TECHNIQUES OF KNOWLEDGE DISCOVERY IN TEXT

By

SHWETA

University Roll No. 2K11/PHD/COE/03

Under the guidance of

Dr. Rajni Jindal

HOD, Dept. of Computer Science & Engineering,
Delhi Technological University

Submitted in fulfilment of the requirement of the degree of

Doctor of Philosophy to the

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(FORMERLY DELHI COLLEGE OF ENGINEERING)

NEW DELHI– 110042

April, 2018

Declaration

I, Shweta, a part time scholar (University Roll No. 2K11/PHD/COE/03), hereby declare that the thesis titled **"Techniques of Knowledge Discovery in Text"** which is being submitted for the award of the degree of Doctor of Philosophy in Computer Science & Engineering, is a record of bonafide research work carried out by me under the supervision of Dr. Rajni Jindal, HOD, Department of Computer Science & Engineering, Delhi Technological University.

I further declare that the work presented in the thesis has not been submitted to any university or institution for the award of any diploma or degree.

Date: April ,2018

Place: New Delhi

Shweta

Roll No. 2K11/PHD/COE/03

Department of Computer Science & Engineering

Delhi Technological University

(Formerly Delhi College of Engineering)

Delhi-110042

DELHI TECHNOLOGICAL UNIVERSITY

(Govt. of National Capital Territory of Delhi)

MAIN BAWANA ROAD, DELHI – 110042

Certificate

Date: April ,2018

This is to certify that the work embodied in the thesis titled **“Techniques of Knowledge Discovery in Text”** submitted to Delhi Technological University for the award of the Degree of **Doctor of Philosophy** by **Shweta,** University Roll No. 2K11/PHD/COE/03 as a **part time scholar** in the Department of Computer Science & Engineering, Delhi Technological University, is an authentic work carried out by her under my guidance. This work is based on the original research and has not been submitted in full or in part for any other diploma or degree of any university to the best of my knowledge and belief.

Supervisor

Dr. Rajni Jindal

HOD, Department of Computer Science & Engineering

Delhi Technological University

(Formerly Delhi College of Engineering)

Delhi-110042

Acknowledgement

This thesis is the result of my dedication towards research work, during which I have been supported and advised by many people. It is a pleasure for me to express my gratitude and respect to all of them.

Firstly, I thank Almighty God for giving me strength, passion and inspiration to complete this study and achieve my dream.

It gives me immense pleasure to thank my mentor and supervisor Dr. Rajni Jindal, HOD, Department of Computer Science & Engineering for her deep insights and dedication to guide me through this research. She is a reputed academician, supportive supervisor and most importantly a good human being. It is her motivation that has always given me support even in the times of stress and encouraged me to move ahead. Without her encouragement and constant support this work would not have been possible. It is her invaluable guidance and suggestions that has shaped my research and led to this dissertation. It is a privilege for me to work in her supervision.

I am thankful to Prof. S. Indu, DRC Chairman, Delhi Technological University for her help, cooperation and support. The Department of Computer Science & Engineering at Delhi Technological University is known for maintaining high academic standards since its inception. It was an honor for me to pursue doctoral study in this prestigious department.

I would like to express special thanks to Prof. Payal Pahwa, Principal, Bhagwan Parshuram Institute of Technology for her constant help and support. I am also thankful to Dr. Deepali Virmani, Head, Computer Science & Engineering Department, Bhagwan Parshuram Institute of Technology and Dr. Bhawna Suri for their guidance and wholehearted support.

Last but not the least I am grateful to my family who have always motivated and encouraged me through this journey. My special thanks to my husband Mr. Madhur Taneja who has helped me throughout the period of research and my children Mehak and Aryan for keeping patience and giving me time to complete the work. My family is my pillar of strength and I am always grateful to all of them for their lifelong support, everlasting love and sacrifices that has led to the successful completion of this study.

Date: April ,2018

Place: New Delhi.

Shweta

Roll No. 2K11/PHD/COE/03

Department of Computer Science & Engineering

Delhi Technological University

(Formerly Delhi College of Engineering)

Delhi-110042

Abstract

In this digital era, a major portion of the information is stored in text documents. The amount of data stored in these documents is exponentially increasing day by day. Analysis of such a vast data is not possible manually. This led to the development of Knowledge Discovery techniques in Text documents (called as KDT). KDT helps us to discover the useful information or knowledge from text documents.

The extraction of useful information from the text documents is termed as text mining. There are a lot of challenges in the field of text mining. Firstly, the text documents occur in a free natural language form like online news stories, e-mail messages, reports, legal documents etc. These documents are unstructured in nature. It is necessary to convert these unstructured text documents into a structured form. Secondly, there is an immense need to organize and manage the text documents efficiently. Text categorization plays an important role in organizing the text documents efficiently. Given a collection of text documents and a set of pre-defined classes/categories, the technique of text categorization assigns a particular class to each text document. There are two types of text categorization: single-label and multi-label. In single-label, each text document belongs to a single category, whereas in multi-label, each text document belongs to more than one category. Most of the real-world documents are multi-label in nature.

This thesis aims at exploring the existing techniques of knowledge discovery in text documents. We studied the existing techniques of knowledge discovery in text documents and came up with the following challenges:

First - The need to convert unstructured text documents to a structured form. To meet this challenge, a framework called as U-STRUCT is proposed that converts an unstructured text document to a structured form. It is a generic framework that can be applied to all domains.

Second - the vast information available in text documents is needed to be organized and managed. For this challenge, the technique of text categorization is taken and a detailed survey of available methods for text categorization technique has been carried out. The existing methods of text categorization have certain limitations. We have to overcome those limitations and suggest a better method for text categorization.

Third - An efficient method is needed for single-label text categorization. To meet this challenge, a Lexical based algorithm called as LKNN is proposed for single-label text categorization. This algorithm is implemented on two datasets: research articles of computer science domain and Ohsumed collection. The standard performance metrics like Recall, Precision, F-measure are calculated to measure the performance of LKNN algorithm. It has shown a good performance.

Fourth - It is required to develop an efficient method for categorization of multi-label documents as most of the real-world documents are multi-label in nature. Therefore, the algorithm proposed for single-label text categorization is extended to multi-label text categorization. And a modified Knowledge Discovery process known as Lexical-Semantics based Knowledge Discovery process for Text documents (LS-KDT) is proposed. The proposed process is divided in seven phases: Text Document Collection, Data Pre-processing, Lexical Analysis, Semantic Analysis, Classification, Ranking of labels and Knowledge Discovery. The proposed LS-KDT process is designed and implemented.

Thereafter, the performance of LS-KDT process is compared in three ways.
Firstly, the performance is compared with ACM Digital Library Results. The research articles are randomly taken from ACM digital library. These articles belong to two domains: computer science and medical domain. ACM digital library uses CCS tool to categorize the research articles. This tool displays the hierarchy of classes to which a research article belongs. Our proposed LS-KDT process also displays the hierarchy of classes and sub-classes to which a research article belongs. The standard performance metrics like Recall, Precision and F-measure are used for comparison.

Secondly, the performance is compared with the results of IEEE Xplore digital library. The articles are randomly selected from IEEE Xplore database. Again, research articles belonging to two domains are taken. One is those that belong to computer science and other is the articles belonging to the medical domain. IEEE Xplore digital library inserts four types of keywords with each research article. These are: IEEE KW, INSPEC Controlled Indexing, INSPEC Uncontrolled Indexing and Author KW. It is noticed that keywords in INSPEC Controlled Indexing includes the keywords of INSPEC Uncontrolled Indexing as well as IEEE KW. Therefore, to prove our work, Controlled Indexing keywords are taken, and their domain/ broad category is identified. The results are compared with the broad categories displayed by our proposed process.

Thirdly, the performance of the proposed process is compared with the existing multi-label methods on standard performance metrics like Accuracy, Precision, Hamming loss and F-measure. The proposed process has shown promising results.

The proposed Knowledge discovery process will help the research community to specify the exact categories to which a research article belongs in a more accurate way. It will aid the journal editors to assign to reviewers the research papers or articles in a systematic manner. The accurate categorization of articles helps the digital libraries, databases, repositories or online resources to efficiently store or search the articles. In future, the proposed LS-KDT process can be tested on research articles of other domains or other text documents like legal documents, reports etc.

Table of Contents

List of Abbreviations

1	U-STRUCT	Algorithm to convert Unstructured text document to a Structured form
2	LKNN Algorithm	Lexical Based K-Nearest Neighbor Algorithm
3	LS-KDT	Lexical-Semantics based Knowledge Discovery process for Text documents
4	CCS Tool	Computing Classification System Tool
5	SOM	Self-Organizing Map
6	KNN Algorithm	K- Nearest Neighbor Algorithm
7	SVM	Support Vector Machines
8	RAkELd	Random k label-set method for multi-label classification
9	ECC	Ensemble Classifier Chains
10	ML-KNN Algorithm	Multi-Label K-Nearest Neighbor Algorithm
11	PoS	Part-of-Speech
12	MESH	Medical Subject Headings
13	SVD	Singular Value Decomposition
14	ESA	Explicit Semantic Analysis
15	BR	Binary Relevance
16	WEKA Tool	Waikato Environment for Knowledge Analysis Tool
17	MEKA Tool	Multi-Label version of WEKA Tool
18	LLSF Method	Linear Least Squares Fit Method

List of Figures

List of Tables

1 INTRODUCTION

This chapter gives a brief introduction of basic concepts like KDD, KDT, Text Mining, Applications and Challenges of text mining. Further, motivation of work is given, and problem statement is defined. This is followed by a brief description of datasets and performance metrics used in research. In the end, contribution of the thesis and organization of the thesis is described.

Nowadays, there is a widespread use of internet and online databases. This has led to the growth of data in textual documents. It is very difficult to manage this data manually. This has resulted in the rise of the field of data mining and knowledge discovery. The field of knowledge discovery and data mining emerged in late 1980's.

1.1 Knowledge Discovery in Databases (KDD)

The term Knowledge Discovery in Databases (KDD) is a process of extracting or mining useful information or knowledge from huge amounts of data (Fayyad, 1996; Chen et al., 1996; Han, Pei & Kamber, 2011). The term data mining is a step in Knowledge Discovery in Databases (KDD) process. The figure 1.1 presents an overview of KDD process.

The steps of KDD process are:

a) **Data cleaning**- It refers to remove noise or any inconsistency in raw data.
b) **Data integration**- In this step, the data from multiple sources is merged and stored.
c) **Data selection**- It refers to the selection of the data that is relevant for task analysis from the database.
d) **Data transformation**- In this step, the data is transformed and converted in a uniform format with the help of certain operations.
e) **Data mining**- It is an essential step of KDD process as here intelligent methods are applied to discover interesting patterns from the data.

f) **Pattern evaluation**- This step refers to the evaluation of interesting patterns in data.

g) **Knowledge presentation**- In this step, visualization methods are used to present the mined knowledge to the user.

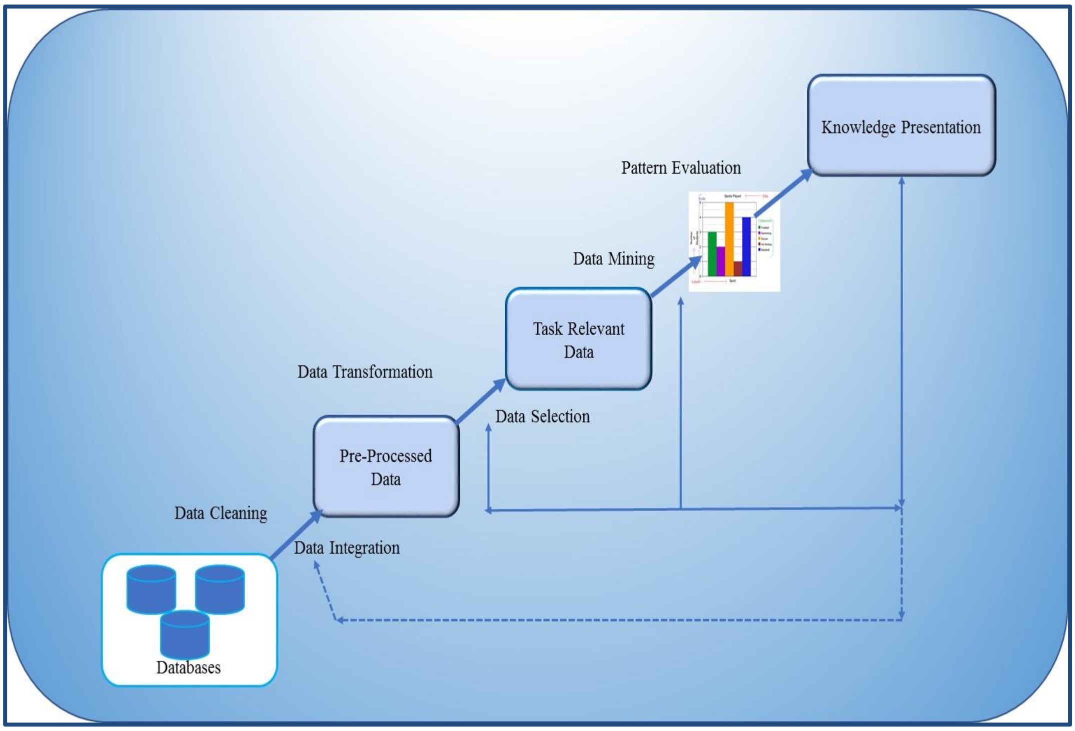


Figure 1.1: Knowledge Discovery in Databases (KDD) Process

There are various techniques of data mining like Classification and Prediction, Clustering, Association Rules, Memory Based Reasoning, Neural Networks, Genetic Algorithm etc. These techniques are applied in almost every field like Education, Sports, Government, Risk Analysis, Market Analysis, Telecommunication industry, Retail industry, Banking and Finance and many more.

The field of data mining deals with structured data like relational tables, transactional tables etc. But in real world, most of the available information is present in the form of text documents. For example, news articles, research papers, books, legal documents, digital libraries, e-mail messages, web pages etc.

Nowadays, there is a rapid growth of information available in electronic form such as electronic documents, government documents, business documents etc. This has led to the growth of text mining.

1.2 Knowledge Discovery in Text (KDT)

To discover useful information or knowledge from the text documents, a process known as Knowledge Discovery in Text (KDT) is used (Stavrianou, 2007; Aggarwal, C., 2012). This process discovers knowledge from text documents.

The whole process of KDT is divided into steps known as phases (Feldman R. & Dagan, 1995; Hearst, 1999). The phases in the process of KDT are shown in figure 1.2.

Following are the phases in the process of KDT: -

a) **Text document collection** - In the beginning, a text document collection is built. These text documents are unstructured in nature.

b) **Text Pre-processing** - In the second step, Text Pre-processing is done to prepare raw data for text mining. Text Pre-processing tasks include - text clean up, stop words removal, stemming, tokenization etc.

c) **Text Transformation** - The next step is Text Transformation. In this step, the text document is represented by the words or features it contains and their frequency of occurrences. Two main approaches are used - Bag of words and Vector Space Model.

d) **Feature Selection** - After this, Feature or Attribute Selection is done, which means to select a subset of features to represent the entire document. This step is also called as Dimension Reduction.

e) **Data Mining** - After this step, Data Mining comes into picture. At this point, the text mining process merges with the traditional data mining process. Now we have a structured database on which the standard data mining techniques can be applied. The different data mining techniques are Classification, Prediction, Clustering, Association Rules, Visualization etc.

f) **Knowledge Presentation** – In this step, knowledge presentation methods are used to represent the mined knowledge to the user.

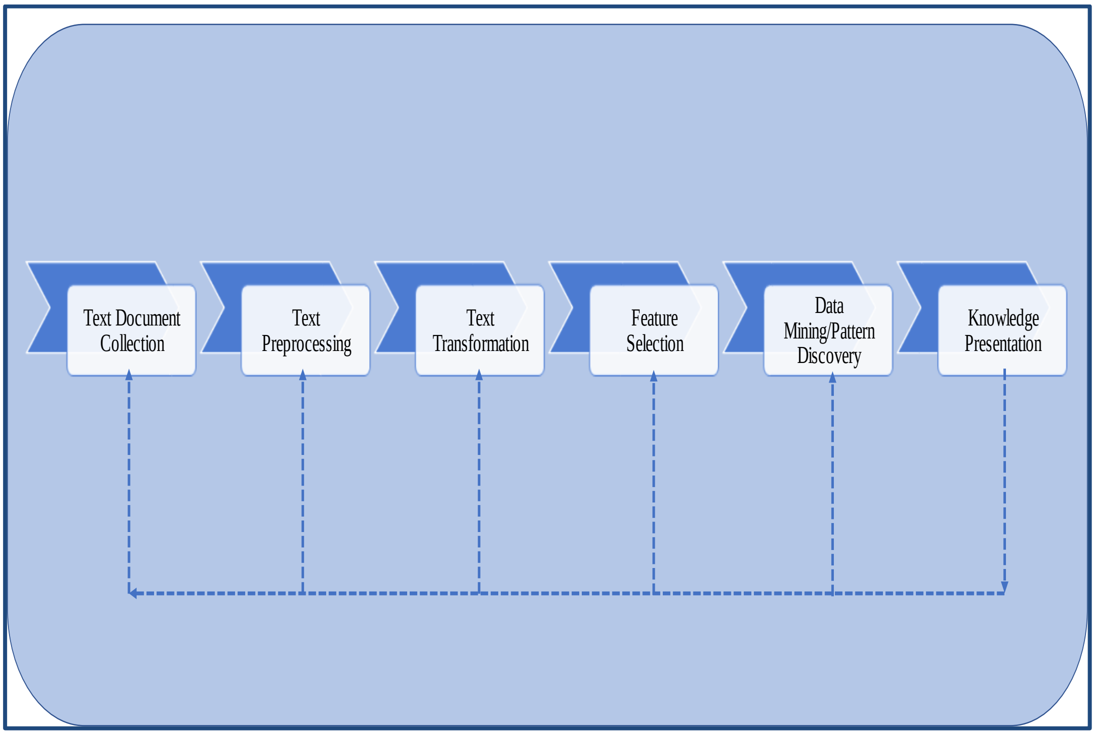


Figure 1.2: Knowledge Discovery in Text (KDT) Process

1.3 Text Mining

The field of text mining has emerged from data mining. In text mining, the basic element is a text document. Nowadays, lot of the information is stored in text documents. The text documents occur in free natural language form i.e. they may be unstructured or semi- structured in nature. For example, news stories, social media data etc. are unstructured in nature. In semi-structured documents, data is neither completely unstructured nor completely structured. For example, a text document contains a few structured fields like title, authors, publication date etc. And, also it may contain some unstructured components like abstract etc. The number of documents in these collections can vary from thousands to crores and even millions.

There are various techniques of text mining like Document Clustering, Link Analysis, Sentiment Analysis, Text Summarization, Text Categorization etc. In Document

Clustering, clustering algorithms are used to organize the text documents. Link analysis deals with finding interesting patterns and relationships in data. Sentiment analysis technique uses algorithms to find opinions in data. In text summarization, methods are used to generate summaries in data. In text categorization technique, algorithms are used to assign the text document in various categories.

1.4 Applications of Text Mining

Text mining has a lot of practical applications. A few of them are given below:

a) Automated organizing of documents in digital libraries: The digital libraries contain vast amount of data and retrieving the information manually from them, is a difficult task. So, text mining tools and techniques are used to automate the organization and management of data in digital libraries.

b) To explore bio-medical research reports: The bio-medical research reports contain huge collection of data. For example, Pub med is an online repository of National library of medicine, USA (Mehnert R., 1997). It consists of huge collection of bio-medical research papers (from 1966 to present). Text mining techniques are used to extract useful information from these documents and manage it. These techniques help in identifying interesting patterns and relationships in data.

c) To evaluate public opinions from social media data: For example, text mining methods can be used in review sites or blogs to develop opinions on data.

d) To filter spam mails from e-mail messages: Spam mails are junk mails. These can be filtered out by using text mining methods and techniques.

e) Cybercrime Prevention: Cybercrimes are the crimes that are based on internet. Text mining methods are used to develop anti-crime applications and thus prevent them.

f) Fraud detection: Text mining techniques are used to identify and detect the frauds. For example, Insurance companies use text analysis methods to prevent frauds.

g) Banking and corporate finance: Many text mining tools are used to analyze the textual data present in banks, help to find relationships and patterns in data, analyzing the trends in specific transactions or persons etc.

h) Text mining techniques are also used in patent research by various large companies i.e. to study the patent development policies and make use of existing patent assets.

1.5 Challenges of Text Mining

The researchers face a number of challenges in the area of text mining. A few of them are given below:

1.5.1 Data is not well organized and labeled

The data in text documents exists in free natural language form. It is either unstructured or semi-structured in nature i.e. the data is not well organized. So, the first step in text mining is to organize or arrange the data in text documents so that some meaningful information can be extracted from it.

1.5.2 Ambiguity in data

The data in text documents suffers from the problem of ambiguity. For example, apple is the name of a fruit and also the name of a company. The ambiguity can occur at many levels: at lexical level, syntax level or semantic level. This problem should be solved so that useful information can be gathered and used in mining the textual data.

1.5.3 Large and Noisy text datasets

The size of text datasets is very large. Apart from that, the real-world data in text documents is raw. It is a challenge to process this large and noisy text data.

1.5.4 Pre-processing of text documents

Pre-processing is an important step in the field of mining. Because it helps in the preparation of raw data for mining. The text documents possess large size and are unstructured in nature. Pre-processing of these documents is another challenge. Efficient methods are needed to pre-process these documents.

1.5.5 Multilingual Text Mining

The text documents are dependent on concepts of language whereas data mining is language independent. It is a challenging task in text mining to process multilingual text documents and obtain useful information from them.

1.6 Motivation of Work

Due to a great increase in amount of internet and digital data, the field of text mining has gained a lot of importance nowadays. Firstly, the nature of text documents is unstructured, so it is a very challenging task to mine useful information from these documents. These text documents must be converted into structured form to obtain useful information or knowledge from them.

Secondly, the text documents possess a huge size and a large number of features. Due to this fact there is an immense need to organize and access this vast information efficiently. The technique of text categorization comes into picture. It assigns a predefined category or class to the text documents. Our focus in this research is on text categorization.

In real world, apart from single-label text documents there are many multi-label text documents also, which can be categorized into more than one category. Therefore, we have worked on both single-label and multi-label text documents.

In this work, we have taken the problem of categorization of text documents. And a Lexical based algorithm (called as LKNN) is proposed for single-label text categorization. Further, the single-label categorization algorithm is extended into a modified knowledge discovery process known as LS-KDT (Lexical-Semantics based Knowledge Discovery process for Text documents). This proposed process helps in automated categorization of multi- label text documents.

1.7 Problem Statement

This thesis aims at exploring the techniques of knowledge discovery in text documents. This overall problem can be divided into following four sub problems addressed in thesis. a) converting unstructured text documents to a structured form, b) conducting a survey of existing methods for text categorization technique, c) suggesting a method for single-label text categorization, d) proposing a knowledge discovery process for multi-label text documents and further implementing the proposed process.

The first sub problem addressed in this thesis is the conversion of unstructured text documents to a structured form. The text documents occur in the form of natural language text that is unstructured in nature. The text documents can be online news stories, legal documents, reports, scientific research papers, contracts, e-mails, spread sheets, medical and healthcare reports. There is a need for the conversion of unstructured text documents to a structured form. In this work, we have proposed a framework called U-STRUCT that converts an unstructured text document to a structured form.

The second part of the problem is to focus on text categorization technique. In this sub problem, we have studied the various methods available for text categorization and performed a survey on these methods. The existing methods of text categorization have certain limitation. We have to overcome these limitations and suggest a better method for text categorization.

Given a set of text documents and a set of predefined categories, the technique of text categorization assigns categories to the text documents. In the third part of the problem, we have proposed Lexical based algorithm (LKNN) for single-label text categorization.

The fourth part of the problem is to extend the single-label text categorization to multi-label text categorization. In this part, a knowledge discovery process is proposed for multi-label text documents. The proposed process consists of seven phases namely Text document collection, Data pre-processing, Lexical Analysis, Semantic Analysis, Classification, Ranking of labels and Knowledge Discovery. In the next, the proposed knowledge discovery process is designed and implemented.

1.8 Datasets Used

In our work, we have used different datasets for the performance evaluation of the proposed work. They are listed below in brief. The details of these datasets are given in the chapters.

a) Research articles of Computer Science domain
b) Ohsumed Collection
c) Research articles from ACM digital library
d) Research articles from IEEE Xplore
e) Multi-label text datasets like Enron, Slashdot and Bibtex

The proposed LKNN algorithm is used for single-label text categorization of documents. It is implemented on two datasets. One is a dataset of research articles of computer science domain and second is Ohsumed Collection. The details of these datasets are given in chapter 4.

The proposed LS-KDT process is used for multi-label text categorization of documents. It is implemented on research articles that are selected from ACM digital library and IEEE Xplore digital library. These articles belong to two domains: Computer Science and medical domain. The details of these datasets are given in chapter 6.

The performance of proposed LS-KDT process is also compared with multi-label datasets like Enron, Slashdot and Bibtex. The details of these datasets are also given in chapter 6.

1.9 Performance Evaluation Metrics

To evaluate the performance of the proposed work, different performance metrics are used. The most common metrics used are: Recall, Precision, F-measure, Accuracy etc.

To understand these metrics, there are some important terms (Rajpathak D.G., 2013). These are: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), confusion matrix etc.

If we have a 2-class classification problem, TP is the number of classes actually positive and also predicted by the classifier as positive. TN is the number of classes actually that are negative and also predicted as negative. FP is the number of classes actually that are negative but are predicted as positive. FN is the number of classes that are actually positive but are predicted by the classifier as negative. These metrics can be shown with the help of a table called as confusion matrix, as shown in table 1.1 (Yuan P. et al., 2008).

Table 1.1: Confusion Matrix

Predicted Class			
		Positive	**Negative**
Actual Class	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Precision (also called as positive predictive value) is a measure of exactness which is, the percentage of tuples that the classifier labelled as positive is actually positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{1.1}$$

Recall (also called as True Positive Rate or Sensitivity) is a measure of completeness and is the percentage of positive tuples that the classifier labelled as positive.

$$\text{Recall}= \frac{\text{TP}}{\text{TP+FN}} \tag{1.2}$$

F-measure is a harmonic mean of Precision and Recall.

$$\text{F-measure}= \frac{\text{2* Precision* Recall}}{\text{Precision +Recall}} \tag{1.3}$$

Accuracy is a measure of showing how good a model is. It is a proportion of the correct predictions made by the classifier out of the total predictions made by it.

$$\text{Accuracy}= \frac{\text{TP+FN}}{\text{TP+FN+FP+TN}} \tag{1.4}$$

In multi-label text categorization, the measures Recall, and Precision are extended to Micro-average and Macro-average. In Micro-average, the values of Recall and Precision are summed up for all individual categories. In Macro-average, the values of Recall and Precision are calculated for each individual category first and then its average is taken.

Similarly, F-measure can also be estimated by taking Micro-average and Macro-average. The details of these performance metrics are given in chapter 4 and chapter 6.

1.10 Contribution of the thesis

The overall objective of this research is to gain deeper insights into the field of text mining and to develop techniques of knowledge discovery from text. The main contribution of work is to a) convert unstructured text documents to a structured form,

b) perform a survey of the state of art methods used for text categorization, c) propose a method for single-label text categorization, d) extend the method for single-label text categorization to multi-label text categorization and therefore propose a knowledge discovery process.

The details of the contribution of the thesis are provided below:

a) To convert unstructured text documents to a structured form.

   i. Pre-processing is an essential step in text mining. As the text documents exist in the form of free natural language so there is a need for efficient pre-processing techniques. More studies should be conducted on various methods used for pre-processing of text documents as this greatly affects the mining results.
   ii. Hence, we have proposed a framework called U-STRUCT that converts an unstructured text document to a structured form.
   iii. These methods of text pre-processing can be used by research community for the conversion of unstructured data to a structured form.

b) To perform a survey of the state of art methods used for text categorization.

   i. There are various text classifiers used in research like K-Nearest Neighbor (KNN) method, Decision tree, Naïve bayes, SVM etc. Our task is to explore the various text classifiers on the basis of certain parameters and suggest a better method for text categorization.

c) To propose a Lexical based algorithm known as LKNN for single-label text categorization.

   i. This method can be used by researchers to solve the problems in the field of automated text categorization.

d) To extend the proposed approach for single-label text categorization to multi-label text categorization.

i. Most real-world text documents are multi-label in nature. In our work, the proposed lexical approach is extended and developed into a knowledge discovery process.
ii. We have proposed a Lexical-Semantics based Knowledge Discovery process for Text documents called as LS-KDT process.
iii. The proposed process can be used by the research community for the automated categorization of multi-label text documents.

1.11 Organization of thesis

The organization of the chapters of the thesis is as follows:

Chapter 2: This chapter describes the related work done in the field of KDT process by different researchers. KDT process has been used in different domain areas. A brief survey about the same has been conducted and given in this chapter.

Next, in this chapter we have focused on the important technique of text categorization. Work done in single-label and multi-label text categorization is given. In multi-label text categorization, the concept of Ranking of labels is also discussed.

After the literature survey, research gaps are identified and then the proposed work is formulated accordingly.
The following paper has been published on this work:

a) Jindal, R. & Shweta (2013). Text Categorization – A Review. Proceedings of Third International Conference on Computational Intelligence and Information Technology, CIIT 2013, held during Oct 18-19, 2013 in Mumbai, India. The proceedings are published in Elsevier. [Online] http://searchdl.org/public/book_series/AETS/7/126.pdf

Chapter 3: This chapter presents our proposed U-STRUCT framework. The proposed framework is used for the conversion of unstructured text documents to a structured form. This chapter gives the need of the framework followed by the detailed explanation of its components.

The following paper has been published on this work:

a) Jindal, R., & Taneja, S. (2013). U-STRUCT: A Framework for Conversion of Unstructured Text Documents into Structured Form. In Advances in Computing, Communication, and Control (pp. 59-69). Springer, Berlin, Heidelberg.

[Scopus indexed]

Chapter 4: Our proposed Lexical KNN (LKNN) algorithm for single-label text categorization is given in this chapter. The proposed algorithm uses the standard ACM Computing Classification system. It is also discussed in this chapter. The detailed flow diagram of the proposed algorithm and datasets used are given. The performance of the proposed LKNN algorithm is compared with traditional KNN algorithm.

The following are the papers to prove our work on LKNN algorithm:

a) Jindal, R., & Taneja, S. (2015). A Lexical Approach for Text Categorization of Medical Documents. Procedia Computer Science, 46, 314-320. The publication is made available on sciencedirect.com. http://www.sciencedirect.com/science/journal/18770509/46.

[Scopus indexed]

b) Jindal, R. & Shweta (2017). A Novel Weighted Linguistic Approach to Text Categorization. Published in International Journal of Computer Applications (IJCA)(0975 – 8887), 80(2), 9-15. December 2017. http://www.ijcaonline.org/archives/volume180/number2/jindal-2017-ijca-915922.pdf

Chapter 5: In this chapter we give our proposed LS-KDT process for knowledge discovery in text documents. The proposed process works for multi-label categorization of text documents. It is subdivided into seven phases. These seven phases are: Text Document Collection, Pre-processing, Lexical Analysis, Semantic Analysis, Classification, Ranking of labels and Knowledge discovery. All the phases of proposed LS-KDT process are discussed in detail in the rest part of this chapter.

In Text Document Collection phase, the text documents are collected and in Pre-processing phase, stop words are removed.

Lexical Analysis which is the third phase of the proposed LS-KDT process is described using the detailed flow diagram of Lexical Analysis phase along with its explanation and the pseudo code.

The next phase which is Semantic Analysis - begins with stating the concept of Dimension Reduction. The complete details of this phase are explained further.

The next two phases are: Classification and Ranking of Labels. The detailed explanation about these phases is given next in this chapter.

And further, the details of the last phase of the proposed LS-KDT process that is, Knowledge discovery is given.

The following papers are published/ accepted on this work.

a) Jindal, R., & Taneja, S. (2017). A lexical-semantics-based method for multi-label text categorization using word net. International Journal of Data Mining, Modelling and Management, 9(4), 340-360. Publisher: Inderscience, [Online]: https://www.inderscienceonline.com/doi/pdf/10.1504/IJDMMM.2017.088412.

[Scopus indexed]

b) Jindal, R. & Shweta (2017). A Modified Knowledge Discovery Process in the Text Documents. Accepted for publication in International Journal of Innovative Computing, Information and Control (IJICIC).

[Scopus indexed]

c) Jindal, R. & Shweta (2016). A Wordnet Based Semantic Approach for Dimension Reduction in Multi-label Text Documents. International Journal of Control Theory and Applications, 9(40), 267-274© International Science Press.
[Online]
http://serialsjournals.com/articles.php?volumesno_id=1145&journals_id=268&volumes_id=848.

[Scopus indexed till 2016]

d) Jindal, R., & Taneja, S. (2015, November). Ranking in multi-label classification of text documents using quantifiers. In 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), (pp. 162-166). IEEE.
[Online]
ieeexplore.ieee.org/iel7/7468595/7482142/07482177.pdf.

[Scopus indexed]

Chapter 6: In this chapter, we have done performance evaluation of our work. The chapter begins with showing the results of the proposed LS-KDT process on a sample research article. Further the results of the proposed LS-KDT process are compared with the results of two digital libraries: ACM digital library and IEEE Xplore. The results are shown on two datasets: one is the dataset of computer science research articles and the other one is medical articles. The standard performance metrics like Recall, Precision and F-measure are calculated to measure the performance which is then compared with the existing multi-label methods.

The following paper is communicated from this work:

a) Rajni Jindal and Shweta (2018). A Novel Method for efficient multi-label text categorization of research articles, Communicated to an International Conference.

Chapter 7: This chapter concludes the thesis with the contributions of our work and outlines some directions for further research in this topic.

Appendix: It gives a list of computer science research articles dataset used in the experimentation work.

List of Publications: This section gives the list of published/ accepted/ communicated papers relating to this research work in International/National Journals/Conferences of repute.

References: This section is the list of references referred in this research work.

2 LITERATURE SURVEY

This chapter deals with the related work done in the field of KDT process by different researchers. KDT process has been used in different domain areas. A brief survey about the same has been conducted and given in this chapter. Next, our focus in this work is on the important technique of text categorization. There are two types of text categorization: single-label and multi- label. Related work done on both the types of text categorization is given. Research questions are formulated, Research gaps are identified and further based on it, proposed work is stated.

2.1 Introduction

The aim of literature survey is to make a detailed study of the existing work, to identify the research gaps and to propose a solution for the same. We have followed the following steps for literature survey.

a) Planning the Review
   - Identify the need of Review

b) Conducting the Review
   - Develop Research Questions (RQs)

c) Reporting the Review
   - Report review category wise
   - Identify the research gaps

d) Concluding the Review
   - Provide future direction

2.2 Planning the Review

In the present real-life scenario, the usage of internet and online text documents have increased to a great extent. There a lot of challenges in text mining. Some of them are:

a) Most of the text documents are unstructured in nature
b) Due to huge size of text documents, there is an immense need to organize and manage this vast information
c) To handle single-label and multi-label text documents

Our research is concerned with the above issues and therefore we have decomposed our research problem into four sub problems given below.

a) To convert unstructured text document to a structured form
b) To study the existing methods of text categorization technique
c) To develop a method for single-label text categorization
d) To design a method for multi-label text categorization

2.3 Conducting the Review

Before starting the literature survey, some research questions are formulated. They are listed below:

- **RQ#1:** What is the need for Knowledge discovery in text?

  (Solution: *Refer Chapter 1***)**

- **RQ#2:** How to organize and access the vast textual information efficiently?

  (Solution: *Refer Chapter 1***)**

- **RQ#3:** What are the existing methods of text categorization technique?

(**Solution:** *Refer Chapter 2*)

- **RQ#4:** What are the existing methods of Knowledge discovery in text?

  (**Solution:** *Refer Chapter 2*)

- **RQ#5:** How to convert a text document (occurring in a free natural language form) to a structured form?

  (**Solution:** *Refer Chapter 3*)

- **RQ#6:** How to categorize single-label text documents?

  (**Solution:** *Refer Chapter 4*)

- **RQ#7:** How to categorize multi-label text documents?

  (**Solution:** *Refer Chapter 5*)

2.4 Reporting the Review

In this section we have assessed the reviews based on four sub problems as listed in section 2.2 and finally the research gaps are identified on this basis.

2.4.1 A Review on Knowledge Discovery in Text (KDT) Process

Nowadays, there is a rapid growth of online data that has led to the need of developing better knowledge discovery systems for retrieving relevant information. The KDT process deals with extraction or mining useful information or knowledge from text documents.

The process of knowledge discovery has been used in various domain areas. An ontology-based system was proposed for automotive domain (Rajpathak, D.G., 2013). The reports so generated were in textual form and were unstructured in nature.

They were used to identify the faults. In another paper, inspection reports were used. These reports belonged to marine domain (Lee, S. et al., 2014). The concept of Self-Organizing Map was used along with linkage approach and concept extraction for document organization. With the help of this system, defects were reported in these reports.

Similar type of work has also been done in bio-medical domain. Song, M. et al. (2015) focused on the extraction of entities and relations and developed a text mining system. The proposed system was tested on five datasets and showed better results. Uramoto, N. et al. (2004) developed MedTAKMI, a set of tools for medical documents. It was an extension of TAKMI (Text Analysis and Knowledge Mining) system that was initially developed for customer relationship management applications. The system was based on using keyword-based search to identify relations among entities.

Another type of work was done in the field of legal documents. A knowledge model was proposed by Wagh, R.S. (2013). In this model, the legal documents were collected, pre-processed and then grouped using clustering technique of data mining. In another work, a new approach was developed to identify criminal networks from a collection of text documents (Al-Zaidy, R. et al., 2012). The method also helped in finding relations among the criminals.

The work done in KDT process in different domains in a detailed manner is shown in table 2.1.

Table 2.1: Work done in KDT process in different domains

S. No	Domain and Name of tool (if any)	Method and Approach	Corpus Taken	Publication Year	Results and Observations
1	Automotive	Based on ontology for annotating key terms in the documents.	From Warranty and Claims Database (WCD) for vehicles (makes and models) from January 01,2009 to March 31, 2010.	2013	Improve in Precision from 0.45 to 0.85 and Recall from 0.43 to 0.81.
2	Marine Structures	Based on concept extraction and linkage along with Self-Organizing Map (SOM) for document organization.	28,873 inspection reports.	2014	The proposed KDT process was useful in understanding the defects in the domain.
3	Bio-medical Documents	Based on a text-mining system that combines dictionary-based entity extraction and rule-based relation extraction in a framework.	Five corpora of different features and relations: BioInfer corpus, AIMed, GAD, PolySearch, CoMAGC.	2015	For entity extraction, average F-measure obtained was 85% and for relation extraction it was 81%.
4	Bio-medical Documents (IBM TAKMI)	Based on the concepts of entity extraction and relation extraction.	MEDLINE database containing 11 million bio-medical journal abstracts.	2004	The tool developed can mine the entire database and it is currently running at a customer site.

5	Legal Documents	Based on the proposal of knowledge model that collects the documents, pre-process them and groups them using Clustering technique. Then the clusters are evaluated.	Legal Documents	2013	The study helps in the use of clustering technique for the grouping of legal documents.
6	Criminal Database	Based on detection of criminal networks from a collection of text documents and extracting useful information for investigation.	Two real life datasets: Enron e-mail corpus and file system of author's personal computer.	2012	The software tool for the proposed method was developed and has received positive feedback from a forensics team in Canada.

It is evident that in the domain of knowledge discovery in research articles, very little or almost no work is done in literature. Therefore, we have taken this domain and developed a modified KDT process. The proposed process is based on lexical and semantics concepts and has shown a good performance.

The next section discusses about an important technique: Text categorization which we have used in our research.

2.4.2 A Review on Text Categorization

Given a collection of text documents and a set of classes or categories, the technique of text categorization assigns a text document to its predefined category. In other words, if D is a set of all text documents and C is a set of predefined categories, then

F is a category assignment function that can be defined as: F: D× C → {0, 1}. The value of F is 1, if a text document d belongs to category c, else it is 0.

Text Categorization is used in a variety of applications. Some examples of its application areas are:

E-mail Classification and Spam Filtering- It is important to distinguish e-mails from spam mails. Text categorization provides methods to classify e-mails and filter spam mails from them. (Carvalho, V.R. & Cohen W., 2005; Cohen, W., 1996)

Text Filtering and Organization- Nowadays there is a trend of online news articles, a large volume of articles is created in a single day (Lang, K, 1995). To handle such voluminous articles, there is a need of some automated tools and methods. This application is called as News Filtering.

Document Retrieval and Organization-This application deals with the use of text categorization methods for the organization of documents. The documents may belong to different domains like scientific literature, web documents, documents in digital libraries, legal documents etc.

Opinion Mining- In this field, customer opinions or reviews are taken into consideration and mined to obtain useful information from them.

2.4.3 A Review on Single-Label Text Categorization

Text Categorization may be Single-label or Multi-label in nature. In single-label, each text document belongs to a single category, whereas in multi-label text categorization, each text document belongs to multiple categories.

There are various types of machine learning and statistical classifiers available for text categorization. These are K-Nearest Neighbor (KNN), Support Vector Machines (SVM), Genetic Algorithm, Naïve Bayes Method, Neural Networks, Decision Trees, Regression based methods, hybrid methods etc. A lot of work is going on in the field of individual classifiers, which is summarized below in a tabular way. Separate tables are made for different classifiers. The work done is shown in a chronological order with respect to time. The fields taken in the tables are dataset used, approach used, results and accuracy obtained.

2.4.3.1 K- Nearest Neighbor (KNN) Classifier

The KNN algorithm is the most widely used algorithm in research. It was first introduced by Cover & Hart (1967). It is a non-parametric, simple and easy to implement method. Non-parametric means that it does not make any theoretical assumptions on the given data. This algorithm inputs a set of labelled training set and uses it to classify the unknown test set.

Given a test document x, the algorithm finds K nearest neighbors of x among the training set and uses their classes. The similarity score of the test document with respect to each nearest neighbor document is calculated. It is used as a weight of the class of nearest neighbor document. If the same class is shared among many K nearest neighbors, then per neighbor weights of that class are added together and the resultant weighted sum is used as the score of that class with respect to the test document. The scores of the classes are then sorted.

The decision rule used in KNN method can be written as

$$\text{Score}\ (x, c_i) = \sum_{x_j \in \text{KNN}(x)} \text{Sim}(x, x_j)\ \S(x_j, c_i) \tag{2.1}$$

where the set of K nearest neighbors of document x are denoted by KNN(x), the classification of document x_j with respect to class c_i is denoted by $\S(x_j, c_i)$. Sim is a cosine function used to calculate similarity between two documents x and x_j.
The function

$$\S(x_j, c_i) = 1, \text{ if } x_j \in c_i$$
$$= 0, \text{ Otherwise}$$

The test document x is assigned a class that has the highest resultant weighted sum.

The table 2.2 shows the survey of work done in KNN method.

Table 2.2: Survey of the work done in KNN method.

Sr No.	Authors	Details of Publication	Dataset Used	Approach Used	Result and Accuracy
1	Yang, Y. & Pederson, J.	1997	Reuters corpus and Ohsumed Collection	They focused on dimensionality reduction. They evaluated five methods: term selection based on document frequency, information gain, mutual information, chi square test and term strength.	Information gain and chi-square methods used with KNN classifier improved the classification accuracy.
2	Tan, S.	Journal of Expert Systems with Applications, Elsevier, 2005.	Two datasets: Reuters 21578 and TDT2.	They proposed a neighbor weighted KNN algorithm for unbalanced datasets. It assigns a big weight to neighbors from small classes and little weight to neighbors of big classes.	The proposed algorithm achieves good performance improvement on imbalanced corpora.
3	Deng, Z.H. & Tang, S.W.	Springer Conference, 2005.	Newsgroups 18828 and Ohscal.	They proposed a non VSM KNN algorithm for text classification based on correlations between	They evaluated the algorithm on two datasets and compared it against traditional KNN.

				categories and features.	Their algorithm performed better.
4	Fayed, H.A. & Atiya, A.F.	IEEE Transactions on Neural Networks, 2009.	Real life datasets from UCI repository.	They have proposed a new condensing approach to remove patterns which cause burden. They have defined a chain of nearest neighbors from alternating classes and then set a cut off for the patterns in the training set.	Their approach has proved to be a simple and fast condensing algorithm. It maintains the same level of classification accuracy as the traditional KNN method.
5	García-Laencina, P. J. et al.	Journal of neurocomputing, Elsevier, 2009.	Two complete datasets and two incomplete datasets from UCI repository.	They propose a novel KNN imputation procedure for missing data using a feature weighed distance metric.	The proposed algorithm is efficient and robust.
6	Toyoma, J. et al.	Journal of Pattern Recognition, Elsevier, 2010.	Japanese phoneme dataset.	They proposed a probably correct approach in which the correct set of k nearest neighbors is obtained in high probability to reduce the search time.	The proposed algorithm has advantage that in it, the searching time is reduced.
7	Bax, E.	IEEE Transactions on information theory, 2012.	-----	They present a method to calculate probably approximately correct error	It presents a bound on out of sample (examples not used for training a classifier) error

				bounds for KNN classifiers.	rate for KNN classifier.
8	Zhang, S.	The Journal of systems and software, Elsevier, 2012.	Real datasets from UCI repository.	They have proposed a novel KNN imputation method to impute missing data: GKNN-grey KNN. It is based on calculating grey distance between missing data and training data.	They have implemented the proposed algorithm and have shown that it is more efficient than existing methods.
9	Jiang, S. et al.	Journal of Expert Systems with Applications, Elsevier, 2012.	Three datasets: Reuters 21578, Fudan University corpus and Ling Spam e-mail spam filtering corpus.	They have improved KNN algorithm by using one pass clustering algorithm in it.	They have tested on three datasets and proved their algorithm to be better.
10	Beliakov, G. & Li, G.	Journal of Pattern Recognition Letters, Elsevier, 2012.	-----	They have improved the speed and stability of KNN method by replacing the sort operation with calculating the order statistics.	Their approach is proved to be superior.

2.4.3.2 SVM Classifier

The SVM Classifier is a more powerful classifier for text categorization (Joachims, T., 1998). In this method, for a 2-class or binary problem, a separating hyper plane is drawn that separates positive examples from negative examples. A major feature of SVM method is its ability to deal with high dimensionality of feature space. The text documents possess high dimensionality of features. The table 2.3 given shows the work done using SVM method.

Table 2.3: Survey of the work done using SVM method

S. No	Authors	Details of Publication	Dataset Used	Approach Used	Result and Accuracy
1.	Joachims, T.	ACM Digital library, 1998.	Two datasets: Reuters 21578 and Ohsumed corpus	He compared SVM method with Naïve Bayes, Rocchio algorithm, KNN and Decision tree.	SVM is the best classifier with 66% accuracy.
2.	Lee, C.H. & Yang, H.C.	IEEE Xplore, 2005.	Random set of documents consisting of five classes.	They used a hybrid approach for measuring semantic relatedness among text. They developed several text classifiers using SVM.	SVM Classifier with a Gaussian kernel is the best having greater than 90% Recall ratio.
3.	Kumar, M.A. &	Journal of Pattern	Three datasets:	They performed empirical	For small scale text

	Gopal, M.	Recognition Letters, Elsevier, 2010.	Web KB, 20 Newsgroups, Industry Sector.	comparison of standard O.A.A and O.A.O together with three improvements made to these standard approaches.	categorization tasks, one against all (O.A.A) will have better performance than other methods.
4.	Li, Z. et al.	Journal of Pattern Recognition Letters, Elsevier, 2011.	Three datasets: Reuters 21578, 20 Newsgroups and Tan Corp.	They proposed a concise semantic analysis technique for text categorization tasks.	Their proposed method reaches a comparable performance with SVM classifier in text categorization tasks.
5.	Mayor, S. & Pant, B.	International Journal of Engineering Science and Technology, 2012.	50 news instances from newspapers or sites.	They have implemented SVM for calculating term frequency. They also have used LIBSVM tool.	The accuracy rate obtained was 66.67%.

2.4.3.3 Naïve Bayes Classifier

The Naïve Bayes Classifier is a probabilistic classifier method. It is based on the concept of Bayes theorem that predicts class membership probabilities. This method gives good results in some cases. But, it suffers from the disadvantage that it assumes class conditional independence. In real world, the classes of text documents are inter-dependent. The table 2.4 shows the related work done in Naïve Bayes method.

Table 2.4: Survey of the work done in Naive Bayes method

S. No	Authors	Details of Publication	Dataset Used	Approach Used	Result and Accuracy
1.	Kim, S. B. et al.	IEEE Transactions on Knowledge and Data Engineering, 2006.	Reuters 21578, 20 Newsgroups.	They have proposed two empirical heuristics to overcome a problem in Naïve Bayes method.	The proposed Naïve Bayes Classifier performs very well.
2.	Chang, Y. H. & Huang, H.Y.	IEEE Conference, 2008.	319 documents from Electronic Thesis and Dissertations.	They have proposed an automatic document classifier system based on ontology and Naïve	The average effectiveness of document classification in 11 categories is about 89%.

				Bayes Classifier.	
3.	Soria, D. et al.	Journal of Knowledge-Based Systems, Elsevier, 2011.	Breast cancer data and on three UCI datasets.	They have proposed a new method which does not restrict the variables to have values which are normally distributed.	Their algorithm performed well in all four datasets.
4.	Zhang, W. & Gao, F.	Journal of Procedia Engineering, 2011.	Two e-mail datasets.	They have proposed an auxiliary feature selection method.	The proposed method improves the performance of Naïve Bayes Classifier.
5.	Balamurugan, et al.	Journal of Knowledge-Based Systems, 2011.	Various databases.	They have proposed a novel algorithm which extends the traditional Naïve Bayes method. They have	The proposed algorithm performs better than traditional algorithm.

				used a partial matching method.	

2.4.3.4 Genetic Algorithms

It is based on famous Darwin's theory of "survival of the fittest". It helps in solving the optimization problems. It involves the concept of cross over and mutation. The table 2.5 below shows the survey of work done in Genetic Algorithms method.

Table 2.5: Genetic Algorithm literature survey

S. No	Authors	Details of Publication	Dataset Used	Approach Used	Result and Accuracy
1	Atkinson, J., Abutridy, Mellish, C., Aitken, S.	IEEE Intelligent Systems, 2004.	Documents of agriculture domain.	Their approach consists of two steps: firstly, pre-processing step that produces training data and initial population of genetic algorithm after information extraction from documents. It will generate	They implemented a prototype of model in Prolog and achieved good accuracy which is assessed by experts.

				rules. The second step is genetic algorithm-based knowledge discovery in which hypothesis are produced.	
2.	Khalessizadeh, S.M. et al.	World Academy of Science, Engineering and Technology, 2006.	Text in Persian language.	They have proposed a new algorithm using genetic algorithm based on concept of standard deviation.	The proposed algorithm has a good performance.
3.	Zhen-fang, Z., Pei-yu, L., & Ran, L.	IEEE Symposium, 2008.	------	They had introduced simulated annealing mechanism of genetic algorithm to solve the problem of text categorization.	The proposed method had good accuracy and recall rate.
4.	Pietramala, A. et al.	Springer Conference, 2008.	Reuters 21578,	They have presented a genetic	The proposed method is equally

			Ohsumed Collection.	algorithm called Olex-GA, based on rule-based text classifiers.	competitive with other classifiers.
5.	Uguz, H.	Journal of Knowledge-Based Systems, 2011.	Reuters 21578 and Classic 3 datasets.	They have used a two-stage feature selection and feature extraction method using genetic algorithms.	The proposed method achieves good effectiveness.

2.4.3.5 Decision Tree

A decision tree is a flowchart like tree structure that helps to solve the problems of classification (Han, J., Pei & Kamber, M., 2011). The internal nodes of the tree represent a test on an attribute and each branch gives the values for the test. The leaves of the tree hold the class labels. Decision trees are used in classification problems. A path is traced starting from the root to the leaves of the tree. The leaves of the tree predict the class/category.

The state of art in this area is as follows: Mehta et al. (1996) had built a classifier, SLIQ. It is a decision tree-based classifier that can handle both numerical and categorical data. It uses a sorting technique in the growth of the tree, which is combined with a breadth first strategy. With the help of this, the classifier can scale large datasets. Vateekul & Kubat (2009) proposed a method for multi-label text categorization. Their method focused on use of decision trees to reduce the computational costs. They proposed an algorithm called as Fast Decision Tree Induction (FDT). Johnson D. E. et al. (2002) proposed a fast decision tree induction algorithm that improves text categorization. In another work, Chen H. et al. (2010) worked on Chinese e-mail classification using association rules.

2.4.3.6 Centroid Based Classifiers

The Centroid based classifiers are widely used due to their simplicity. The Centroid vectors are calculated for the documents belonging to similar class. Cardoso-Cachopo & Oliveira (2007) used Expectation Maximization method with Centroid classifiers for single-label text categorization. Tan S. (2008) gave a batch-based approach to overcome the inductive bias problem of centroid based classifiers. In another work, Liu C. et al. (2017) proposed a new classification model known as gravitational model to solve the problem of imbalanced classes in data.

2.4.3.7 Combination of multiple Classifiers

Nowadays, a hybrid approach is used in the area of text categorization. Here the classifiers are combined together based on their advantages and disadvantages. The table 2.6 shows the state of art of work done by combining multiple classifiers.

Table 2.6: Survey of the work done in hybrid method

S. No	Authors	Details of Publication	Dataset Used	Approach Used	Result and Accuracy
1.	Li,Y.H. & Jain, A.K.	The Computer Journal, 1998.	Yahoo newsgroups dataset having 7 classes.	They took a combination of four classifiers: Naïve Bayes, Nearest Neighbor, decision tree and subspace method. They used three approaches: simple voting, dynamic	Naïve Bayes and subspace methods are better having an accuracy of 83%. The approach of adaptive combination is the best.

				classifier selection and adaptive classifier combination.	
2.	Yang, Y. & Liu, X.	ACM Digital Library, 1999.	Reuters 21578	They combined five classifiers: SVM, KNN, Neural Networks, LLSF (linear least squares fit) and Naïve Bayes and compared the results.	SVM, KNN and LLSF are better, if data is less and all methods perform equally, if data is large.
3.	Fujino, A. et al.	Journal of Information Processing & Management, 2007.	They took four test collections.	They have focused on a hybrid of classifiers using probabilistic generative and discriminative approaches. They have used Naïve Bayes method.	Their proposed method achieved good classification accuracy.
4.	Lee, C.H. &	Expert Systems with Applications, Elsevier, 2009.	They took 6000 news documents.	They combined SVM	The proposed hybrid

	Yang, H.C.			(supervised technique) and SOM (self-organizing maps-unsupervised technique) for multilingual text categorization.	method provided good accuracy.
5.	Wan, C.H. et al.	Journal of Expert Systems with Applications, 2012.	Benchmark text datasets.	They have combined KNN with Support Vector Machines.	Their method has achieved good classification accuracy.

2.4.4 A Review on Multi-Label Text Categorization

Most real-world text documents are multi-label in nature. For example, a book may be classified in multiple categories like science, engineering, computers etc. A movie can belong to various film genres like science, drama, action, horror etc. Multi-label categorization is more popular nowadays. This factor motivated us to select this topic. So, our focus in this work is on multi-label text categorization.

There are two main techniques used for multi-label text document categorization in literature. These are problem transformation methods and algorithm adaptation methods. The first approach transforms or converts a multi-label problem into one or more single-label problems. And then builds binary classifier for each single-label problem. There are many problem transformation methods proposed in research. Binary Relevance or BR is a popular method given by Boutell et al. in 2004. This method builds m binary or single-label classifiers for each label. Tsoumakas (2011) gave a method called as RAkELd. It is a random k label-set method for multi-label classification. Another method is ECC method given by Read, J. et al. (2011). It is an ensemble classifier chains algorithm.

The second approach modifies the existing algorithms for text categorization to suit the multi-label data. A popular example of this approach is ML-KNN algorithm (Zhang & Zhou, 2007).

Different researchers have given various methods in multi-label text categorization. In 2014, Yu, et al. used rough sets and local correlation for multi-label categorization of text documents. Firstly, location of test data is found with the help of rough sets and then probability of test data belonging to a class is calculated using the concept of correlation. In another work, Chang, et al. (2008) has built a multi-label classifier by using the method of weighted indexing. And further, a threshold value was used to compute the value of degree of similarity between the categories. There also have been modifications made in single-label classifiers to support multi-label learning. For example, ML-KNN is a multi-label version of traditional KNN algorithm. And also, a multi-label version of SVM, that is fuzzy support vector machines was given by Abe, S. in 2015.

The concept of hierarchical classification is also used with multi- label categorization. Both these concepts are linked. In hierarchical classification, classes are hierarchically structured. A lot of work has been done in this area. Cerri et al. (2014) have used the concept of multi-layer perceptron for each level of hierarchy of classes. In 2015, Serafino, et al. gave a new framework MultiWebClass by extending the concept of Web Class III. Web Class III was a classification framework for HTML pages. Another work using the concept of transductive learning was done in hierarchical text categorization (Ceci, M. in 2008). He gave a method for classification of text documents in internal nodes and leaf nodes present in the hierarchy of classes.

In multi-label categorization, the concept of ranking of labels also needs to be discussed. It is given in the next section.

2.4.4.1 Ranking of Labels in Multi-Label Text Categorization

Ranking of class labels is an important task in multi-label text categorization. For example, in case of news articles, a single article may belong to many classes like science, technology, politics etc. The concept of ranking then plays an important role, for which the topic is more relevant.

According to Gibaja & Ventura (2014), the term ranking in multi-label learning means an ordering or strict ordering of class labels according to their importance. For example: if there is a text document and L is a set of class labels that the document belongs to L = {l1, l2, l3, l4, l5…} where l1, l2, l3, l4, l5…. are individual class labels. The result of ranking of class labels is a strict sequence of class labels as r(l5) < r(l3) < r(l2) <r(l1) <r(l4) <…….

In literature, ranking is done on the basis of two factors (Tsoumakas, 2009). It can be label-based or example-based. In example-based method, ranking is done individually for each test document and then average is taken, whereas in label-based method, ranking is done separately for each label and then average is taken. The selection of the method of ranking depends on the approach or technique used to solve multi-label categorization problem. In problem transformation method, ranking is done by single-label learning, pair wise comparison, calibrated label ranking etc. In algorithm adaptation method, different approaches are followed by different algorithms for ranking of class labels.

In our work, we have used the concept of quantifiers to calculate the ranking of class labels in multi-label text categorization. We have proposed eight quantifiers: none, almost none, very low, low, high, higher, highest and all. These are discussed in chapter 5.

2.4.5 Identification of Research Gaps

Based on literature survey, some gaps are identified in the existing methods. To fill these gaps, following solutions have been proposed.

- There is a challenge to mine useful information from text documents due to their unstructured nature.

    i. To the best of our knowledge, none of the existing methods have used lexical and semantic concepts. So, we have used lexical and semantic concepts to analyze text documents. We have devised a method for conversion of a raw text document to a structured form, so that useful information can be extracted from it.

- There is lack of efficient methods to organize and access the vast information available in text documents.

    i. We have to study the existing methods of text categorization and tried to find a better method that will help to organize and access the vast textual information available in text documents in a more systematic manner.

- The existing methods for single-label text categorization suffer from some limitations like KNN Classifier, which has a major drawback that it uses all the features in computing the similarity measure. The similarity measure helps in identifying the neighbors of a particular text document. The Naïve Bayes classifier gives good results with small dataset as it keeps Recall and Precision values low. The genetic algorithm needs a proper design and definition of fitness function.

    i. We have developed an algorithm (LKNN) for single-label text categorization that finds tokens in the text documents. The tokens are associated with their frequency of occurrence. The frequency of occurrence of tokens is the weight of a token. So, we have assigned the weights to the features(tokens) in our proposed algorithm. The proposed LKNN algorithm gives good results with large dataset also.

- In multi-label text categorization technique, there is little or almost no work done on research articles.
    i. We have proposed a modified KDT process called as LS-KDT for research articles. To the best of our knowledge, no work has been done in categorization of text documents using lexical and semantic concepts. The proposed process focuses on lexical and semantic aspects of a text document. The results of proposed LS-KDT process are compared with the results of standard digital libraries. It has shown promising results.

2.5 Concluding the Review

The literature survey conducted, gave us future directions in research. We studied the work done in the four sub problems and further proposed our work in the next chapters.

3 PROPOSED U-STRUCT FRAMEWORK

The nature of the documents in the real world is unstructured. Therefore, it is the need of the hour to convert them into structured form, so that further, data mining techniques can be applied. In this chapter, a framework called as U-STRUCT, is proposed, that is used for the conversion of unstructured text documents into a structured form. The framework is divided in two major phases: Text Analysis phase and Text Synthesis phase. These phases are further divided into steps which have been described in brief in this chapter.

Text mining is an upcoming field nowadays. The reason for the rising importance of this field is that most of the documents contain text data. On internet, there is an enormous amount of information that is too large to be read and analyzed. In this field, there is a need for efficient pre-processing methods and techniques, so that useful knowledge can be obtained from text documents.

3.1 Need of Proposed Framework

Text documents contain natural language text which is unstructured and unorganized. Moreover, this unstructured data does not follow any format, sequence or rules. The unstructured data is unpredictable as well as difficult to use. Therefore, there is a tremendous need to convert the text documents to structured form. So, a generalized framework, known as U-STRUCT, is proposed which converts unstructured data into structured form and this can be used for making future predictions and analysis.

3.2 Proposed U-STRUCT Framework

A generic framework, known as U–STRUCT has been described in this section, which converts unstructured text document into structured form. This framework analyses the text documents from different views: lexical, syntactic and semantics and produces a generalized intermediate form of documents. The figure 3.1 shows the proposed U-STRUCT framework.

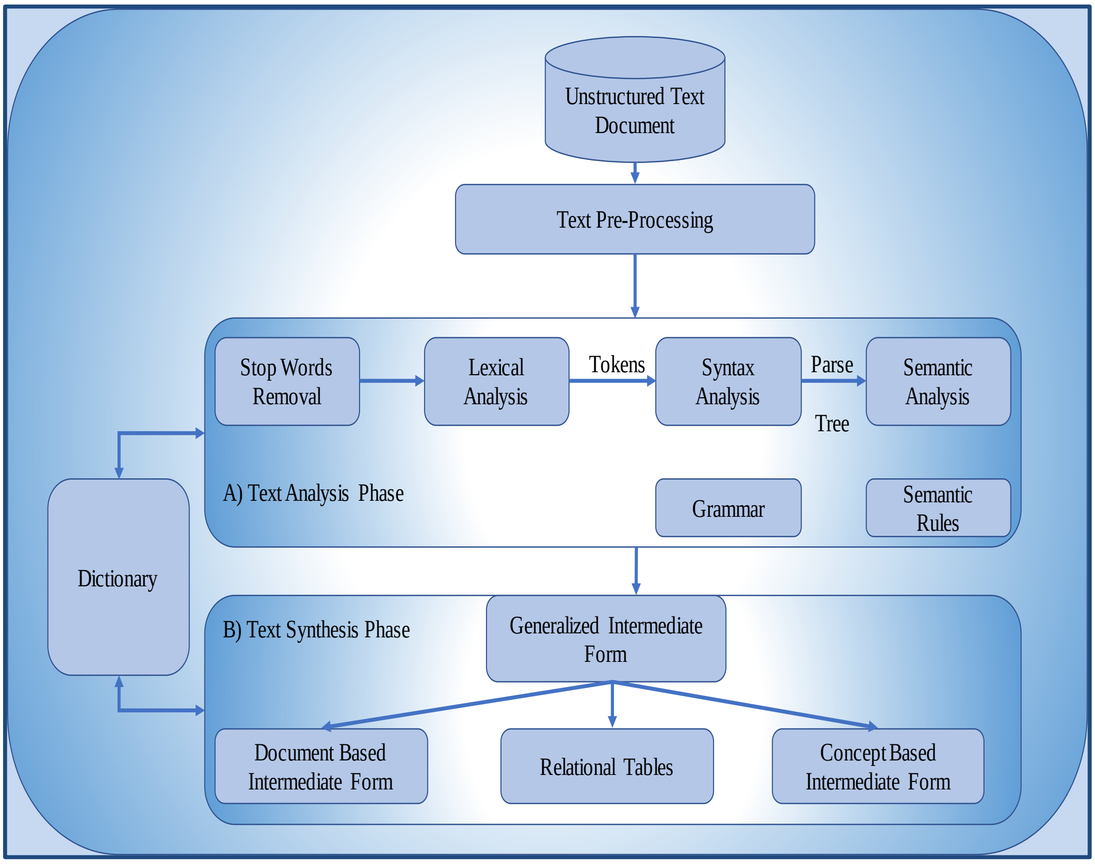


Figure 3.1: Proposed U-STRUCT framework

The proposed framework is divided into two phases: Text Analysis phase and Text Synthesis phase. In the Text Analysis phase, the raw unstructured text document undergoes text pre-processing. This phase is further divided into four steps: Stop words removal, Lexical Analysis or Scanner, Syntax Analysis or Parser and Semantic Analysis. These steps are further explained in detail in the following section. The output of first phase i.e. Text Analysis phase is fed as input to the next phase which is Text Synthesis phase. In this phase, we generate a generalized intermediate form of documents. This intermediate form can be represented in three forms: firstly, documented form, secondly in the form of relational tables and thirdly it can be in a conceptual form. In parallel to the two phases, a dictionary or a bookkeeping component is also required for storing the tokens. It is used in both the phases.

.

3.3 Components of Proposed U-STRUCT Framework

The following table 3.1 shows the components of U-STRUCT framework.

Table 3.1: Components of proposed U-STRUCT Framework

1. Text Analysis Phase	
Stop words removal	Stop words are the commonly used words like ‘a’, ‘the’, ‘of’ etc. These are usually considered irrelevant as they do not play any role in the knowledge extraction and hence are ignored.
Lexical Analysis	In this step, the text document is scanned character by character, and grouped into tokens. Tokens can be Nouns, Verbs, Article, Adjective, Preposition, Number and special symbols.
Syntax Analysis	This step performs two functions, firstly it checks whether the incoming token is according to the specifications of the grammar or not. Secondly, if the token follows the specification of the grammar, it generates a syntax tree or a parse tree of a noun phrase, verb phrase, prepositional phrase, adjective phrase or a clause otherwise it gives an error.
Semantic Analysis	In this step, a semantic action is executed which leads to intermediate representation of the document directly.
2. Text Synthesis Phase	
Document based Intermediate form	In Document based Intermediate form, each entity represents a document.
Relational tables	In this form, each entity represents a relational table.
Concept based Intermediate form	In concept based Intermediate form, each entity represents an object or concept of interest in a specific domain.

3. Dictionary/Bookkeeping	
Dictionary	It is required mainly in Lexical Analysis and Syntax Analysis step to store the tokens and check for their spellings.

The components of the proposed U-STRUCT framework and their detailed functions are given below.

3.3.1 Text Analysis Phase

The input to the system is a raw unstructured text document. This document can be of any form like a news story, a business or a legal report, an e-mail etc. The document firstly undergoes for text pre-processing using the pre-processing techniques. In this phase there are four steps: Stop words removal, Lexical Analysis/Scanner, Syntax Analysis/Parser and Semantic Analysis.

3.3.1.1 Stop Words Removal

Stop words are the words like ‘a’, ‘an’, ‘the’ etc. These words are considered immaterial and are therefore removed from the document. There is a standard Stop words list used which is given in the references section.

3.3.1.2 Lexical Analysis / Scanner

This step is the key feature of our proposed framework. A lot of work has been done in this direction. But, to our best knowledge, none of the previous works have extracted tokens for the identification of major features in text documents.

Lexical Analysis is also known as Scanner. In this step, the text document is scanned character by character and decomposed into chapters, sections, paragraphs, sentences and words. The most frequent method used for this involves breaking the text into sentences and words, which is called tokenization.

Another function of Lexical Analysis is Part-of-Speech(PoS) Tagging. It is the tagging of tokens with the appropriate PoS tags depending upon the context in which they appear. PoS tags divide words into categories depending on the role they play in the sentence in which they appear. They provide information about the semantic content of a word. Nouns usually denote “tangible and intangible things,” whereas prepositions express relationships between “things”. Most PoS tag sets make use of the same basic categories. According to Shatkay & Feldman (2003), there are seven commonly used set of tags. These are Proper Noun, Article, Noun, Adjective, Verb, Preposition and Number. Some systems contain a much more elaborate set of tags. These PoS tags capture syntactic category or variable like noun, verb, adjective etc. and they can be used for the identification of noun phrase, verb phrase or other parts of speech in the next Syntax Analysis step. The output of Lexical Analysis is a sequence of tokens which is passed onto the next step.

3.3.1.3 Syntax Analysis/ Parser

The input to Syntax Analysis or Parser step is the sequence of tokens from the previous step. This step performs two functions: firstly, it checks whether the incoming tokens are according to the specified Grammar or not. So, the concept of Grammar comes into picture. And, if the tokens are in accordance with the specifications of the grammar then a Parse tree is drawn, else, there is an error. A Grammar or a Natural Grammar (G) is a formal specification of the language used, and is represented as

$$G = (VNT, VT, P, S)$$

where VNT denotes a set of non-terminal symbols, VT a set of terminal symbols, P is a set of production rules, and S represents a sentence. A non-terminal symbol is a symbol that does not appear in an input string but is a syntactic category or a variable in G. Examples of VNT are NP (Noun Phrase), VP (Verb Phrase), PP (Preposition Phrase), and so on. A terminal symbol is a symbol that represents a class of basic and indivisible symbols in input strings; it represents a part-of-speech symbol. Examples of terminal symbols are Noun, Verb, Adjective, Preposition etc. A production is a rule of the form $\alpha \rightarrow \beta$, where α is a non-terminal symbol, and β represents a set of terminal symbols or non-terminal symbols. When a sentence is successfully parsed, a structure called a parse tree of the sentence is generated. For example, a parse tree of the sentence ‘He holds bat with his hand ‘is shown in the figure 3.2 given below:

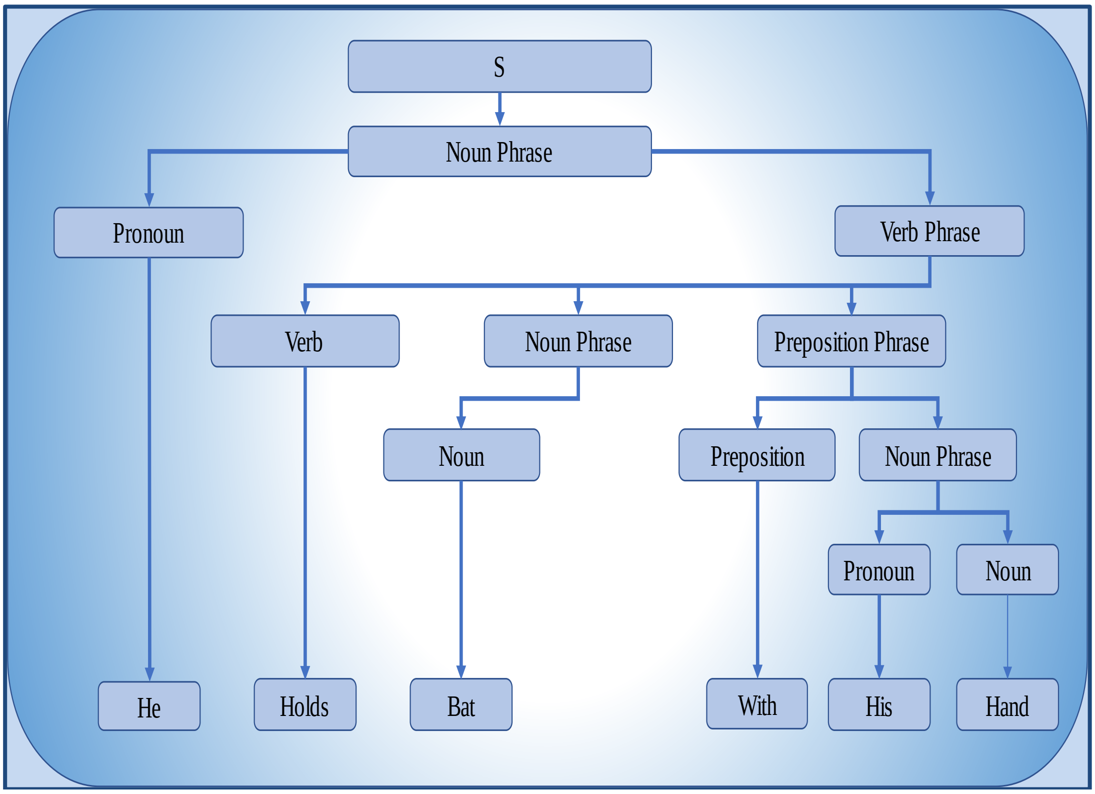


Figure 3.2: Parse tree of the sentence ‘He holds bat with his hand’

The major work in syntax analysis is sentence parsing. During this, a sentence is broken down into phrases and sub phrases.

3.3.1.4 Semantic Analysis

This is the next step after Syntax Analysis. The input of this step is the parse tree which is generated in the previous phase. Semantic Analysis is a process of converting a parse tree into a semantic representation that is unambiguous and clear. In this step, a semantic action is associated with each production rule of the grammar. For example, α →β {α1.sem, α2.sem…. αn.sem}, where the expression in {} is the semantic action. The semantic action is a piece of code that generates the intermediate form of text document directly.

3.3.2 Text Synthesis Phase

The Text Synthesis Phase generates the intermediate form of text document. This intermediate form can be document-based, in the form of relational tables or a concept based. In a document based intermediate form, each entity is a document. It helps to discover patterns and relationships across documents. In relational tables based intermediate form, techniques like categorization, clustering etc. can be used to infer useful knowledge. In concept based intermediate form, each entity represents an object or concept of interests in a specific domain and derives patterns and relationships across concepts.

There is a challenging issue in Text Synthesis Phase which is the complexity of the intermediate form and needs to be dealt with.

3.3.3 Dictionary /Bookkeeping component

The dictionary or Bookkeeping component is used by both the phases of the proposed framework. It works in two modes: store and search. Firstly, it stores the tokens found in Lexical Analysis and Syntax Analysis step, so that there is no redundancy of the tokens. Secondly, it helps to search any token, which may be required by other components.

In this chapter, U-STRUCT framework has been proposed. This framework converts unstructured text documents to structured form. The proposed framework has two phases: Text Analysis and Text Synthesis. Further these phases are decomposed into steps, which are explained in this chapter.

4 SINGLE-LABEL TEXT CATEGORIZATION

This chapter describes the proposed Lexical KNN (LKNN) algorithm. The proposed algorithm helps in single- label categorization of text documents. It has been evaluated on two datasets: dataset of research articles belonging to computer science domain and Ohsumed Collection (articles belonging to medical domain). The performance of the proposed LKNN algorithm has been evaluated with the help of standard metrics like Recall, Precision, F1-measure and Accuracy. The proposed algorithm(LKNN) is compared with the traditional KNN algorithm. Our proposed algorithm has shown significantly good results in terms of Recall, Precision, F1-measure and Accuracy.

In single-label text categorization, a text document belongs to exactly one category. In this chapter, we have proposed a lexical based KNN algorithm (LKNN) for single-label categorization of text documents.

4.1 Proposed LKNN Algorithm

The proposed Lexical KNN (LKNN) algorithm automatically classifies research articles into their categories. The algorithm identifies tokens and their weights in the Abstract of research articles. Tokens are considered to be the major source of information in our work. Each research article is represented as a vector of tokens and their weights. The weight of a token is defined as frequency of its occurrence.

The proposed LKNN algorithm is implemented on two datasets. Firstly, on the collection of research articles of computer science domain and secondly on Ohsumed Collection.

The proposed LKNN algorithm works in two steps:

- Build Classification Model or Classifier using a Training set: In this step, a model or classifier is built using a training set. The Abstract of the research article is scanned and stop words are removed from it, using a standard list of stop words. In Lexical Analysis module, tokens are identified in the Abstract of research article using the standard ACM Computing Classification system (given in the References section). The frequency of occurrence of a token is stored as the weight of the token.

The parameter distance is used to build KNN Classifier. It is defined as a basis to calculate the contribution of each k neighbor in the class allocation process. Then we calculate the predicted class of the research article according to the formula given in equation 4.1 (mentioned in the proposed algorithm in figure 4.1).

- Text Categorization: In the second step, a test research article is classified using the KNN Classifier with the help of CosSim function given in equation 4.2 (mentioned in the proposed algorithm in figure 4.1).

The proposed LKNN algorithm is given in Figure 4.1 below.

Step 1: Build Classification Model or Classifier using a Training set

// Input a set of research articles. Let J = {j_1, j_2, j_3… j_n), where n is the maximum number of research articles taken.

For i = 1 to n

Repeat

i. Scan the abstract section of the research article (j_i) and remove stop words from it. The standard list of stop words is used and is given as Stop words in the References section.

ii. Using Lexical Analysis, scan the abstract and identify the tokens in the abstract of the article j_i. The standard list of keywords (tokens) is specified in ACM Computing Classification System (2012). Also, find the weight of the token. The weight w_i of a token t_i = $freq_{i,}$ where freqi is the frequency of occurrence.

iii. Create a table of tokens to record the name of token and its weight.

iv. Each research article j_i called an instance, is represented as a vector: <w_1 (j), w_2 (j), w_3 (j), w_4 (j)...> where w_i (j) is the weight of the ith term. This weight is set according to its frequency of occurrence.

v. To build KNN Classifier, we use distance as a basis to calculate the contribution of each k neighbor in the class allocation process.

vi We define the predicted class of a journal article j_i belonging to class c as:

$$Pred_{class}(c, j_i) = \frac{\sum_{k_i \in K[Class(k_i=c)]} Sim(k_i, j_i)}{\sum_{k_i=K} Sim(k_i, j_i)} \tag{4.1}$$

where Sim is a similarity function which returns a value after comparing an article with its neighbor. That is, we sum up the similarities of each neighbor belonging to a particular class c and divide by all similarities of k neighbors irrespective of the class.

Step 2: Text Categorization

// classify the test research article using the KNN Classifier

To compare article j with instance i, we define the CosSim function which is defined using our token weight approach as follows:

$$CosSim(i, j) = \frac{S}{\sqrt{A * B}} \tag{4.2}$$

where S is the number of terms that i and j have in common, A is the number of terms in i and B the number of terms in j.

Figure 4.1: Proposed LKNN Algorithm

4.2 About ACM Computing Classification System

It is a standard classification system for the computing field used by ACM digital library. The traditional 1998 version of the ACM Computing Classification System (CCS) is revised by ACM Computing Classification System in 2012. The taxonomy exists in a hierarchical form. It consists of broad categories, which are further organized into sub categories. We have used a subset of ACM taxonomy in our work. Out of total 14 broad categories, there are 11 broad categories under computer science domain. These are: Computer systems organization, Networks, Software and its engineering, Theory of computation, Mathematics of computing, Information systems, Security and privacy, Human-centered computing, Computing methodologies, Applied computing and Social and professional topics.

4.3 Flow Diagram of Proposed LKNN Algorithm

The flow diagram of the proposed LKNN algorithm is given in the figure 4.2.

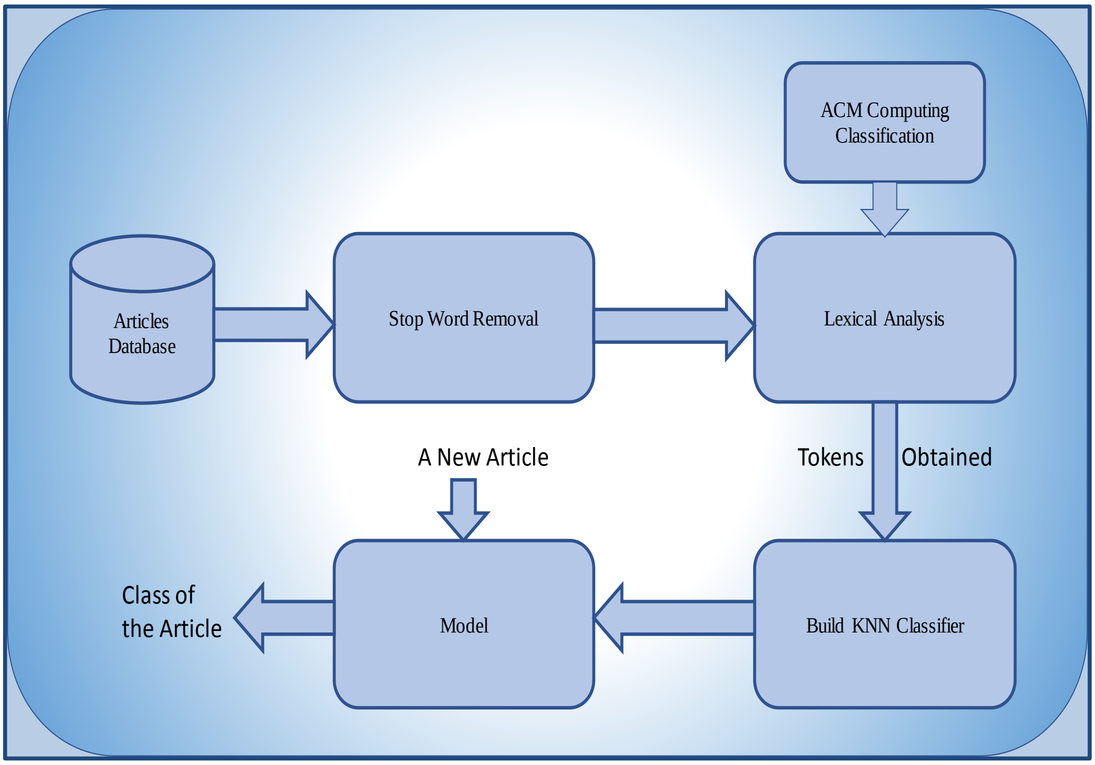


Figure 4.2: Flow diagram of the proposed LKNN Algorithm

The above flow diagram can be explained as follows: In the first part, a collection of research articles is taken as input. Stop words are removed from the abstracts of the articles. Then the articles are sent to Lexical Analysis module. The job of lexical analysis module is to scan the characters in the given input and group them into tokens. We have taken a standard list of keywords given in ACM Computing Classification System, 2012 for identifying the tokens. The output of this part is a list of tokens generated. The frequency of occurrence of tokens is also taken into consideration. It is considered as weight of a token. After this, we have built KNN Classifier using these tokens and their weights. And the last part is testing, in which a new test document arrives, it goes through the whole process and is classified. In the result we obtain the category or the discipline in which article belongs.

The proposed LKNN algorithm is implemented on two datasets. First is a collection of research articles of computer science domain and the second is a collection of medical documents (Ohsumed Collection). In case of medical documents, MESH tree structure is used for identifying the tokens in the medical documents. The flow diagram for the Ohsumed Collection dataset is given in figure 4.3.

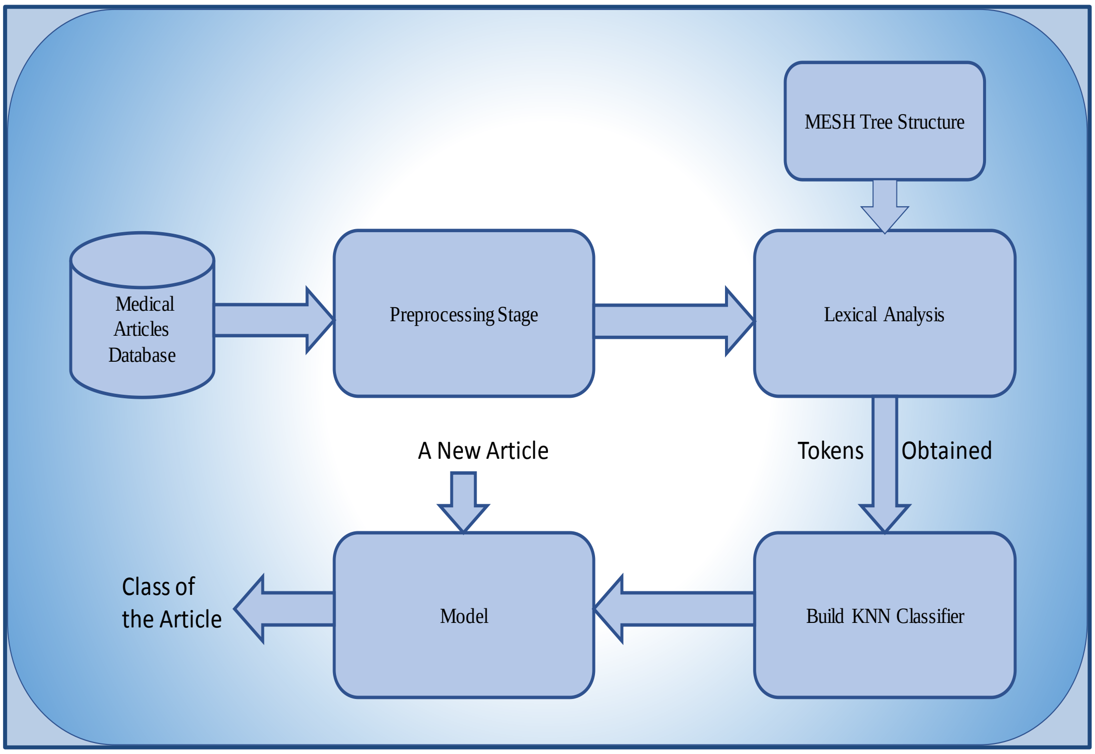


Figure 4.3: Flow diagram of the proposed LKNN algorithm for medical documents

MESH stands for Medical Subject Headings. It is the controlled vocabulary thesaurus of US's National Library of Medicine. It is used for indexing the bio-medical journals of MEDLINE and PUBMED database. PUBMED contains journal articles of life sciences and bio-medical domain since 1996. It contains more than 27 million references. MEDLINE is the largest subset of PUBMED database. It contains more than 5600 reputed journals of biomedical domain.

4.4 Datasets Used

The proposed LKNN algorithm has been evaluated on two datasets. One is set of research articles of computer science discipline and the other is a collection of medical documents (Ohsumed collection).

4.4.1 Computer Science Research Articles Dataset

In computer science domain, we have taken a total of 80 research articles from reputed journals. The articles belong to different sub disciplines of computer science domain like databases, networks, compilers, security and encryption. This is a small dataset taken for a pilot study. The details of computer science research articles dataset are shown in Table 4.1.

Table 4.1: Computer Science Research Articles Dataset

S. No	Sub discipline	Number of articles
1	Databases	20
2	Networks	20
3	Compilers	20
4	Security	10
5	Encryption	10

4.4.2 Ohsumed Collection

The Ohsumed Collection is a subset of MEDLINE database. It was compiled by Hersh, W. et al. (1994). It is a database of significant, peer-reviewed medical literature maintained by the United States National Library of Medicine. We have used the dataset in which the first 20,000 documents were divided as 10,000 for training and 10,000 for testing (Joachims T., 1998).

4.5 Comparison of Proposed LKNN Algorithm with KNN Algorithm

To evaluate the performance of the proposed LKNN algorithm, we have compared it with the traditional KNN algorithm. The most common performance metrics used in text categorization are Recall, Precision, F1 – measure and Accuracy.

These can be calculated as follows: In an experiment, if A is the number of true positive samples predicted as positive, B is the number of true positive samples predicted as negative, C as the number of true negative samples predicted as positive and D as the number of true negative samples predicted as negative, then Precision, Recall, F1-measure can be expressed as follows (Li, B., Yu, S., & Lu, Q., 2003).

$$Precision = \frac{A}{A+C} \tag{4.3}$$

$$Recall = \frac{A}{A+B} \tag{4.4}$$

$$F1 - measure = \frac{(2*Precision*Recall)}{(Precision+Recall)} \tag{4.5}$$

$$Accuracy = \frac{A+D}{A+B+C+D} \tag{4.6}$$

The performance comparison is shown with both the datasets below:

4.5.1 Computer Science Research Articles dataset

A comparative study of the performance of traditional KNN and LKNN is conducted with different K values. The figures 4.4, 4.5, 4.6 and 4.7 show the Recall, Precision, F1- measure and Accuracy values for various values of K for computer science research articles dataset. From the results, it can be analyzed that the Recall values of proposed LKNN algorithm increase linearly with the increasing values of K. The Precision values of the proposed algorithm first increase linearly as compared to traditional KNN algorithm and then it becomes constant. The values of F1-measure and Accuracy of the proposed algorithm increases initially linearly with the increasing values of K but then it becomes constant.

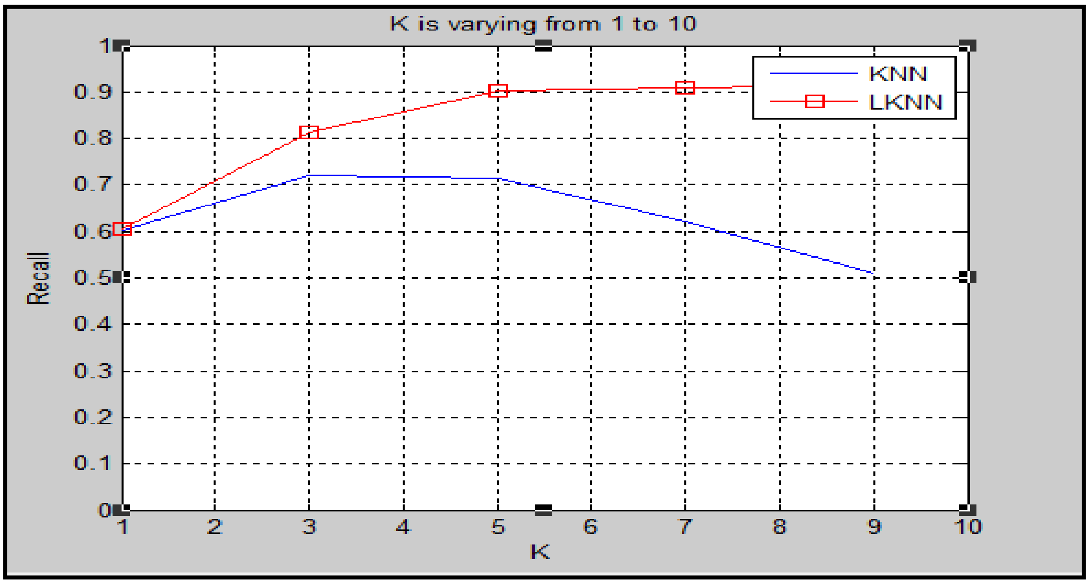


Figure 4.4: Recall values for different values of K in computer science journal articles dataset

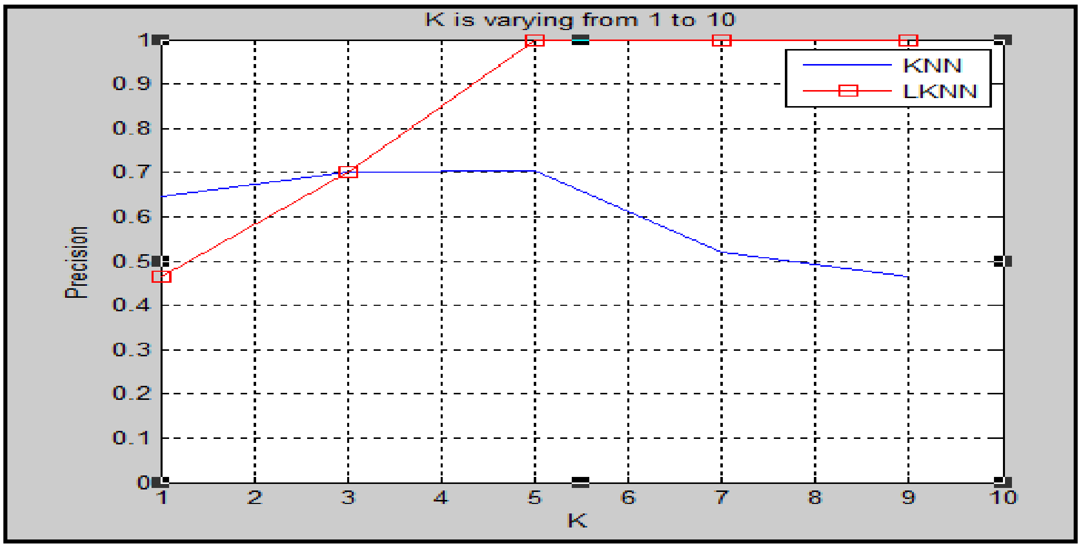


Figure 4.5:Precision values for different values of K in computer science journal articles dataset

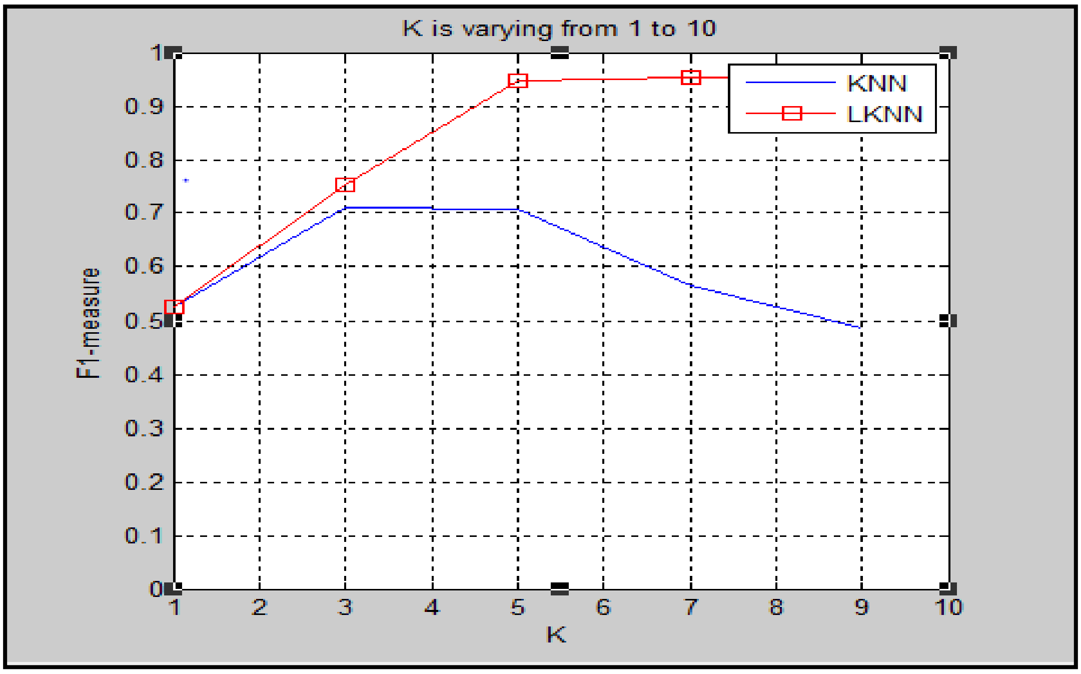


Figure 4.6: F1- measure values for different values of K in computer science research articles dataset

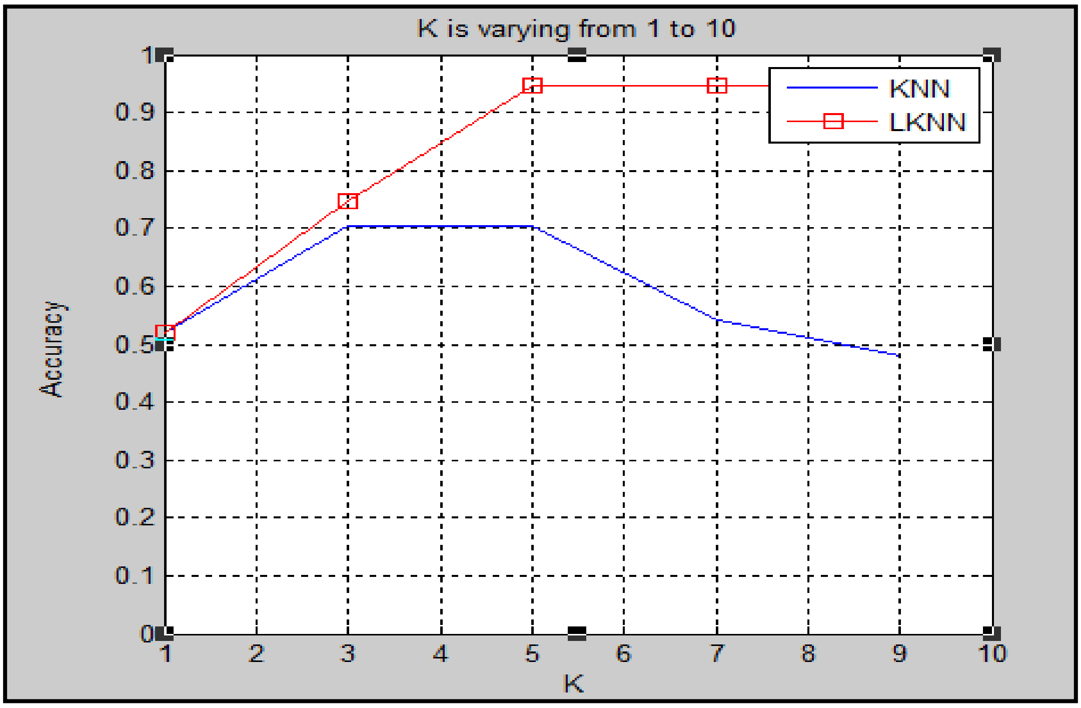


Figure 4.7: Accuracy Values for different values of K in computer science research articles dataset

4.5.2 Ohsumed Collection

Also, the experiments are conducted with Ohsumed Collection dataset for measuring the parameters: Recall, Precision, F1-measure and Accuracy. In figures 4.8, 4.9, 4.10 and 4.11, the results for Recall, Precision, F1- measure and Accuracy values for various values of K for Ohsumed Collection are shown respectively. The tables 4.2, 4.3, 4.4 and 4.5 also show the values of these parameters for Ohsumed Collection respectively. The results listed are the best results which we get for each algorithm in our experiments.

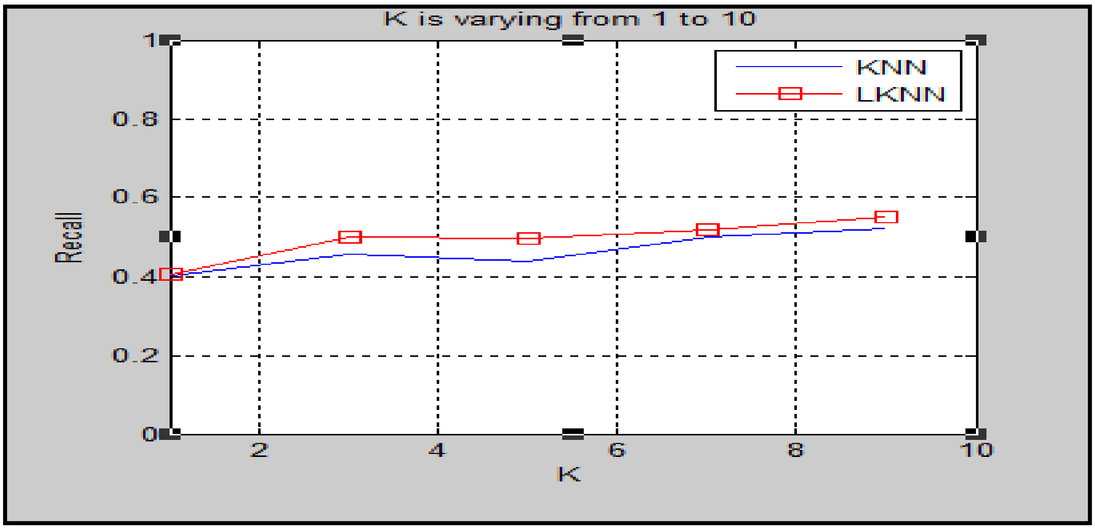


Figure 4.8: Recall Values for different values of K in Ohsumed Collection dataset

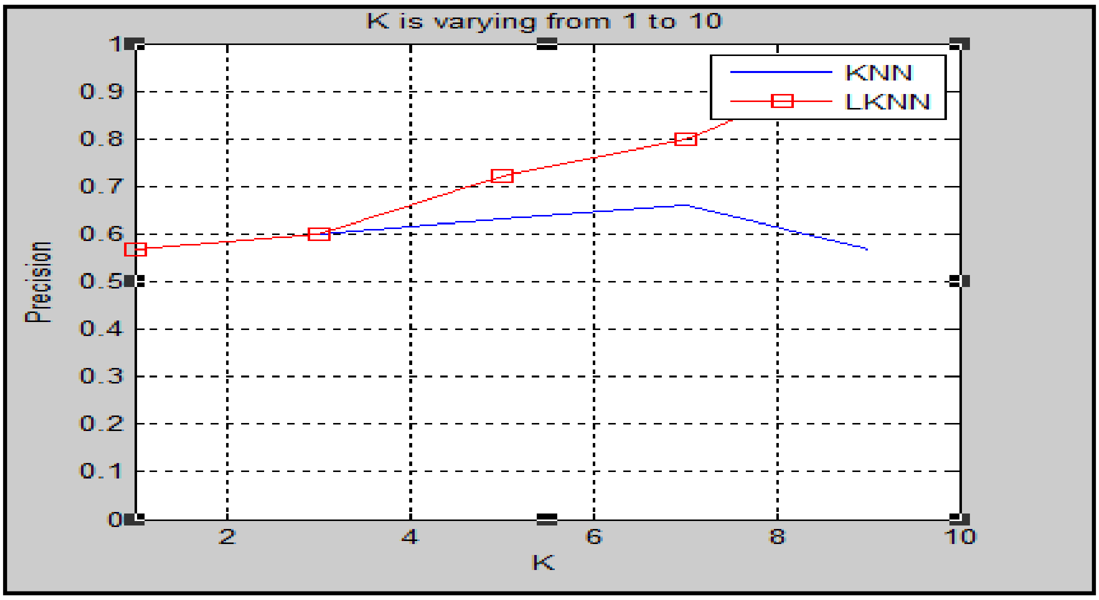


Figure 4.9: Precision Values for different values of K in Ohsumed Collection Dataset

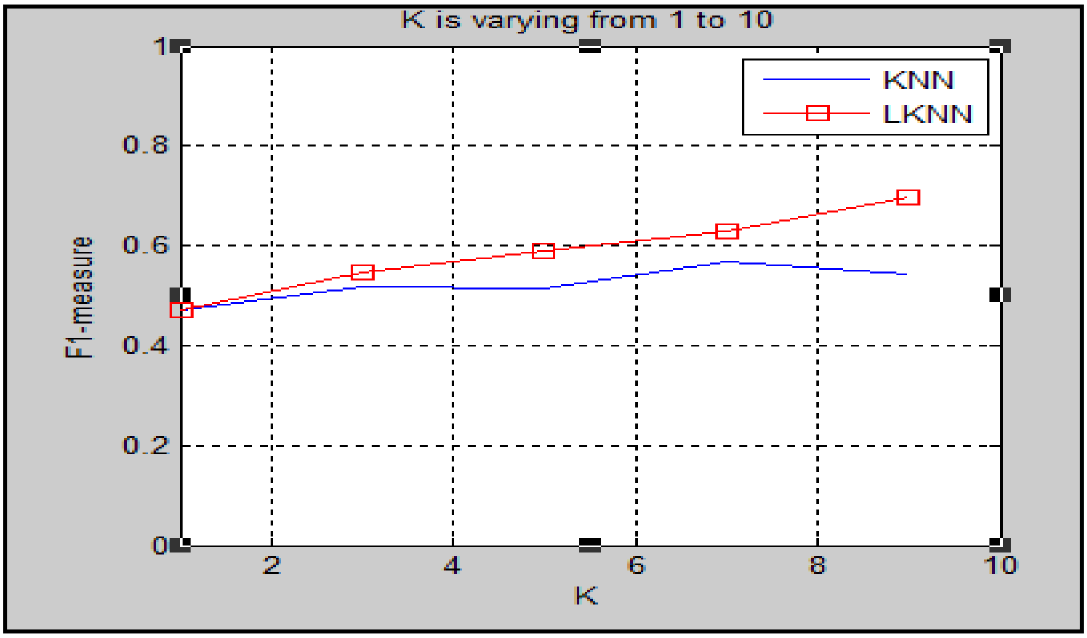


Figure 4.10: F1- measure Values for different values of K in Ohsumed Collection dataset

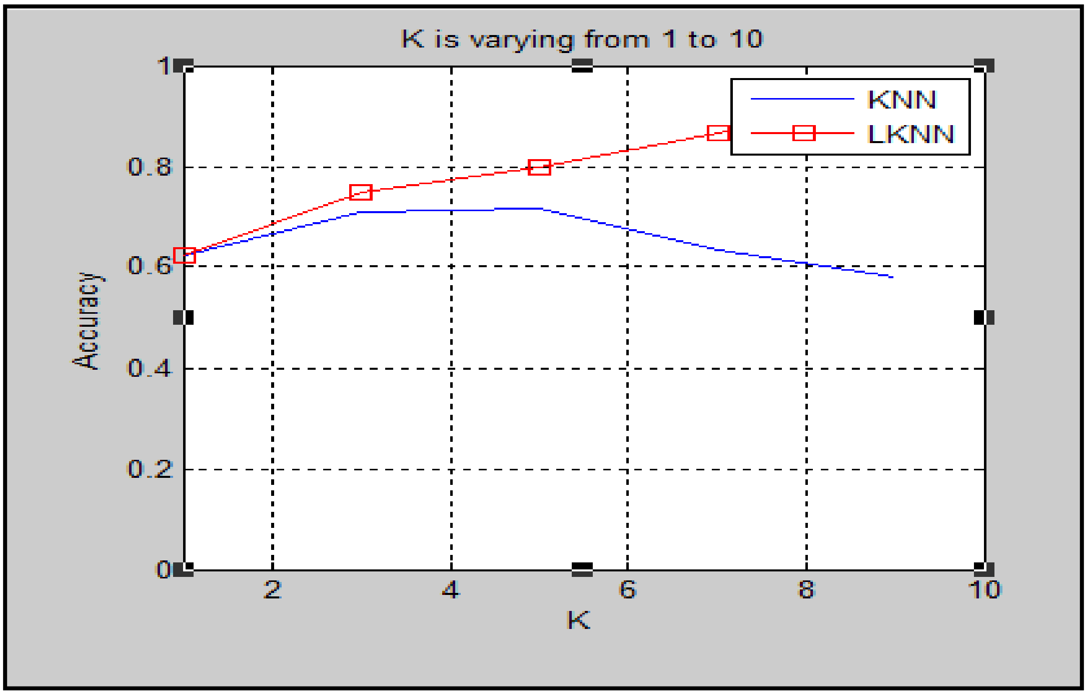


Figure 4.11: Accuracy Values for different values of K in Ohsumed Collection dataset

From the results it can be noticed that the values of Recall, Precision, F1-measure and Accuracy increase with the increase in the values of K.

Table 4.2: Values of Recall for different values of K in Ohsumed Collection

Values of K	KNN	LKNN
1	0.402	0.405
3	0.456	0.501
5	0.437	0.498
7	0.501	0.520
10	0.521	0.550

Table 4.3: Values of Precision for different values of K in Ohsumed Collection

Values of K	KNN	LKNN
1	0.566	0.566
3	0.601	0.601
5	0.631	0.721
7	0.660	0.800
10	0.566	0.956

Table 4.4: Values of F1-measure for different values of K in Ohsumed Collection

Values of K	KNN	LKNN
1	0.470	0.472
3	0.517	0.546
5	0.516	0.589
7	0.569	0.630
10	0.542	0.698

Table 4.5: Values of Accuracy for different values of K in Ohsumed Collection

Values of K	KNN	LKNN
1	0.625	0.625
3	0.710	0.750
5	0.715	0.800
7	0.635	0.867
10	0.581	0.947

In this chapter, LKNN algorithm has been proposed. The proposed algorithm is used for single-label categorization of text documents. It is based on a lexical approach. The proposed algorithm has been tested on two datasets: research articles of computer science domain and Ohsumed Collection. The performance of the proposed algorithm is compared with the standard KNN algorithm in terms of parameters like Recall, Precision, F1-measure and Accuracy. The proposed algorithm has shown a good performance.

5 MULTI-LABEL TEXT CATEGORIZATION

In this chapter, a Lexical-Semantics based KDT process (LS-KDT) is proposed. The proposed process is divided into a series of sub processes called as phases. There are seven phases in the proposed LS-KDT process. These are: Text Document Collection, Data Pre-processing, Lexical Analysis, Semantic Analysis, Classification, Ranking of labels and Knowledge Discovery. As well as new grammar rules for Classification phase are also proposed in this chapter.

The details of all the phases are described in this chapter. The description includes flow diagram of phase, Algorithm, pseudo code etc.

During the research, our main focus was on Text Analysis phase of the proposed U-STRUCT framework. For single-label text categorization, we proposed a lexical based KNN algorithm. The proposed algorithm is extended to multi-label categorization. A modified Knowledge discovery process, i.e. Lexical-Semantics based Knowledge Discovery process (called as LS-KDT) is proposed. The proposed process is used for the automated categorization of multi-label text documents. It works for both single-label and multi-label text categorization.

5.1 Proposed LS-KDT Process

We have proposed a Lexical and Semantics based knowledge discovery process for the automated categorization of text documents. It is called as LS-KDT. The proposed process is subdivided into a series of phases. The proposed LS-KDT process consists of seven phases, i.e. Text Document Collection, Data Pre-processing, Lexical Analysis, Semantic Analysis, Classification, Ranking of labels and Knowledge Discovery. The figure 5.1 given below shows the proposed LS-KDT process. The details of phases are given below.

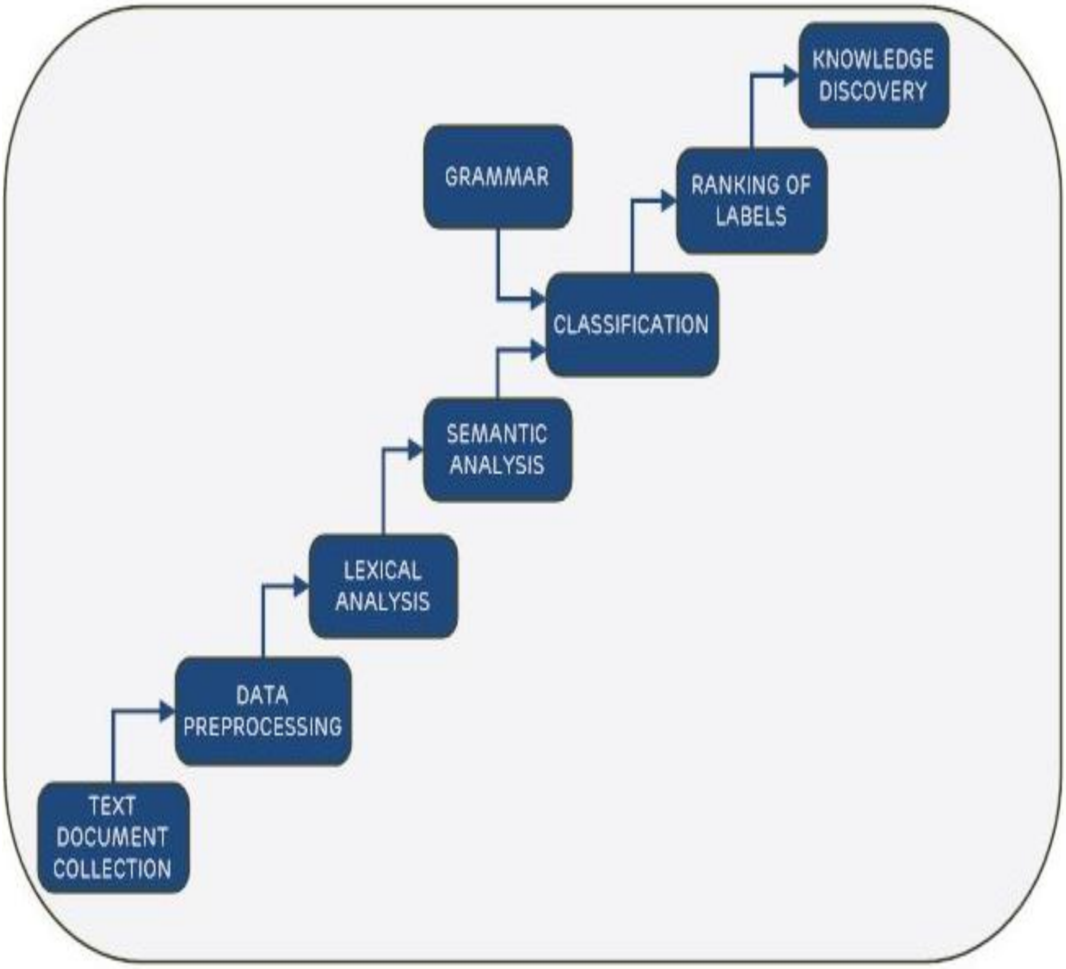


Figure 5.1: Proposed LS-KDT Process

5.2 Phases of Proposed LS-KDT Process

The phases of the proposed LS-KDT process are described below in detail.

5.2.1 Text Document Collection

In the first phase, the text documents are collected. The text documents may be journal articles, medical documents, legal documents etc. The documents may be single-label or multi-label in nature.

5.2.2 Data Pre-processing

Second phase of the proposed LS-KDT process is data pre-processing. In this phase, data is cleaned and prepared for mining. It is an important phase as quality of results depends on the quality of data. Further stop words are also removed by using a predefined Stop words list (given in References section).

5.2.3 Lexical Analysis

Third phase of the proposed LS-KDT process is Lexical Analysis. The function of this phase is to scan the text document and identify tokens. This phase identifies the tokens and stores them along with their frequency (of occurrence) in a table. The ACM Computing Classification System, 2012 is a standard system used to identify the tokens. It consists of broad categories of computer science, which are further organized into sub categories.

This phase extracts the tokens from the Abstract of the article, Title as well as from the given Keyword list. The output of this phase is a list of tokens along with their frequency, for each journal article.

The working of Lexical Analysis phase is shown with the help of flow diagram in figure 5.2. It can be explained as follows:

A set of research articles is fed as input to this phase. It removes Stop words from the abstract of the research articles by using a Stop words list. It extracts the tokens from the Title of the article, Keywords list as well as from the Abstract. If some token is found in the title or in the keyword list, then its frequency is increased by 2 as it is more important. The output of this phase is a list of tokens along with their frequency, for each article. A research article can be represented as a set of tokens along with their frequency and the entire collection can be represented formally as per the definition 1 and 2 given below:

Definition 1(Research Article). A research article denoted as $J_i = \{(t_1, f_{i1}), (t_2, f_{i2}) \ldots (t_m, f_{im})\}$is represented as a set of tokens t_i together with corresponding frequency f_{ij}.

Definition 2 (Research Articles Set). A Research articles set, denoted as J = {J_1, J_2…. J_i …. J_n}, is a set of articles, where n is the total number of articles in J.

The output of this phase is a sequence of tokens (along with their frequency) for each research article. These tokens are passed as input to Semantic Analysis phase.

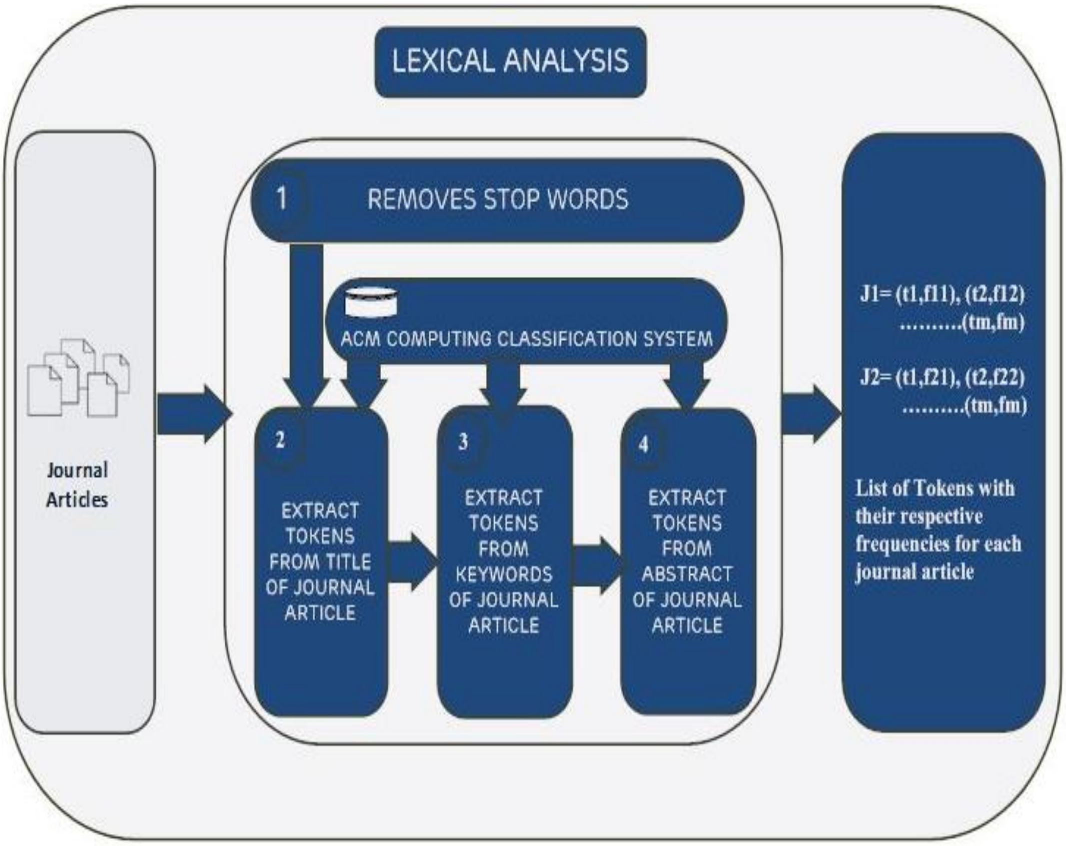

LEXICAL ANALYSIS
Journal Articles
1 REMOVES STOP WORDS
ACM COMPUTING CLASSIFICATION SYSTEM
2 EXTRACT TOKENS FROM TITLE OF JOURNAL ARTICLE
3 EXTRACT TOKENS FROM KEYWORDS OF JOURNAL ARTICLE
4 EXTRACT TOKENS FROM ABSTRACT OF JOURNAL ARTICLE
J1= (t1,f11), (t2,f12) ……….(tm,fm)
J2= (t1,f21), (t2,f22) ……….(tm,fm)
List of Tokens with their respective frequencies for each journal article

Figure 5.2: Flow Diagram of Lexical Analysis Phase

The algorithm of proposed LS-KDT process is shown in figure 5.3 Different phases of the proposed process are called one by one in the algorithm.

```
Algorithm: Proposed LS-KDT process
Input: Title - T <String>, Set of Keywords - K = {k1, k2, k3…….kn} <List[Strings]>, Abstract- Ab <String>, The 2012 ACM Computing Classification System - A<String>, WordNet – W<String>, Set of research articles - J [T, K, Ab, A, W].
Output: {<$T_i$, $C_j$> | $T_i$ is the title belonging to a set of classes $C_j$ where i>0, j>=1}
for (each Title $T_i$)
Begin
    1   LAM = Lexical Analysis (T, K, Ab, A, J)
//LAM is a variable used to store the output of Lexical Analysis Phase
    2   <$T_i$, µ > = Semantic Analysis (LAM)
// µ is a variable used to store the reduced set of tokens
    3   <$T_i$, C> = Classification (Ti, µ)
//C is a variable used to store set of classes or categories
    4   Ordering of Class Labels = Ranking ($T_i$, C)
End
```

Figure 5.3: Algorithm of proposed LS-KDT process

The third phase of proposed LS-KDT process is Lexical Analysis. The output of this phase is the set of tokens which are identified from the title, Abstract and Keywords of the research article.

Figure 5.4 shows the algorithm of Lexical Analysis phase of the proposed LS-KDT process.

```
Algorithm:  Lexical Analysis Phase
Input:   Title - T <String>, Set of Keywords - K = {$k_1$, $k_2$, $k_3$…….$k_n$}
                <List[Strings]>, Abstract- Ab <String>, The 2012 ACM
                Computing Classification System - A<String>, Set of research
                articles - J [T, K, Ab, A].
Output:  LAM= {<$T_i$, Token, Freq>| for each $T_i$, Ǝ Token = {$t_{i1}$, $t_{i2}$, $t_{i3…}t_{im}$},
         & Freq = {$f_{i1}$, $f_{i2}$, $f_{i3…}f_{im}$}}
  Begin
    1. for each $J_i$ Є J
  Begin
1.1 Parse the title string T to extract the tokens $t_i$ using A and insert it into
      set Token. Also record the frequency $f_i$ of $t_i$ in Freq respectively.
              // identify the tokens in the title.
1.2. Extract the tokens from K and add them to set Token. Also append the
      frequency $f_i$ for each $t_i$.
              // identify the tokens in Keywords.
1.3. Extract the tokens from Ab. Append ($t_i$) in set Token and append ($f_i$) in Freq
     respectively.
             // identify the tokens in the Abstract.
 1.4.   For each $f_i$ and $t_i$, for all i= 1……., m till there are no repeated tokens
                        Begin
                            if any $t_i$ equals $t_j$
                             then $f_i$ = $f_i$ +1
                                remove $t_j$ and $f_j$
                         End    // removes redundant tokens

                      End
                 End
```

Figure 5.4: Algorithm of Lexical Analysis phase

The pseudo code of Lexical Analysis phase is given below in figure 5.5.

The input to this phase is a file containing title, Abstract and keywords of a research article.

```
        lexicalAnalyser.analyseArticle(file);
          //Read contents of each file

       lexicalAnalyser.analyseJournalArticle(articleText);
       //Return object containing title, abstract and keywords

       StopWordsRemover.removeStopWords(article);
       //Remove stop words from article

       LexicalAnalyser.analyseContent(textContent, journalArticle, type);
       //The type in the function signature is the type   of the text passed
        (e.g. title, keywords of abstract)

        Execute the following steps for each journal article
                1. Tokenize content and execute below steps for each token.
                2. Use MorphologicalProcessor(from wordnetlibray) to get the
                   base form of the word.
                3. Call acmClassificationSystem.ContainsToken(word)) to
                   check if the word is a valid token
                4. Call updateFrequency(journalArticle, updatedToken, word,
                   type) to update journal Article object with the frequency of
                   the token.
```

Repeat the above steps for keywords and abstract of each research article.

The above steps result in the list of journal Article objects containing tokens along with their frequencies.

Figure 5.5: Pseudo code of Lexical Analysis phase

5.2.4 Semantic Analysis

The next phase of the proposed LS-KDT process is Semantic Analysis. This phase receives a stream of tokens with their frequency as input from Lexical Analysis phase. The main aim of this phase is to reduce the dimensionality of text documents by using the concept of semantics. The goal is to minimize the number of tokens required to categorize a journal article. To reduce the dimensionality, two concepts have been used in this phase: Dimension Reduction and WordNet.

The real-world text documents are multi-label in nature. For example, a book may belong to multiple categories like science, engineering, computers etc. A movie can belong to various film genres like science, drama, action, horror etc. Also, the text documents possess large size and huge number of features. This leads to a great challenge in categorization of text documents. So, to meet this challenge, the dimension reduction method is used.

5.2.4.1 Dimension Reduction

Dimension reduction (Joachims, 1998; Godbole, 2004) is a method used to reduce the features as well as the dimensions of text documents. It is used in many applications like image recognition problems, biological data etc. Many researchers have contributed in this field. Feature selection (Koller, 1996) and feature reduction (Mika et al., 1999; Pearson, 1901) are concepts used in dimension reduction. Feature selection selects a subset of features; Feature reduction reduces the dimensionality by combining certain features. The choice of the method depends on the application domain and the type of problem. Different researchers have contributed and suggested their methods for dimension reduction. Table 5.1 shows a summary of work done by different authors in dimension reduction and characteristics of our proposed approach.

Table 5.1: Work done in dimension reduction by different authors and features of our proposed approach

S. No	Authors	Problem Addressed	Proposal	Dataset	Results
1	Deerwester, S. (1990)	Dimension Reduction	Singular value decomposition (SVD) based on decomposing a large term by document matrix.	MED- medical abstracts, CISI- information science abstracts.	Results show that it is a promising method.
2	Ram kumar, A. S. & Poorna B. (2016)	Dimension Reduction	Proposed Document Clustering Using Dimension Reduction.	BBC Sports Dataset.	This method shows significant improvement in Accuracy, Precision and Recall.
3	Kim, H. et al. (2005)	Dimension Reduction	Support Vector Machines are used for reducing the dimensions of documents.	MEDLINE dataset, Reuters 21578	It achieves better efficiency in both training and testing the data.
4	Gabrilovich, E. & Markovitch, S. (2009)	Semantic Interpretation of Natural Language texts	Explicit Semantic Analysis technique(ESA)	Used concepts derived from Wikipedia.	Significant improvements over existing algorithms.

5	Hou, C. et al. (2010)	Dimension Reduction	Constraints are used for multiple view dimension reduction problems.	WebKB, 20 NewsGroup and Sonar data.	Their approach outperformed other approaches.
6	Li, Z. et al. (2011)	Dimension Reduction	Concise semantic Analysis technique	Reuters 21578, 20 NewsGroup and Tancorp.	Their approach reaches a comparable performance with SVM.
7	Mallick, K. & Bhattacharyya S. (2012)	Dimension Reduction	Distance between data points is counted and scatter matrix is calculated.	Reuters dataset	Their approach is more efficient than other state of art algorithms.
8	Guan, H. et al. (2013)	Dimension Reduction	Imprecise Spectrum Analysis for fast dimension reduction.	Web KB, Reuters 21578 and 20 News Group	Their approach achieves fast and competitive classification accuracy with state of art algorithms.

9	Proposed LS-KDT Approach	Dimension Reduction	Semantic Analysis using WordNet.	Ohsumed dataset, Computer Science research articles dataset.	Our approach achieves a significant reduction in the number of tokens.

In our proposed LS-KDT process, we have used a new semantic approach for dimension reduction in text documents. Our approach is based on the use of WordNet.

5.2.4.2 WordNet

It is a lexical database in English language which stores words as nouns, verbs, adjectives and adverbs (Miller, G. A. et al., 1990). We have considered nouns in our work. In WordNet, similar words or synonyms are grouped together in the same set called as synset. For example, in figure 5.6, a sub graph of relationships used in WordNet is shown. We can notice that {dwelling, abode} and {house, home} exist in a synset. These synsets are linked to each other with the help of relationships.
There are different relationships that exist between the synsets. These are synonyms, hypernyms, hyponyms, meronyms, antonyms etc.

- Hypernymy and Hyponymy: These are ‘ISA’ relationships. Hypernymy links specific synsets like {house, home} to general synsets like {dwelling, abode}. Hyponymy is the opposite of hypernymy. It links general synsets to specific ones.
- Meronymy: It is ‘PART-WHOLE’ relationship. For example: Meronymy holds between synsets like {chair} and {back, backrest}, {seat} and {leg} etc.

Many authors have used WordNet in the field of text categorization. The pioneers in this area were the authors Scott and Matwin in 1998. They used Ripper algorithm in text categorization. Then Jensen and Martinez improved categorization in 2000 by using contextual and conceptual features. Further in 2008, Wang & Domeniconi used Wikipedia to build semantic kernels. In 2012, authors Li C. H. et al. used WordNet and thesaurus for text categorization. Maciołek P. (2013) used graph-based approach for semantic analysis of text documents. Wei et al. (2015) used WordNet in text clustering. They have used many concepts like ontology, relations and lexical chains for word sense disambiguation. Patil & Ravindran (2015) used hypernyms (of WordNet) for unsupervised multi-label classification.

We have focused on hypernym relationship of tokens. This was used to obtain more general and conceptual meaning of the words. And secondly, the authors Dave, et al. (2003) and Sedding, (2004), have already stated that hypernyms give better results than synonyms.

In our work, we have used WordNet 2.1.

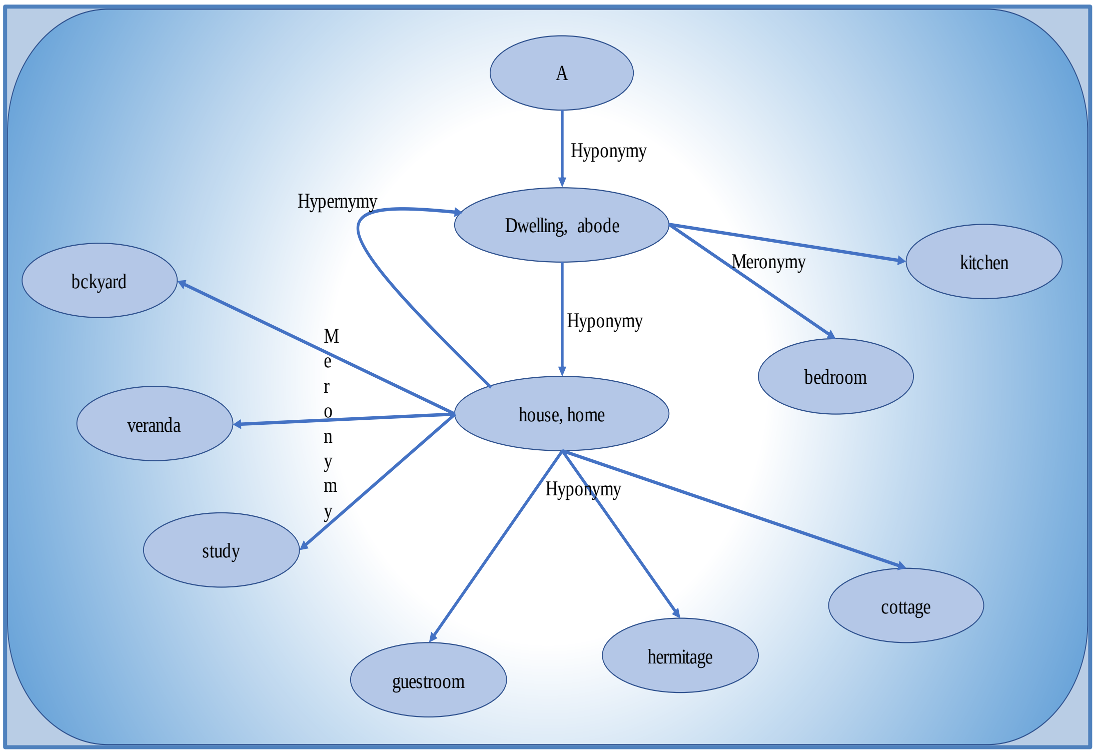


Figure 5.6: A sub graph from WordNet

The following figures 5.7(a) and 5.7 (b) show the details of Semantic Analysis phase step by step.

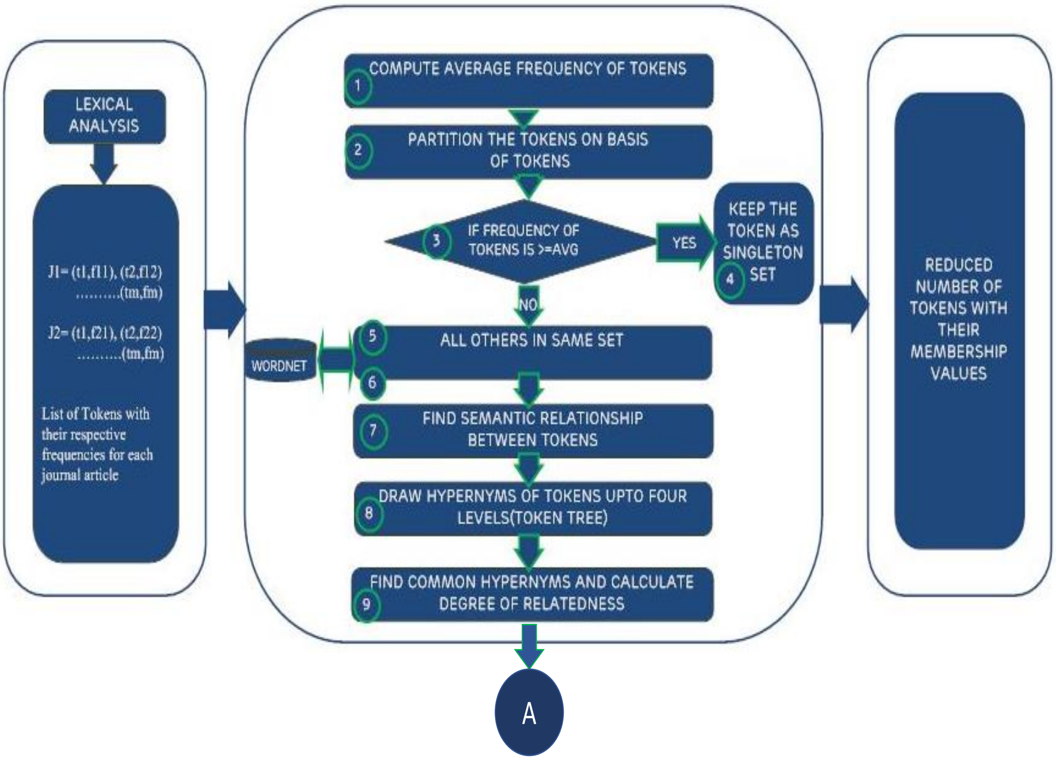

LEXICAL ANALYSIS
J1= (t1,f11), (t2,f12) ....(tm,fm)
J2= (t1,f21), (t2,f22) ....(tm,fm)
List of Tokens with their respective frequencies for each journal article
1 COMPUTE AVERAGE FREQUENCY OF TOKENS
2 PARTITION THE TOKENS ON BASIS OF TOKENS
3 IF FREQUENCY OF TOKENS IS >=AVG
YES
KEEP THE TOKEN AS SINGLETON SET
4
NO
WORDNET
5 ALL OTHERS IN SAME SET
6
7 FIND SEMANTIC RELATIONSHIP BETWEEN TOKENS
8 DRAW HYPERNYMS OF TOKENS UPTO FOUR LEVELS(TOKEN TREE)
9 FIND COMMON HYPERNYMS AND CALCULATE DEGREE OF RELATEDNESS
REDUCED NUMBER OF TOKENS WITH THEIR MEMBERSHIP VALUES
A

Figure 5.7 (a): System Flow of Semantic Analysis Phase

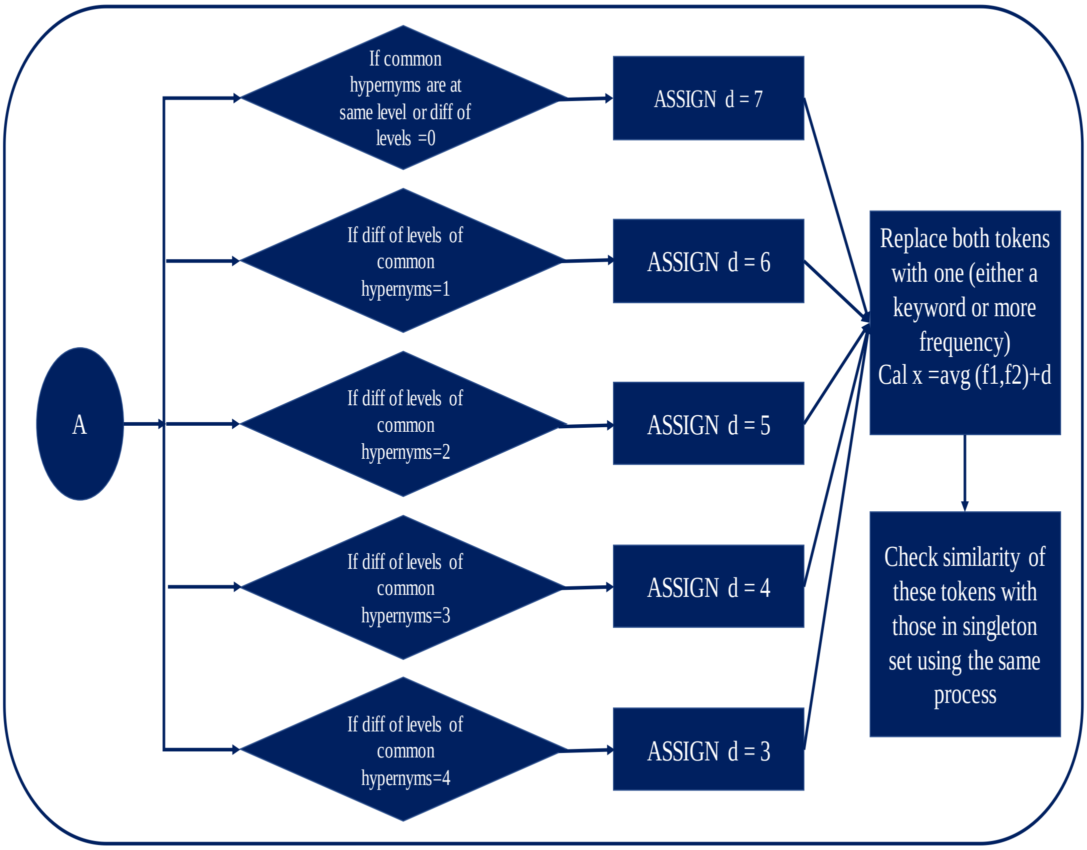

A
If common hypernyms are at same level or diff of levels =0
ASSIGN d = 7
If diff of levels of common hypernyms=1
ASSIGN d = 6
If diff of levels of common hypernyms=2
ASSIGN d = 5
If diff of levels of common hypernyms=3
ASSIGN d = 4
If diff of levels of common hypernyms=4
ASSIGN d = 3
Replace both tokens with one (either a keyword or more frequency)
Cal x =avg (f1,f2)+d
Check similarity of these tokens with those in singleton set using the same process

Figure 5.7 (b): System Flow of Semantic Analysis Phase

In this phase, the input is a list of tokens along with their frequency for each research article, obtained from the previous phase. Firstly, the average frequency of tokens is calculated. The tokens are partitioned on the basis of their frequency. The tokens whose frequency is more than the average frequency are kept as singleton sets (as they are important). The rest of the tokens are put in the same set. These tokens (which are in the same set) are then checked for similarity. Using WordNet, hypernyms of tokens are drawn up to four levels to form hypernym trees. If two tokens have a common hypernym, that means they are related. This relationship is measured with the help of degree of relatedness. To calculate the degree of relatedness (denoted by d), a set of rules are defined as shown in figure 5.8.

1. IF (common hypernym of two tokens are at the same level) or (difference of levels is 0) THEN degree of relatedness is VERY HIGH, d=7.
2. IF (common hypernym of two tokens are at a difference of level 1) THEN degree of relatedness is ALMOST HIGH, d=6.
3. IF (common hypernym of two tokens are at a difference of level 2) THEN degree of relatedness is HIGH, d=5.
4. IF (common hypernym of two tokens are at a difference of level 3) THEN degree of relatedness is LOW, d=4.
5. IF (common hypernym of two tokens are at a difference of level 4) THEN degree of relatedness is VERY LOW, d=3.

Figure 5.8: Rules to calculate degree of relatedness

This can be explained as follows: If t1 and t2 are two tokens whose similarity is to be checked. The membership value of a token (denoted by x) is calculated by the following formula:

$$x= \text{average}\ (f1, f2) +d,$$

where f1 and f2 are frequencies of the two tokens t1 and t2 respectively and d is degree of relatedness.

Out of the two tokens t1 and t2, one representative is selected and assigned the membership value x. If the token is a keyword, it is given high preference otherwise the token that is more frequent, is preferred. This process is repeated. Finally, the reduced number of tokens with their membership values is obtained as the output. So, basically in semantic analysis phase, we focus on reducing the dimensionality of text documents using the concept of semantics.

For example, if two tokens are Learning and Algorithm as shown in figure 5.9, their hypernym trees can be drawn by using hypernym relationship (of WordNet) as follows:

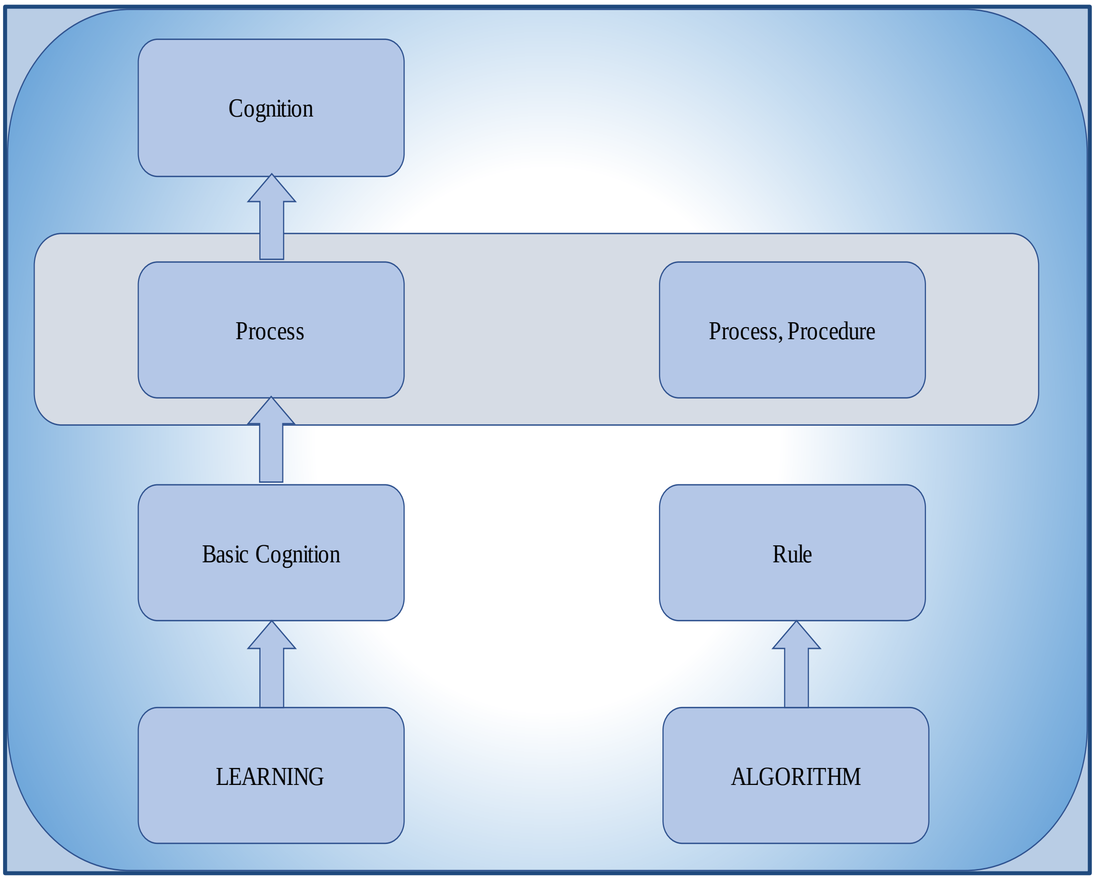


Figure 5.9: Example to find similarity between two tokens

The token LEARNING comes under the category Basic Cognition which is further a child of Process and Process is the child of Cognition (root). Similarly, ALGORITHM is the child of Rule, which is a child of the root Process, Procedure. Now, it is clear that in both the above trees, there is a common hypernym which is Process. And both the tokens have it at same level. So, rule 1 is satisfied i.e. degree of relatedness is very high (d=7). And suppose we assume that frequency of Learning is 3 and frequency of Algorithm is 2. So, token Learning is retained with membership value x calculated in following way.

$$x = 3 \text{ (average of 3 and 2)} + 7 = 10$$

Figure 5.10 shows the Algorithm of Semantic Analysis phase.

```
Algorithm:  Semantic Analysis Phase

Input: LAM= {<$T_i$, Token, Freq>| for each $T_i$ ∃ Token = {$t_{i1}$, $t_{i2}$, $t_{i3...}$, $t_{im}$} & Freq = {$f_{i1}$, $f_{i2}$, $f_{i3}$ .$f_{im}$}
Output:<$T_i$, µ >, where µ   is the reduced set of tokens along with their membership values
for each research article $T_i$
  Begin
      1. Compute avg = ($f_{1+}$ $f_{2+}$ $f_{3+}$….$f_m$) /(m).  //m is the number of tokens
      2. Construct the partitions $\pi_i$ of set of tokens till the partitions are singleton or they cannot
         be further reduced (or replaced) by their hypernyms.

          2.1 Initially $\pi_0$ = {< Token, Freq> | Token = {$t_{i1}$, $t_{i2}$, $t_{i3...}$, $t_{im}$}, Freq = {$f_{i1}$, $f_{i2}$, $f_{i3...}f_{im}$}}
          2.2 Next, we construct $\pi_{i+1}$ from $\pi_i$ as follows:
                $\pi_{i+1} =$ {$\pi_{(i+1,0)}$, $\pi_{(i+1,1)}$, $\pi_{(i+1,2)}$ ……. $\pi_{(i+1, k)}$}, where $\pi_{(i+1, k)}$ =

                $\begin{cases} |\{\pi_{(i+1, k)}\}| = 1 & \text{//singleton sets containing important tokens} \\ |\{\pi_{(i+1, k)}\}| > 1 & \end{cases}$

                       where k = 0, 1….
        2.3 Consider| $\pi_{(i+1, k)}$ |   > 1

        2.3.1 For each token $t_i$ Є$\pi_{(i+1, k)}$
                     Find the hypernyms up to four levels using WordNet.
                     Create a hypernym tree for each $t_{i,}$ which is the leaf node of the tree and its
                       root is the general area or broad area to which the token belongs.

                          If there is a match between the hypernyms of tokens
                             then replace the tokens $t_i$ with the one that is either the keyword
                             or if it has more frequency.
                             And calculate membership value of the token by the following
                             formula:
                             If t1 and t2 are two tokens with frequency f1 and f2
                             respectively, then
                                x = average (f1, f2) +d, where d is the degree of relatedness
                                      that is computed with the help of proposed rules in
                                   figure 5.8. // Find semantic relations between the tokens.

       2.4 Consider| $\pi_{(i+1, k)}$ | = 1     // Singleton sets are considered to look for semantic relations
               2.4.1 For each $\pi_{(i+1, k)}$
                 Draw the hypernyms for each token $t_i$
                  If (there is a match found in hypernyms) then restore the token in the singleton
                 set and remove the token $t_i$ Є   | π (i+1, k) | > 1
             // Tokens in the singleton sets are important tokens.
    3.  Finally reduced set of tokens is stored in µ.
End
```

Figure 5.10: Algorithm of Semantic Analysis phase

The pseudo code of Semantic Analysis phase is given in figure 5.11.

```
This phase takes as input a list of research articles along with list of tokens
identified from the previous phase.
 semanticAnalyzer.performSemanticAnalysis(List<JournalArticle> articles);
    //Call next   function for each article containing tokens with their frequencies
 semanticAnalyzer.performSemanticAnalysis(JournalArticle article);

semanticAnalyzer.removeTokensWithFrequencyOne(JournalArticlejournalArticle)
 semanticAnalyzer.createTokenSetsBasedOfAverageFrequency(article);
 semanticAnalyzer.performSemanticAnalysisOfNonSingletonTokens(process,
   processes);
    // Process models each pass (e.g. π0, π1, …) through the process of matching
   hypernyms. processes is the list of such processes.
semanticAnalyzer.performSemanticAnalysisWithSingletonTokens(process,
processes);
semanticAnalyzer.matchTokensHypernyms(tokenFrequency1,
tokenFrequency2);
wordnetService.retrieveHypernymeTrees(token,type);
   //type is either noun or verb
semanticAnalyzer.matchTokensHypernyms(tokenFrequency1,
tokenFrequency2, hypernymTree1, hypernymTree2);
```

Figure 5.11: Pseudo code of Semantic Analysis Phase

5.2.5 Classification

The next phase of the proposed LS-KDT process is Classification. The aim of this phase is to identify classes/categories of research articles. The new grammar rules are proposed in this phase for classification of research articles.

The flow diagram of Classification phase is given in figure 5.12.

This phase receives as input tokens with their membership values from the previous Semantic Analysis phase. It performs two functions, first, it merges tokens (till now that were single) using the Keyword List (of that research article). Multi word tokens can also be handled. The membership values of individual tokens are summed up. Second, this phase identifies classes of tokens using the standard ACM Computing Classification System and proposed grammar rules generated.
The output of this phase is a list of tokens (merged) with their membership values(x) and classes.

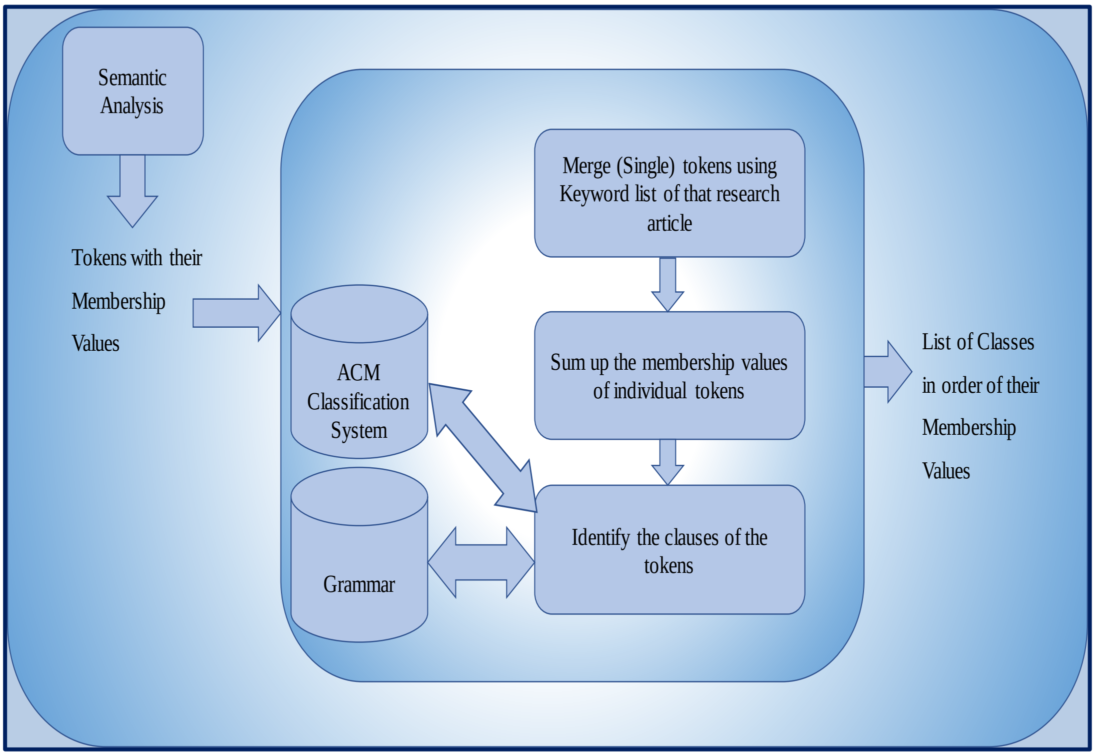

Semantic Analysis
Tokens with their Membership Values
Merge (Single) tokens using Keyword list of that research article
ACM Classification System
Sum up the membership values of individual tokens
List of Classes in order of their Membership Values
Grammar
Identify the clauses of the tokens

Figure 5.12: Flow diagram of Classification Phase

The algorithm of Classification phase is given in figure 5.13.

```
Algorithm: Classification Phase
    Input:   <$T_i$, μ >, where, $T_i$ is the title of the research article and
                         μ is the reduced set of tokens
    Output:  <$T_i$, C >, where $T_i$ is the title of the research article and
                         C is set of respective classes to which $T_i$ belongs
    Begin
     1. Input the reduced set of tokens (μ) for the title $T_i$.
     2. Merge (single) tokens using keywords list of that research article.
     3. Sum up the membership values of individual tokens.
     4. Identify the classes of the tokens using the standard ACM Computing
        Classification System and the Grammar generated.
     5. Display the set of classes present in a research article in their order of
        membership as the result.
    End
```

Figure 5.13: Algorithm of Classification Phase

In step 4 of the above algorithm, the grammar rules are generated using ACM Computing Classification System. This grammar rules generation is given in the next section.

5.2.5.1 Proposed Grammar Rules

This section proposes grammar rules for Classification phase of the proposed LS-KDT process. A Grammar G is constructed in this phase using the standard ACM Computing Classification System.

The formal definition of G = (N, T, P, S),

where N denotes the set of Non-terminals,

T denotes the set of Terminals,

P is the set of production rules

and S ε N is the start symbol.

Now, in our case, N represents the broad categories or classes in standard ACM Computing Classification system.

N= {Computer System Organization, Networks, Software and its Engineering, Theory of Computation, Mathematics of Computing, Information Systems, Security and Privacy, Human Centered Computing, Computing Methods, Applied Computing, Social and Professional Topics}

T represents the lowest level categories under the broad categories shown in N.

The next attribute is P, which is the set of production rules.

The production rules are of the form A → α,

where A ε N and α ε N ∪ T.

Hence, the production set P is as follows:

P = {1. S→ A | B |C |D……|K

where A = Computer System Organization
B = Networks
C= Software and its Engineering
D = Theory of Computation
E= Mathematics of Computing
F= Information Systems
G = Security and Privacy
H = Human Centered Computing
I = Computing Methods
J= Applied Computing
K = Social and Professional Topics

2. A→ A1|A2|A3……|A_m

3. B→ B1|B2…. |B_n

Similarly, we have defined the productions for all the broad categories- A to K.}.

The snapshot of the grammar for the broad category A that is, Computer System Organization is shown in figure 5.14.

1. S → Computer System Organization | Networks | Software and its Engineering| Theory of Computation| Mathematics of Computing | Information Systems | Security and Privacy | Human Centered Computing |Computing Methods | Applied Computing | Social and Professional Topics

2. Computer System Organization → Architecture | Embedded and cyber-physical systems | Real-time systems | Dependable and fault-tolerant systems and networks

3. Architecture→ Serial Architecture | Parallel architectures | Distributed architectures | other architectures

4. Serial Architecture → Reduced instruction set computing | Complex instruction set computing | Superscalar architectures | Pipeline computing | Stack machines

5. Parallel architectures→ Very long instruction word | Interconnection architectures | Multiple instruction, multiple data | Cellular architectures | Multiple instruction, single data |Single instruction, multiple data | Systolic arrays | Multi core architectures

Figure 5.14: Snapshot of grammar generated

The pseudo code of Classification phase is given in figure 5.15.

This phase takes as input a list of research articles along with reduced set of tokens with their membership values from the previous phase.

```
public void performClassification(List<JournalArticle>journalArticles)
      {if (journalArticles! = null &&journalArticles.size() > 0)
{for (JournalArticlejournalArticle: journalArticles)
{performClassification(journalArticle);}}}
private void performClassification(JournalArticlejournalArticle)
      {Article article = journalArticle.getArticle();
         String originalKeywords = article.getOriginalKeywords();
          String keywords [] = originalKeywords.split(",");
if (keywords! = null)
{for (String keyword: keywords)
  {String normalizedKeyword=
CommonUtilities.normalizeString(keyword.trim());
processKeyword(normalizedKeyword, journalArticle);}}
processNonKeywordTokens(journalArticle);}
   private void processKeyword(String keyword,JournalArticlejournalArticle)
{ClassificationSystemacmClassificationSystem=ClassificationSystem.getCurrent
ClassificationSystem();
ClassElementacmClassElement = acmClassificationSystem.getClass(keyword);
            List<String> tokens = getKeywordTokens(keyword);
if (tokens! = null &&tokens.size() > 0)
             {journalArticle.addClass(keyword, tokens, acmClassElement);}}
```

Figure 5.15: Pseudo code of Classification Phase

5.2.6 Ranking of Labels

The next phase is Ranking of class labels. In multi-label categorization of text documents, ranking of class labels is an important concept. Ranking means a strict ordering of class labels according to their priority. This helps in categorizing the text document that is, the most appropriate category has the highest ranking and then going down to other lower categories in the ranking. This phase receives a list of tokens with their membership values and classes. In this phase, we have proposed eight quantifiers – none, almost none, very low, low, high, higher, highest and all (Jindal & Taneja, 2015) as shown in table 5.2. These quantifiers help in the process of ranking of labels in multi-label categorization of text documents. The output of this phase is ranking or ordering of class labels on the basis of membership values of tokens.

5.2.6.1 Proposed Quantifiers

This phase receives a list of tokens with their membership values and classes from the Classification Phase. Let z be the total value of a class label which is taken as sum of its frequency and degree of relatedness. We assume the value of z as 50. Let x be the membership value of a particular class label. This membership value is obtained from the previous phase that is Classification. These quantifiers help in the process of ranking of labels in multi-label categorization of text documents. A linguistic mapping of membership values of tokens is done as shown in table 5.2.

Table 5.2: Definition of proposed Quantifiers

Quantifier	Definition
None	x=0
Almost none	x>=1 and x<z-45
Very low	x>=z-45 and x<z-40
Low	x>=z-40 and x<z-30
High	x>=z-40 and x<z-20

Higher	x>=z-20 and x<z-10
Highest	x>=z-10 and x<z
All	x=z

The following figure 5.16 describes the process used for ranking phase. The working of this phase can be explained as follows:

This phase receives tokens with their classes and membership values from the previous phase that is classification. Firstly, similar classes are merged together, and their corresponding membership values are also added up. Then, percentage membership value is calculated. And further, quantifiers are assigned according to the proposed rules given in table 5.2. The output of this phase is a ranking of class labels, their membership values and the corresponding quantifiers.

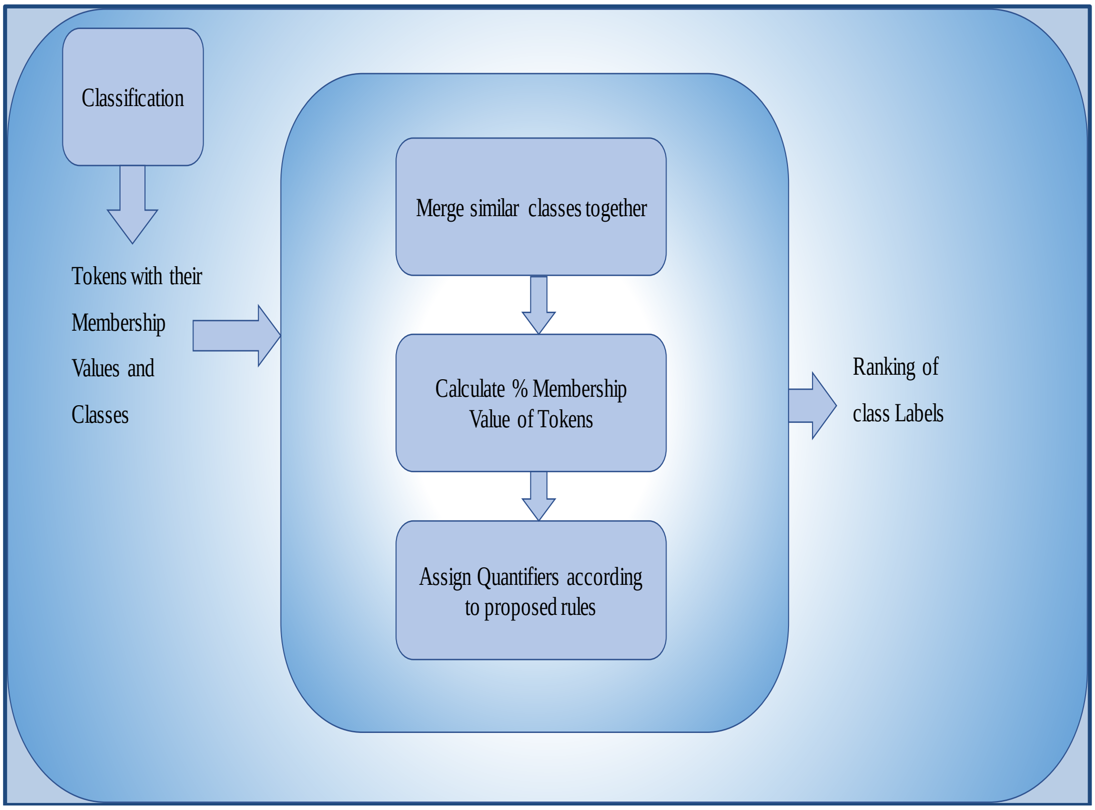

Classification
Tokens with their Membership Values and Classes
Merge similar classes together
Calculate % Membership Value of Tokens
Assign Quantifiers according to proposed rules
Ranking of class Labels

Figure 5.16: Flow Diagram of Ranking of Labels Phase

The algorithm of this phase is given in figure 5.17.

```
Algorithm:   Ranking Phase
Input:  <$T_i$, C >, where, $T_i$ is the title of the research article and
                    C is set of respective classes to which $T_i$ belongs
Output: Strict Ordering of Classes C1>C2>C3>C4……
  Begin
      1. Input the class labels (classes) along with their membership values from
         the previous phase.
      2. On the basis of membership values, assign the quantifiers.
      3. Display the class labels for each title of the research article in an order.
  End
```

Figure 5.17: Algorithm of Ranking Phase

5.2.7 Knowledge Discovery

Knowledge Discovery is the last phase of the proposed LS-KDT process. The knowledge obtained is an ordering of class labels of a text document. For example, take the case of a research article belonging to Computer Science domain/discipline. Under Computer Science discipline, there are various sub disciplines. The article may belong to a hybrid of sub disciplines. It is an example of multi-label categorization. Using the concept of quantifiers, we are able to calculate the membership degree of various class labels in a single research article. It helps in automated categorization of articles, thus helping the editors finding the best reviewers or experts to review them.

In this chapter, our proposed Lexical-Semantics based KDT process (LS-KDT) is discussed. The proposed process has seven phases namely Text Document Collection, Data Pre-processing, Lexical Analysis, Semantic Analysis, Classification, Ranking of labels and Knowledge Discovery. The working of all these phases is explained in detail in this chapter.

6 PERFORMANCE EVALUATION

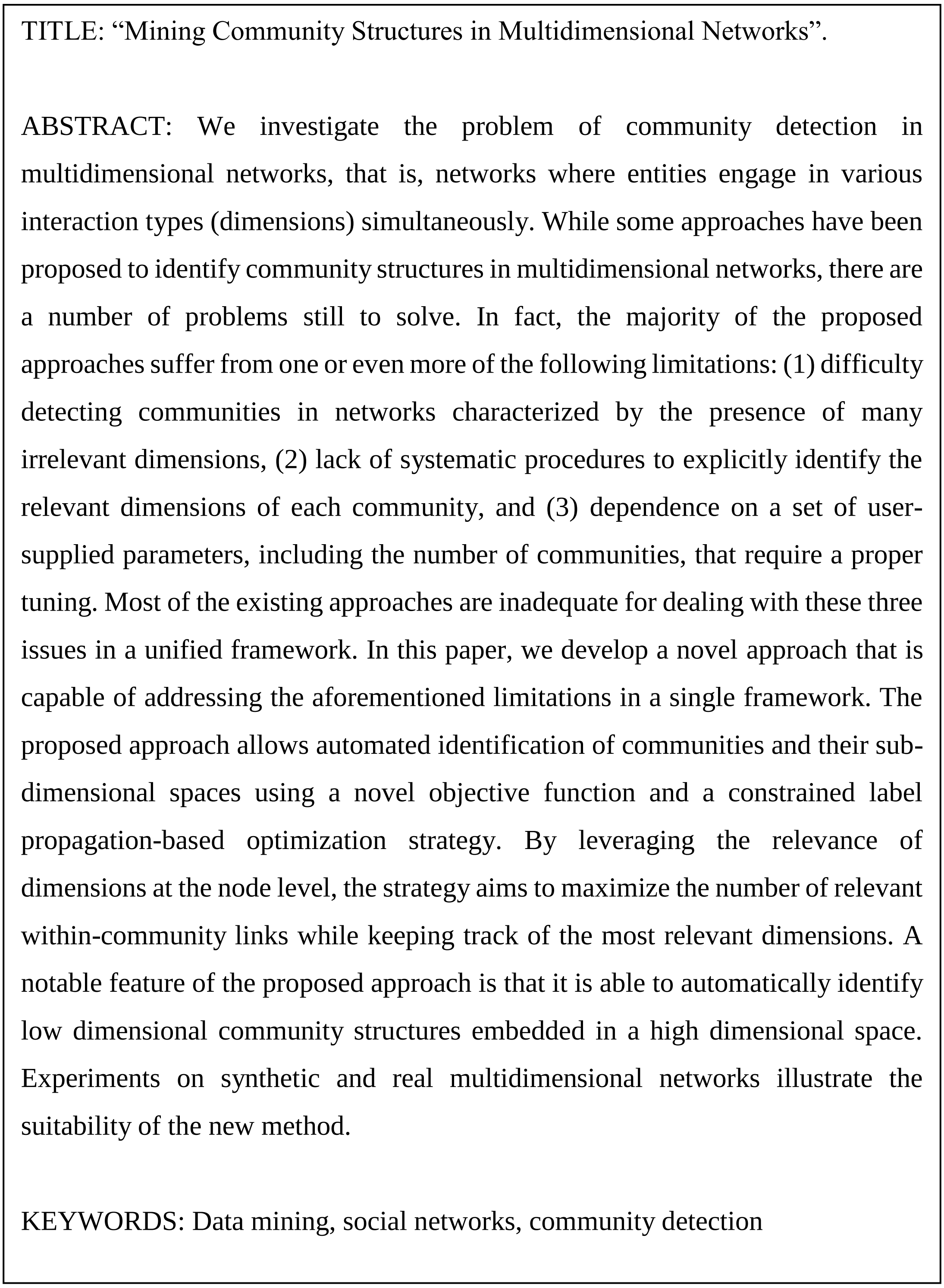
In this chapter, the performance of the proposed LS-KDT process is evaluated. It is calculated in three ways: firstly, the performance of proposed process is compared with the results of ACM digital library, secondly with IEEE Xplore results and thirdly with existing multi-label algorithms. The performance metrics like Recall, Precision, F-measure etc. are calculated. The proposed process has shown significantly good results.

In this chapter the performance of the proposed LS-KDT process is evaluated for various parameters. For implementation of the proposed LS-KDT process, we have used Java 1.7 (Schildt, H., 2007). All the experiments were performed on OS Type: Microsoft Windows XP Professional, Memory size: 3.5 GB, Processor: Intel[R] Core[TM]2 Duo CPU T9300 @ 2.50 GHz.

In the first section, results of different phases of proposed LS-KDT process are shown with the help of a sample research article. Then, performance of the proposed process is compared with two standard digital libraries. These are ACM digital library, IEEE Xplore and existing multi-label methods. And standard performance metrics like Recall, Precision and F-measure are calculated.

6.1 Results of Proposed LS-KDT Process on a sample research article

The figure 6.1 given shows a sample research article. The title, Abstract and keywords of the research article are taken.



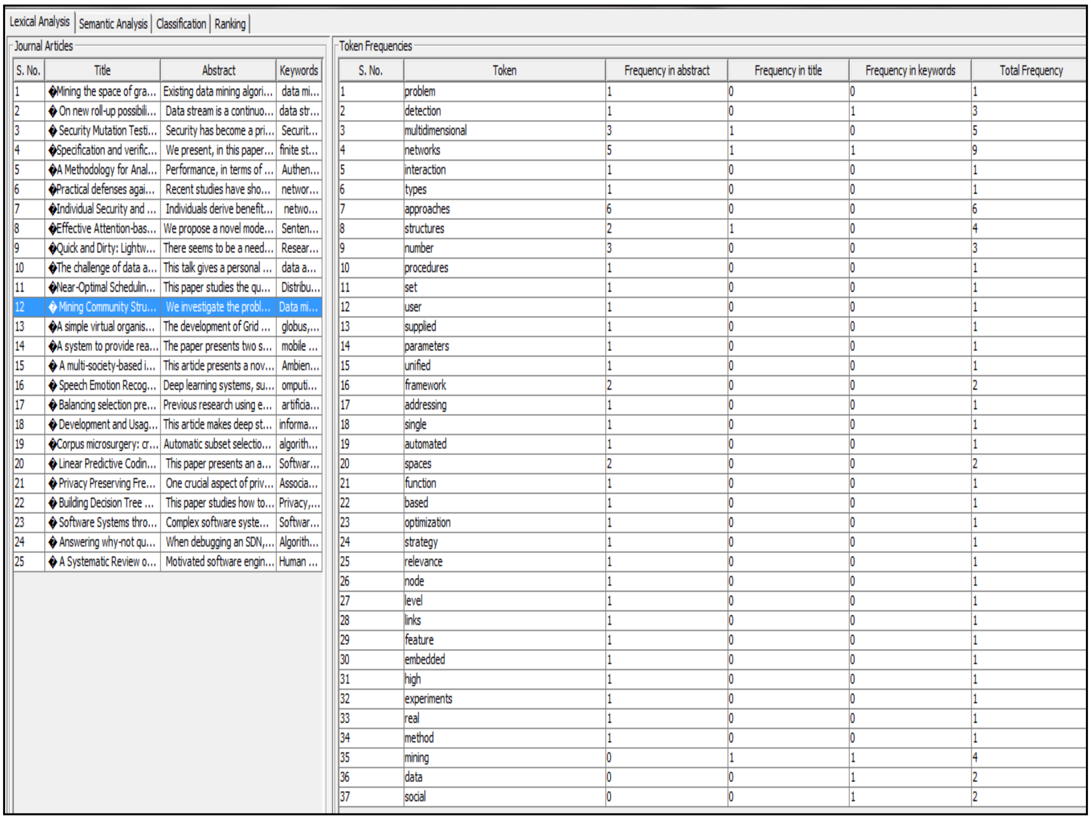
TITLE: “Mining Community Structures in Multidimensional Networks”.

ABSTRACT: We investigate the problem of community detection in multidimensional networks, that is, networks where entities engage in various interaction types (dimensions) simultaneously. While some approaches have been proposed to identify community structures in multidimensional networks, there are a number of problems still to solve. In fact, the majority of the proposed approaches suffer from one or even more of the following limitations: (1) difficulty detecting communities in networks characterized by the presence of many irrelevant dimensions, (2) lack of systematic procedures to explicitly identify the relevant dimensions of each community, and (3) dependence on a set of user-supplied parameters, including the number of communities, that require a proper tuning. Most of the existing approaches are inadequate for dealing with these three issues in a unified framework. In this paper, we develop a novel approach that is capable of addressing the aforementioned limitations in a single framework. The proposed approach allows automated identification of communities and their sub-dimensional spaces using a novel objective function and a constrained label propagation-based optimization strategy. By leveraging the relevance of dimensions at the node level, the strategy aims to maximize the number of relevant within-community links while keeping track of the most relevant dimensions. A notable feature of the proposed approach is that it is able to automatically identify low dimensional community structures embedded in a high dimensional space. Experiments on synthetic and real multidimensional networks illustrate the suitability of the new method.

KEYWORDS: Data mining, social networks, community detection

Figure 6.1: Sample Research Article

The figure 6.2 shows the output obtained from the third phase that is, Lexical Analysis. It takes as input the title, abstract and keywords of the article. And produces a list of tokens along with their frequency. There are total of 37 tokens in this research article. The lexical Analysis phase uses the standard ACM Computing Classification system for identifying the tokens.



Journal Articles

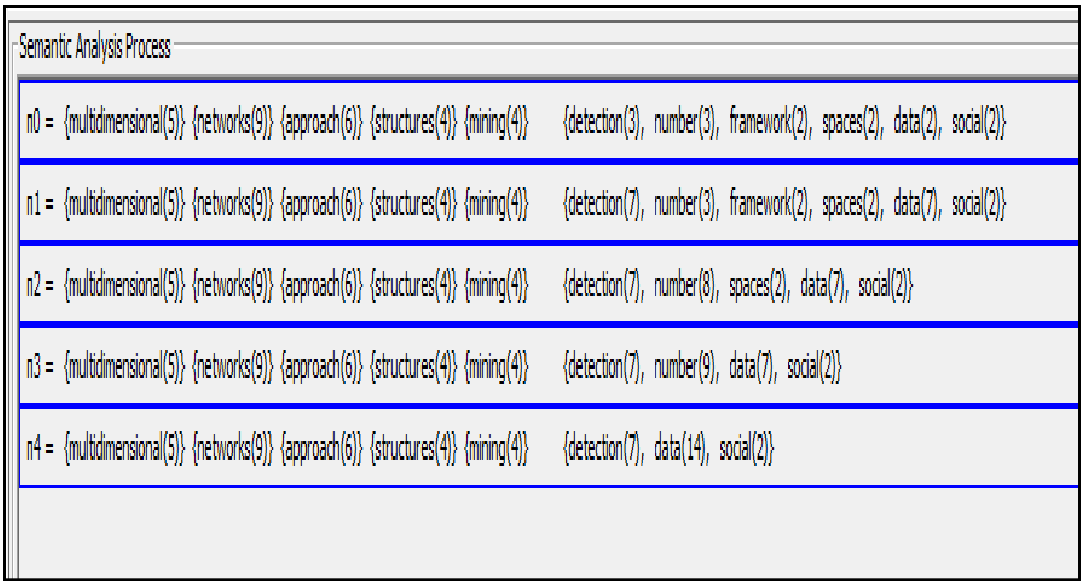
S. No.	Title	Abstract	Keywords
1	�Mining the space of gra...	Existing data mining algori...	data mi...
2	� On new roll-up possibili...	Data stream is a continuo...	data str...
3	� Security Mutation Testi...	Security has become a pri...	Securit...
4	�Specification and verific...	We present, in this paper...	finite st...
5	�A Methodology for Anal...	Performance, in terms of ...	Authen...
6	�Practical defenses agai...	Recent studies have sho...	networ...
7	�Individual Security and ...	Individuals derive benefit...	netwo...
8	�Effective Attention-bas...	We propose a novel mode...	Senten...
9	�Quick and Dirty: Lightw...	There seems to be a need...	Resear...
10	�The challenge of data a...	This talk gives a personal ...	data a...
11	�Near-Optimal Schedulin...	This paper studies the qu...	Distribu...
12	� Mining Community Stru...	We investigate the probl...	Data mi...
13	�A simple virtual organis...	The development of Grid ...	globus,...
14	�A system to provide rea...	The paper presents two s...	mobile ...
15	� A multi-society-based i...	This article presents a nov...	Ambien...
16	� Speech Emotion Recog...	Deep learning systems, su...	omputi...
17	� Balancing selection pre...	Previous research using e...	artificia...
18	� Development and Usag...	This article makes deep st...	informa...
19	�Corpus microsurgery: cr...	Automatic subset selectio...	algorith...
20	� Linear Predictive Codin...	This paper presents an a...	Softwar...
21	� Privacy Preserving Fre...	One crucial aspect of priv...	Associa...
22	� Building Decision Tree ...	This paper studies how to...	Privacy,...
23	� Software Systems thro...	Complex software syste...	Softwar...
24	� Answering why-not qu...	When debugging an SDN,...	Algorith...
25	� A Systematic Review o...	Motivated software engin...	Human ...

Token Frequencies

S. No.	Token	Frequency in abstract	Frequency in title	Frequency in keywords	Total Frequency
1	problem	1	0	0	1
2	detection	1	0	1	3
3	multidimensional	3	1	0	5
4	networks	5	1	1	9
5	interaction	1	0	0	1
6	types	1	0	0	1
7	approaches	6	0	0	6
8	structures	2	1	0	4
9	number	3	0	0	3
10	procedures	1	0	0	1
11	set	1	0	0	1
12	user	1	0	0	1
13	supplied	1	0	0	1
14	parameters	1	0	0	1
15	unified	1	0	0	1
16	framework	2	0	0	2
17	addressing	1	0	0	1
18	single	1	0	0	1
19	automated	1	0	0	1
20	spaces	2	0	0	2
21	function	1	0	0	1
22	based	1	0	0	1
23	optimization	1	0	0	1
24	strategy	1	0	0	1
25	relevance	1	0	0	1
26	node	1	0	0	1
27	level	1	0	0	1
28	links	1	0	0	1
29	feature	1	0	0	1
30	embedded	1	0	0	1
31	high	1	0	0	1
32	experiments	1	0	0	1
33	real	1	0	0	1
34	method	1	0	0	1
35	mining	0	1	1	4
36	data	0	0	1	2
37	social	0	0	1	2

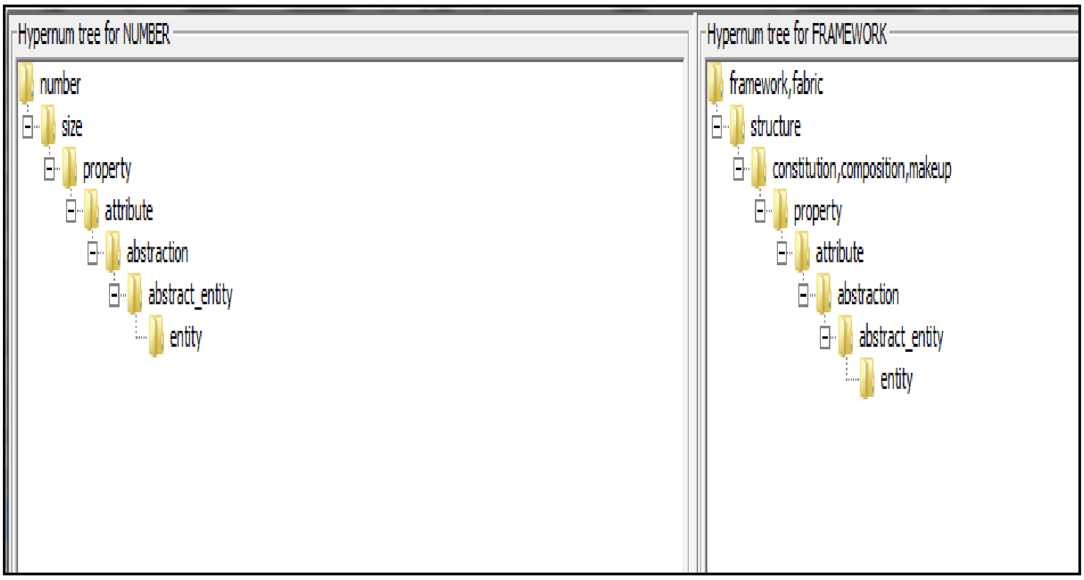
Figure 6.2: Output of Lexical Analysis Phase

The next figures- 6.3, 6.4 and 6.5 show the output of the next phase Semantic Analysis. The figure 6.3 shows the partition of the tokens made. The terms n0, n1, n2, n3 and n4 show the partitions made in each step. The figure 6.4 shows the hypernym trees for two tokens- Number and framework. These are drawn with the help of WordNet. And figure 6.5 shows the final output of Semantic Analysis phase. It gives the reduced number of tokens. The tokens are reduced to a count of 8 in this phase.



```
Semantic Analysis Process
n0 = {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(3), number(3), framework(2), spaces(2), data(2), social(2)}
n1 = {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(7), number(3), framework(2), spaces(2), data(7), social(2)}
n2 = {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(7), number(8), spaces(2), data(7), social(2)}
n3 = {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(7), number(9), data(7), social(2)}
n4 = {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(7), data(14), social(2)}
```

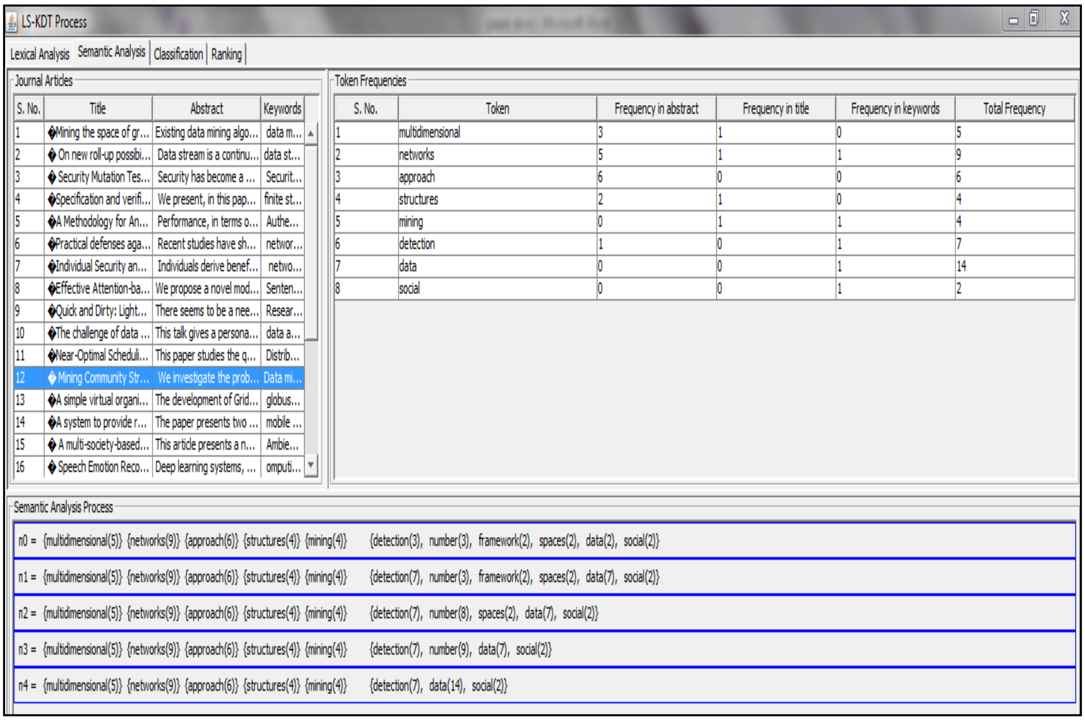
Figure 6.3: Output of Semantic Analysis Phase showing partitions



```
Hypernum tree for NUMBER
number
  size
    property
      attribute
        abstraction
          abstract_entity
            entity
```

```
Hypernum tree for FRAMEWORK
framework,fabric
  structure
    constitution,composition,makeup
      property
        attribute
          abstraction
            abstract_entity
              entity
```

Figure 6.4: Output of Semantic Analysis Phase showing hypernym trees between tokens NUMBER and FRAMEWORK



Journal Articles

S. No.	Title	Abstract	Keywords
1	�Mining the space of gr...	Existing data mining algo...	data m...
2	� On new roll-up possibi...	Data stream is a continu...	data st...
3	� Security Mutation Tes...	Security has become a ...	Securit...
4	�Specification and verifi...	We present, in this pap...	finite st...
5	�A Methodology for An...	Performance, in terms o...	Authe...
6	�Practical defenses aga...	Recent studies have sh...	networ...
7	�Individual Security an...	Individuals derive benef...	netwo...
8	�Effective Attention-ba...	We propose a novel mod...	Senten...
9	�Quick and Dirty: Light...	There seems to be a nee...	Resear...
10	�The challenge of data ...	This talk gives a persona...	data a...
11	�Near-Optimal Scheduli...	This paper studies the q...	Distrib...
12	� Mining Community Str...	We investigate the prob...	Data mi...
13	�A simple virtual organi...	The development of Grid...	globus...
14	�A system to provide r...	The paper presents two ...	mobile ...
15	� A multi-society-based...	This article presents a n...	Ambie...
16	� Speech Emotion Reco...	Deep learning systems, ...	omputi...

Token Frequencies

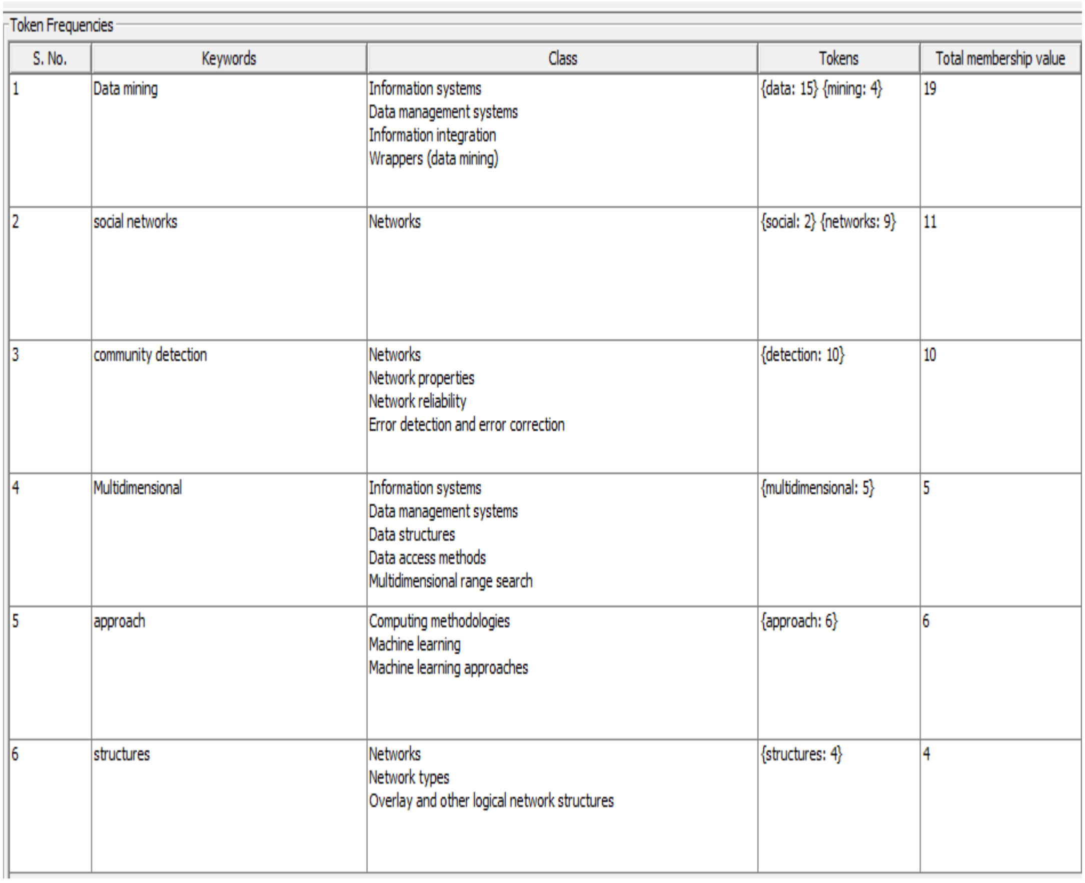
S. No.	Token	Frequency in abstract	Frequency in title	Frequency in keywords	Total Frequency
1	multidimensional	3	1	0	5
2	networks	5	1	1	9
3	approach	6	0	0	6
4	structures	2	1	0	4
5	mining	0	1	1	4
6	detection	1	0	1	7
7	data	0	0	1	14
8	social	0	0	1	2

Semantic Analysis Process

```
n0 =  {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(3),  number(3),  framework(2),  spaces(2),  data(2),  social(2)}
n1 =  {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(7),  number(3),  framework(2),  spaces(2),  data(7),  social(2)}
n2 =  {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(7),  number(8),  spaces(2),  data(7),  social(2)}
n3 =  {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(7),  number(9),  data(7),  social(2)}
n4 =  {multidimensional(5)} {networks(9)} {approach(6)} {structures(4)} {mining(4)}     {detection(7),  data(14),  social(2)}
```

Figure 6.5: Final Output of Semantic Analysis Phase

This list of reduced tokens with their membership values is sent as input to the next phase that is, Classification phase. The figure 6.6 shows the output of Classification phase. In this phase, two functions are performed. First, the single tokens are merged with the help of the given keywords of the research article. And second, the classes of the tokens are obtained from the standard ACM Computing Classification system. The broad classes along with the hierarchy of sub classes are also displayed. So, the output of this phase is a list of classes to which the research article belongs along with the membership values.

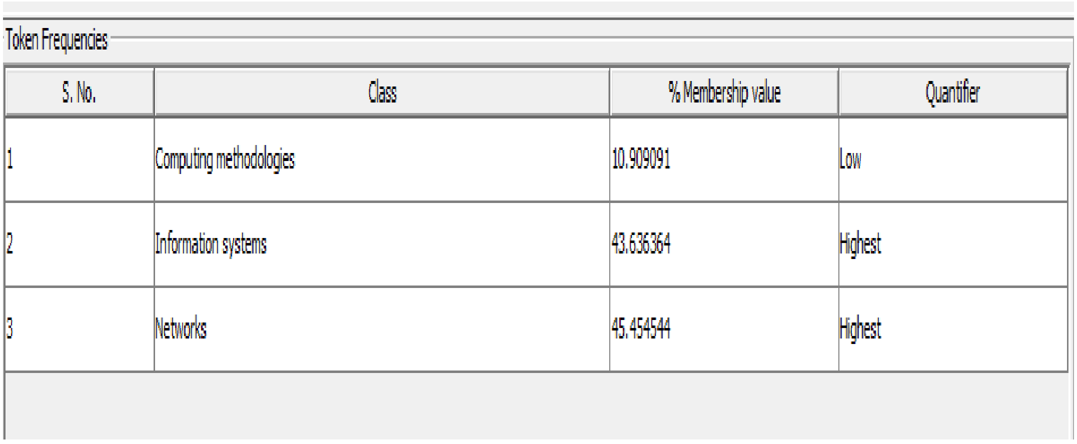


Token Frequencies

S. No.	Keywords	Class	Tokens	Total membership value
1	Data mining	Information systems Data management systems Information integration Wrappers (data mining)	{data: 15} {mining: 4}	19
2	social networks	Networks	{social: 2} {networks: 9}	11
3	community detection	Networks Network properties Network reliability Error detection and error correction	{detection: 10}	10
4	Multidimensional	Information systems Data management systems Data structures Data access methods Multidimensional range search	{multidimensional: 5}	5
5	approach	Computing methodologies Machine learning Machine learning approaches	{approach: 6}	6
6	structures	Networks Network types Overlay and other logical network structures	{structures: 4}	4

Figure 6.6: Output of Classification Phase

The next figure 6.7 shows the output of Ranking of labels phase. In this phase, firstly, similar classes obtained from the previous phase, are merged together. Their membership values are also added. And, then percentage membership value of each class is computed. Secondly, quantifiers are assigned to each class depending on its membership value. So, the output of this phase is a list of classes (along with the percentage membership values) to which a research article belongs and quantifiers.



Token Frequencies

S. No.	Class	% Membership value	Quantifier
1	Computing methodologies	10.909091	Low
2	Information systems	43.636364	Highest
3	Networks	45.454544	Highest

Figure 6.7: Output of Ranking Phase

6.2 Performance Comparison of Proposed LS-KDT Process

The results of the proposed LS-KDT process are compared with the results of two digital libraries: ACM digital library and IEEE Xplore for performance comparison.

6.2.1 With ACM digital library results

In this section, we show the comparison with ACM Digital library.

We have used two datasets:

- Dataset of research articles belonging to computer science domain
- Dataset of research articles belonging to medical domain

And standard performance metrics like Recall, Precision and F-measure are calculated.

The research articles are randomly selected from ACM digital library. In total, we have considered 250 articles of computer science domain. Following table 6.1 shows the details of the dataset.

Table 6.1: Details of dataset of computer science research articles

S. No	Category	Number of Articles
1.	Computer System Organization	20
2.	Networks	30
3.	Software and Engineering	20
4.	Theory of Computation	20
5.	Mathematics of Computing	20
6.	Information Systems	30
7.	Security and Privacy	30
8.	Human Centered Computing	20
9.	Computing Methodologies	20
10.	Applied Computing	20
11.	Social and Professional Topics	20

The table 6.2 below shows the details of dataset of research articles of medical domain. The articles are randomly selected from ACM digital library. We have taken a total of 275 articles.

Table 6.2: Details of dataset of medical domain articles

S. No	Category	Number of Articles
1.	Computer System Organization	20
2.	Networks	20
3.	Software and Engineering	20
4.	Theory of Computation	20
5.	Mathematics of Computing	25
6.	Information Systems	20
7.	Security and Privacy	30
8.	Human Centered Computing	30
9.	Computing Methodologies	30
10.	Applied Computing	30
11.	Social and Professional Topics	30

ACM digital library uses Computing Classification System (CCS) tool for the categorization of research articles. The CCS tool displays the result in a hierarchical manner i.e. broad category of an article followed by levels of sub categories to which an article belongs. Our proposed LS-KDT process shows the levels of broad categories as well as sub categories to which an article belongs along with their membership values.

Recall is the ratio of related retrieved documents to relevant documents. Precision is the ratio of relevant retrieved documents to retrieved documents. F-measure is the harmonic mean of both Precision and Recall.

In our proposed process, we have obtained levels of broad categories of a research article. So, Precision, Recall and F-measure can be calculated as follows:

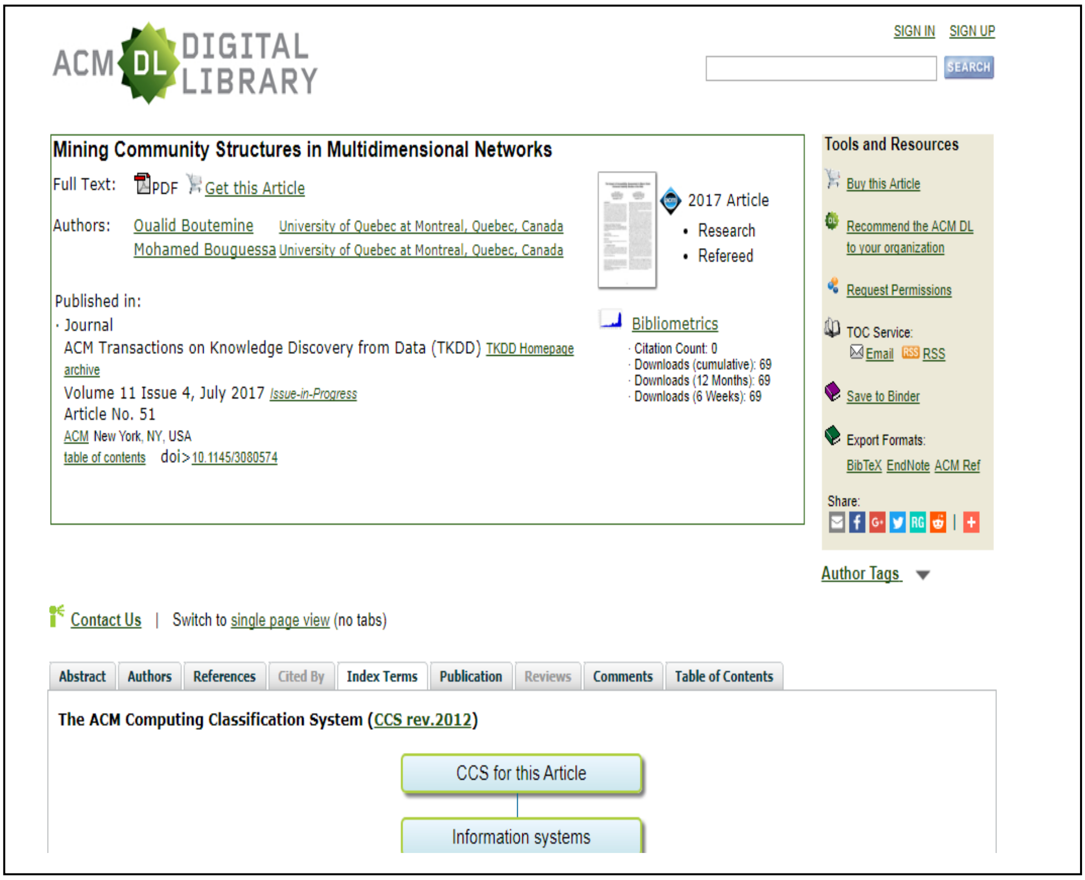
$$Recall = \frac{\sum RelevantLevels}{\sum AllTheLevelswithintheCategory} \tag{6.1}$$

$$Precision = \frac{\sum Relevant\ Levels}{\sum Retrieved\ Levels} \tag{6.2}$$

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.3}$$

First, we have calculated the performance metrics of the categories one by one. Then, by taking the average of all the categories, we have calculated micro precision, micro recall and micro F-measure. The values of micro precision, micro recall and micro F-measure have shown increase by our proposed LS-KDT process.

Figure 6.8 shows the sample research article of ACM digital library.



Mining Community Structures in Multidimensional Networks

Full Text: PDF Get this Article

Authors: Oualid Boutemine University of Quebec at Montreal, Quebec, Canada
Mohamed Bouguessa University of Quebec at Montreal, Quebec, Canada

2017 Article
- Research
- Refereed

Published in:
· Journal
ACM Transactions on Knowledge Discovery from Data (TKDD) TKDD Homepage archive
Volume 11 Issue 4, July 2017 Issue-in-Progress
Article No. 51
ACM New York, NY, USA
table of contents doi>10.1145/3080574

Bibliometrics
· Citation Count: 0
· Downloads (cumulative): 69
· Downloads (12 Months): 69
· Downloads (6 Weeks): 69

Tools and Resources
Buy this Article
Recommend the ACM DL to your organization
Request Permissions
TOC Service:
Email RSS
Save to Binder
Export Formats:
BibTeX EndNote ACM Ref
Share:

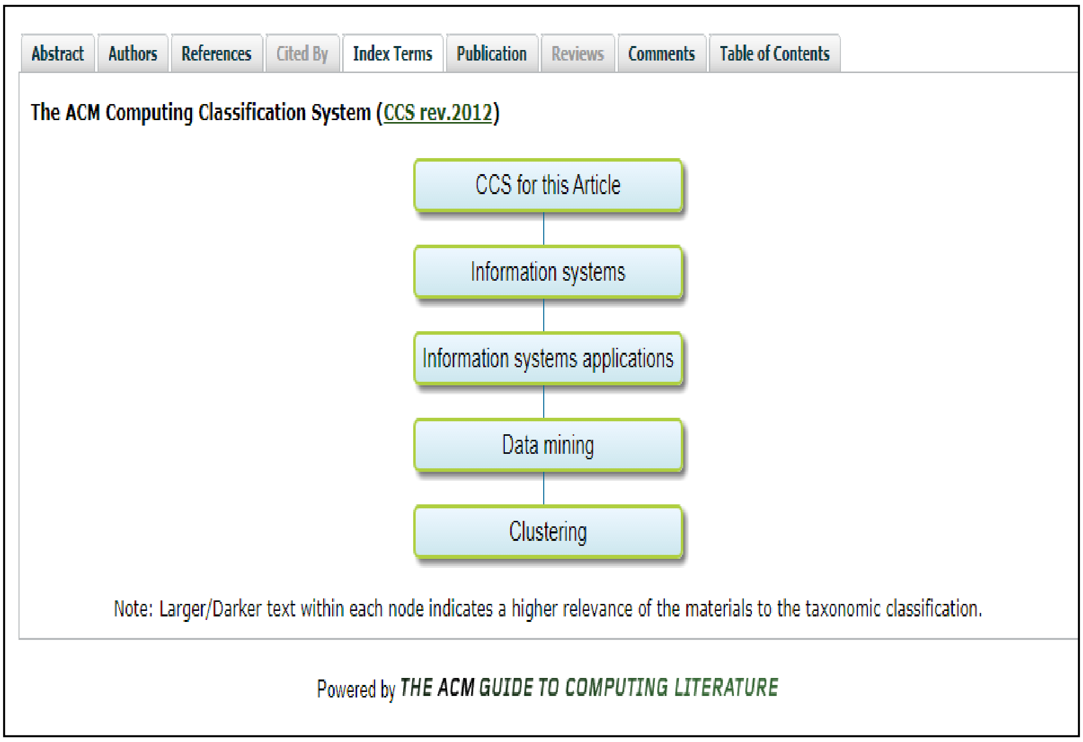
Author Tags

Contact Us | Switch to single page view (no tabs)

The ACM Computing Classification System (CCS rev.2012)

Figure 6.8: Sample Research article in ACM digital library

Figure 6.9 shows the categorization of the sample research article by ACM CCS tool. So, as per the existing ACM CCS tool the research article is categorized in one broad category: Information Systems.



The ACM Computing Classification System (CCS rev.2012)

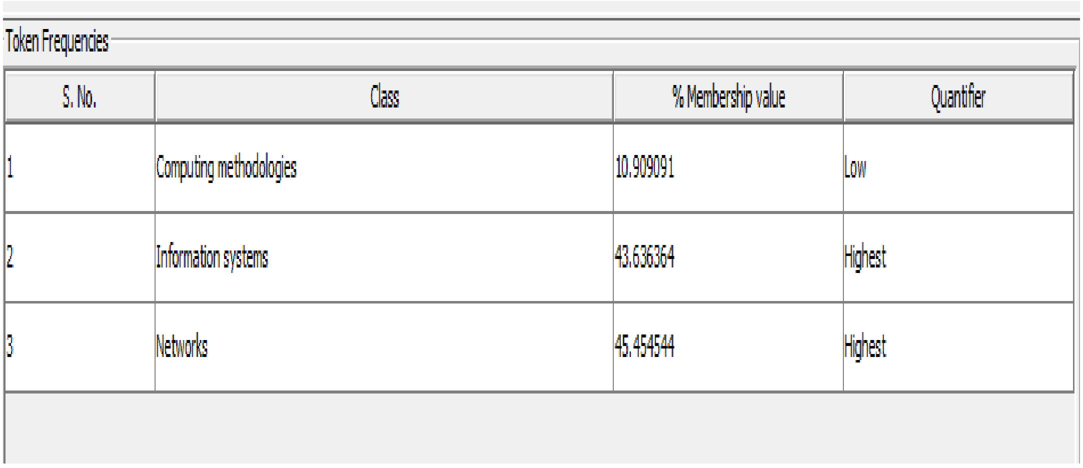
CCS for this Article
Information systems
Information systems applications
Data mining
Clustering

Note: Larger/Darker text within each node indicates a higher relevance of the materials to the taxonomic classification.

Powered by THE ACM GUIDE TO COMPUTING LITERATURE

Figure 6.9: Sample Research article in ACM CCS tool

Figure 6.10 shows the output of sample research article by proposed LS-KDT process.



Token Frequencies

S. No.	Class	% Membership value	Quantifier
1	Computing methodologies	10.909091	Low
2	Information systems	43.636364	Highest
3	Networks	45.454544	Highest

Figure 6.10: Output of Sample Research article by proposed LS-KDT process

As clearly shown by figure 6.10 that proposed LS-KDT process displays three classes for the sample research article along with the membership values.

The performance metrics- Recall and Precision for the same research article are calculated using equations 6.1 and 6.2.

ACM CCS Tool

$$\text{Recall} = \frac{4+3+2+1}{5+4+3+2+1} * 100$$

$$= 67\%$$

$$\text{Precision} = \frac{4+3+2+1}{4+3+2+1} * 100$$

$$= 100\%$$

Proposed LS-KDT Process

$$\text{Recall} = \frac{5+4+3+2+1}{5+4+3+2+!} * 100$$

$$= 100\%$$

$$\text{Precision} = \frac{5+4+3+2+1}{5+4+3+2+1} * 100$$

$$= 100\%$$

The table 6.3 shows the comparison of values of Precision, Recall and F-measure for dataset of research articles of computer science domain for both ACM CCS tool and the proposed LS-KDT process. Around 20 – 30 research papers from each category are used for comparison. The sample research article shown belongs to broad category: Information Systems. The Precision, Recall and F–measure for the broad category Information Systems has been increased to 87, 88 and 88 respectively by the proposed LS-KDT process as compared to 84, 51 and 63 respectively with the existing ACM CCS tool.

Table 6.3: Comparison of Results for Computer Science domain articles

		ACM CCS tool			**Proposed LS-KDT Process**		
S. No	**Class/Category**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
1	Computer System Organization	60	60	60	90	80	85
2	Networks	100	35	51	100	54	70
3	Software and Engineering	92	75	82	100	100	100
4	Theory of Computation	100	100	100	90	90	90
5	Mathematics of Computing	100	100	100	100	100	100
6	**Information Systems**	**84**	**51**	**63**	**87**	**88**	**88**
7	Security and Privacy	100	40	57	100	67	80
8	Human Centered Computing	90	60	72	95	75	84
9	Computing Methodologies	100	70	73	90	80	85
10	Applied Computing	94	55	69	95	67	79
11	Social and Professional Topics	85	50	63	90	55	68
	Average Values	Micro Precision= 91	Micro Recall = 63	Micro F-Measure = 72	Micro Precision= 94	Micro Recall = 78	Micro F-Measure = 85

From the table 6.3, it is clear that Micro Precision, Micro Recall and Micro F-measure for the proposed LS-KDT process is increased to 94, 78 and 85 respectively with respect to 91, 63 and 72 respectively for the existing ACM CCS tool.

Figures 6.11-6.13 show comparison of Precision, Recall and F-measure values for the proposed LS-KDT process with the existing ACM CCS tool.

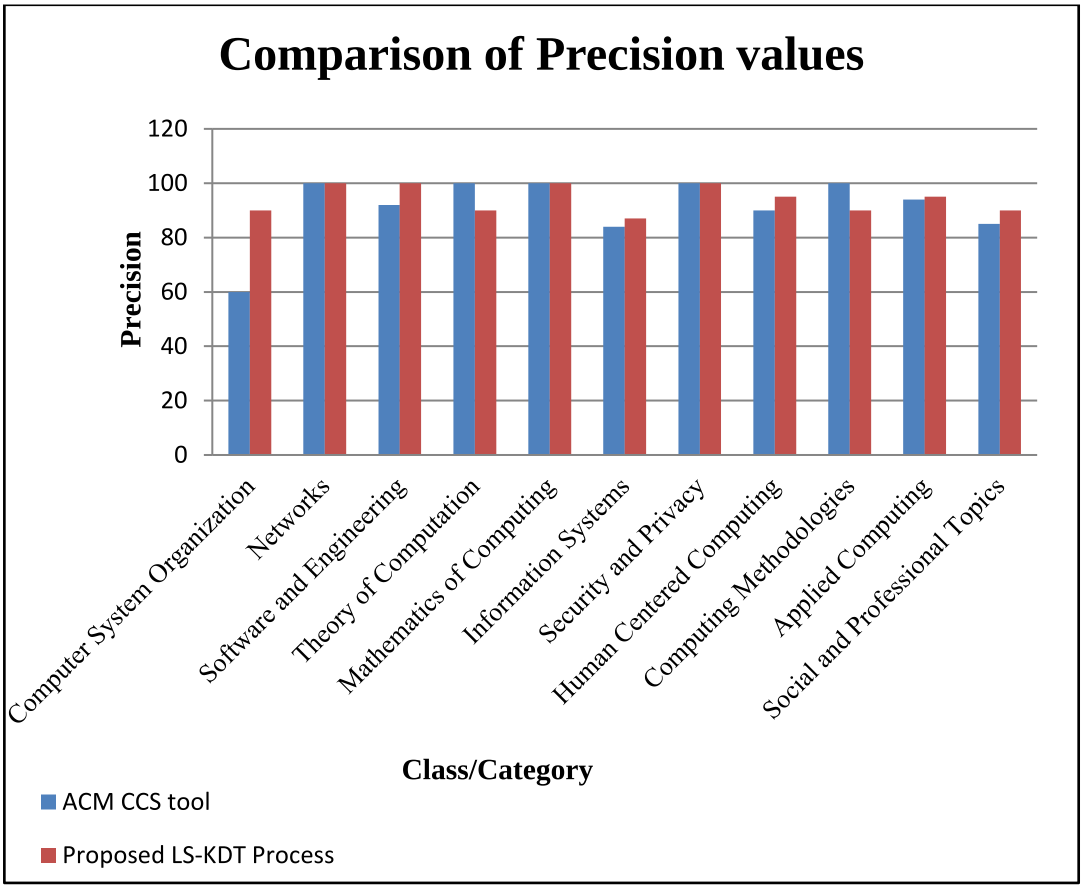


Figure 6.11: Comparison of Precision values for research articles of computer science domain

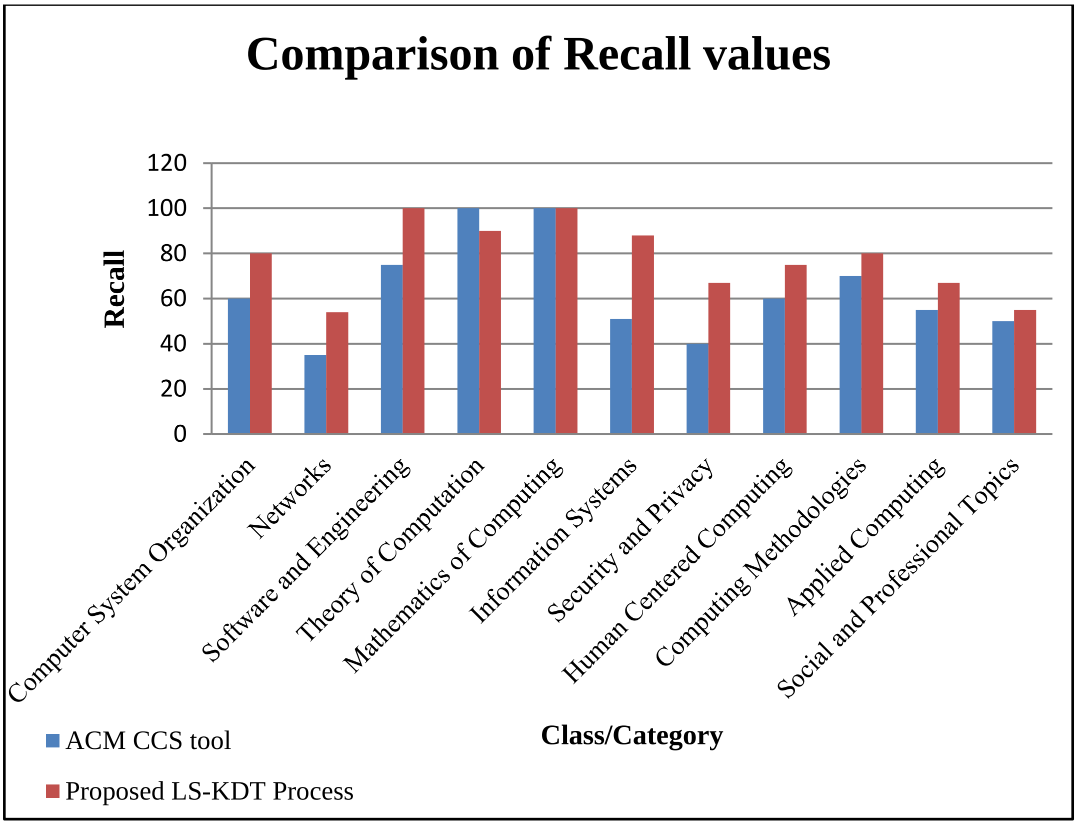


Figure 6.12: Comparison of Recall values for research articles of computer science domain

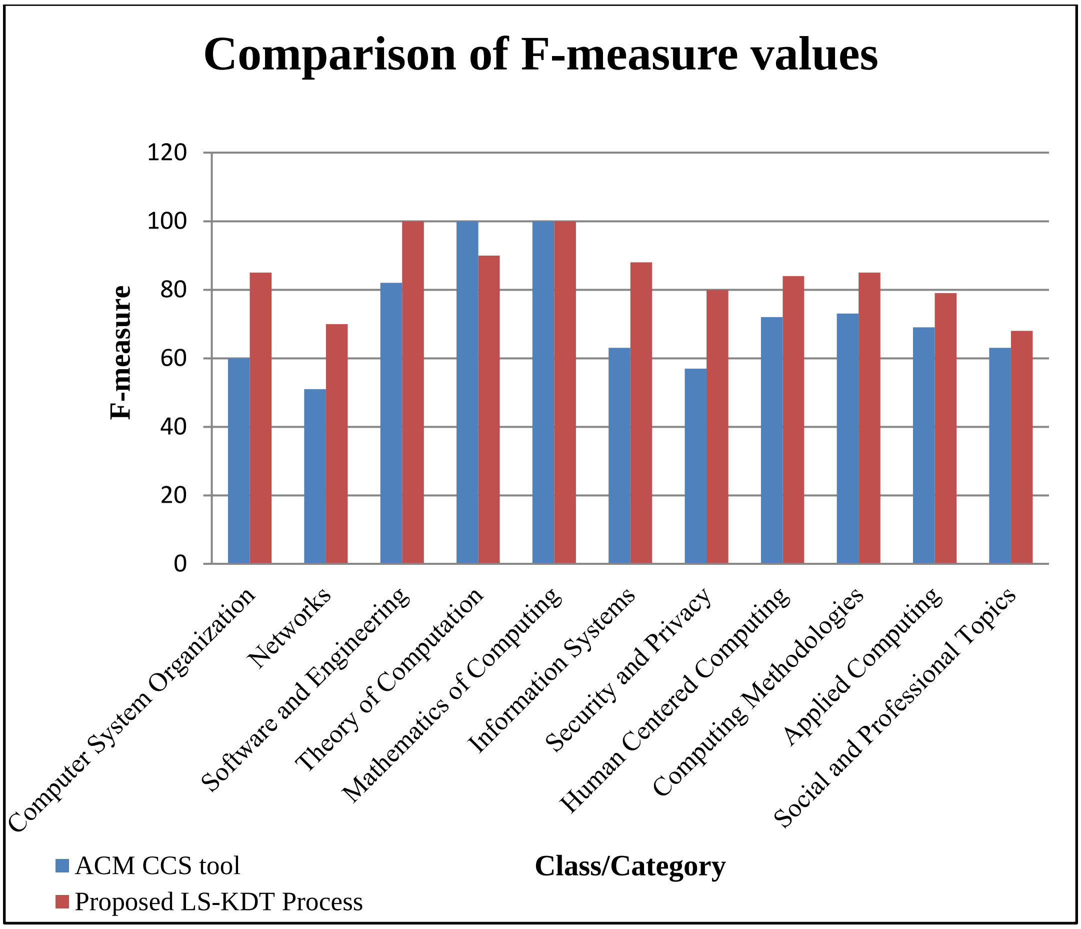


Figure 6.13: Comparison of F-measure values for research articles of computer science domain

The table 6.4 shows the comparison of values of Recall, Precision and F- measure for the dataset of research articles of medical domain for both ACM CCS tool and the proposed LS-KDT process.

Table 6.4: Comparison of Results for medical domain articles

		ACM CCS tool			Proposed LS-KDT Process		
S. No	**Class/Category**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
1	Computer System Organization	70	60	65	80	80	80
2	Networks	100	45	68	100	54	70
3	Software and Engineering	92	75	83	90	70	79
4	Theory of Computation	100	100	100	90	90	90
5	Mathematics of Computing	95	90	92	100	100	100
6	Information Systems	90	50	64	90	50	64
7	Security and Privacy	100	40	57	100	78	88
8	Human Centered Computing	90	60	72	95	75	84
9	Computing Methodologies	100	90	95	100	90	95
10	Applied Computing	95	65	77	95	67	79
11	Social and Professional Topics	85	50	63	90	55	68
	Average Values	Micro Precision =92	Micro Recall =66	Micro F-Measure =76	Micro Precision =94	Micro Recall =76	Micro F-Measure =82

The figures 6.14, 6.15 and 6.16 show the comparison of values of Precision, Recall and F-measure respectively for medical domain articles.

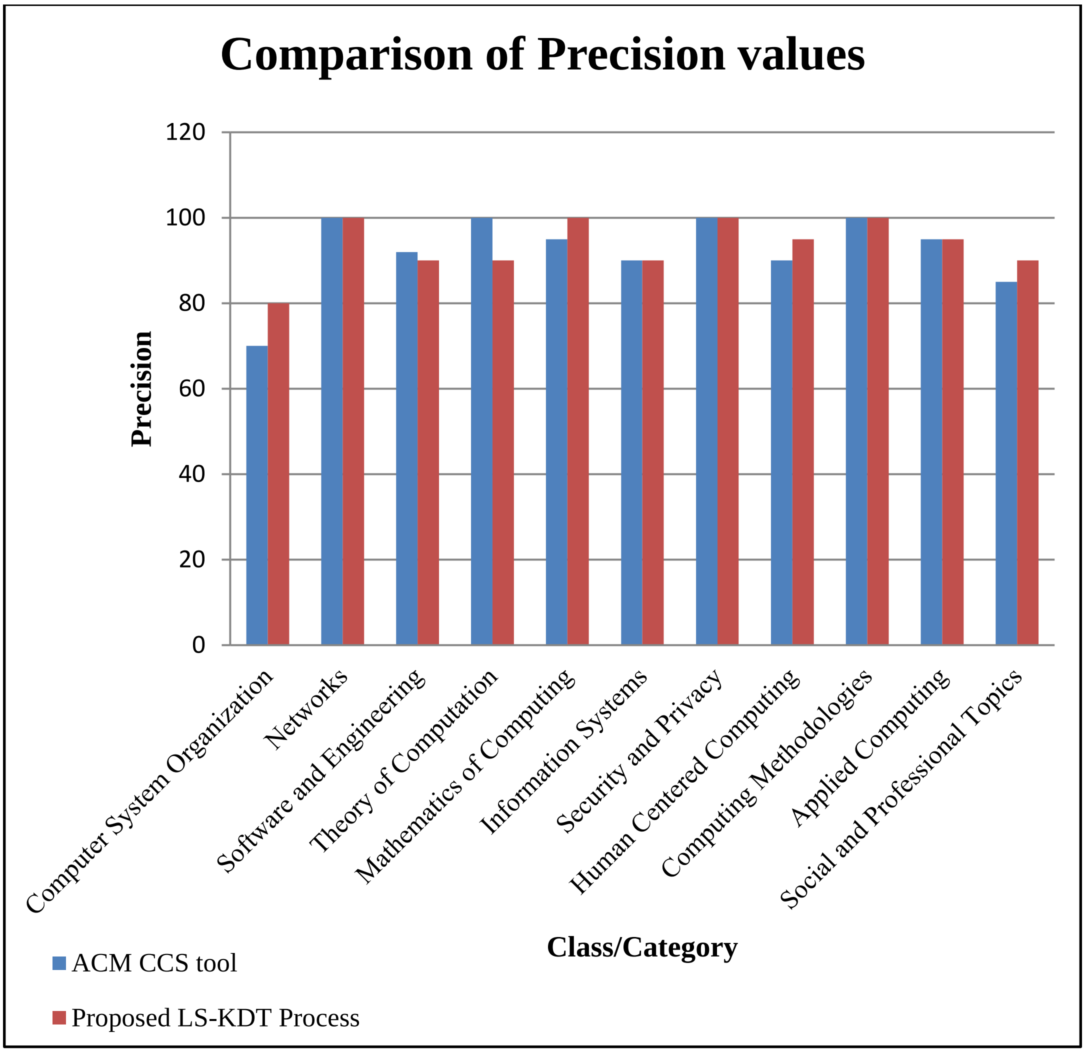


Figure 6.14: Comparison of Precision values for research articles of medical domain

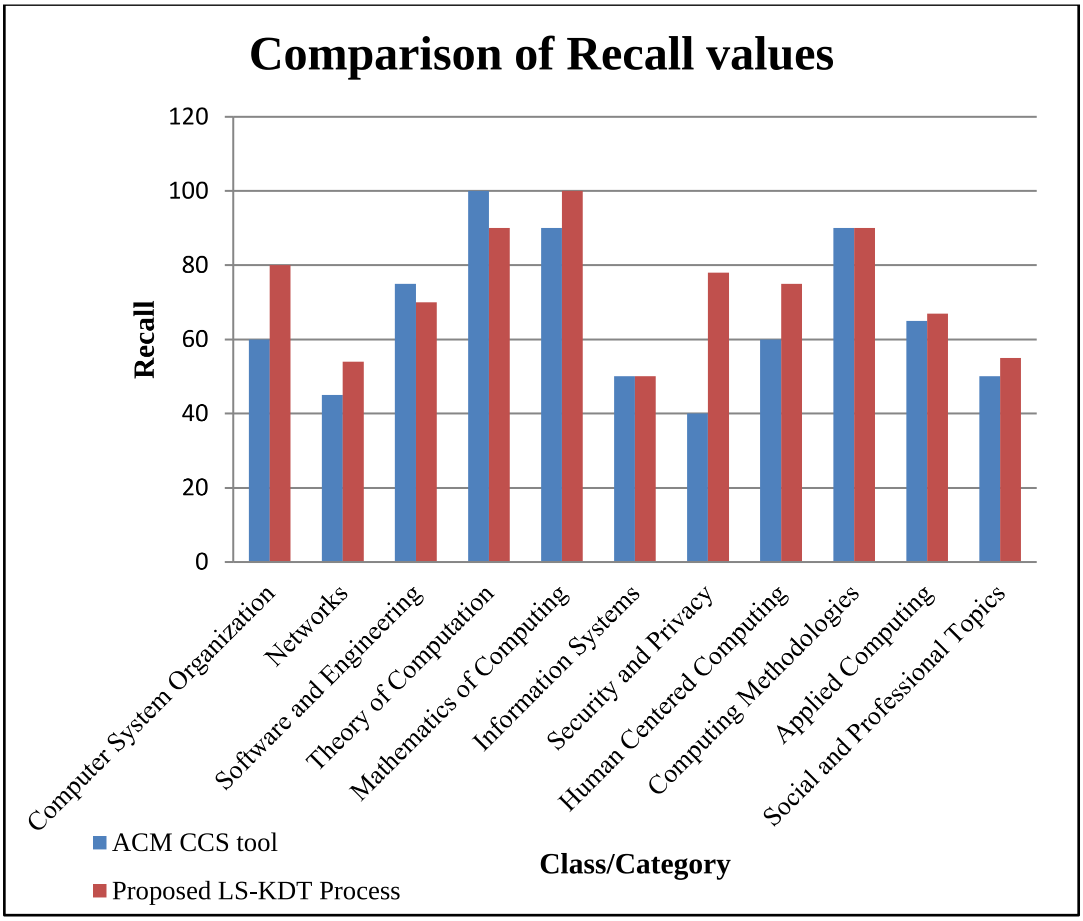


Figure 6.15:Comparison of Recall values for research articles of medical domain

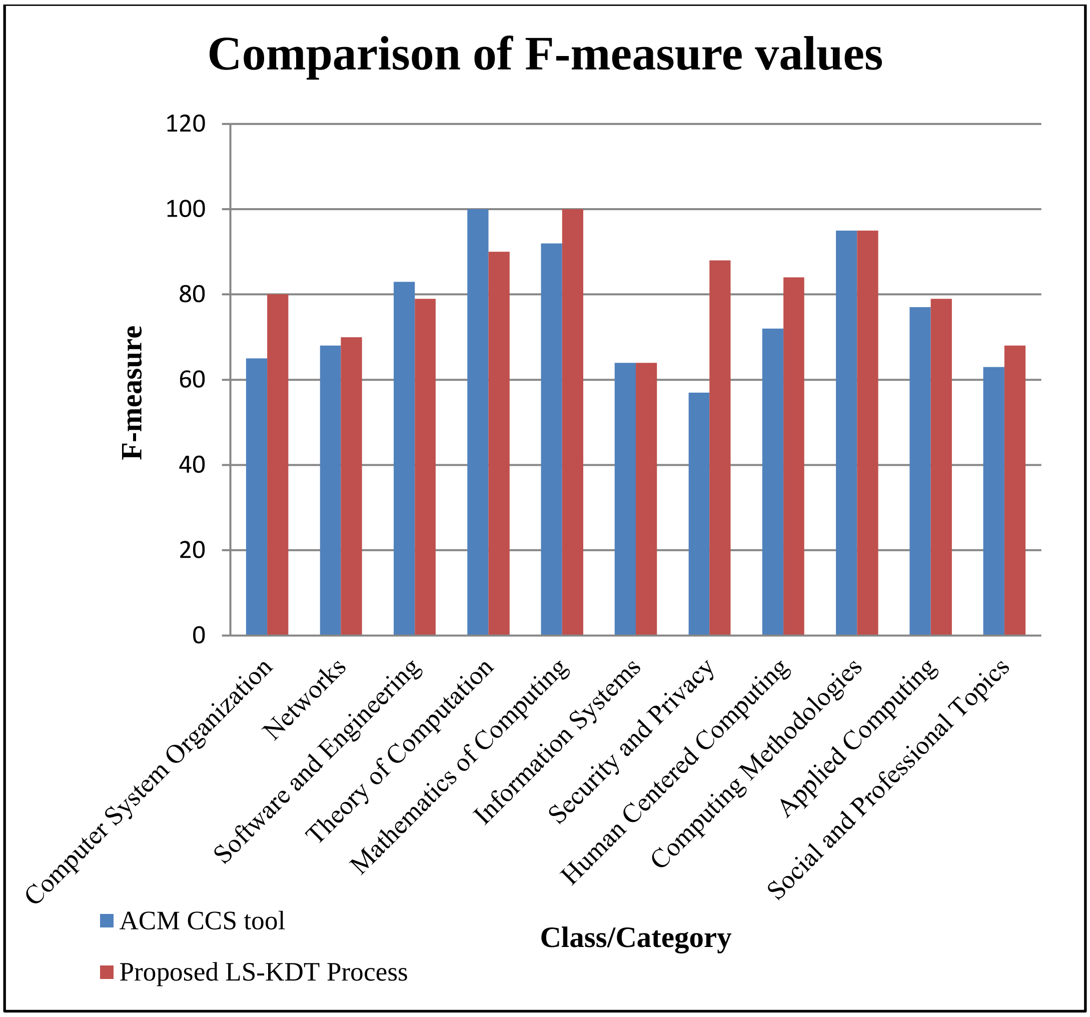


Figure 6.16: Comparison of F-measure values for research articles of medical domain

6.2.2 With IEEE Xplore Results

In this section, we give the performance comparison of the proposed LS-KDT process with IEEE Xplore results. Figure 6.17 shows a sample research article in IEEE Xplore digital library. It shows the title, abstract and keywords of the article.

TITLE: “MFZ-KNN - A modified fuzzy based K nearest neighbor algorithm”.

ABSTRACT: KNN is amongst the simplest top ten classification algorithm of data mining. Being effective and efficient it has some drawbacks which cannot be overlooked. Moreover, real world data is fuzzy in nature. To overcome this drawback fuzzy KNN was introduced which was based on fuzzy membership. But, it had large time complexity as the membership is calculated at the classification period. To improve this, we have proposed a modified fuzzy based KNN algorithm MFZ-KNN whereby fuzzy clusters are obtained at pre-processing step and the membership of the training data set is computed in reference with the centroid of the clusters. This reduces the complexity of time remarkably. We have implemented the algorithm in MatLAB and NetBeans IDE using standard UCI data set-Wine. The results prove that it is better than both conventional KNN and fuzzy KNN in terms of accuracy and time.

KEYWORDS: Fuzzy C-means (FCM), KNN (K-Nearest Neighbor), Fuzzy KNN (FKNN)

Figure 6.17: A sample research article

IEEE Xplore digital library uses the standard IEEE Taxonomy for the classification purpose. The details of the taxonomy are discussed in section below.

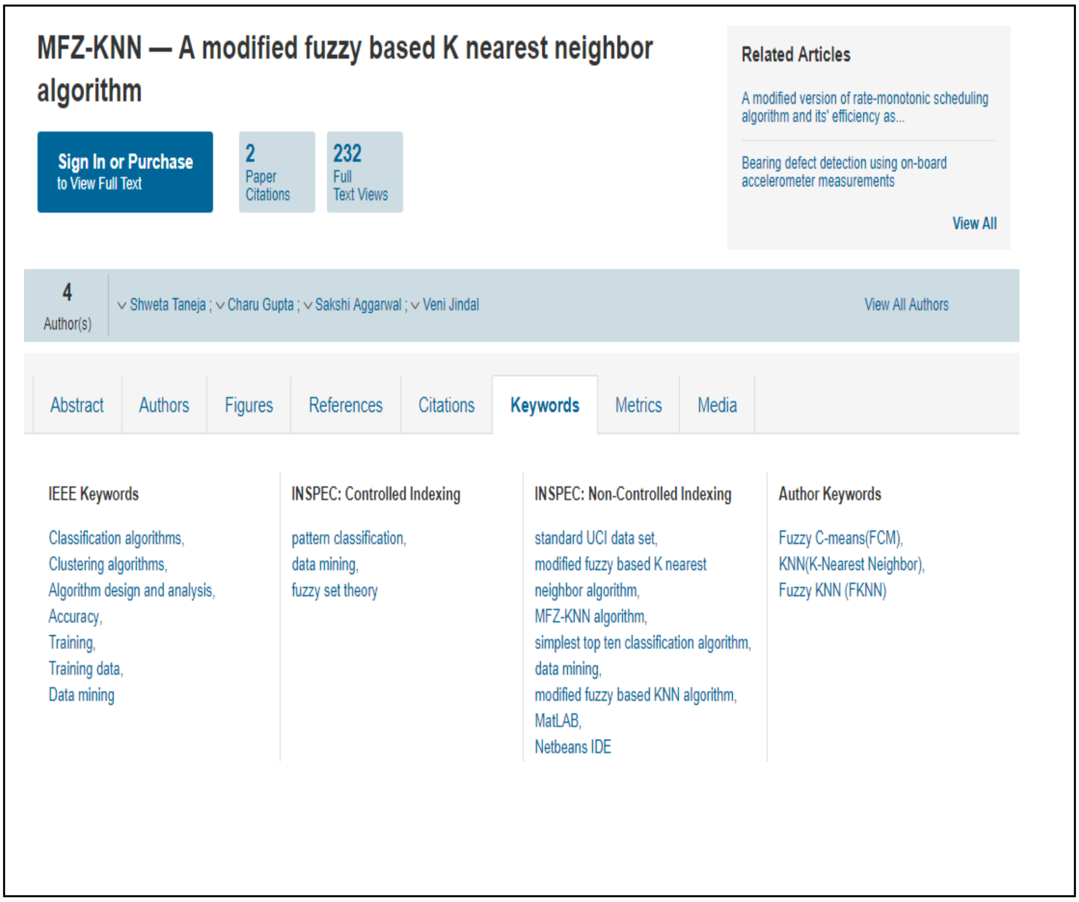
6.2.2.1 IEEE Taxonomy

IEEE Xplore uses the standard IEEE Taxonomy for the indexing of research articles. In our work, we have used 2017 IEEE Taxonomy. The taxonomy is also hierarchical in structure. That is broad categories/classes and further sub categories under them.

It has a total of 42 categories. We have considered a subset of seven categories that belong to computer science domain. The categories are:

- Communication Technology
- Computational and Artificial Intelligence
- Engineering Management
- Information Theory
- Professional Communication
- Reliability
- Robotics and Automation

Figure 6.18 shows the sample research article in IEEE Xplore digital library.



MFZ-KNN — A modified fuzzy based K nearest neighbor algorithm

Related Articles

A modified version of rate-monotonic scheduling algorithm and its' efficiency as...

Bearing defect detection using on-board accelerometer measurements

Shweta Taneja ; Charu Gupta ; Sakshi Aggarwal ; Veni Jindal

IEEE Keywords

Classification algorithms,
Clustering algorithms,
Algorithm design and analysis,
Accuracy,
Training,
Training data,
Data mining

INSPEC: Controlled Indexing

pattern classification,
data mining,
fuzzy set theory

INSPEC: Non-Controlled Indexing

standard UCI data set,
modified fuzzy based K nearest neighbor algorithm,
MFZ-KNN algorithm,
simplest top ten classification algorithm,
data mining,
modified fuzzy based KNN algorithm,
MatLAB,
Netbeans IDE

Author Keywords

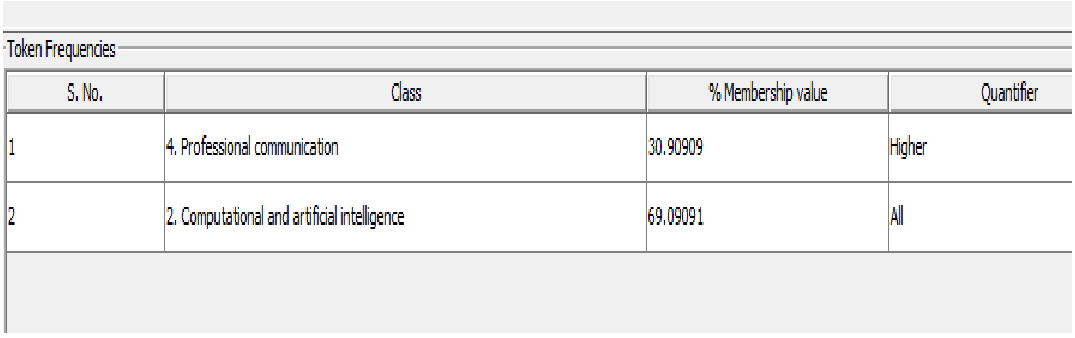
Fuzzy C-means(FCM),
KNN(K-Nearest Neighbor),
Fuzzy KNN (FKNN)

Figure 6.18: Sample research article in IEEE Xplore

It can be noticed in the figure that IEEE Xplore inserts four types of keywords with the research articles. These are: IEEE Keywords, INSPEC Controlled Indexing, INSPEC Non-Controlled Indexing and Author Keywords.

INSPEC is a database of scientific and technical research maintained by the Institution of Engineering and Technology. INSPEC covers areas like science, engineering etc.

In IEEE taxonomy, we have seen that in INSPEC Non-Controlled Indexing, new keywords are present, but they belong to same broad domain as Controlled Indexing. So, to prove our work, we have taken Controlled Indexing into consideration. IEEE Keywords are also covered in Controlled Indexing. In the above research article shown in figure 6.18, three key words are seen in Controlled Indexing: Pattern Classification, Data Mining and Fuzzy set theory. They all belong to same broad class: Computational and Artificial Intelligence.



Token Frequencies

S. No.	Class	% Membership value	Quantifier
1	4. Professional communication	30.90909	Higher
2	2. Computational and artificial intelligence	69.09091	All

Figure 6.19: Output of proposed LS-KDT process for sample research article

Figure 6.19 shows the output of proposed LS-KDT process for the same research article. It can be seen that the proposed process has displayed two classes: Computational and AI and Professional Communication with the membership values.

6.2.3 With existing Multi-label methods

In this section, the performance of the proposed LS-KDT process is compared with popular multi-label algorithms like ML-KNN, BR, ECC and RAkELd. ML-KNN algorithm is a modified version of KNN algorithm to suit the multi-label data (Zhang & Zhou, 2007). BR or Binary Relevance (Boutell, et al., 2004) is another famous multi-label algorithm that constructs multiple single-label or binary classifiers for each label. For multi-label categorization, there is another random k label-set algorithm known as RAkELd (Tsoumakas et al., 2011). Read J. in 2011 gave another ensemble classifier chains algorithm called as ECC.

The experiments are conducted on five text datasets containing multi-label instances. These are Computer Science research articles, Ohsumed dataset, Enron, Slashdot and Bibtex. Enron dataset consists of e-mails. It belongs to 1700 labelled e-mails from UC Berkeley Enron E-mail Analysis Project (Enron dataset). Slashdot dataset is made up of articles from Slashdot.org (Slashdot dataset). BibteX data set is made up of metadata like paper's title, authors etc. (BibteX data set).

The performance is evaluated with the help of standard measures like Accuracy, Hamming Loss, Average Precision, F1 micro average and total time taken (Godbole & Sarawagi, 2004).

Accuracy measures the closeness of the estimated labels(Y) to actual labels (Z). It is given in equation 6.4 below, where N is the total number of examples. The equations 6.4, 6.5 and 6.6 given below show the formulae used for Accuracy, Hamming Loss and Precision respectively:

$$\textit{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{Y_i \cap Z_i}{Y_i \cup Z_i}\right] \qquad (6.4)$$

$$\textit{Hamming Loss} = \frac{1}{N}\sum_{i=1}^{N}\frac{(Y_i \Delta Z_i)}{L}$$

$$\textit{(where}\Delta \textit{ stands for symmetric difference of two sets.)} \qquad (6.5)$$

$$\textit{Precision} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{Y_i \cap Z_i}{Z_i}\right] \qquad (6.6)$$

The implementation is done using a WEKA-based framework (WEKA tool) running under Java JDK 1.7 along with the libraries of MEKA and Mulan. Experiments are conducted on 64-bit machines with 2.6 GHz of clock speed. Evaluation is done in the form of training and test split on each dataset. The split into training and test is done on a random basis and repeated multiple times. 10-fold cross validation is used each time. The values of Accuracy, Hamming Loss, Average Precision, F-measure and Total time taken are calculated for each dataset. The total time includes build time as well as the test time of the classifier. The results obtained are shown in the form of tables from table 6.5 to table 6.9. And the performance metrics are shown graphically from figure 6.20 to figure 6.24.

Table 6.5: LS-KDT vs multi-label algorithms: Accuracy

Datasets	**BR**	**ML KNN**	**ECC**	**RAkELd**	**LS-KDT**
Enron	0.357	0.414	0.422	0.430	0.436
Ohsumed	0.482	0.460	0.450	0.400	0.482
Bibtex	0.400	0.400	0.412	0.452	0.493
Slashdot	0.600	0.612	0.623	0.632	0.646
Computer Science articles	0.520	0.531	0.530	0.550	0.563

Table 6.6: LS-KDT vs multi-label algorithms: Hamming Loss

Datasets	**BR**	**ML KNN**	**ECC**	**RAkELd**	**LS-KDT**
Enron	0.061	0.060	0.061	0.062	0.059
Ohsumed	0.035	0.035	0.036	0.036	0.021
Bibtex	0.490	0.500	0.500	0.501	0.48
Slashdot	0.16	0.16	0.17	0.17	0.15
Computer Science articles	0.250	0.251	0.251	0.252	0.245

Table 6.7: LS-KDT vs multi-label algorithms: Average Precision

Datasets	BR	ML KNN	ECC	RAkELd	LS-KDT
Enron	0.102	0.084	0.088	0.077	0.221
Ohsumed	0.165	0.092	0.095	0.082	0.231
Bibtex	0.132	0.212	0.236	0.301	0.343
Slashdot	0.212	0.259	0.301	0.335	0.361
Computer Science articles	0.312	0.339	0.359	0.401	0.423

Table 6.8: LS-KDT vs multi-label algorithms: F1-measure

Datasets	BR	ML KNN	ECC	RAkELd	LS-KDT
Enron	0.100	0.050	0.002	0.035	0.05
Ohsumed	0.550	0.420	0.400	0.392	0.560
Bibtex	0.480	0.481	0.482	0.481	0.482
Slashdot	0.612	0.623	0.635	0.635	0.636
Computer Science articles	0.710	0.700	0.700	0.685	0.690

Table 6.9: LS-KDT vs multi-label algorithms: Total Time (build time+ test time) (in seconds)

Datasets	BR	ML KNN	ECC	RAkELd	LS-KDT
Enron	480.087	704.061	660.566	558.976	552.23
Ohsumed	3148.977	3320.522	3101.23	3523.22	3102.20
Bibtex	2132.32	2111.30	2212.60	1950.30	1120.30
Slashdot	919.3	952.66	889.23	912.34	862.33
Computer Science articles	415.61	459.33	512.55	569.43	312.33

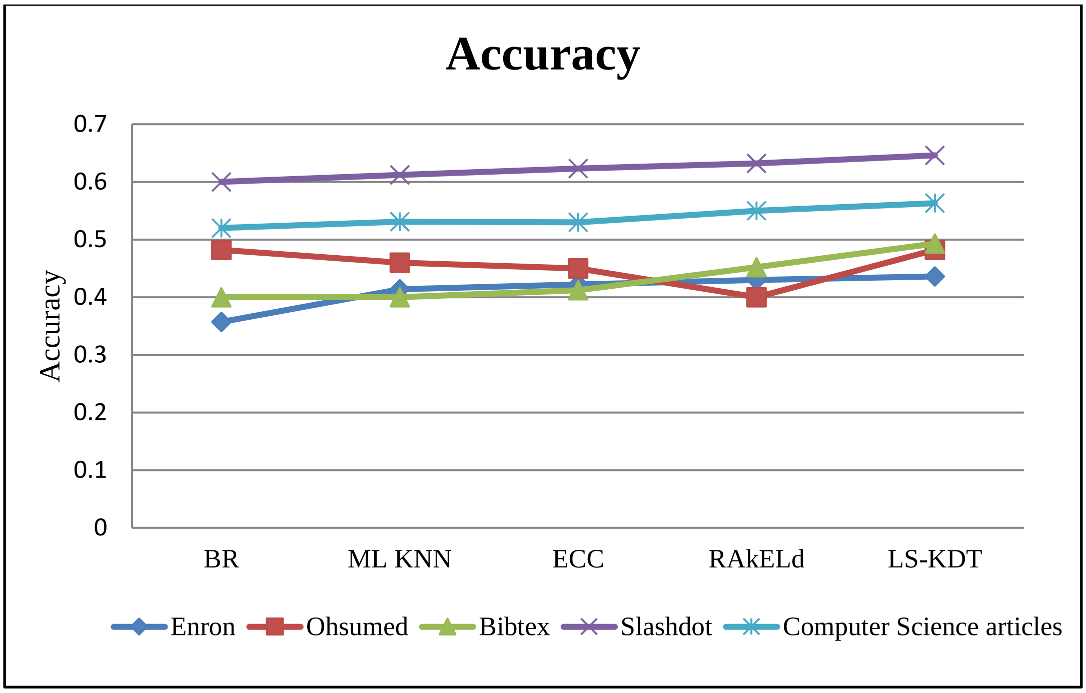


Figure 6.20: LS-KDT vs multi-label algorithms: Accuracy

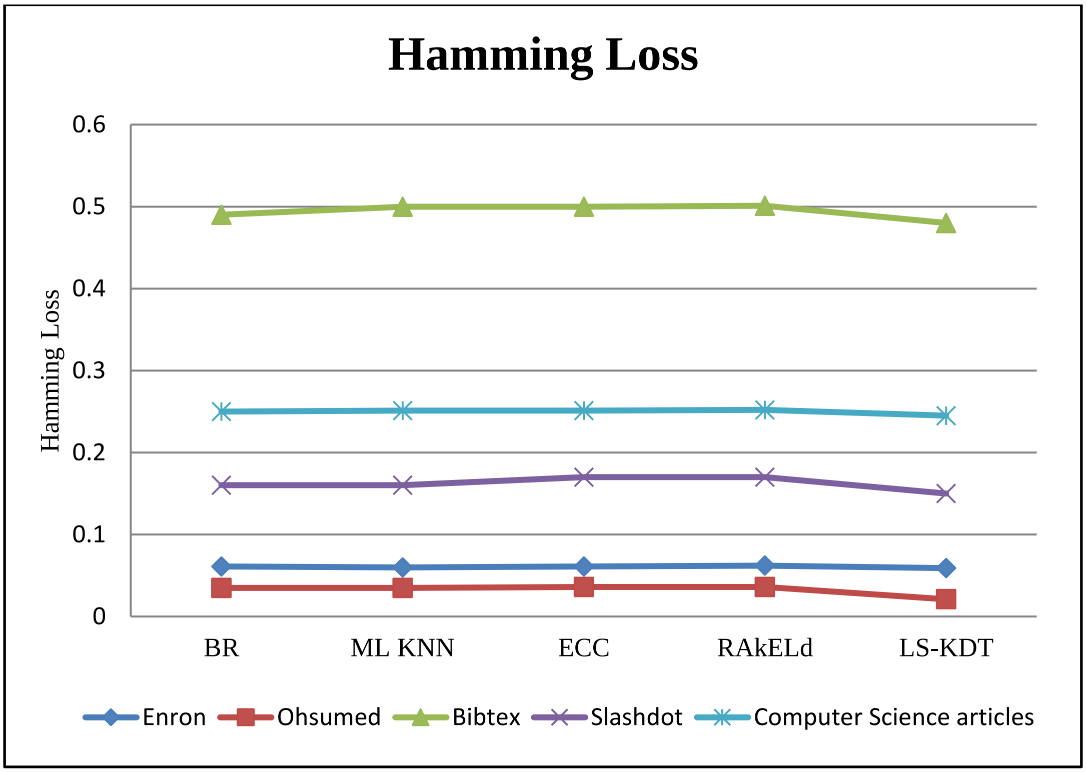


Figure 6.21: LS-KDT vs multi-label algorithms: Hamming Loss

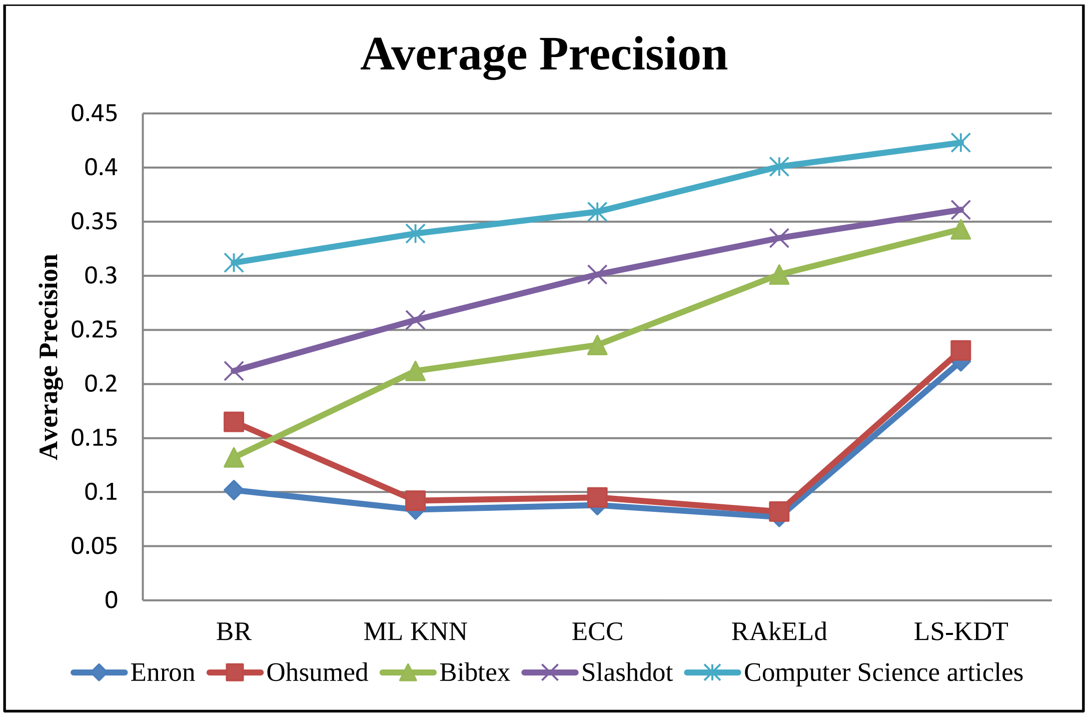


Figure 6.22: LS-KDT vs multi-label algorithms: Average Precision

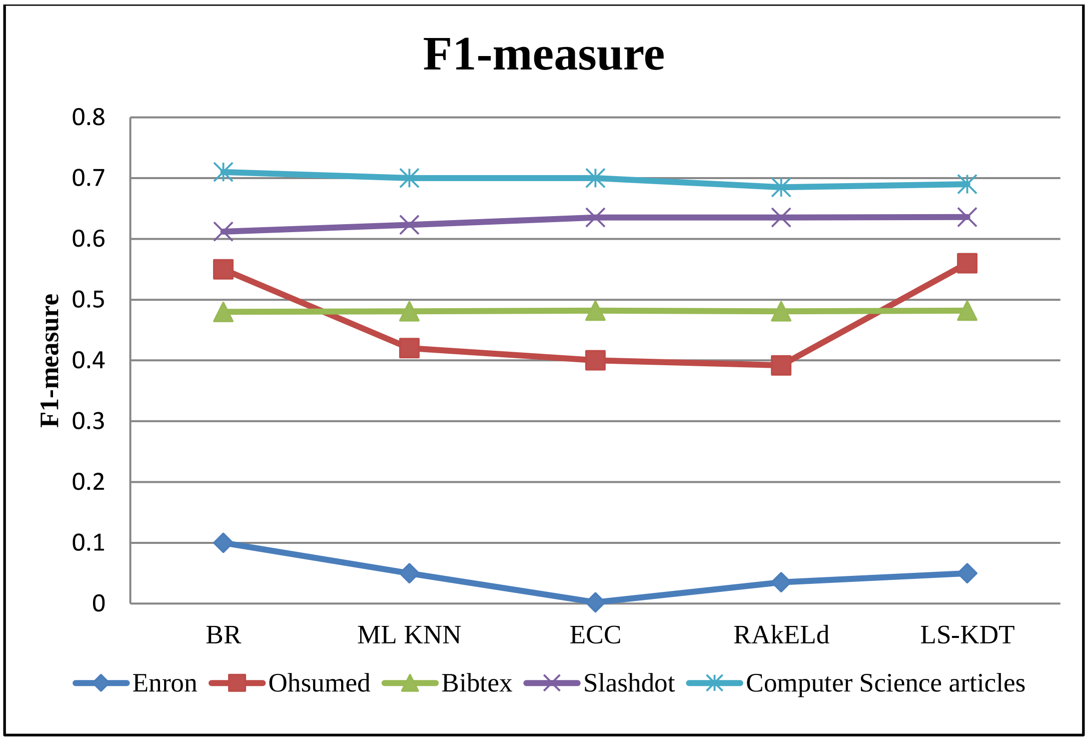


Figure 6.23: LS-KDT vs multi-label algorithms: F1- measure

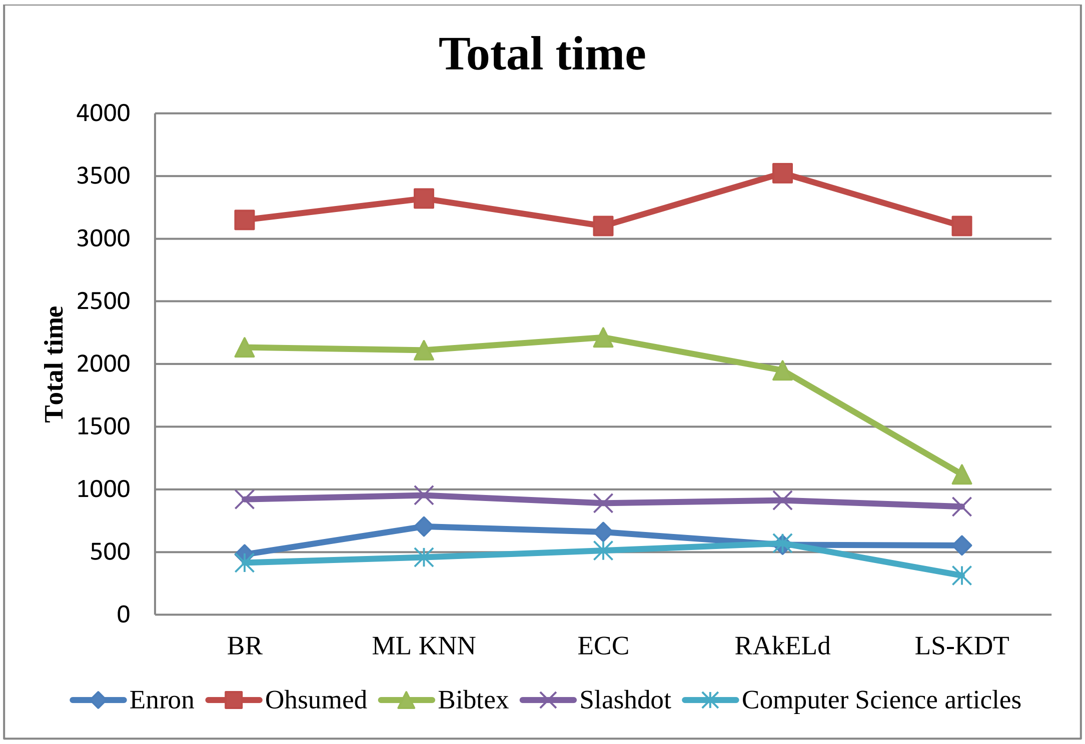


Figure 6.24: LS-KDT vs multi-label algorithms: Total Time (in seconds)

In multi-label categorization of text documents, there is not a single method that performs best on all datasets. Like the nearest neighbor methods like ML KNN and RAkELd perform well on metric-hamming loss as compared to other methods. Then in case of other metrics like Accuracy, Average Precision, F1- measure and total time taken, RAkELd performs better than rest of the algorithms. ECC performs well in most of the situations. Our proposed process gives good results on metrics like Accuracy, Hamming loss and Average Precision and on all datasets.

7 CONCLUSION AND FUTURE WORK

This chapter concludes the thesis. It gives the applications and contribution of our work and gives direction for future research in this area.

7.1 Conclusion of the Research

In this work, we have tried to achieve the basic aim of exploring techniques for knowledge discovery in text.

A framework called U-STRUCT is proposed that converts an unstructured text document to a structured form. It is a generic framework that can be applied in any type of domain. The proposed framework consists of two phases namely, Text Analysis phase and Text Synthesis phase. Thereafter, we conducted a survey of existing methods and algorithms of text categorization technique and identified their limitations. Further in our work, an algorithm for single-label categorization is proposed, that is, Lexical KNN or LKNN. It is based on lexical concepts. It is compared with the standard KNN algorithm. The performance of the proposed algorithm was good in terms of Recall, Precision and F1-measure.

Further, we have extended the single-label categorization to multi-label categorization. A modified knowledge discovery process known as Lexical - Semantics based Knowledge Discovery Process (LS-KDT) for multi-label text documents is developed. The proposed process is divided into seven phases: Text Document Collection, Data Pre-processing, Lexical Analysis, Semantic Analysis, Classification, Ranking of Labels and Knowledge Discovery. The proposed process is tested on research articles of computer science domain and articles of medical domain. The performance of the proposed process is compared with the results of two standard digital libraries: ACM digital library and IEEE Xplore database. The proposed process has shown significantly good results in terms of performance and accuracy. And further, the performance of the proposed process is compared with state of art multi- label algorithms.

7.2 Application of the Research

The applications of the research are as follows:

- The proposed Knowledge discovery process will help the research community to specify the exact categories to which a research article belongs.
- It will aid the journal editors in assigning reviewers to the research papers or articles.
- It will facilitate efficient search and categorization of journal articles. The accurate categorization of articles helps the digital libraries, databases, repositories or online resources to efficiently store or search the articles.

7.3 Future Work

The future work of proposed process is that it can be tested on journal articles of other domains of engineering like Electronics, Electrical etc. or on other text documents like legal documents, reports etc. The performance of our proposed process can be compared with other standard digital libraries or repositories that store research articles.

APPENDIX

Alidousti, S., Nazari, M., & Ardakan, M. A. (2008). A study of success factors of resource sharing in Iranian academic libraries. *Library Management*, 29 (8), 711-728.

Altena, A.J.V., Moerland, P.D., Zwinderman, A.H. & Olabarriaga, S.D. (2016). Understanding big data themes from scientific biomedical literature through topic modeling, *Journal of Big Data*, 3(23), 1-21. doi: 10.1186/s40537-016-0057-0.

Avanzi, R.M. (2005). The Complexity of Certain Multi-Exponentiation Techniques in Cryptography. *Journal of Cryptology*, 18(4), 357–373.

Barzilai-Nahon, K. (2008). Toward a theory of network gate keeping: A framework for exploring information control. *Journal of the American Society for Information Science and Technology*, 59(9), 1-20.

Bissyande, F.T. et al. (2013). Implementing an embedded compiler using program transformation rules. *Journal of Software: Practice and Experience* , 45(2), 177-196.

Borgatti, S. P. & Li, X. (2009). On social network analysis in a supply chain context. *Journal of Supply Chain Management*, 45 (2), 5-22.

Cohen, A.E. & Parhi, K.K. (2010). Fast Reconfigurable Elliptic Curve Cryptography Acceleration for GF (2m) on 32-bit Processors. *Journal of Signal Processing System,* 60(1), Springer, 31–45.

Chakraborty, P., Saxena, P.C. & Katti, P.C. (2010). A Compiler-Based Toolkit to Teach and Learn Finite Automata. *Wiley Periodicals, Computer Applications in Engineering Education*, 21(3), 467–474.

Chakraborty, P., Saxena, P.C., Katti, P.C., Pahwa, G. & Taneja, S. (2011). A New Practicum in Compiler Construction. *Wiley Periodicals, Computer Applications in Engineering Education*, 22(3), 429-441.

Dahiya, N., Bhatnagar, V., Singh, M. (2015). An empirical experimentation towards predicting understandability of conceptual schemas using quality metric. *Int. J. of Big Data Intelligence*, 2(1), 9-22.

Diamantoulakis, P.D., Kapinas, V.M. , Karagiannidis, G.K. (2015). Big Data Analytics for Dynamic Energy Management in Smart Grids. *Journal of Big Data Research*, 2(3), 94-101.

Dudin, E.B. & Smetanin, Y.G. (2010). Problems and Prospects of Modeling Computer Information Networks. A Review. *Journal of Automatic Documentation and Mathematical Linguistics*, 44 (6), 287–296.

Ellison, N.B., Steinfield, C. & Lampe, C. (2011). Connection strategies: Social capital implications of Face book-enabled communication practices, *New Media & Society*, 13(6), 873-892.

Farella, E. et al. (2008). Interfacing human and computer with wireless body area sensor networks: the WiMoCA solution. *Journal of Multimedia Tools Applications*, 38(3), 337–363.

Feng, G. (2014). Finding k shortest simple paths in directed graphs: A node classification algorithm. *Networks - An International Journal, Wiley Online Library*, 64(1), 6-17.

Fritsch, M., & Kauffeld-Monz, M. (2010). The impact of network structure on knowledge transfer: An application of social network analysis in the context of regional innovation networks. *The Annals of Regional Science*, 44 (1), 21-38.

Grcar, M., Trdin, N. & Lavrac, N. (2012). A Methodology for Mining Document-Enriched Heterogeneous Information Network. *The Computer Journal*, Oxford University Press, 56(3), 321-335.

Guerra, L., Bielza, C., Robles, V., Pedro (2014). Semi supervised projected model-based clustering. *Data Mining and Knowledge Discovery*, 28 (4), Springer, 882-917.

Haythornthwaite, C. (1996). Social network analysis: An Approach and technique for the study of information exchange. *Library & Information Science Research*, 18 (4), 323-342.

Jamali, Hamid, R. (2013). Citation relations of theories of human information behaviour, *Webology*, 10(1), Article 106. Available at http://www.webology.org /2013/v10n1/a106.html.

Jamali, H. R., Nooshinfard, F., Baghestani, G., & Asadi, S. (2010). Evaluation of the interlibrary loan services in Iran: A case study of the AMIN service. *Interlending& Document Supply*, 38 (4), 218-222.

Jalalimanesh, Ammar, Yaghoubi & Majid S. (2013). Application of social network analysis in interlibrary loan services. *Webology,* 10(1), Article 108. Available at: http://www.webology.org/2013/v10n1/a108.html.

Jantz, R.M. & Kulkarni, A.P. (2013). Analyzing and addressing false interactions during compiler optimization phase ordering. *Journal of Software: Practice and Experience* , Wiley Online Library, 44(6), 643-679.

Jayaraman, B. (2012). Special Issue on Security and Performance of Networks and Cloud. *The Computer Journal*, Oxford University Press on behalf of The British Computer Society, 55(8), 907-908.

Kannan, S.R., Ramthilagam, S., Devi, S., Huang & Y. M. (2012). Novel Quadratic Fuzzy c-Means Algorithms for Effective Data Clustering Problems. *The Computer Journal*, Oxford University Press, 56(3), 393-406.

Krall, A. & Barany (2012). Compilers for Parallel Computing. *Journal of Concurrency and Computation: Practice and Experience* , 24(5), Wiley Online Library.

Larivière, V., Gingras, Y. & Archambault, E. (2006). Comparative analysis of networks of collaboration of Canadian researchers in the natural sciences, social sciences and the humanities. *Scientometrics*, 68(3), 519-533.

Lomotey, R.K. & Deters R. (2015). Terms analytics service for CouchDB: a document-based NoSQL. *Int. J. Big Data Intelligence*, 2 (1), 23-36, Inderscience Enterprises Ltd.

Memar, M. et al. (2012). An Efficient Frequent Item set Mining Method over High-speed Data Streams. *The Computer Journal*, Oxford University Press on behalf of The British Computer Society, 55(11), 1357-1366.

Moskovkin, V.M. (2009). The potential of using the Google Scholar search engine for estimating the publication activities of universities. *Scientific and Technical Information Processing*, 36(4), 198-202.

Moskovkin, V.M., Delux, T., & Moskovkina, M.V. (2012). Comparative analysis of university publication activity by Google Scholar (on example of leading Czech and Germany universities). Cybermetrics: *International Journal of Scientometrics, Informetrics and Bibliometrics*, (16), 2-9.

Moskovkin, Vladimir, M., Fraser, Jason, K., & Moskovkina, Maria, V. (2013). University networks in the context of their academic excellence and openness: A comparative study of leading Czech and German universities. *Webology*, 10(1), Article 107. Available at: http://www.webology.org/2013/v10n1/a107.html.

Nakano, I.S., Uehara, R., & Uno, T. (2013). Efficient algorithms for a simple network design problem. *Wiley Periodicals, Inc. Networks*, 62 (2), 95–104.

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32 (3), 245-251.

Otte, E., & Rousseau, R. (2002). Social network analysis: A powerful strategy, also for the information sciences. *Journal of Information Science*, 28 (6), 441-453.

Padmaja, B., Rama Prasad, V.V. & Sunitha, K.V.N (2016). TreeNet analysis of human stress behavior using socio-mobile data. *Journal of Big Data*, 3(1), 24. doi: 1.1186/s40537-016-0054-3.

Salles, R.M. & Donato, A. (2011). Strategies and Metric for Resilience in Computer Networks. *The Computer Journal*, Oxford University Press on behalf of The British Computer Society, 55(6), 728-739.

Shabeera, T.P. & Madhu Kumar, S.D. (2015). Optimising virtual machine allocation in MapReduce cloud for improved data locality. *Int. J. of Big Data Intelligence*, 2 (1), 2 – 8.

Stárka, J. et al. (2011). Analyzer: A Complex System for Data Analysis, *The Computer Journal*, Oxford University Press, 55(5), 590-615.

Taylor, R. S. (1968). Question-negotiation and information seeking in libraries. *College and Research Libraries*, 29, 178-194.

Tew, C., Carrier, C. G., Tanner, K., & Burton, S. (2014). Behavior-based clustering and analysis of interestingness measures for association rule mining. *Data Mining and Knowledge Discovery*, 28 (4), Springer, 1004-1045.

Tian, B.et al. (2011). Self-Healing Key Distribution Schemes for Wireless Networks: A Survey. *The Computer Journal*, and Oxford University Press on behalf of The British Computer Society, 54(4), 549-569.

Turner, T. C., Smith, M. A., Fisher, D., & Welser, H. T. (2005). Picturing Usenet: Mapping computer-mediated collective action. *Journal of Computer-Mediated Communication*,10(4), article7, Retrieved from http://jcmc.indiana.edu/vol10/issue4/turner.html.

Varga, A., & Parag, A. (2009). Academic knowledge transfers and the structure of international research networks. University knowledge transfers and regional development: Geography, entrepreneurship and policy, *Edward Elgar Publishers*, 138-159.

Wang, H., & Wellman, B. (2010). Social connectivity in America: Changes in adult friendship network size from 2002 to 2007. *American behavioural Scientist*, 53(8), 1148- 1169.

Weisz, C., & Wood, L.F. (2005). Social identity support and friendship outcomes: A longitudinal study predicting who will be friends and best friends 4 years later. *Journal of Social and Personal Relationships*, 22(3), 416-432.

Williams, A.L., & Merten, M.J. (2008). A review of online social networking profiles by adolescents: Implications for future research and interventions. *Adolescence,* 43(170), 253-274.

Wildemuth, B. M. (1992). An empirically grounded model of the adoption of intellectual technologies. *Journal of the American Society for Information Science*, 43 (3), 210-224.

Zhang, H., Raitoharju, J., Kiranyaz, S. & Gabbouj, M. (2016). Limited random walk algorithm for big graph data clustering. *Journal of Big Data*.3(26). doi: 10.1186/s40537-016-0060-5.

Zhecheng, Z. (2016). Application of Geographical Information System and Interactive Data Visualization in Healthcare Decision Making. *International Journal of Big Data and Analytics in Healthcare (IJBDAH),* 1(1), 49-58. doi: 10.4018/IJBDAH.2016010104.

List of Publications from the thesis

Papers Accepted/Published in International Journals:

- Jindal, R., & Taneja, S. (2017). A lexical-semantics-based method for multi-label text categorization using word net. International Journal of Data Mining, Modelling and Management, 9(4), 340-360. Publisher: Inderscience,
  [Online]:
  https://www.inderscienceonline.com/doi/pdf/10.1504/IJDMMM.2017.088412.

  [Scopus]

- Rajni Jindal and Shweta (2017). A Modified Knowledge Discovery Process in the Text Documents. Accepted for publication in International Journal of Innovative Computing, Information and Control (IJICIC).

  [Scopus]

- Rajni Jindal and Shweta (2015). A Lexical Approach for Text Categorization of Medical Documents", proceedings published in Elsevier Procedia Computer Science , Volume 46, Pages 1-1834, proceedings of the International Conference on Information and Communication Technologies, ICICT 2014, 3-5 December 2014 at Bolgatty Palace & Island Resort, Kochi, India. The publication is made available on sciencedirect.com.
  [Online]:
  http://www.sciencedirect.com/science/journal/18770509/46.

  [Scopus]

- Rajni Jindal and Shweta (2016). A Wordnet Based Semantic Approach for Dimension Reduction in Multi-label Text Documents. I J C T A, 9(40), pp. 267-274 © International Science Press.
  [Online]

http://serialsjournals.com/articles.php?volumesno_id=1145&journals_id=268&volumes_id=848.

[Scopus till 2016]

- Rajni Jindal and Shweta (2017). A Novel Weighted Linguistic Approach to Text Categorization, International Journal of Computer Applications (IJCA), 180(2), pp.9-15.

  [UGC Approved]

Papers Accepted/Published in International Conferences:

- Rajni Jindal and Shweta (2013). Text Categorization – A Review Published in Third International Conference on Computational Intelligence and Information Technology, CIIT 2013, held during Oct 18-19, 2013 in Mumbai, India. The proceedings are published in Elsevier.
  [Online]
  http://searchdl.org/public/book_series/AETS/7/126.pdf

  [Elsevier]

- Rajni Jindal and Shweta (2013) U-STRUCT: A Framework for Conversion of Unstructured Text Documents into Structured Form. Published in Advances in Computing, Communication and Control. Communications in Computer and Information Science book series, CCIS, 361, Springer, Berlin, Heidelberg, pp. 59-69.
  [Online]
  https://link.springer.com/chapter/10.1007/978-3-642-36321-4_6.

  [Scopus]

- Rajni Jindal and Shweta (2015). Ranking in Multi-Label Classification of Text Documents Using Quantifiers. Published in the proceedings of 5th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2015, held at Park Royal Penang Resort, Batu Ferringhi, 27-29 Nov 2015, Malaysia. The proceedings are published in IEEE Xplore digital library.
  [Online]
  ieeexplore.ieee.org/iel7/7468595/7482142/07482177.pdf.

  [Scopus]

- Rajni Jindal and Shweta (2015). A Lexical Approach for Text Categorization of Medical Documents, proceedings of the International Conference on Information and Communication Technologies, ICICT 2014, 3-5 December 2014 at Bolgatty Palace & Island Resort, Kochi, India. The publication is made available on sciencedirect.com.

  [Online]
  http://www.sciencedirect.com/science/journal/18770509/46.

  [Scopus]

- Rajni Jindal and Shweta (2016). A Wordnet Based Semantic Approach for Dimension Reduction in Multi-label Text Documents, proceedings of the second International Conference on Sustainable Computing techniques in Engineering, Science and Management held in January 2017 near Goa, India.

- Rajni Jindal and Shweta (2018). A Novel Method for efficient multi- label text categorization of research articles, Communicated to an International Conference.

REFERENCES

Abe, S. (2015). Fuzzy support vector machines for multilabel classification. *Pattern Recognition*, 48(6), 2110-2117.

ACM Computing Classification System (2012). [Online] Available at: https://www.acm.org/publications/class-2012.

Aggarwal, C., & Zhai, C.X, (Eds.) (2012). *Mining Text Data*. London: Springer, ISBN 978-1-4614-3222-7.

Al-Zaidy, R. et al. (2012). Mining criminal networks from unstructured text documents. *Digital Investigation*, 8(3-4), 147–160.

Atkinson J., Abutridy, Mellish, C. & Aitken, S. (2004). Combining Information Extraction with Genetic Algorithms for Text Mining, *IEEE Intelligent Systems*, 19(3), 22-30.

Balamurugan, et al. (2011). NB+: An improved Naïve Bayesian algorithm, *Journal of Knowledge-Based Systems*, 24(5), 563-569.

Bax E. (2012). Validation of K- Nearest Neighbor Classifiers, *IEEE Transactions on information theory*, 58(5), 3225-3234.

Beliakov G. & Li G. (2012). Improving the speed and stability of the k-nearest neighbors method, *Journal of Pattern Recognition Letters*, Elsevier, 33(10), 1296-1301.

BibteX data set. Source: Katakis I., Tsoumakas G., & Vlahavas I. (2008). Multilabel Text Classification for Automated Tag Suggestion. *Proceedings of the ECML/PKDD 2008 Discovery Challenge*, Antwerp, Belgium.
[Online] http://mulan.sourceforge.net/datasets.html.

Boutell, M.R., et al. (2004). Learning multi-label scene classification. *Pattern Recognition,* 37(9), 1757–1771.

Cardoso-Cachopo A. & Oliveira A.L. (2007). Semi-supervised single-label text categorization using centroid-based classifiers. *Proceedings of the 2007 ACM Symposium on Applied Computing (SAC),* Seoul, Korea, March 11-15, doi: 10.1145/1244002.1244189.

Carvalho, V. R., & Cohen, W. W. (2005). On the collective classification of email speech acts. *Proceedings of the 28th Annual International ACM SIGIR conference on Research and development in information retrieval,* 345-352, ACM.

Ceci, M. (2008). Hierarchical text categorization in a transductive setting. In *IEEE International Conference on* Data *Mining Workshops, ICDMW'08,* 184-191. IEEE.

Cerri, R. et al. (2014). Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences, 80*(1), 39-56.

Chang, Y. H., & Huang, H. Y. (2008). An automatic document classifier system based on naive bayes classifier and ontology. Proceedings of *2008 International Conference on* Machine *Learning and Cybernetics,* 6, 3144-3149, IEEE.

Chang, Y. C. et al. (2008). Multi label text categorization based on a new linear classifier learning method and a category-sensitive refinement method. *Expert Systems with Applications*, 34(3), 1948-1953.

Chen, H. et al. (2010). The Application of Decision Tree in Chinese Email Classification. *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, Qingdao.

Chen et al. (1996). Data Mining- An Overview from a Database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8(6), 866-883.

Cohen, W. W. (1996). Learning rules that classify e-mail. In *AAAI Spring symposium on Machine Learning in Information Access*, 18, 25.

Cover, T.M. & Hart, P.E. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions Information Theory*, 13(1), 21-27.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, 519-528, ACM.

Deerwester, S. et al. (1990). Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6), 391-407.

Deng, Z.H. & Tang, S.W. (2005). A Non – VSM Algorithm for Text Classification, *LNAI*, 3584, 339-346, Springer.

Enron dataset. UC Berkeley Enron Email Analysis Project. [Online] http://mulan.sourceforge.net/datasets.html.

Fayed, H. A., & Atiya, A. F. (2009). A novel template reduction approach for the K -nearest neighbor method. *IEEE Transactions on Neural Networks*, *20*(5), 890-896.

Fayyad, et al. (1996). From Data Mining to Knowledge Discovery in databases. *AI Magazine*, 17(3), 37-54.

Feldman, R. & Dagan, I. (1995). Knowledge discovery in textual databases (KDT. *Proceedings of First International Conference on Knowledge Discovery and Data Mining, KDD 95*, Canada, 112-117.

Fujino, A. et al. (2007). A hybrid generative/discriminative approach to text classification with additional information, *Journal of Information Processing & Management*, 43(2), 379–392.

Gabrilovich, E. & Markovitch, S. (2009). Wikipedia-based Semantic Interpretation for Natural Language Processing. *Journal of Artificial Intelligence Research*, 34, 443-498.

García-Laencina, P. J., Sancho-Gómez, J. L., Figueiras-Vidal, A. R., & Verleysen, M. (2009). K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing*, *72*(7-9), 1483-1493.

Gibaja, E., & Ventura, S. (2014). Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(6), 411-444.

Godbole, S., & Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*, 22-30, Springer, Berlin, Heidelberg.

Guan, H. et al. (2013). Fast dimension reduction for document classification based on imprecise spectrum analysis. *Information Sciences*, *222*, 147-162.

Han J., Pei J. & Kamber M. (2011), *Data Mining Concepts and Techniques*, Elsevier.

Hearst, M. A. (1999). Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 3-10, Association for Computational Linguistics.

Hersh, W., et al. (1994). OHSUMED: an interactive retrieval evaluation and new large test collection for research. In *SIGIR'94*,192-201, Springer, London.

Hou, C., Zhang, C., Wu, Y., & Nie, F. (2010). Multiple view semi-supervised dimensionality reduction. *Pattern Recognition*, *43*(3), 720-730.

IEEE Taxonomy. https://www.ieee.org/documents/taxonomy_v101.pdf

INSPEC database. https://www.theiet.org/resources/inspec/

Jensen, L. S., & Martinez, T. (2000). *Improving text classification by using conceptual and contextual features* (Doctoral dissertation, Brigham Young University. Department of Computer Science).

Jiang, S. et al. (2012). An improved K-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, *39*(1), 1503-1509.

Jindal, R., & Taneja, S. (2015). Ranking in multi label classification of text documents using quantifiers. *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE),* 162-166, IEEE.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning,* 137-142, Springer, Berlin, Heidelberg.

Johnson, D. E. et al. (2002). A decision-tree-based symbolic rule induction system for text categorization. *IBM Systems Journal*, *41*(3), 428-437.

Khalessizadeh, S. M., Zaefarian, R., Nasseri, S. H., & Ardil, E. (2006). Genetic mining: using genetic algorithm for topic based on concept distribution. *Transactions on Engineering Computing and Technology*, *13*, 44-147.

Kim, S. B. et al. (2006). Some effective techniques for naive bayes text classification. *IEEE transactions on knowledge and data engineering*, *18*(11), 1457-1466.

Kim, H., Howland, P., & Park, H. (2005). Dimension reduction in text classification with support vector machines. *Journal of Machine Learning Research, 6*, 37-53.

Koller, D., & Sahami, M. (1996). *Toward optimal feature selection*. Stanford InfoLab.

Kumar, M. A., & Gopal, M. (2010). A comparison study on multiple binary-class SVM methods for unilabel text categorization. *Pattern Recognition Letters*, *31*(11), 1437-1444.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings*, 331-339.

Lee S. et al. (2014). Knowledge discovery in inspection reports of marine structures. *Expert Systems with Applications, 41*(4), 1153-1167.

Lee, C. H., & Yang, H. C. (2009). Construction of supervised and unsupervised learning systems for multilingual text categorization. *Expert Systems with Applications*, *36*(2), 2400-2410.

Lee, C. H., & Yang, H. C. (2005). A classifier-based text mining approach for evaluating semantic relatedness using support vector machines. *International Conference* on *Information Technology: Coding and Computing*, *ITCC,* 1, 128-133 IEEE.

Li, Y. H., & Jain, A. K. (1998). Classification of text documents. *The Computer Journal*, *41*(8), 537-546.

Li, C. H., Yang, J. C., & Park, S. C. (2012). Text categorization algorithms using semantic approaches, corpus-based thesaurus and WordNet. *Expert Systems with Applications*, *39*(1), 765-772.

Li, Z. et al. (2011). Fast text categorization using concise semantic analysis. *Pattern Recognition Letters*, *32*(3), 441-448.

Li, B., Yu, S., & Lu, Q. (2003). An improved k-nearest neighbor algorithm for text categorization. arXiv preprint cs/0306099.

Liu, C. et al. (2017). A new Centroid-Based Classification model for text categorization. *Knowledge-Based Systems*, *136*, 15-26.

Maciołek, P., & Dobrowolski, G. (2013). Using shallow semantic analysis and graph modelling for document classification. *International Journal of Data Mining, Modelling and Management*, *5*(2), 123-137.

Mallick, K., & Bhattacharyya, S. (2012). Uncorrelated local maximum margin criterion: an efficient dimensionality reduction method for text classification. *Procedia Technology*, 4, 370-374.

Mayor, S., & Pant, B. (2012). Document classification using support vector machine. *International Journal of Engineering Science and Technology*, *4*(4).

Mehnert, R. (1997). Federal Agency and Federal Library Reports, National Library of Medicine: Providence. [Online] http://www.nlm.nih.gov/

Mehta, M., Agrawal, R., & Rissanen, J. (1996). SLIQ: A fast scalable classifier for data mining. In *International Conference on Extending Database Technology*, 18-32, Springer, Berlin, Heidelberg.

Meka tool [Online] http://MEKA.sourceforge.net

MESH tree structure- https://www.nlm.nih.gov/mesh

Mika, S. et al. (1999). Fisher discriminant analysis with kernels. *Proceedings of the 1999 IEEE signal processing society workshop*, in *Neural networks for signal processing IX*, 41-48, IEEE.

Miller, G. A. et al. (1990). Introduction to Word Net: An on-line lexical database. *International journal of lexicography*, *3*(4), 235-244.

Patil, S., & Ravindran, B. (2015). Active learning based weak supervision for textual survey response classification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 309-320, Springer, Cham.

Peason, K. (1901). On lines and planes of closest fit to systems of point in space. *Philosophical Magazine*, *2*(11), 559-572.

Pietramala, A. et al. (2008). A genetic algorithm for text classification rule induction. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 188-203, Springer, Berlin, Heidelberg.

PubMed Database: https://www.ncbi.nlm.nih.gov/pubmed/

Rajpathak, D. G. (2013). An ontology-based text mining system for knowledge discovery from the diagnosis data in the automotive domain. *Computers in Industry*, *64*(5), 565-580.

Ramkumar, A. S., & Poorna, B. (2016). Text document clustering using dimension reduction technique. *International Journal of Applied Engineering Research*, *11*(7), 4770-4774.

Read, J. et al. (2016). Meka: a multi-label/multi-target extension to weka. *The Journal of Machine Learning Research*, *17*(1), 667-671.

Read, J. et al. (2011). Classifier chains for multi-label classification. *Machine learning*, *85*(3), 333.

Schildt, H. (2007). *Java: The complete reference*. McGraw-Hill.

Scott, S., & Matwin, S. (1998). Text classification using WordNet hypernyms, In Association for Computational Linguistics,38-44.

Sedding J., Kazakov D. (2004), ‘Word Net-based text document clustering’, *Proc. of COLING- Workshop on Robust Methods in Analysis of Natural Language Data*, 104-113.

Serafino, F. et al. (2015). Hierarchical Multidimensional Classification of web documents with MultiWebClass. In *International Conference on Discovery Science*, 236-250, Springer, Cham.

Shatkay, H., & Feldman, R. (2003). Mining the biomedical literature in the genomic era: an overview. *Journal of computational biology*, *10*(6), 821-855.

Slashdot dataset. [Online] http://MEKA.sourceforge.net.

Song, M. et al. (2015). PKDE4J: Entity and relation extraction for public knowledge discovery. *Journal of biomedical informatics*, *57*, 320-332.

Soria, D. et al. (2011). A ‘non-parametric’version of the naive Bayes classifier. *Knowledge-Based Systems*, *24*(6), 775-784.

Stavrianou, A., Andritsos, P., & Nicoloyannis, N. (2007). Overview and semantic issues of text mining. *ACM Sigmod Record*, *36*(3), 23-34.

Stop words list [Online list] Available at

http://www.db-net.aueb.gr/gbt/resources/stopwords.txt

Tan, S. (2005). Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, *28*(4), 667-671.

Tan, S. (2008). An improved centroid classifier for text categorization. *Expert Systems with Applications*, *35*(1-2), 279-285.

Tsoumakas, G. et al. (2011). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, *12*(Jul), 2411-2414.

Tsoumakas, G., & Zhang, M. L. (2009). Learning from multi-label data. Tutorial at ECML/PKDD, Bled, Slovenia.

Toyama, J., Kudo, M., & Imai, H. (2010). Probably correct k-nearest neighbor search in high dimensions. *Pattern Recognition*, *43*(4), 1361-1372.

Uguz, H. (2011). A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems, 24*(7), 1024-1032.

Uramoto, N. et al. (2004). A text-mining system for knowledge discovery from biomedical documents. *IBM Systems Journal*, *43*(3), 516-533.

Vateekul, P., & Kubat, M. (2009). Fast induction of multiple decision trees in text categorization from large scale, imbalanced, and multi-label data. *IEEE International Conference on* Data *Mining Workshops, ICDMW'09,* 320-325, IEEE.

Wagh, R. S. (2013). Knowledge Discovery from Legal Documents Dataset using Text Mining Techniques. *International Journal of Computer Applications*, *66*(23).

Wan, C. H. et al. (2012). A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. *Expert Systems with Applications*, *39*(15), 11880-11888.

Wang, P., & Domeniconi, C. (2008). Building semantic kernels for text classification using wikipedia. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 713-721, ACM.

Wei, T., Lu, Y., Chang, H., Zhou, Q., & Bao, X. (2015). A semantic approach for text clustering using WordNet and lexical chains. *Expert Systems with Applications*, *42*(4), 2264-2275.

WordNet (2012). http://wordnet.princeton.edu/.

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Icml*, 97, 412-420.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 42-49, ACM.

Yu, Y. et al. (2014). Multi-label classification by exploiting label correlations. *Expert Systems with Applications*, *41*(6), 2989-3004.

Yuan, P. et al. (2008). MSVM-kNN: Combining SVM and k-NN for Multi-class Text Classification. *IEEE International Workshop on Semantic Computing and Systems, 2008. WSCS'08*, 133-140, IEEE.

Zhang, W., & Gao, F. (2011). An improvement to naive bayes for text classification. *Procedia Engineering*, *15*, 2160-2164.

Zhang, S. (2012). Nearest neighbor selection for iteratively kNN imputation. *Journal of Systems and Software*, *85*(11), 2541-2552.

Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, *40*(7), 2038-2048.

Zhen-fang, Z., Pei-yu, L., & Ran, L. (2008). Research of text classification technology based on genetic annealing algorithm. *International Symposium on* Computational *Intelligence and Design*, *2008. ISCID'08*, 1, 265-269, IEEE.

Biography of Authors

Ms. Shweta

Ms. Shweta is a research scholar in Department of Computer Science & Engineering at Delhi Technological University. She received her MTech (Information Systems) degree from Netaji Subhas Institute of Technology, Delhi University in 2009. Her areas of interest are Data Mining, Text Mining, Data Warehousing and Database Management Systems. She has published around 30 research papers in various journals and conferences, both national and international. She has a total experience of 15 years in Academics and Research.

Dr. (Mrs.) Rajni Jindal

Dr. (Mrs.) Rajni Jindal is currently heading the Department of Computer Science & Engineering at Delhi Technological University (erstwhile Delhi College of Engineering). She is working here as faculty since 1992. She completed her PhD (Computer Engineering) from Faculty of Technology, Delhi University in the area of Data Mining. She received her M.E. (Computer Technology & Applications) degree from Delhi college of Engineering and her Master of Computer Applications (MCA) degree from Jawaharlal Nehru University (JNU), Delhi. Prior to joining teaching, she has also worked with Tata Consultancy Services (TCS).

She joined Indira Gandhi Technical University for Women (IGDTUW) as Professor in 2012 on lien. She worked as Head (IT) and Dean (Research & Collaboration) at IGDTUW till Feb. 2015 before returning back to DTU.

She possesses a work experience of more than 25 years in research and academics. Her major areas of interest are Database Systems, Data Mining and Operating systems. She has taught various subjects at UG and PG Level. She has supervised more than 50 ME/MTech thesis. One of her guided ME Thesis “Semantic Web in Web Advertising and Software Engineering” is published by LAP LAMBERT Academic Publishing, Germany.

She has authored/coauthored around 80 research papers and articles for various national and international journals/conferences. Recently one of her papers is published in IT Professional published by IEEE Computer Society. There are 12 PhD students working under her supervision. 2 PhD have already been awarded to her students.

She has visited and presented papers abroad at places like San Francisco, Las Vegas and Spain. She has completed AICTE sponsored project in the area of education data mining as Co- Principal Investigator. She has authored books on “Data Structures using C” and “Compiler-Construction and Design”. She has also coauthored books on “Computer and Communication Technology “developed by NCERT for class XI and XII.

Other than organizing various workshops and lectures she successfully organized an IEEE International Conference on Data mining and Intelligent Computing (ICDMIC-2014) held at IGDTUW and another IEEE International Conference on information processing (IICIP 2016) at DTU. She is a life member of professional bodies like CSI, ISTE and senior member of IEEE, USA.