task of textual document clustering. Topic maps (Pepper & Moore,
2010); (Garshol, 2002) is a standard for knowledge representation and exchange. It is a similar
concept to representing back-of-the-book index structures. The main idea is to facilitate the
searching process, integration of information and find ability of information pieces. It is a meta-
model (Lachica & Karabeg, 2008) which is becoming very popular for internet applications
(Steiner et al., 2001). The approach based on topic maps first transforms the document into a
topic maps based representation. It also defines a similarity measure (Rafi & Shaikh, 2013) on
the topic maps data structures like: topic tag, association and tag-values. The algorithm reports
improvement over FIHC and CFWS on three standard text mining datasets. It uses an online
tool Wandora (Team) for creating topic maps for a given document. There are some very well-
known issues with the application of topic maps (Rath, 1999) such as describing, maintaining
and using topics. Checking consistency of the topics, constraints on topic and automatic
integration of topics are some of the challenging aspect of topic maps. The work suggested that
topic maps can very quickly created for each document (Küster & Moore, 2008; J. Park,
Hunting, & Foreword By-Engelbart, 2002) hence it will not take much time for a web scale
clustering.

Dependency Graph based Document Clustering (DGDC) (Y. Wang et al., 2011) is a method in
which each document is represented as a dependency graph where word corresponds to nodes.
It is used as a meta-descriptor of the document. The semantics between pairs of words are
captured by using edges between nodes. A novel similarity measure is also proposed for
calculating the similarity of documents based on their corresponding dependency graphs. The approach has the following steps: (i) construct a dependency graph for each document, (ii) calculate the similarity of each pair of documents based on the novel similarity measure (iii) generate the final clusters of documents by using Group-average Agglomerative Hierarchical Clustering (GAHC) algorithm.

2.2.2 Document clustering using semantic features

The document clustering approach using simple features typical does not produce high quality clusters as it only considers the features independent of each other i.e., its dependency is only to adjacent terms like in phrases or sequences. In order to use semantic features from documents there are two major approaches: (i) using an external dictionary of lexical related terms, and (ii) using a computational language algorithm to discover relationships among simple features. The first approach is easy to use in the clustering process and does not require much computational efforts but its major drawback is the coverage of the features. The second approach is computationally very expensive and there are many algorithms to deal with the relationship issues of simple terms that appear in a document.

2.2.2.1 Semantics through WordNet

WordNet (Miller, 1995) is one of the most widely used and largest lexical databases of English language. It can be considered a dictionary as well as a thesaurus. It contains a collection of terms from domains that are stemmed into their lexical categories. It has been used since its introduction for a number of different purposes such as semantic similarity measure between words, word-sense disambiguation, information retrieval, automatic text classification/clustering, automatic text summarization, machine translation and automatic crossword puzzle generation. In this approach the WordNet 3.0 is used which contains 158000
terms, 125000 senses, and 210000 pairs of term-sense. It defines synsets as groups of nouns, verbs, adjectives and adverbs. It is about 12 megabytes in size. The synsets are organized into senses, providing a hierarchical tree-like structure for each term. It gives synonyms of each word, and also into hyponym/hypernym (i.e. Is-A), and meronym/holonym (i.e. Part-Of) relationships. In document clustering WordNet lexical database has been widely used and its uses shown improvement in clustering. The work in (Dave, Lawrence, & Pennock, 2003) used synsets as features for document representation and subsequent clustering. As they have not applied word sense disambiguation, synsets actually decreased clustering performance. A subsequent work from (Hotho et al., 2003) used WordNet in document clustering for word sense disambiguation to improve upon (Hotho, Maedche, & Staab, 2002; Hotho et al., 2003) clustering performance. Another work that reports improvement is (Fodeh, Punch, & Tan, 2009). The work proposed in (Kim, Kim, & Cho, 2017; Rao & Mishra, 2017; Stanchev, 2016; Vidhya & Geetha, 2017) also used semantic features using concept, term relationship and semantic relationship using external knowledge based like WordNet.

The main limitation of all these approaches is the increase in dimensionality of the data. Our work is similar to these works as we are also using WordNet to extract the lexical chains from the documents. Our approach is very similar to (Pedersen, Patwardhan, & Michelizzi, 2004), as we extracted the words that are linked in WordNet together for defining their lexical classes. Researchers have also used Wikipedia as a knowledge resource for document clustering (Gabrilovich & Markovitch, 2007; Hu et al., 2008), and have reported improved results through semantics incorporated through Wikipedia hierarchical knowledge. Moreover, use of text mining methods to extract semantics is also well evolved now. Researchers have used statistical characteristics (Matsuo & Ishizuka, 2004) of terms to get the semantics (Farahat & Kamel, 2011).
2.2.2.2 Lexical Chains

Lexical chains (Halliday & Hasan, 2014) are collection of related words that are commonly used in language to convey a message in oral and written communication. These sequences of words can be used to determine the semantics of the document. The use of lexical chains for text summarization (Barzilay & Elhadad, 1999) is quite established research but its use in document clustering is quite new and still need to be investigated. These lexical chain may span short or long distance in the text. Lexical Chains (Morris & Hirst, 1991) generally exploit the semantic relatedness of words in a document. It is also independent on grammatical structure of the text. We strongly believe that in order to take advantage of semantics of textual documents in representation scheme we need to collect these chains from the document first. These chains offer lexical cohesion from a document which is a promising feature for document clustering.

The process of identifying lexical chains can be divided into two categories. The first uses a lexical dictionary like WordNet (Miller, 1995) to identify the relations between a pair of words (Pedersen et al., 2004). The other approach uses computational linguistics algorithm to identify the lexical sense through co-location and co-occurrences of words. In this thesis, we apply a technique similar to (Kang, Kim, & Lee, 2005) to extract lexical chains from a given document. Our work is very similar to the work reported in (Wei, Lu, Chang, Zhou, & Bao, 2015) and we have also used lexical chains to determine the core semantics or themes of a document, but unlike their work we extend the semantics through topic maps structure. We next describe the topic maps (Garshol, 2002) structures and its utility to linking different terms semantically.
2.2.2.2 Topics and Topic maps

Another approach to identify the semantics of a document is to use topics for the identification of important terms. It is a well develop text mining area in which topics are extracted using high cohesion between groups of words, from documents that are used to convey the meaning of a textual document. There are many algorithms through which one can get important, emerging –say hot topics (K.-Y. Chen, Luesukprasert, & Seng-cho, 2007), new topics (Ayad & Kamel, 2002) and probabilistic topics (Bun & Ishizuka, 2002) from a collection of text. The study of topic extraction from scientific literature for understanding core and competency is reported in (Buitelaar & Eigner, 2008) and a similar work in which hierarchical structure of topics are studied is (Gollapudi & Panigrahy, 2006). There is another work through which topics are identified and extracted from source code (Kuhn, Ducasse, & Girba, 2007). All these work emphasis that the topics from a text can be used to represent contextual information which is good for performing document clustering (Kuhn et al., 2007; Quan, Liu, Lu, Ni, & Wenyin, 2010; Silva, Mexia, Coelho, & Lopes, 2001).

The most influential work in topic based clustering is (Anaya-Sánchez, Pons-Porrata, & Berlanga-Llavori, 2008, 2010) motivated by text mining, they have used topic mining techniques (Pons-Porrata, Berlanga-Llavori, & Ruiz-Shulcloper, 2007). Topic maps (Pepper & Moore, 2010) on the other hand is an emerging area in knowledge management and it has been reported in (Rafi et al., 2011) for efficiently performing document clustering. However, learning the topic term and structures from the text is a challenging activity (Bloehdorn, Cimiano, Hotho, Subasic, & Huettner, 2006). The topic specific document clustering is studied in (Ayad & Kamel, 2002). We believe that the topic maps structures carry some important
contextual information and that should be merged with lexical chains to obtain more semantically rich information from the simple words.

2.2.2.3 Semantic Similarity

Semantic similarity is a measure (Zezula et al., 2006) that exploits the meaning of a term rather than its lexical form. It has a wide application in natural language processing, information retrieval, text clustering and classification, text summarization and automatic text comprehension. There are two approaches to define a semantic similarity in general, one is based on lexical dictionary like WordNet and the other is based on computational linguistic and text mining method to come up with a semantic relationship between words. Use of WordNet is quite popular among researchers to get the semantics from the simple term features of the documents. Many semantic similarity measures have been proposed by using WordNet. In general, we can classify these measures into four distinct groups: (i) path-length based measure, (ii) feature based measure, (iii) information content based measure, and (iv) hybrid measure. For a complete overview we recommend (Meng, Huang, & Gu, 2013; Pedersen et al., 2004).

The work in (Stanchev, 2016) performs semantic document clustering using semantic graphs. These graphs are extracted from documents using WordNet. This work is similar in the sense that it is also using WordNet for semantic relationship identification. They only used Reuters-21578 dataset for their experimentation. An important aspect of document clustering is to utilize world knowledge as enrichment process as proposed in (C. Wang, Song, Roth, Zhang, & Han, 2016) they proposed a framework for incorporating domain specific knowledge in the process of document clustering. This work identifies three challenges for such a knowledge integration through machine learning (1) domain specification, (2) knowledge representation, and (3) propagation of indirect supervision. Our work also identifies that the document
representation is very sensitive to actual effective document clustering. Following the cited overview, we will highlight the research gap and present an analysis of the progress so far for the task of document clustering.

2.2.3 Document clustering – Evaluation

The most challenging and difficult part of document clustering is validation of the clustering results. The cluster analysis can provide valuable knowledge about the document collection such as whether it is non-random in nature or not, what is the quality of the cluster produced, and how many distinct clusters are there in the document collection (Zhao & Karypis, 2002). There are three different ways through which we can validate the clustering outcomes (Rand, 1971). The first one is called external validity measure. It is generally validated against a gold standard reference. It means a manual-human annotated cluster is already there, and we need to compare the resultant clusters with this human annotated cluster. It is usually the evaluation technique for document clustering as there are several standard text mining datasets for clustering and classification task. The second technique for cluster validation is called internal. It uses the actual data set for the clustering to validate the process. It is an intrinsic process and Dunn Index (Bezdek & Pal, 1998) is one such evaluation. The third process through which we validate a clustering outcome is relative measure. It is the analysis of the sensitivity of internal measure during the clustering process.

2.2.4 Research Gap – Document clustering

We identify the following research gap by investigating the above cited works.

- Document clustering is highly sensitive to the document representation schemes. A document representation which implicitly captures the semantics of the document is
highly desirable. The lexical chains alone, or phrases or sequences are not enough to glue the semantic of the document.

- A similarity measure is an essential function for the document clustering task. A similarity measure that assigns high score to semantically related pair of documents is what is required for better semantic clustering.
- The evaluation of a better clustering algorithm is justified when it has high purity and low entropy as external evaluation measures for any experiment. In offline document clustering task standard datasets are used to highlight the clustering objectives.

2.2.5 Research Contribution- Document clustering

The thesis proposes a document clustering approach that alleviates all mentioned problems and its contribution can be summarized as follows.

- We introduce the concept of extracting the core semantics of a document using lexical chains. The lexical chains can be used as a small subsets of semantic features to identify document theme. On the other hand, this small subset also reduces the dimensionality of the feature space. The lexical chains offer prospects of incorporating semantics into the clustering process. We propose a simple algorithm using WordNet to extract lexical chains from the documents.
- We propose a hybrid document representation model that uses lexical chains and topic maps structures to overcome the non-linear relationships among simple features. Our
topic maps features are rich in power to describe the semantics and can be used to create better understandable labels for resulting clusters.

- We propose a semantic based similarity measure with no extra computational burden to assign a more appropriate score to a pair of documents. Our similarity measure uses lexical chain and topic maps structures as common features for similarity.
- Extensive series of experiments are performed to evaluate the effectiveness of our proposed approach and encouraging results are obtained.
Chapter 3

Document Representation

Document representation is a central problem to many text mining tasks such as document processing, understanding, retrieval, classification, and clustering. The document representation is essential for defining, solving and evaluating the proposed solution of the general document analysis problem. Documents in their raw form, simply consist of plain text written in human language. A document can be decomposed into atomic units as word/lexeme or token, but not every atomic unit of text is meaningful. Documents need to be processed and represented in a concise and identifiable logical atomic/structure unit. Generally, the text written in human language is preprocessed in a pipeline fashion with multiple choices of processing function at each step. The documents are parsed as per the defined unit/structure for representation.

The task of document clustering is very sensitive to the actual document representation model. There are numerous efforts in research to model documents in easy, concise, low computational cost and yet high semantic representation scheme. This chapter discusses various document representation schemes from the literature which are comparable to proposed topic maps based representational model.
3.1 Document Pre-processing

The general idea of document representation is to get the most describable features for document. There are several functions of a document pre-processing pipeline. It is very much dependent on the feature on which one wants to represent the documents. The pipeline starts with a Tokenization process; it separates out each lexeme from the text. The tokenizer returns an array list of words. The next phase of pipeline is stop-words removal.

**Stop words** usually refer to the most common words in a language, which are short functional words in nature for example: is, am are, the, it, etc. These words generally do not provide any valuable information in the text. These words are generally filtered out before or after processing on Natural Language Processing (NLP) data. In different languages, there is always a list of stop words available for removing them out for further processing. Stop word removal is not always fruitful for text processing task, as its removal can cause problem when searching for phrases. Hence it is not recommended to remove stop words when performing any phrase-based information processing. It is generally a choice whether to remove or not to remove stop words from the processing.

**Stemming** is the term used in information retrieval to describe the process of reducing inflected (or sometimes derived) words to their word stem (base/root form of the word). Stemming programs are commonly referred to as stemming algorithms or stemmers. The Porter stemming algorithm (‘Porter stemmer’) is a process for removing the commoner morphological and inflexional endings from words in English.

For example if the following list is given to porter's stemming algorithm (Porter, 1980)
operation, operator, operand, operational and operable

We will get the following stem- words:

oper oper operand oper and oper

**Lemmatization** is a linguistics process that group together different inflected forms of a word into a single representational item. Unlike, stemming it is based on contextual information and hence it is a complex process. The text pre-processing pipeline ensures that one will get the descriptors (features) as per one’s definition and processing capabilities.

In order to explain the different representation models for illustration purpose, we consider the following set of simplified documents.

D1: I love to play cricket. I hate hockey.

D2: Pakistani boys love cricket and hockey.

D3: Hockey is the national game of Pakistan. Cricket is mainly popular and loved by majority.

### 3.2 Simple Boolean Model for Documents

The simple Boolean Model for information retrieval (Salton, 1971a) is based on Boolean algebra and Set theory. This is a model based on simple, intuitive, understandable formalism and has a simple and very easy implementation approach. Documents in this model are being represented by terms (features) extracted from them. The model does not impose any fixed definition of terms (features). These terms (features) are equally weighted for the representation (Salton, Wong, & Yang, 1975) of documents. The model supports simple exact information retrieval. The problem associated with this model is that it can retrieve too few or too many documents and all the documents are at the same level. There is no ranking in the retrieved documents. The simple Boolean model is sometimes called bag-of-words or bag-of-terms. It is simply a multi-set of terms (features). One more limitation of this model is that it does not
consider the order of the words used in the document. In this representation model the given documents are represented as follow:

\[
\begin{align*}
D_1 &: \{\text{love, play, cricket, hate, hockey}\} \\
D_2 &: \{\text{Pakistan, boy, love, cricket, hockey}\} \\
D_3 &: \{\text{hockey, nation, game, Pakistan, cricket, main, popular, love, major}\}
\end{align*}
\]

### 3.3 Vector Space Model

Vector Space Model comes from algebraic mathematical structures. A document in this model is represented as n-dimensional vector of terms. The term definition is not imposed by the model (Salton et al., 1975), it can be a word (dictionary term), a phrase or a sequence of words. The document dataset is used to collect terms to form a dictionary called Vocabulary-(V). Each term from the dictionary is considered an independent dimension. Hence each document is represented by a high dimensional vector with the dimension being equal to the fixed size of vocabulary.

Here are the simple representations of our example documents in vector space model. Documents are fixed length vectors of terms from the vocabulary in dictionary order.

\[
\begin{align*}
\text{Vocabulary (V)} &= \{\text{boy, cricket, game, hate, hockey, love, main, major, nation, Pakistan, play, popular}\} \\
D_1 &: <0,1,0,1,1,0,0,0,0,1,0> \\
D_2 &: <1,1,0,0,1,1,0,0,1,0,0> \\
D_3 &: <0,1,1,0,1,1,1,1,0,1,0>
\end{align*}
\]

Vector space model has numerous advantages over simple Boolean model. The Boolean model generally supports exact match for information retrieval, whereas this vector based representation allows partial matching. It can also support ranking of documents as per the relevance with the query for information retrieval. Another advantage of using vector space
model is to allow continuous degree of similarity calculation for queries and documents. There are several schemes that modify the document vector to overcome challenges in effective computations. Term frequency (tf) is the number of occurrences of each term in the collection. The inverse document frequency (idf) is a function that represents number of documents where a term appears. These two together form (tf*idf), a scheme which is most widely used for effective representation of documents.

However, this simple model comes with a drawback. The large document vectors with very sparse document-to-document matrix make the computation really expensive thus making this model impractical for large corpus. The shortcoming of vector model is addressed with various dimension reduction techniques such as Latent Semantic Analysis (LSA) (Dumais, 2004). LSA provides a mathematical tool to map a vector space model into a more compact relevant space based on synonyms and related words. Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Indexing (PLSI) and Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) all use the vector space model to reduce the dimension by exploiting a latent space (semantic sub-space). These methods are based on linear algebra model and are computationally very expensive.

### 3.4 Suffix Tree Model

In suffix tree model a document (Zu Eissen, Stein, & Potthast, 2005) is represented as a suffix tree in which nodes are common term from documents and leaves are the vocabulary of the documents. In order to form a suffix tree for a given document, the document is parsed and the extracted terms are inserted in an initially empty suffix tree. Here is an example from (Chim & Deng, 2007) to describe the suffix tree model.
The example suffix tree of three documents (i) "cat ate cheese", (ii) "mouse ate cheese too" and (iii) "cat ate mouse too". Suffix tree is a good data structure to extract order sequences from a document. From our sample dataset, consider the document D1, here is a suffix tree constructed for this document. The suffix tree contains 14 branching nodes. It clearly preserves the order of the terms appears in the given document.
3.5 Sequence (Phrase) Model

Sequence based representation of documents are proposed by many researchers (C.-L. Chen et al., 2010; Fung et al., 2003; Li et al., 2008; Zhuang & Dai, 2004). These sequences can be easily extracted from a pre-built suffix tree from the documents or by applying data mining techniques for sequence extraction from textual documents.

Another approach which can extract sequences from documents is based on market basket analysis in which each document is treated as transactional basket and each word is treated as
an item selected for purchase. The idea is to select frequent interesting item-sets that appears in the document collection. In the given example from D1: I love to play cricket. I hate hockey.

We can easily extract sequences of two words as follow:

D1: {love play, love cricket, love hate, love hockey, play cricket, play hate, play hockey, cricket hate, cricket hockey, hate hockey}

### 3.6 Graph model

A generic graph is denoted by $G (V, E)$, where $V$ is the set of vertices and $E$ is the set of edges. Edges represent some relationship with a pair of vertices. In a graph model representation terms are vertices and edges represent relationship between terms. A graph of the three example documents is represented below.

![Figure 2- Graph model for example documents](image-url)
The main benefit of graph-based technique is that it allows keeping the inherent structural information of the original document. The above diagram represents the three overly simplified documents in a single graph. It is evidently clear that it can captures the order of the terms from the documents but it is a very complex representation.

### 3.7 Lexical Chain Model

A lexical chain can be defined as a sequence of related words generally used to convey a message in written or oral form in communication. These textual features can span from short adjacent sentences to the entire text document. A lexical chain is independent of grammatical structure of the text (Wei et al., 2015); it is a collection of related lexical tokens (words). The good part of this group of words (chains) is that it can be used as a document level lexical unit, which is lexically cohesive. This can be used to determine the context of the lexical terms for disambiguation and identification of the concept that the term represents (Ercan & Cicekli, 2007). There are numerous algorithms to extract lexical chains from a given document (Barzilay & Elhadad, 1999), (Mochizuki, Iwayama, & Okumura, 2000) and (Silber & McCoy, 2002), which mostly use a lexical knowledge base to facilitate extraction process. These algorithms may produce different chains. Consider the document below from the sample. There are two chains \{love, hate\}, \{play, cricket, hockey\} extracted from sample document.

D1: I love to play cricket. I hate hockey.

Lexical Chains = \{\{love, hate\}, \{play, cricket, hockey\}\}.

### 3.8 Topic Model

A document can be represented in a topic model by only selecting the topic through using a topic extraction algorithm. The main idea behind this model is to extract autonomously term
pairs from a document collection that are likely to represent homogenous contents. These terms are considered as topic of the document (Anaya-Sánchez et al., 2010) can be extracted by using probabilistic criteria (Quan et al., 2010).

3.9 Topic Maps Model

Topic maps (Pepper & Moore, 2010) is a new standard for information representation and exchange. It is web 3.0 enabling technology. The idea of topic maps is motivated from the back-of-book-indices and their merging. It is a meta-model that supports compact information representation, exchange and later information retrieval (Steiner et al., 2001). Finding information is real ambient function of topic maps. The need for automated textual analysis is growing and there are many fields in which it is becoming a big issue. The digital journalism is one area where automatic text analysis is critical and Elisabet & Thorsten (Günther & Quandt, 2016) proposed a topic model based analysis. Their Topic models refer to the idea of an imaginary generative process, in which the distribution of topics in a document and the distribution of words in a topic are drawn from a probability distribution. Hence an automatic analysis is possible. This work differs from our approach mainly in the creation of topic model. The topic maps document model uses topic maps extracted from a document. A document thus transformed into a topic maps and the structure of this map of the topics that appears in a document. The topic maps contains topics, association and occurrences (Rafi et al., 2011). A topic maps based similarity measure is quite effective in determining semantic relationships among topic terms. Here are the examples topic maps from our example documents: $D_1$: I love to play cricket. I hate hockey.
3.10 Lexical + Topic Maps Model

We propose in this thesis a new document representation model, which is a kind of hybrid model that utilizes lexical chains and topic maps together. First the lexical chains are extracted from the given documents, next the lexical terms are mapped to the topic maps structures present in the document. Our scheme takes advantage of lexical cohesion structure along with topic map relationships to get a semantic based representation of document. This representation implicitly captures the lexical terms from a documents, which is a natural abstract of the document. These lexical terms, when they occur in the topic maps structures, innately encapsulate the non-linear relationship from the lexical terms. Thus this representation offers easy contextual and semantics information in meta-descriptor of the documents.

Consider the document below from the sample, we have

\( D_1 \): I love to play cricket. I hate hockey.

Lexical Chains = \{ \{love, hate\}, \{play, cricket, hockey\} \}. 
Topic Maps = \{(1:2;love, cricket, person), (1:2;hate, person, hockey), (1:3;play, cricket, hockey, outdoor), (1:2; experience, love, hate)\}

Lexical Chains + Topic Maps = \{(1:2;love, cricket, person), (1:2;hate, person, hockey), (1:3;play, cricket, hockey, outdoor)\}

We discussed nine different document representations for the task of document clustering. Table 2 briefly summarized the advantages and disadvantages of each. The representation scheme proposed in this thesis is based on Lexical Chain + Topic Maps. This representation is easy to compute, rich in contextual and semantic information and also captures non-linear relationships among terms.

Figure 4- Lexical terms based topic maps model for document D1
<table>
<thead>
<tr>
<th>Representation</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean Model</td>
<td>- mathematical model</td>
<td>- no context</td>
</tr>
<tr>
<td></td>
<td>- simple implementation</td>
<td>- problem with collocation</td>
</tr>
<tr>
<td></td>
<td>- easy to comprehend</td>
<td>- query formulation is difficult</td>
</tr>
<tr>
<td>Vector Space Model</td>
<td>- mathematical model</td>
<td>- large and spare vectors</td>
</tr>
<tr>
<td></td>
<td>- simple representation</td>
<td>- average statistical quality (not more support)</td>
</tr>
<tr>
<td></td>
<td>- context and semantic quality</td>
<td></td>
</tr>
<tr>
<td>Suffix Tree Model</td>
<td>- mathematical model</td>
<td>- low statistical quality</td>
</tr>
<tr>
<td></td>
<td>- context and semantic quality</td>
<td>- complex document processing</td>
</tr>
<tr>
<td></td>
<td>- collocation can be supported</td>
<td>- high cost</td>
</tr>
<tr>
<td>Sequence (Phrase) Model</td>
<td>- mathematical model</td>
<td>- average statistical quality</td>
</tr>
<tr>
<td></td>
<td>- simple representation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- context and semantic quality</td>
<td>- high cost</td>
</tr>
<tr>
<td></td>
<td>- collocation can be supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graph Model</td>
<td>- mathematical model</td>
<td>- low statistical quality</td>
</tr>
<tr>
<td></td>
<td>- good context and semantic quality</td>
<td>- complex document processing</td>
</tr>
<tr>
<td></td>
<td>- non-linear relationship can be supported</td>
<td>- high computational cost</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lexical Chains</td>
<td>- simple document level processing</td>
<td>- low statistical quality</td>
</tr>
<tr>
<td></td>
<td>- simple representation</td>
<td>- low computational cost</td>
</tr>
<tr>
<td></td>
<td>- good context and semantic quality</td>
<td>- lack of word sense disambiguation</td>
</tr>
<tr>
<td></td>
<td>- linear relationship</td>
<td></td>
</tr>
<tr>
<td>Topic Model</td>
<td>- most probable terms from the collection</td>
<td>- low statistical quality</td>
</tr>
<tr>
<td></td>
<td>- collocation can be supported</td>
<td>- high computational cost</td>
</tr>
<tr>
<td></td>
<td>- context and semantic quality</td>
<td></td>
</tr>
<tr>
<td>Topic Maps Model</td>
<td>- topic oriented terms</td>
<td>- computational cost</td>
</tr>
<tr>
<td></td>
<td>- context and semantic quality</td>
<td>- ontology</td>
</tr>
<tr>
<td></td>
<td>- non-linear linked terms via topic maps</td>
<td></td>
</tr>
<tr>
<td>Lexical + Topic Maps Model</td>
<td>- lexical + topic oriented terms</td>
<td>- average statistical quality</td>
</tr>
<tr>
<td></td>
<td>- context and semantics</td>
<td>- lack of word sense disambiguation</td>
</tr>
<tr>
<td></td>
<td>- non-linear linked terms</td>
<td></td>
</tr>
</tbody>
</table>

Table 2- Comparison of different document representation models
Chapter 4

Document Similarity

Almost all clustering algorithms require a basic function that gives the similarity between a pair of objects, from the collection that need to be clustered. Generally, this functional requirement is called a similarity measure. A major computational burden, while performing document clustering, is the calculation of similarity measure between a pair of documents. Similarity measure is a function that assigns a real number between 0 and 1 to a pair of documents, depending upon the degree of similarity between them. A value of zero means that the documents are completely dissimilar whereas a value of one indicates that the documents are practically identical. Similarity calculation heavily depends on document representation scheme hence most functions that define similarity calculation for a pair of documents exploit implicit document representation (Huang, 2008). Accurate document clustering is mainly based on the effectiveness of the defined similarity measure. The function for similarity measure should be easy to compute, it should implicitly capture the relatedness of the documents, and it should also be explainable (Strehl et al., 2000). The document to document similarity poses many challenges such as size of the documents: small and large text documents require focused analysis (Aiken, 2001; Mohler & Mihalcea, 2009) in defining a similarity functions for them.

In this chapter, we discuss various similarity measure proposed for general task of document clustering. Later, we discuss our proposed similarity measure for exploiting topic maps based representation of the documents. We define a similarity function based on frequent common tree patterns extracted from the topic maps of documents, hence these pattern (sub-trees) also inherently capture semantics in similarity calculation. We believe that this similarity measure
implicitly includes the semantics of the textual features in similarity calculations. Thus, it offers better similarity measure for textual documents.

**4.1 Jaccard Similarity**

The simplest type of similarity measure that can be used for clustering task is Jaccard Similarity. It is also called Jaccard Index or Jaccard similarity coefficient. It is a statistical measure that can be used to know the similarity or diversity of a pair of sample sets. The Jaccard coefficient computes the similarity between finite sample sets. Mathematically, in theoretical set algebra, it is defined as the size of the intersection divided by the size of the union of the sample sets.

\[
Jaccard\ similarity(A,B) = \frac{(A \cap B)}{(A \cup B)}
\]

In our example documents, the Jaccard similarity between a pair of documents given below can be computed as:

D1: {love, play, cricket, hate, hockey}

D2: {Pakistan, boy, love, cricket, hockey}

\[
Jaccard\ similarity(D1, D2) = \frac{(D1 \cap D2)}{(D1 \cup D2)}
\]

\[
Jaccard\ similarity(D1, D2) = \frac{3}{7} = 0.428
\]

It is generally used in document clustering when the documents are represented as simple bag of words (BOW) model or simple collection of features.
4.2 Cosine Similarity

A similarity measure that is defined on the vector representation of the documents is termed as Cosine similarity. It basically represents the cosine of angle between the two vector representations of the documents. It is a judgment of orientation and not on the magnitude of the vectors. Since all the components in vector representation of a document are non-negative, the cosine is positive and lies in [0,1]. Two vectors of same orientation will produce a cosine value equal to one. Mathematically it is computed by the given equation:

\[ \text{Cosine}(d_1, d_2) = \frac{d_1 \cdot d_2}{|d_1| \cdot |d_2|} \]

In the above equation both d1 and d2 represents document vectors.

In our example documents, the cosine similarity between a pair of documents given below can be computed as:

D1: \(<0,1,0,1,1,0,0,0,1,0>\>

D2: \(<1,1,0,0,1,1,0,0,0,1,0>\>

Hence the similarity value is calculated from above equation as

\[ \cos(D_1 , D_2 ) = \frac{(D_1 \cdot D_2)}{|D_1| \cdot |D_2|} \]

\[ \cos(D_1 , D_2 ) = 3/5 \quad = 0.6 \]

It is a most commonly used similarity measure in information retrieval and text mining methods. It has been considered the best technique for calculating similarity between objects in high dimensional spaces.
4.3 Kullback-Leibler Similarity

Kullback-Leibler (KL) divergence is also an approach to compute similarity between a pair of documents. In this a document is considered as a probability distribution of terms. The distance between the two corresponding probability distributions is measured as similarity between the two documents. The Kullback-Leibler (KL)-divergence, also called the relative entropy, is a widely applied measure for evaluating the difference between two probability distributions. This measure is more conveniently used for word sense disambiguation. It is very effective in calculating similarity of words and their collocation. For document similarity, it is seldom used as it is asymmetric in nature. There is another issue that it is computationally very expensive due to probability calculations involved in it. The equation for this is given below:

\[
Sim_{KL}(A, B) = \sum_{t=1}^{m} w(t, A) \times \log \frac{w(t, A)}{w(t, B)}
\]

Here A and B are two sequence of words or collocations or documents, and t represents the term appears in either A and B.

4.4 Semantic Similarity

Semantic similarity measure is calculated based on matching concepts and their hierarchy in the two documents. This work is very closely related to ours, as we are also using the semantic terms via lexical chains and topic maps. Semantic similarity refers to the similarity of two documents through their features like words, phrases and sequences and how close they are in a taxonomy such as the WordNet.

The semantic similarity must consider three intuitions (Lin, 1998) related to a pair of documents, which are: (i) the similarity between documents are higher if they share more common concepts or terms, (ii) the difference between documents is inversely related to their
similarity which means that if their concepts or terms more different then their similarity score will be less, and (iii) the maximum similarity between a pair of documents is 1, which is the case when the two documents are identical or have same concepts or terms used in them.

Many semantic similarity measure have been proposed by using WordNet. In general, we can classify these measures into four distinct groups: (i) path-length based measure, (ii) feature based measure, (iii) information content based measure, and (iv) hybrid measure. For a complete overview we recommend (Meng et al., 2013; Pedersen et al., 2004).

The semantic similarity between two concepts like ‘Dinner’ and ‘Lunch’ refers to the most generic common term for the two words which is ‘Food’, let $x_1 = \text{Dinner}, x_1 \in C_1$ and $x_2 = \text{Lunch}, x_2 \in C_2$. The commonality between the two is $C_0$, which is $C_1 \cap C_2$. Hence the similarity between the two words can be given by

$$Sim(x_1, x_2) = \frac{2 \times \log P(C_0)}{\log P(C_1) + \log P(C_2)}$$

There has been many proposal to define similarity over a common parent in a tree like hierarchy, there has been strong observation that these measure to do conform or correlate to human judgment (Rada, Mili, Bicknell, & Blettner, 1989).

A similarity measure proposed by Wu and Palmer (Wu & Palmer, 1994) is mainly accepted to more correlated to human based similarity measure, this similarity measure uses IS-A links between the two words along with the most specific common superclass exists in WordNet. Let $N_1$ and $N_2$ are the number of IS-A links from $x_1$ and $x_2$, and $N_3$ is the number of IS-A link from superclass to top of the taxonomy. The similarity can be given by the following equation:

$$Sim(x_1, x_2) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3}$$
4.4 Semantic Similarity Using Lexical Chains and Topic Maps

We propose a new similarity measure using a hybrid document representation. This hybrid representation is a combination of lexical chains and topic maps constructs, extracted from the document using algorithms proposed in this thesis. This lexical chain carries core lexical semantics of the document in similarity calculation (S. Andrews, 1992). Given a document, we first process the document to extract terms from it, and then the lexical chains are formed by using WordNet through an API. The lexical chains are collected when the pair of terms are n length apart, where n is the number of unique terms from document Di. We are not using any ranking on lexical chains, all the terms in the lexical chains are collected from a collection to form a vector space. The lexical similarity between a pair of documents is computed by using cosine similarity function. The main difference is the lexical terms space which generally contains core lexical terms from the document hence we compute similarity on reduced dimensional space which is relevant to lexical semantics of the document (Jijkoun & Rijke, 2005). We are not explicitly using any word sense disambiguation; hence lexical chains may have very little possibility of having an ambulation (Manabu & Takeo, 1994).

\[ \text{Cosine_lexical chains} \left( d_i, d_j \right) = \frac{d_i \cdot d_j}{\left| d_i \right| \cdot \left| d_j \right|} \]

The topic maps constructs are extracted from each document using Wandora and an API OpenCalais. These topic maps are filtered on lexical terms so that only those topic maps constructs that contains a lexical term from the chains are retained for hybrid document representation. These topic maps implicitly capture non-linear relationships among lexical terms and hence are able to capture semantics of the document. The similarity between a pair
of documents on these lexical cohesions based topic maps constructs are computed using Jaccard coefficient.

\[
JC_{\text{topic maps pattern}}(d_i, d_j) = \frac{(d_i \cap d_j)}{(d_i \cup d_j)}
\]

The combined similarity measure is computed as the average of the Cosine and Jaccard similarities. Using the equation below:

\[
\text{Similarity} (d_i, d_j) \text{ from lexical and topic maps} = \frac{\text{Cosine_lexical chains}(d_i, d_j) + JC_{\text{topic maps pattern}}(d_i, d_j)}{2}
\]

### 4.5 Conclusion

Similarity is an important concept for document clustering. Traditional similarity functions are heuristics-based in nature and tied to a particular form of representation. We define a hybrid representation of documents, using lexical chains and topic maps constructs extracted and transformed for compact document representation which also carries core lexical semantics and non-linear relationships among terms. We also define a composite similarity function that averages out the similarity between lexical chains and topic maps. This similarity function implicitly assigns high scores to those document pairs that have high lexical similarity and high topic maps patterns common in them.
Chapter 5

Document Clustering Algorithms

Document clustering is a specialized clustering technique in which textual documents are clustered in distinct groups (clusters). Document clustering is generally a three steps process (i) we need to define a meta representation of documents (ii) a similarity measure is a basic computational function that define relatedness between a pair of documents, and (iii) a clustering algorithm that utilize this similarity measure to produce the final clusters.

In this chapter, we discuss various document clustering algorithms. On a very high level of abstraction, we can classify the document clustering algorithms in two distinct classes (i) hard clustering algorithms and (ii) soft or fuzzy clustering algorithms. In hard clustering each document must belong to a single cluster hence it has a fixed or hard membership of one cluster only. On the other hand, in soft clustering a document may belong to more than one clusters and the final assignment will be based on the likelihood of the membership. Another very common characterization of clustering algorithms is based on the underlying model through which it infers clusters. There are many models through which different clustering results can be obtained.
5.1 Centroid Model

In the centroid model of document clustering, the clustering process utilizes the average similarity among all documents in a cluster. The partition clustering algorithms fall into this category. The partition clustering techniques, when provided with a document dataset, partitions it into a group of disjoint sets of documents. These algorithms generally require a predefined number of clusters to produce. These algorithms are iterative in nature and operate on centroid measure to change the assignment of documents in each iteration. Once there is no change in assignments within an iteration the algorithm stops and returns the resultant clusters. There are many algorithms that are representative of partition approach. k-Means (Anil K Jain, 2010), k-Medoids (H.-S. Park & Jun, 2009), CLARA and CLARANS (Ng & Han, 2002) are few of the examples in this category. The main advantage of this kind of algorithm is its running time which is proportional to the number of documents in the clustering dataset. The main disadvantage is to provide number of clusters as an input to the algorithm. It is a challenging problem to know a prior about exactly how many clusters exist in the given dataset. Centroid model based algorithms separate the documents into distinct groups that are marginally cut-off from the centroid values. It is generally, preferred over all other clustering algorithms as it has simple centroid based explanation for distinct clusters.

5.1.1 k-Means

The simplest and basic algorithm for clustering is k-Means. In this algorithm each document is represented as a vector in a d-dimensional vector space assuming that a collection of document has d-distinct terms. It will take k seeds to start the process. These k-seeds are the example documents that represent the prototype of the clusters we are interested in. The rest of the
documents are assigned to these prototype clusters based on which of the k seed documents they are closest to, which is generally a measure in d-dimensions. After going through all the documents, the centroids (cluster prototype or seeds) are updated as the mean of all the documents assigned to that cluster. The process iteratively continues until there is no change in prototype assignments. A variation to k-means is called Bisecting k-means. Bisecting k-mean first selects a cluster to split and then applies k-mean algorithm to get the two resultant clusters. The process iteratively applies till the desired numbers of clusters are obtained. Steinbach’s (Steinbach et al., 2000) work shows that the bisecting k-mean algorithm outperforms the basic k-mean and agglomerative hierarchical clustering in terms of accuracy and efficiency.

Algorithm 5.1.1 Basic k–means algorithm

Initialize k centroids

repeat
    for all objects in input do
        Assign each element to its closest centroid
    end for
    for all centroids do
        Compute the mean of the assigned points
        This mean now becomes the new centroid
    end for
until all centroids remains unchanged or other termination criteria
5.2 Connectivity Model

Connectivity model based document clustering uses the idea that if the distance between any pair of documents is short then they must belong to the same cluster or they are related. There is large possibility of distances between clusters, hence there are many variations of connectivity model based clustering algorithms. These algorithms are also called Hierarchical Algorithms.

5.2.1 Hierarchical Clustering

Hierarchical clustering method works by grouping data elements into a tree like structure (hierarchy). There are two variations of hierarchical clustering (i) agglomerative and (ii) divisive. In agglomerative hierarchical document clustering, initially each document is assumed to be a single cluster. Iteratively, a pair-wise similarity measure is applied to select the two most similar clusters and merge them to reduce the number of clusters. The algorithm works in this bottom-up fashion until all documents fall into a single cluster. In divisive hierarchical clustering, initially all documents are considered to be in a single cluster. The cluster is split into two by using some dissimilarity criterion. These splits continue till each document finally becomes a cluster by itself.

There are two main objections to agglomerative methods. First, they have low clustering accuracy as the algorithm cannot adjust once the merge step is performed. Second, they also have high computational cost. The main advantage of using this type of algorithm is that these algorithms produce all levels of possible clusters and hence the required level can easily be selected from it.

Algorithm 5.2.1 Hierarchical Agglomerative Clustering-Bottom up
Compute the similarity matrix

repeat
    Find two best candidates according to criterion
    Save these two in the hierarchy as sub clusters
    Insert new cluster containing elements of both clusters
    Remove the old two from the list of active clusters
until one cluster remains
There are many variations of hierarchical clustering algorithms based on distance functions and linkage types like: (i) complete-linkage clustering, (ii) single-linkage clustering, (iii) average linkage clustering, or Unweighted Pair Group Method with Arithmetic Mean (UPGMA), (iv) centroid linkage clustering, or Unweighted Pair Group Method Centroid UPGMC and (v) minimum energy clustering, etc.

5.3 Distribution Model

In the distribution model, clusters are arranged based on statistical distribution within the dataset. The main objective of distribution model based clustering is to find maximum a posteriori (MAP) estimates of parameters in statistical models defined on unobserved latent variables. Distribution-based clustering is able to capture correlation and dependence between latent variables and is able to produces complex models for clusters. One of the drawbacks of these types of clustering is that it is often not the case that for a given dataset follows certain statistical distribution.

5.3.1 Expectation Maximization

The Expectation Maximization (EM) algorithm is a type of distribution model based clustering. The main idea of EM algorithm is based on existence of an analytical model for the data and the knowledge of the functional form of the model. The algorithm tries to find the functional form and model that best describes the data. It is an iterative algorithm.

Algorithm 5.3.1 Expectation Maximization (EM) Clustering

| initialize k clusters centers |
| repeat |
| - E-step assigns objects to clusters according to the current parameter estimates |
| - M-step finds the new clustering or parameters that maximize the sum of squared error(SSE) or the expected likelihood. |
| until convergence-condition-for-model |
5.4 Density Model

The density model based clustering tries to find the denser point in a high dimensional space of data points to form clusters. These algorithms can be used to learn clusters of any arbitrary shape. These algorithms are also insensitive to noise in data. The best known algorithm in this density model based clustering is Density-based spatial clustering of applications with noise (DBSCAN) (Ester, Kriegel, Sander, & Xu, 1996) and Model-based clustering (Fraley & Raftery, 2002), with discriminant analysis and density estimation.

5.4.1 DBSCAN

DBSCAN algorithm is a density based clustering algorithm that looks for maximum local density of objects in order to group them into a cluster. Hence, the algorithm can find any arbitrary shaped cluster in a large spatial data set. It uses two parameters to learn the maximum density region. These parameters are (i) Eps - is a distance that is used to group the data points, and (ii) MinPts - is the minimum number of such data points to form a cluster.

Algorithm 5.4.1 DBSCAN(Dataset D, Eps, MinPts)
1. Select an arbitrary object \( P \) in \( D \)
2. Retrieve all objects density-reachable from \( P \) by arbitrary/random \( Eps \) and \( MinPts \);
3. if \( P \) is a core object then a cluster is formed;
4. if \( P \) is a border object then no objects are density reachable from \( P \) and DBSCAN visit the next object of the data set;
5. else assign \( P \) to noise object;
6. Continue the process(from step 1) until all the objects are processed

5.5 Graph Model

In graph model for clustering the given object is abstractly represented as a graph. An implicit pattern based on vertices (V) and edges (E) connectivity is extracted to define a similarity
between a pair of documents. Graph model is considered as a good semantic based model for
document clustering.

5.5.1 Highly Connected Subgraphs (HCS)

HCS clustering algorithm finds all the subgraphs with n vertices such that the minimum cut of
those subgraphs contains more than n/2 edges, and identifies them as clusters. Such a subgraph
is called a Highly Connected Subgraph (HCS). Single vertices are not considered clusters and
are grouped into a singletons set S.

Algorithm 5.5.1 Highly Connected Subgraphs (Graph g)

HCS(Graph g)
if g is highly connected
then return (g)
else
(H1,H2,C) ← MINIMUMCUT(g)
    HCS(H1)
    HCS(H2)
end if

5.6 Conclusion

Cluster analysis is an important problem and has vast applications in almost all areas of modern
day life. There are many algorithms proposed in literature for cluster analysis, ranging from
most general to very domain specific. In this thesis, we propose a document clustering approach
that tries to cluster documents by exploiting semantics of the textual contents in both, document
representation and document-to-document similarity measure. Both of these things are
essential for performing the task in hand. As an unsupervised approach we do not want to
provide any input to the approach, the approach itself extracts the hidden patterns through
which the clusters are to be produced. In this thesis, we will be using external evaluations for
evaluating the proposed approach, in which we only cut the solution at some specified level as indicated in the ground truth of dataset. Hence a variant of hierarchical clustering algorithm is used.

The Group-average Agglomerative Hierarchical Clustering (GAHC) algorithm is adopted in this thesis as clustering algorithm for the task of document clustering. The algorithm initially considers each document as a unique cluster iteration, and selects a pair of clusters to merge repeatedly in the merging procedure. In each turn, the pair of most similar clusters are selected to be merged. The similarity of two clusters is calculated by the group-average measure.
Chapter 6

Document Clustering on Semantic Notions

6.1 Introduction

Document clustering is a challenging problem as discussed in earlier chapters. An ideal solution to this problem produces clusters with high intra-cluster similarity, which means the documents that fall in a given cluster should be semantically relevant to each other. The documents in two different clusters should be highly contrastable in term of understanding and semantics of the text. However, all existing approaches to document clustering ignore the relationship between terms (words) from the documents. These approaches use features like terms, phrases, sequences and do not consider the relationships between different features. Documents written in human language follow a context and the use of words mainly depends on the same context.

The main contribution of this thesis is a document clustering approach named MTS-DC. The approach is also following the three major steps suggested by (i) a document representation scheme is proposed, (ii) a similarity functions is defined and (iii) an actual document clustering algorithm is used. The proposed approach is summarized in Figure 5.
The main idea behind this approach is to incorporate the semantic rich features from lexical chains and topic maps data structures into representation of documents. Figure 5, indicates how each document is transformed using algorithm and Wandora API into semantic feature based representation. We further process this representation to exploit the lexical cohesive patterns in topic maps representation. This combining approach offer excellent semantics across both lexical chains and topic maps. The meta-descriptors for documents still reduce the dimension and only carry refine semantic based non-linear features. Next, we define a similarity measure based on inferred common semantics between a pair of documents. Thus creating a document-
to-document similarity matrix for eventual document clustering using agglomerative hierarchical clustering.

6.2 Multi-layer Topic maps based Semantic-rich hierarchical Document Clustering (MTS-DC)

The approach proposed MTS-DC has three steps that we want to discussed in details next, these are (i) Document representation, (ii) Similarity measure and (iii) the clustering algorithm.

6.2.1 Document Representation

The representation of documents is a critical problem in document clustering. It is a knowledge representation problem that should be inherently captures the semantics of the document. Our document representation scheme uses two kinds of features from the document: (i) a collection of lexical features in the form of lexical chains and (ii) topic maps features. These features provide hierarchical linkage between terms. In document representation we use all such topic maps patterns that contain any term from the identified lexical chains.

6.2.1.1 Lexical Chain

Lexical chains are a collection of lexical terms extracted from a document. These terms are strongly related to each other and collectively used to transfer the message/intent of the text. There are many algorithms for extracting lexical chains but most effective are the ones that use some lexical databases like WordNet to identify the lexical chains. We also propose a lexical chain algorithm using Wordnet 3.0, where the algorithm is applied to a set of pre-processed documents. The preprocessing first tokenizes the text form a given document and there removes the stop words from the tokenized collection. After this we apply Porter's stemming
algorithm (Porter, 1980) to obtain a refine set of tokens form the document. The algorithm uses each processed document and for every term, it will check for every other terms, by using WordNet. Every pair of terms is compared for the same synset type. If the term pairs are linked with each other not more than n edges, it is included in the lexical chain. Where n is number of distinct terms in the document d. In WordNet graph, it uses a breadth first search to see the link-distance for each pair of terms. If the other term is related it is added to the chain of the first term. The algorithm is presented as Algorithm 6.1 below:

**Algorithm 6.1 Generating Lexical Chains from a document using WordNet 3.0**

**Input:** a document d after preprocessing, with terms n  
**Output:** a collection of identified lexical chains (LC [ ])
**Using:** WordNet Database 3.0 + JAWS API

```
LC[ ] = Φ;  
chain[]= Φ;  
for each token ti in d  
    (locate_in_wordnet(ti))  
    chain.add(ti);  
    for each token tj in d  
        if ((ti.SynsetType == tj. SynsetType) && ti.containsPath(tj) < n)  
            chain.add(tj);  
    endfor  
    LC.add(chain);  
endfor
```

Algorithm 6.1 is used to generate lexical chains from a given document. Each document is parsed and for each pair of terms appears in the document, if these terms are related to each other through WordNet 3.0 and their SynsetType is same and the path that like these terms are less than the total number of terms in the document, we add this into a lexical chain collection.

Algorithm 6.1 complexity analysis suggests that the running time of this algorithm is quadratic polynomial as there are N documents and each document has n term. The outer loop runs for every document terms and inner loop get the relationship via WordNet. Hence the in worst case it will look for every item from the other document. The “if condition” of the inner loop suggest
that the processing is only required for the item whose SyssetType is matched. The lexical
chains for a collection of N documents with n terms can be generated using Algorithm 6.1 in
O(n^2) time. Hence the running time is quadratic for this algorithm.

6.2.1.2 Topic Maps

Topic maps is becoming an international standard for knowledge representation that facilitates
the searching of information. It is based on formal model and modern information management.
The model is very well defined in ISO standard (ISO13250). The use of topic maps is
escalating in the projects that perform enterprise information integration, knowledge
management, digital libraries and web-based information integration and management. The
subject-centric nature of topic-maps paradigm is what we perceive the very nature of human
centered clustering process. We wish to exploits the very nature of this subject centric
information in topic maps for performing the final clustering. The documents are transformed
into a topic maps representation by using Wandora. We have used OpenCalis to access and
transform these documents through web. We have extracted topics and their related information
constructs from these transformed documents in XML Topic maps representation by using
XPath expressions. The algorithm 6.2 is used to get the topic maps structures from a given text
document.
Algorithm 6.2 Generating Topic Maps from a given document

Input: Document \( d \)

Output: Topic maps collection from a document - TM[]

Using: Wandora + OpenCalis API

\[
\begin{align*}
TM[] &= \emptyset; \\
XML_d &= \text{OpenCalisAPI.BuildTopicMaps}(d); \\
&\text{for each topicType } \, tt \, \text{ in XML}_d \\
&\quad \text{for each associationType } \, at \, \text{ in XML}_d \\
&\quad \quad \text{if } (at.\text{associationType}(tt)) \\
&\quad \quad \text{for each occurrenceType } \, ot \, \text{ in XML}_d \\
&\quad \quad \quad \text{if } (ot.\text{occurrenceType}(at) \&\& \text{linked}(tt, at, ot)) \\
&\quad \quad \quad \text{TM.add(tt);} \\
&\quad \text{endfor} \\
&\quad \text{endfor} \\
&\text{TM.add(tt);} \\
&\text{endfor}
\end{align*}
\]

Algorithm 6.2 is used to generate topic maps from a given document. Initially a collection of topic maps is empty TM[], for each document using OpenCalisAPI we try to build a topic maps for the entire document. The OpenCalisAPI gives a XML based representation of document. For this XML document, we extract topic types, its association and occurrence for each topic type in document. The extracted topic type is added into TM[].

The complexity analysis of Algorithm 6.2 tells that the running time of this algorithm is also \( O(n^2) \) as for every document we are only creating a relationship of terms appears in the document by using an external Topic Maps-Wandora using an API. The topic extracted from a given document is always less than the terms in the document. Hence for a collection of \( N \) document with \( n \) unique terms, this algorithm runs in \( O(n^2) \) time. The space analysis also suggests that there is only a need to hold \( n \) relationships. Hence the space complexity is of \( O(n) \).
6.2.1.3 Lexical Chains + Topic Maps Hybrid Model

We propose a hybrid model for representing documents that employs both lexical chains and topic maps constructs for meta-descriptors for the documents. The major motivation behind creating this hybrid model is to get the semantics from the document. The lexical chains from a document represent the lexical cohesion of the document. The topic maps structure generally describe how different topic terms are related. A hybrid representation captures document level core semantics from lexical chains and the non-linear relationships between core words are captured through topic maps structures. Hence this representation carries document level core semantics along with lexical cohesion. It offers more meaningful information for the task of clustering. For one of the example documents D₁ it is a simplified document hence demonstrate how to create a hybrid document representation.

D₁: I love to play cricket. I hate hockey.

Lexical Chains = { {love, hate}, {play, cricket, hockey} }.

Topic Maps = {(1:2;love, cricket, person), (1:2;hate,person, hockey), (1:3;play, cricket, hockey, outdoor), (1:2; experience, love, hate)}

Lexical Chains + Topic Maps = { (1:2;love, cricket, person), (1:2;hate,person, hockey), (1:3;play, cricket, hockey, outdoor)}

We present one more document from a standard text mining dataset. The document is selected from 20 Newsgroup dataset from category soc.religion.christian and the document ID is 20361.

Lexical Chains = { {education, teaching, instructor, school, student}, {sex, married, AIDS, family, condom, contraception, prevention}, {religious, catholic}}

Topic Maps = {(1:2; teaching, instructor, student), (1:1; school, education), (1:1; school, student), (1:3; sex, condom, contraception, prevention), (1:2; married, sex, family), (1:1; religion, catholic)}
Lexical Chains + Topic Maps = \{(1:2; teaching, instructor, student), (1:1; school, education), (1:1; school, student), (1:3; sex, condom, contraception, prevention), (1:2; married, sex, family), (1:1; religion, catholic)\}

**Algorithm 6.3** Topic maps with lexical cohesion

**Input:** Document \(d\) + LC \([]\) + TM \([]\)

**Output:** Lexical Cohesion LTM \([]\)

**Local variables:** Queue \(Q\), int \(matches=0\); bool \(matched=false\);

\{
LTM\[\] = \(\phi\);
for each lexicalterm \(lt\) in LC
  for each topicterm \(tt\) in TM
    if ( \(lt == tt\))
      \{ matches++; matched= TRUE; Q.addNode(tt);\}
  endfor
endfor
LTM = Q;
\}

Algorithm 6.3 is used to refine the Topic Maps with lexical cohesion. The input to this algorithm is a particular document \(d\), its lexical chain LC\[\] and topic maps TM \[]. The algorithm refines all those topic maps structures that contains lexical chains from the given document.

The complexity of Algorithm 6.3 is also \(O(n^2)\) as this algorithm take a document of “a” terms with lexical chain of “b” terms and a topic maps of term “c” where a, b, and c are all positive integers. The analysis of lexical cohesion is performed for every lexical terms and topic terms.
Algorithm 6.4 Common Topic Patterns in a pair of documents

**Input:** TM for document Di with n and Dj with m, LTM Di and Dj

**Output:** Common Pattern Tree;

**Local variables:** \( \text{int matches}=0; \text{bool matched}=\text{FALSE}; \)

{ 
    for each topicmi \( tmi \) in TM.Di
        for each topicmj \( tmj \) in TM.Dj
            { 
                if \( (tmi == tmj) \text{ and ( LTMDi.contains(tmj) or LTMDj.contains(tmi))} \)
                    \( \text{matches}++; \text{matched}= \text{TRUE}; \text{Tree.add(lti);} \)
                
            } 
            while \( \text{matches} \leq \text{Max}(m,n) \)
                \( \text{ExtractAndMatchedChildern(Tree, LTMDi); matches}++ \)
            \( \text{matched=} \text{FALSE}; \)
        
    
    return Tree;
}

Algorithm 6.4 is used to find common patterns from a pair of documents such that for any lexical term appears in a document, the algorithm tries to find the maximum hierarchical match subtree from the topic maps structures in the other document. The input to this algorithm is a pair of documents in topic maps format with n and m topic terms in document D_i and D_j respectively. The algorithm returns all common pair of topic maps term along with the hierarchical match found in the documents.

The complexity analysis of Algorithm 6.4 is performed, this algorithm extract topic maps based hierarchical patterns from a document representation produced by algorithm 6.3, the lexical cohesive patterns already reduced the dimensionality of the text as its only select terms that are lexical chains and their contributions of topic structures enforce semantic relationship between reduced terms. The input to this algorithm are lexical topic maps LTM documents contains n and m terms. Hence the running time of this algorithm is also \( O(n^2) \).

Hence the entire process of transforming a document into lexical topic maps based pattern is taking polynomial time of order 2.
When I do programs, I spend about half the time talking about abstinence [...] I find that most people who object to sex education actually object to the teaching *anything* other than abstinence, and that IMO is just as irresponsible as only talking about condom use.

I'm under the impression that most sex ed instructors and/or policy makers actually object to making any more than a passing reference to abstinence, wishing to spend time only on the "realistic" choices.

In the "sex ed" portion of the high school "health" course I took in 1984, it was impressed that the only 100% positive way to *not* get pregnant was to *not* have sex.

Other methods of contraception were discussed, in the framework of a chart which showed both the _expected_ failure rate (theoretical, assumes no mistakes) and the _actual_ failure rate (based on research).

Top of the chart was something like this:

<table>
<thead>
<tr>
<th>Method</th>
<th>Expected Failure Rate</th>
<th>Actual Failure Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstinence</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

And NFP (Natural Family Planning) was on the bottom. The teacher even said, "I've had some students tell me that they can't use anything for birth control because they're Catholic. Well, if you're not married and you're a practicing Catholic, the *top* of the list is your slot, not the *bottom*. Even if you're not religious, the top of the list is safest."

Yes, this was a public school and after Dr Koop's "failing abstinence, use a condom" statement on the prevention of AIDS.

-jen
6.2.2 Similarity Measure

We define a similarity measure for our hybrid document representation in two separate ways and combing them by taking average. The first part of similarity is computed by lexical chains, we have used cosine similarity for computing the core semantic similarity between a pair of documents. We define a vector space of all the terms that appears in document collection using popular term frequency and inverse document frequency weighting scheme (tf*idf) for each term as a weight. Each document is now represented as the feature vector from this space. This part of similarity is calculated by cosine similarity of the documents vectors as below:

$$\text{Cosine_lexical\ chains\ (di, dj)} = \frac{di \cdot dj}{|di| \cdot |dj|}$$

The similarity from the topic maps portion of the hybrid representation is computed using Jaccard Coefficient or similarity. We extracted the common patterns from a pair of documents using algorithm 6.4 and the similarity is computed as below:

$$\text{JC_topic\ maps\ pattern\ (di, dj)} = \frac{(di \cap dj)}{(di \cup dj)}$$

The combine similarity measure is computed by the average of the two similarities. Using the equation below:

$$\text{Similarity\ (di, dj)\ from\ lexical\ and\ topic\ maps} = \frac{\text{Cosine_lexical\ chains}(di, dj) + \text{JC_topic\ maps\ pattern\ (di, dj)}}{2}$$

This similarity gives equal weights to both lexical chains and topic maps construct, we tried to learn some convex average for the similarity through different computational experiments but we have learned that it requires some smoothing factors for both the components of similarity functions. The smoothing factors are important criteria for information retrieval task and it can
be an extension of this thesis work to try some existing and smoothing techniques with our proposed approach.

6.2.3 Clustering Algorithm

The clustering algorithm that we used for performing actual clustering of documents is Group-Average Agglomerative Clustering (GAAC) (El-Hamdouchi & Willett, 1989) and (Voorhees, 1986). It evaluates cluster quality based on all similarity between documents of the candidate clusters, thus avoiding the pitfalls of complete-link and single-link criteria for hierarchical clustering. GAAC requires (i) documents represented as vectors, (ii) length normalization of vectors, so that self-similarities are 1.0, and (iii) the dot product as the measure of similarity between vectors and sums of vectors which is different from both single-link and complete-link HAC.

6.2.4 Example

Consider the two simplified documents in order to understand the similarity calculation proposed by lexical chains and topic maps constructs. Here we are given the two documents:

D₁: I love to play cricket. I hate hockey.
D₂: Pakistani boys love cricket and hockey.

We define a vector space of all the terms that appears in document collection using tf*idf for each term as a weight. Each document is now represented as the feature vector from this space. The lexical chains extracted from these documents are given below:

Lexical Chains = {{love, hate}, {play, cricket, hockey}}.
By using the weight vectors calculated from the collection over TF*IDF scores of the lexical chains from the collection is given below:

\[
\text{Cosin}_\text{lexical chains}(d_i, d_j) = \frac{d_i \cdot d_j}{|d_i| \cdot |d_j|}
\]

\[
\text{Cosin}_\text{lexical chains}(D_1, D_2) = \frac{3}{\sqrt{15}} = 0.774
\]

It is evident that lexical chain based cosine similarity is much higher than simple cosine similarity. This is due to the fact that irrelevant dimensions are reduced from the vector space.

When we consider the topic maps pattern extracted from these two documents, we will get the following:

Topic Maps = \{ (1:2; love, cricket, person), (1:2; hate, person, hockey), (1:3; play, cricket, hockey, outdoor), (1:2; experience, love, hate) \}

For the similarity, we need lexical cohesion based topic maps pattern, hence the lexical cohesion based topic maps patterns are as below:

Lexical Chains + Topic Maps = for \( D_1 \) = \{ (1:2; experience, love, hate), (1:2; love, cricket, person), (1:2; hate, person, hockey) \}

Lexical Chains + Topic Maps = for \( D_2 \) = \{ (1:2; experience, love, hate), (1:2; love, cricket, person) \}

The similarity from the topic maps portion of the hybrid representation is computed using Jaccard Coefficient or similarity. We extract the common patterns form a pair of documents using algorithm 6.4 and the similarity is computed as below:
\[ JC_{\text{topic maps pattern}}(d_i, d_j) = \frac{(d_i \cap d_j)}{(d_i \cup d_j)} \]

\[ JC_{\text{topic maps pattern}}(D_1, D_2) = \frac{2}{3} = 0.666 \]

The combine similarity measure is computed by the average of the two similarities. Using the equation below:

\[ \text{Similarity (} d_i, d_j \text{) from lexical and topic maps} = \frac{\text{Cosine lexical chains}(d_i, d_j) + JC_{\text{topic maps pattern}}(d_i, d_j)}{2} \]

\[ \text{Similarity (} D_1, D_2 \text{) from lexical and topic maps} = \frac{\text{Cosine lexical chains}(D_1, D_2) + JC_{\text{topic maps pattern}}(D_1, D_2)}{2} \]

\[ \text{Similarity (} D_1, D_2 \text{) from lexical and topic maps} = \frac{0.774 + 0.666}{2} \]

\[ \text{Similarity (} D_1, D_2 \text{) from lexical and topic maps} = 0.720 \]

The result obtained from this approach is far better than cosine similarity (=0.6) and Jaccard similarity (=0.42) respectively, for the same pair of documents that is \( D_i \) and \( D_j \) from the simplified collection. On a subset of NEWS20 dataset, we humanly annotated 50 documents in pair and it is humanly verifiable that the score assigned to the pair of documents \( D_i \) and \( D_j \) through our proposed similarity function is very close to human assigned similarity scores for the said 50 document dataset.
Chapter 7

Experimental studies

7.1 Introduction

This chapter describes the complete experimental studies performed in order to evaluate the efficiency of the proposed approach. The proposed approach MTS-DC is designed and implemented in Java programming language using integrated development environment NetBeans 8.0.2. The evaluation process comprises of extensive experimental evaluations and comparisons of our proposed method with some of the recently proposed approaches on four standard text mining datasets. The result and findings from these series of experiments will be discussed in next chapter. Here we explain our selection of datasets, preprocessing, comparable approaches, and evaluation measures in order to justify the performance of our proposed approach.

7.2 Datasets

Four standard document datasets are used to compare the quality of our clustering method. These four data sets are selected mainly due to the fact that most researchers, whose work is related to this study, have used the same datasets to report their results and comparisons. These datasets are:

(i) Reuters: The Reuters-21578, test collection of Distribution 1.0 is used. The collection appeared in Reuter’s newswire in the year 1987. The collection consists
of 22 data files. These files are in Standard Generalized Markup Language (SGML). It is a standard for how to specify a document markup language or tag set. Such a specification is itself a document type definition (DTD), which describes how documents are composed. The format of the available data, and six files describing the categories used to index data. The collection is available at http://www.daviddlewis.com/resources/testcollections/reuters21578/

(ii) NEWS20: It is also a popular data set among text mining community. It is mainly used for text classification and clustering measure for machine learning techniques. The data set consists of approximately 20,000 newsgroup documents. These documents are small text, which are further partitioned in 20 different classes. The data set is available at http://people.csail.mit.edu/jrennie/20Newsgroups/

(iii) OHSUMED: The OHSUMED collection consists of the 1987-1991 abstract from 270 journals and contains over 348,566 references from the MEDLINE database. MEDLINE database is a database of medical literature maintained by the National Library of Medicine (NLM) of U.S.A. Most of the references have abstracts and all have associated MeSH (Medical Subject Headings) indexing terms, with some of the MeSH terms marked as primary. The data set is available at http://davis.wpi.edu/xmdv/datasets/ohsumed.html

(iv) WebKB: The dataset is collected in January 1997 by Carnegie Mellon University (CMU) text learning group for World Wide Knowledge Base (WebKb) project. The 8,282 pages were manually classified into 7 classes: 1) student, 2) faculty, 3) staff, 4) department, 5) course, 6) project and 7) other. For each class the data set contains
pages from the four universities: Cornell, Texas, Washington, Wisconsin and 4,120 miscellaneous pages from other universities. The files are organized into a directory structure, one directory for each class. Each of these seven directories contains 5 subdirectories, one for each of the 4 universities and one for the miscellaneous pages. These directories in turn contain the Web-pages.

The key characteristics of these datasets are given in Table 4

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Name</th>
<th>Type</th>
<th>Source</th>
<th>Number of Documents</th>
<th>Number of Terms</th>
<th>Number of Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>20NG</td>
<td>20 newsgroups</td>
<td>Usenet</td>
<td>Newsgroups posts</td>
<td>18,824</td>
<td>91,652</td>
<td>20</td>
</tr>
<tr>
<td>Reuters</td>
<td>Reuters -21578</td>
<td>PlainText</td>
<td>NEWSwire</td>
<td>10,369</td>
<td>24,501</td>
<td>90</td>
</tr>
<tr>
<td>OHSUMED</td>
<td>MEDLINE</td>
<td>PlainText</td>
<td>Medical Journal</td>
<td>20,500</td>
<td>58,345</td>
<td>34</td>
</tr>
<tr>
<td>WebKB</td>
<td>CMU WebKB</td>
<td>web-pages</td>
<td>HTML pages</td>
<td>8,228</td>
<td>68,346</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 4- Key Characteristics of Dataset

We have selected subsets of data from these datasets. Why these subsets from the datasets are selected? the answer to this question is, the semantic processing is a time consuming activity and we like to expedite the experimental, hence a subset from these datasets are selected by sampling based on equal representation from different classes available from the dataset. We have selected three sample datasets from Reuters-21578. We named them Re1, Re2, and Re3, where Re is abbreviated for Reuters and 1, 2, and 3 represents three distinct datasets. The subsets N20a, N20b, and N20c are similarly selected from NEWS20. The dataset OHS1 and OHS2 are selected from OHSUMED and Kb1,Kb2 and Kb3 from WebKB, in the very same fashion. Each document of the test data sets has been pre-classified into one or more classes. This information is hidden during the clustering processes and is used to evaluate the clustering quality of each clustering algorithm in terms of accuracy. The summary of the selected subsets of data for the experiments are given in the Table 5.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>#of doc.</th>
<th>#classes</th>
<th># of unique terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re1</td>
<td>350</td>
<td>8</td>
<td>6218</td>
</tr>
<tr>
<td>Re2</td>
<td>700</td>
<td>10</td>
<td>7988</td>
</tr>
<tr>
<td>Re3</td>
<td>1400</td>
<td>15</td>
<td>12646</td>
</tr>
<tr>
<td>N20a</td>
<td>400</td>
<td>8</td>
<td>6570</td>
</tr>
<tr>
<td>N20b</td>
<td>800</td>
<td>10</td>
<td>9875</td>
</tr>
<tr>
<td>N20c</td>
<td>2000</td>
<td>18</td>
<td>16765</td>
</tr>
<tr>
<td>OHS1</td>
<td>400</td>
<td>8</td>
<td>8700</td>
</tr>
<tr>
<td>OHS2</td>
<td>800</td>
<td>16</td>
<td>12436</td>
</tr>
<tr>
<td>Kb1</td>
<td>500</td>
<td>4</td>
<td>8630</td>
</tr>
<tr>
<td>Kb2</td>
<td>1200</td>
<td>7</td>
<td>15400</td>
</tr>
<tr>
<td>Kb3</td>
<td>2000</td>
<td>7</td>
<td>28745</td>
</tr>
</tbody>
</table>

Table 5- Characteristics of Selected Datasets

7.2.1 Dataset Pre-processing

The selected datasets are preprocessed as per the requirement of the experiments. We have to use four different preprocessing routines for each kind of data, as the format and genre of each dataset is different. The general preprocessing includes tokenization of the documents to get the terms, stop-word removal in which we have used famous stop-word list available with Onix text retrieval toolkit. We have also used Porter stemmer (Porter, 1980) for stemming. The document representation as per our proposed approach is created using lexical chain algorithm and topic maps API Wandora. The details of document representation process along with a running example is explained in the chapter 3.
7.3 Comparison methods

We compare our MTS-DC, with four recently proposed approaches in the literature for the task of document clustering. It is a hierarchical clustering approach and the methods we have selected for comparison also use hierarchical clustering.

7.3.1 Baseline I

The baseline I for this experiment is set for Bag-of-Words (BOW) (Salton, 1971b) representation model for documents. It uses simple terms after the preprocessing as a feature. This baseline is set for experiment as this representation does not use any relationship with other terms and all the terms are independent. There is no word order or meaning representation in BOW (Zhang, Jin, & Zhou, 2010). Hence it can be used as a simple baseline for our experimental study.

7.3.2 Baseline II

The Baseline II for this experiment is a document representation that uses both lexical chains and topic maps terms for the documents. The process for lexical chains and topic maps extraction are run for each document and a combined representation is created form these two feature sets. Lexical chain contains an implicit relationship among words and topic maps also contains non-linear relationship but this representation does not support the non-linear relationship. Hence it is also a simple baseline for the experiments. One can think of this baseline as BOW which only contains lexical and topic maps features but no hybrid composition of each.
7.3.3 Group Average Hierarchical Agglomerative Clustering (GAAC)

Hierarchical clustering is a method of cluster analysis which tries to build a hierarchy of clusters. There are two main types hierarchical clustering: "top-down” version also called Divisive Hierarchical Clustering and "bottom up" version also called Agglomerative Hierarchical Clustering. The greedy approach to merge and split is further categorized into different types like single-link, complete-link and group average. In our experiment, Group Average Hierarchical Agglomerative Clustering (GAAC), where we have used a bottom-up approach. Initially each document is considered as one cluster and we continue to merge the two highly similar clusters (documents) in each iterative step and finally all documents contained in one single cluster. Obviously, the choice of Group Average Hierarchical Agglomerative Clustering (GAAC) is due to the fact that it produces a hierarchy of clusters which contains more information. On the other hand, it also suitable for text document clustering when we do not know much about the structures of the clusters, number of exact clusters, and membership of each cluster. Hence, in a nondeterministic situation it is the best choice for clustering. GAAC computes the average similarity of all pairs of documents, including pairs from the same cluster but self-similarities are ignored. GAAC evaluates cluster quality based on all similarities between documents, thus avoiding the pitfalls of the single-link (a variation of HAC where the similarity of two clusters is the similarity of their most similar members) and complete-link (another variation of HAC where the similarity of two clusters is the similarity of their most dissimilar members), which equate cluster similarity with the similarity of a single pair of documents.
7.3.4 Comparative Approaches

Our proposed MTS-DC approach is compared with four recently proposed approaches named as Frequent Itemset-based Hierarchical Clustering (FIHC) (Fung et al., 2003), Clustering based on Frequent Word Sequences (CFWS) (Li et al., 2008), Document clustering based on Topic maps (TMHC) (Rafi et al., 2011) and Dependency Graph based Document Clustering (DGDC) (Y. Wang et al., 2011).

Frequent Itemset-based Hierarchical Clustering (FIHC) technique proposed for document clustering is based on the idea of frequent itemsets mining technique that is very useful in market basket analysis. The main idea of this approach, is to consider each term of the document as item for shopping and different set of documents (cluster) share some frequent terms. A frequent itemset is a set of words that occur together in some minimum fraction of documents that can be grouped as a cluster. The approach has some clear advantages over previous proposed techniques for document clustering. First of all, it helps in reduce the dimensionality of the problem. The clusters can have a good description from these frequent terms. It was highly competitive approach at the time it was proposed (Fung et al., 2003) (Zhuang & Dai, 2004). We have implemented this approach as outline by the researchers as it is not available for performing experiments.

Clustering based on Frequent Word Sequences (CFWS)(Li et al., 2008) treats the text document as a sequence of words, and the frequent word sequences among documents are used as an implicit feature to perform clustering. The approach reduces each document to a compact frequent sequence of words (sequence length=2). It exploits the sequential occurrences of words in the document to extract the semantics richness from the document into the representation. It uses a generalized suffix tree (Zu Eissen et al., 2005) from all the compact documents to get the frequent word sequences that are shared among different sets of
documents. The set of documents that contain more common such sequences are grouped to form a candidate cluster. The researchers proposed two variations of this algorithm: one that uses common frequent sequences of words and other that uses common frequent word meaning sequences.

Document clustering based on Topic maps (TMHC) is another approach that exploits the lexical semantics for the task of textual document clustering. It first transforms the document into a topic maps based representation. It also defines a similarity measure on the topic maps data structures like: topic tag, association and tag-values. The algorithm reports improvement over FIHC and CFWS on three standard text mining datasets (Rafi et al., 2011).

In Dependency Graph based Document Clustering- (DGDC) (Y. Wang et al., 2011), each document is represented as a dependency graph where words correspond to nodes. It is used as a meta-descriptor of the document. The semantics between pairs of words are captured by using edges between nodes. A novel similarity measure is also proposed for calculating the similarity of documents based on their corresponding dependency graphs. The approach has the following steps: (i) construct a dependency graph for each document, (ii) calculate the similarity of each pair of documents based on the novel similarity measure, and (iii) generate the clusters. It uses Group-average Agglomerative Hierarchical Clustering (GAHC) algorithm for performing final clustering.

**7.4 Evaluation Metrics**

There are several measures for evaluating the quality of document clustering task. There are three main kinds of cluster evaluation (cluster analysis). Internal Evaluation is a process in which a clustering algorithm is evaluated on the data/objects that performed self-clustering. External evaluation is a process in which a clustering algorithm is evaluated on the data/objects
that are not used for clustering. These are the benchmark datasets, often human created, which consists of a set of pre-classified items. Relative evaluation is a validation criterion for two different clustering algorithms. Generally, in document clustering external evaluation is used to perform cluster analysis. We also justify the effectiveness of our proposed method by using standard cluster quality measures like F-measure, purity and entropy.

7.4.1 F-measure

F-measure is a commonly used measure in evaluating the effectiveness of clustering and classification algorithms. The F-measure uses a combination of precision and recall values of clusters. Assume that the document base (D) comprises of N documents and the clustering algorithm produces C={C_1, C_2,.. C_k} clusters. The document base has actual correct clusters C^*=\{C^*_1, C^*_2,.. C^*_l\}. Then, the recall of cluster j with respect to class i, rec(i,j) is define as |C_j \cap C^*_i| / |C^*_i|. The precision of cluster j with respect to class i, prec(i,j) is defined as |C_j \cap C^*_i| / |C_j|. F-Measure combines both precision and recall with the following formula:

\[ F(i, j) = \frac{2 \times \text{prec}(i, j) \times \text{rec}(i, j)}{\text{prec}(i, j) + \text{rec}(i, j)} \]

The F-measure for overall quality of cluster set C is defined by the following formula:

\[ F = \sum_{i=1}^{k} \frac{|G_i|}{N} \times \max_{i=1,2,..k} \{F(i, j)\} \]
7.4.2 Purity

Purity provides a measure of “goodness” for a clustering arrangement. The higher value of purity represents that the clusters are more homogeneous. Purity can be defined as the maximal precision value for each class $j$. The cluster purity indicates the percentage of the dominant class members in the given cluster. The overall purity of the cluster $C$, can be computed as the weighted average purity by the following formula:

$$ Purity = \sum_{j=1}^{k} \frac{|C_j|}{N} \cdot \max_{i=1,2,\ldots,k} \{Prec(i,j)\} $$

7.4.3 Entropy

Entropy provides a measure of “goodness” for a clustering arrangement. The lower value of entropy represents the clusters are more homogeneous. Given the clustering result $C$, we first calculate the class distribution data, i.e. for each cluster $j$ at some level, we compute $p_{ij}$ (probability that a member of cluster $j$ belong to class $i$). The entropy of each cluster $j$ can be calculated by the formula given below:

$$ E_j = -\sum_i p_{ij} \log p_{ij} $$

The total entropy of the clustering arrangement $C$ can be calculated by the sum of the entropies of each cluster weighted by the size of each cluster. This can be given by the formula:

$$ Entropy = -\frac{1}{\log k} \sum_{j=1}^{k} \frac{C_j}{N} \sum_{i=1}^l p_{ij} \log p_{ij} $$

Where $p_{ij}$ is the probability that a member of cluster $C_j$ belong to class $C^*_i$. 

93
Chapter 8

Results and Findings

8.1 Introduction

This chapter discusses the results of the extensive series of experimental studies carried out for evaluating the proposed approach “Multi-layer Topic maps based Semantic-rich Hierarchical Document Clustering” (MTS-DC). In order to justify the effectiveness of our proposed approach, we have selected 11 sampled datasets from the four standard text mining datasets as described in Chapter 7 on and implemented 7 algorithms for the task of clustering, along with one proposed algorithm. The experimental studies demand an independent and reliable evaluation, assessment, and verification of the clustering results. The evaluation of the clustering task is performed by using external cluster measures like purity, entropy and F-measure. The outcome of these evaluations clearly established the fact that MTS-DC is practical in producing clusters of high semantic quality. The improvement on purity, decrease in entropy is a clear indication of better clustering results.
8.2 External Evaluation of Clustering Task

The proposed approach to document clustering is evaluated by using external measures. The experimental data used for the study is already labeled. These data sets are standard and are widely used for the task of clustering and classification in text mining and machine learning domains. The basic idea of this is to compare the results obtained by unsupervised clustering with the ground truths of the given datasets, which is often manually annotated by the domain experts. The external evaluations used for this study are F-measure, Purity and Entropy. These measures are good to support the quality analysis of clustering task like: homogeneity, completeness, cluster size, quality and having a miscellaneous class (rag bag).

8.2.1 F-Measure

Document clustering as an unsupervised learning task often poses more challenges as neither the number of clusters nor the mapping between eventually know classes (target clusters) are provided to the clustering algorithm. The F-measure is a statistical measure often used in information retrieval for measuring the quality and accuracy of search results for both clustering and classification. The F-measure (or F-score) is calculated based on the precision and recall (Chapter 8). The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall which equally favors both. Figure 8.1 shows F-measure for our experiments, where the proposed MTS-DC performs better than other schemes, e.g., FIHC, CFSW, TMHC and DGDC.
Figure 6- F-Measure from the experiments

F-measure represents the accuracy of the clustering scheme. Our MTS-DC algorithm has better F-measure because we use a better document representation model for text documents. Our proposed document representation implicitly uses the relationships between words in similarity measure and hence assigns high similarity scores to document pairs which have more common semantics in term of lexical cohesiveness. The next best F-score comes from the topic maps based representation, as our approach uses lexical terms based topic maps construct our approach is better than TMHC and an increases in F-measure is an evidence of picking better semantics based patterns from the documents for clustering task. Both FIHC and CFSW use frequent patterns and CFSW obviously uses order information implicitly to perform better than FIHC. These two do not use semantics rich features like our lexical topic maps patterns that offers better non-linear relationships among terms hence these methods could not score higher on F-Measure. The FIHC and CFSW use a support level of 15% for the experiments. The results are compared with all methods by using GAHC and cutting the dendogram at a level
where it matches the actual number of clusters in the sampled datasets. The high values of F-Measures of Reuters, News20 and WebKB are due to the fact that these documents contain natural language text for the transmission of an implicit message or idea and hence the lexical chains are obvious from these datasets. OHSUMED datasets do not contain a lot of English lexical dictionary terms and generally a good way to extract lexical chains from these documents does not exist, hence F-measure are relatively low in this dataset.

### Table 6- F-Measure from the experiments

<table>
<thead>
<tr>
<th></th>
<th>BASE-I</th>
<th>BASE-II</th>
<th>FIHC</th>
<th>CFSW</th>
<th>TMHC</th>
<th>DGDC</th>
<th>MTS-DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re1</td>
<td>0.34</td>
<td>0.3</td>
<td>0.38</td>
<td>0.41</td>
<td>0.51</td>
<td>0.31</td>
<td>0.55</td>
</tr>
<tr>
<td>Re2</td>
<td>0.4</td>
<td>0.32</td>
<td>0.44</td>
<td>0.49</td>
<td>0.53</td>
<td>0.33</td>
<td>0.56</td>
</tr>
<tr>
<td>Re3</td>
<td>0.44</td>
<td>0.33</td>
<td>0.48</td>
<td>0.55</td>
<td>0.58</td>
<td>0.35</td>
<td>0.66</td>
</tr>
<tr>
<td>N20a</td>
<td>0.41</td>
<td>0.32</td>
<td>0.41</td>
<td>0.44</td>
<td>0.55</td>
<td>0.34</td>
<td>0.55</td>
</tr>
<tr>
<td>N20b</td>
<td>0.45</td>
<td>0.32</td>
<td>0.44</td>
<td>0.49</td>
<td>0.58</td>
<td>0.36</td>
<td>0.58</td>
</tr>
<tr>
<td>N20c</td>
<td>0.49</td>
<td>0.37</td>
<td>0.48</td>
<td>0.53</td>
<td>0.63</td>
<td>0.38</td>
<td>0.72</td>
</tr>
<tr>
<td>OHS1</td>
<td>0.33</td>
<td>0.28</td>
<td>0.19</td>
<td>0.22</td>
<td>0.34</td>
<td>0.31</td>
<td>0.44</td>
</tr>
<tr>
<td>OHS2</td>
<td>0.41</td>
<td>0.3</td>
<td>0.21</td>
<td>0.24</td>
<td>0.38</td>
<td>0.35</td>
<td>0.53</td>
</tr>
<tr>
<td>Kb1</td>
<td>0.39</td>
<td>0.28</td>
<td>0.49</td>
<td>0.53</td>
<td>0.56</td>
<td>0.39</td>
<td>0.58</td>
</tr>
<tr>
<td>Kb2</td>
<td>0.44</td>
<td>0.32</td>
<td>0.55</td>
<td>0.58</td>
<td>0.61</td>
<td>0.45</td>
<td>0.66</td>
</tr>
<tr>
<td>Kb3</td>
<td>0.5</td>
<td>0.38</td>
<td>0.58</td>
<td>0.61</td>
<td>0.62</td>
<td>0.44</td>
<td>0.73</td>
</tr>
</tbody>
</table>

### 8.2.2 Purity

In cluster analysis, purity is an external evaluation criterion for clustering. Purity is a simple, transparent and effective measure for cluster quality. In the external evaluation setting it refers to the amount of truly classified classes per cluster over the sample data. A purity value 1, indicates that the clusters obtained from the algorithm are identical with the sample data ground truth structures. High purity values indicate a good result. Figure 8.2 shows purity values from the experiments.
Figure 7- Purity from the experiments

The purity values from MTS-DC are clearly better than all comparative algorithms which is due to the fact that our document representation scheme implicitly takes care of lexical word chains and their relationships to other words and thus builds a higher semantics relatedness among features. Perhaps, it will be better if we say that our document representation scheme captures document level semantics from the lexical and topic maps, terms and structures. MTS-DC is much better than CFSW and FIHC as both use frequent words as features to cluster documents. CFWS purity is better than FIHC which is obviously mentioned by the researchers that it uses sequences rather than frequent pattern, hence the order of the words are implicitly there in the similarity calculation. Our experiments also validate the claims from CFWS regarding FIHC. The reason for poor performance of DGDC on purity is because the dependency graph from the text could not resolve the word sense. TMHC is again second best as it only uses topic maps features that is tags, values and associations for similarity measures and hence fails to build a lexical cohesion from the text. FIHC and CFWS both can only use term sequences (phrases or item-sets), hence cannot differentiate non-linear relationships.
between these units. On a close analysis we find that the terms from a cluster, labeled as "Tea Consumption" - have phrases like: "tea culture", "tea popularity", "tea lover", and "tea export" and when we link these phrases with topic maps patterns we get "tea export -> Kenya" and "Pakistan, Kenya" -> Countries, which can clearly distinguishes between class labels "Tea Industry" and "Tea Consumption".

<table>
<thead>
<tr>
<th></th>
<th>BASE-I</th>
<th>BASE-II</th>
<th>FIHC</th>
<th>CFSW</th>
<th>TMHC</th>
<th>DGDC</th>
<th>MTS-DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re1</td>
<td>0.54</td>
<td>0.38</td>
<td>0.58</td>
<td>0.68</td>
<td>0.71</td>
<td>0.53</td>
<td>0.81</td>
</tr>
<tr>
<td>Re2</td>
<td>0.51</td>
<td>0.45</td>
<td>0.53</td>
<td>0.62</td>
<td>0.68</td>
<td>0.51</td>
<td>0.79</td>
</tr>
<tr>
<td>Re3</td>
<td>0.48</td>
<td>0.48</td>
<td>0.48</td>
<td>0.61</td>
<td>0.62</td>
<td>0.53</td>
<td>0.78</td>
</tr>
<tr>
<td>N20a</td>
<td>0.58</td>
<td>0.46</td>
<td>0.66</td>
<td>0.71</td>
<td>0.74</td>
<td>0.66</td>
<td>0.83</td>
</tr>
<tr>
<td>N20b</td>
<td>0.54</td>
<td>0.47</td>
<td>0.63</td>
<td>0.72</td>
<td>0.73</td>
<td>0.65</td>
<td>0.81</td>
</tr>
<tr>
<td>N20c</td>
<td>0.51</td>
<td>0.49</td>
<td>0.61</td>
<td>0.71</td>
<td>0.72</td>
<td>0.61</td>
<td>0.79</td>
</tr>
<tr>
<td>OHS1</td>
<td>0.51</td>
<td>0.38</td>
<td>0.53</td>
<td>0.53</td>
<td>0.61</td>
<td>0.54</td>
<td>0.68</td>
</tr>
<tr>
<td>OHS2</td>
<td>0.48</td>
<td>0.41</td>
<td>0.51</td>
<td>0.53</td>
<td>0.64</td>
<td>0.51</td>
<td>0.65</td>
</tr>
<tr>
<td>Kb1</td>
<td>0.71</td>
<td>0.53</td>
<td>0.62</td>
<td>0.73</td>
<td>0.75</td>
<td>0.77</td>
<td>0.84</td>
</tr>
<tr>
<td>Kb2</td>
<td>0.63</td>
<td>0.59</td>
<td>0.62</td>
<td>0.71</td>
<td>0.73</td>
<td>0.73</td>
<td>0.82</td>
</tr>
<tr>
<td>Kb3</td>
<td>0.58</td>
<td>0.63</td>
<td>0.68</td>
<td>0.69</td>
<td>0.71</td>
<td>0.73</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Table 7- Purity from the experiments

8.2.3 Entropy

In cluster analysis, entropy is an external evaluation criterion for clustering. Entropy is also a simple, transparent and effective measure for cluster quality. Entropy provides a measure of “goodness” for a clustering arrangement. The lower value of entropy represents that the clusters are more homogeneous which means that the approach is able to identify the true classes from the sample data in substantial portion.
The relationship between purity and entropy is important for the document clustering task. Table 7 represents purity values for different approaches for selected sample datasets, similarly Table 8 represents entropy values for the same. High purity refers to the amount of truly classified class per cluster over the pre-categorized sample data while low value of entropy refers to how homogenous the resultant clusters are. Basically, a good clustering approach should maximize the purity and minimize the entropy of clusters in order to achieve high quality clustering. Figure 8.3 shows entropy values from the experiments. It is clearly evident from the Table 7 and Table 8 that the purity is higher than all comparative approaches and at the same time entropy is lower for the same.
It is clear from the Fig. 8 that MTS-DC achieved lower values of entropy in all datasets. The main reason is due to our document representation scheme which implicitly carries semantic relationships between words in linear and non-linear fashion, thus being able to produce more homogenous clusters in all datasets and perform much better than all comparative algorithms for the task of document clustering.
Chapter 9

Conclusion and Future Works

In this thesis a set of methods and algorithms are proposed to innovate the task of document clustering. The continuous and unbound growth of textual documents in proprietary and public information systems requires an effective and fast approach for taking knowledge out from these collections. Document clustering is a specialized clustering technique that has founded its niche in effectively browsing, filtering, managing and summarizing these collections. Information searching has become more important than the information itself. An ideal document clustering algorithm should produce clusters of high quality in terms of semantically relevant and apparently understandable results. This thesis proposed a document clustering approach named as "Multi-Layer Topic Maps based Semantic-Rich Hierarchical Document Clustering" (MTS-DC). There are two major contributions of our works: (i) a compact document representation is proposed which implicitly captures the semantics of the textual document, (ii) a novel similarity measure based on inferred information through topic maps data and structures is proposed. The representation scheme and similarity measure both take advantage of lexically cohesive structure along with topic map relationships. The methodology is scalable i.e. when provided a large document collection it is able to produce clusters of high semantic quality.

The main contribution of the thesis is a novel document representation scheme that is specifically suited for the task of document clustering. Our scheme first extracts lexical chains from the documents and exploits topic maps structure for the lexical chains. It takes advantage
of lexical cohesive structure along with topic map relationships to get a semantic based representation of document without any extra overhead. This representation implicitly overcomes the issue of word-order problem for the document clustering task. A collection of lexical chains for a document is generally regarded as an automatic abstracting of a document, hence it carries the contextual information about the document. On the other hand, topic maps transformation of a documents implicitly carries the non-linear relationships among groups of terms that also add semantics to the representation. Hence we are able to exploits the two fold advantages of semantic rich representations from the textual documents. We proposed a novel similarity measure that combine the effect of lexical chain and topic maps structures in the similarity calculations. An extensive series of experiments have been performed to justify the effectiveness of the proposed document clustering approach.

9.1 Conclusion

The proposed "Multi-Layer Topic Maps based Semantic-Rich Hierarchical Document Clustering" (MTS-DC) is very effective for the task of document clustering. The experimental studies on 11 different sampled datasets from four standard text mining datasets of different web-genre produce high purity value which is a clear indication of high quality clusters. The proposed approach clearly out performs some of recently introduced approaches for the task of document clustering. Frequent Itemset-based Hierarchical Clustering (FIHC), Clustering based on Frequent Word Sequences (CFWS), Document clustering based on Topic maps (TMHC) and Dependency Graph based Document Clustering- (DGDC) all four produced low F-Measure scores, low purity scores and high entropy scores on all 11 datasets. "Multi-Layer Topic maps based Semantic-rich hierarchical Document Clustering" (MTS-DC) is clearly a winner in all experiments. This is because our proposed document representation scheme
implicitly captures semantics via lexical chains and exploits topic maps structures for the lexical chains. Our representation is better in terms of utilizing document level semantics hence the compact and meta-descriptor of each document carries more information in the clustering process. It is equally applicable to offline and online settings of document clustering task and the approach has low computational cost as compared to other methods in experimental studies.

9.2 Future Works

The proposed work "Multi-layer Topic maps based Semantic-rich hierarchical Document Clustering" (MTS-DC) has several directions of possible extensions for future work. The document representation exploits lexical chains along with topic maps and the scheme takes advantage of lexical cohesive structure along with topic map relationships to get a semantic based representation of document. The meta-representation thus creates an automatic abstracting for each document. There are several possible extensions to this work:

- The proposed document clustering approach has many practical applications. One direction is to apply this technique on some specific application area along with application specific optimizations to see the outcome. For example: web search results can be clustered using this approach. The snippets for each cluster are generated to see the quality of these snippets.

- In the proposed approach each term, whether it is from lexical chain or from topic maps, has an equal effect on similarity calculation for a pair of documents. One possible direction is to introduce discriminative feature weighting for the features in this approach. Discriminative feature weighting has encouraging results for both text clustering and classification tasks.
The actual clustering is carried out using Group Average Hierarchical Agglomerative Clustering (GAAC), where we have used a bottom-up approach. There are several clustering algorithms and experimenting with them using the proposed document representation and similarity function would be a good direction. A linear time document clustering algorithm (Larsen & Aone, 1999) will be a smart choice to extend the approach to online document clustering.

Another possible extension of this thesis work can be using a semi-supervised approach to document clustering, using available information in terms of constraints. Some time background information is readily available prior to the clustering process in terms of known document pairs that can fall together into a single cluster (must-link pair) or that cannot be put into a single cluster (cannot-link pair). This information can be transformed into a constraint representation to guide the clustering process using these constraints. Topic maps constraint language can be used to model these constraints.
Bibliography


Affect analysis of text using fuzzy semantic typing. *From Data and Information Analysis to Knowledge Engineering*, 9(4), 334-341.


Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. IEEE Transactions on Knowledge and Data Engineering, 14(5), 1003-1016.


Wang, C., Song, Y., Roth, D., Zhang, M., & Han, J. (2016). World knowledge as indirect supervision for document clustering. ACM Transactions on Knowledge Discovery from Data (TKDD), 11(2), 13.


