CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Multi-label Classification of Research Articles using Word2Vec and Identification of Similarity Threshold

by

Ghulam Mustafa

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Computing
Department of Computer Science

2020

My dissertation work is devoted to My Family, My Teachers and My Friends. I have a special feeling of gratitude for My beloved parents, brothers and sisters. Special thanks to my supervisor whose uncountable confidence enabled me to reach this milestone.

CERTIFICATE OF APPROVAL

Multi-label Classification of Research Articles using Word2Vec and Identification of Similarity Threshold

by

Ghulam Mustafa

(MCS181037)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Ayyaz Hussain	QAU, Islamabad
(b)	Internal Examiner	Dr. Muhammad Shahid Iqbal Malik	CUST, Islamabad
(c)	Supervisor	Dr. Muhammad Tanvir Afzal	CUST, Islamabad

Dr. Muhammad Tanvir Afzal
Thesis Supervisor
March, 2020

Dr. Nayyer Masood
Head
Dept. of Computer Science
March, 2020

Dr. Muhammad Abdul Qadir
Dean
Faculty of Computing
March, 2020

Author's Declaration

I, **Ghulam Mustafa** hereby state that my MS thesis titled "**Multi-label Classification of Research Articles using Word2Vec and Identification of Similarity Threshold**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Ghulam Mustafa)
Registration No: MCS181037

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Multi-label Classification of Research Articles using Word2Vec and Identification of Similarity Threshold**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Ghulam Mustafa)
Registration No: MCS181037

Acknowledgements

Say, "He is Allah, [who is] One. Allah, the Eternal Refuge. He neither begets nor is born. Nor is there to Him any equivalent." Al-Quran [112:1-4]. I would like to said ***Alhamdulillah***, for blessing me, with strength to finished this work.

My heartfelt thanks to my parents for her love, prayers and everything what I need. A special thanks to my teachers, brothers and sisters for their support. Last but not the least my friends, with whom this journey becomes a garden with full of flowers to me.

Special thanks to my esteemed supervisor ***Dr. Muhammad Tanvir Afzal*** for the support, for his assistance, inspiration and guidance in the field of research. I have learned not only research under his supervision, but also his attitude towards others. Sir "You are one of those persons in my life, whom I love to follow".

May Allah shower his countless blessing upon all of you, ***JazakAllah***

(Ghulam Mustafa)

Registration No: MCS181037

Abstract

In scientific literature, a publication is deemed as one of the potential indicators to determine scientific contributions in different disciplines. The process of research publication never stops, instead a gradual increase is witnessed after every year. From past several years, scientific community has been producing a huge plethora of research documents and it is believed that the amount will get doubled after every five year. Around 28,100 journals are publishing 2.5 million research articles every year. These documents are searched by a wider population using search engines, digital libraries and citation indexes. When a user poses a query, its returns a bulk of documents in which few of them are relevant. This is due to the fact that these documents are highly un-structured as they have not been indexed properly. The existing systems normally index research papers using keywords not on the basis of some subject hierarchy. In literature, varieties of techniques have been proposed to perform single (SLC) or multi-label classification (MLC) using content and metadata based features. Content based approaches produce comparatively better results due to richness of features. However, content of scientific paper is not always freely available. One alternative to the content is utilization of metadata-based parameter, however, the existing metadata based approaches report low accuracy. Further, the approaches have employed conventional statistical measures to represent textual features due to which semantic context could not be identified. Also, existing MLC approaches require a predefined value for threshold to map articles onto predefined categories for which domain expert's knowledge is required. This thesis focuses on overcoming all of the said deficiencies by proposing a model which performs SLC and MLC of research articles. The model uses word2vec model for textual representation to capture the semantic and contextual information of terms. We also introduce a method to obtain a threshold values by rigorous analysis of data which omits the need of domain experts. For experimentation we have used two Computer Science datasets (JUCS & ACM). The outcomes revealed good performance of a model than contemporary state-of-the-art. For SLC, the average accuracy of 0.87 & 0.84 for JUCS and ACM respectively. For MLC, the average accuracy is 0.82 & 0.80 for JUCS and ACM respectively.

Contents

List of Figures

List of Tables

Abbreviations

ACM	Association of Computing Machinery
BOW	Bag of Word
JUCS	Journal of Universal Computer Science
MLC	Multi-Label Classification
SLC	Single Label Classification
TF	Term Frequency
W2V	Word2Vec

Symbols

AS_c	Average Similarity for individual category
SS_{CnCm}	Average Similarity Score between two categoires papers
$M_T(D_n)$	Represent a list of diagonal values
T_p	Represent a test paper
P_{c_i}	Represent individual category papers
SS	Similarity Score

Chapter 1

Introduction

Researchers have presented their ideas/innovations/breakthroughs in the form of research articles. From the past few decades, the production of research articles have been increased enormously. According to Larsen and Von [1], research articles have been doubled after every five years. This process of production of research articles have never been stopped, instead it is increased day by day [2]. Ware and Mabe [3] have reported in 2015, that almost 28,100 journals are publishing 2.5 million research articles every year. Peoples have an opportunity to search these articles over the internet using search engines, digital libraries and citation indexes etc. The huge amount of data on web hinders the recommender systems to extract the relevant research papers against the posed query. Typically, users explore different repositories to extract the relevant research papers, such as, Digital Library, Google Scholar etc. However, the existing data over the web is unstructured in nature that adversely affects the process of finding relevant information against the posed query. These systems return millions of generic hits in which some of them are related to the posed query. Let us consider an example, the query of "Ontology Engineering" was poses at 25/10/2019 on Google Scholar, it returns around 1.4 million relevant research papers. Reading all of these papers requires a lot of time. Almost 144 years are required to read 20 papers per day. This is due to the fact that papers over these repositories are not properly classified or indexed

according to their respective classes. We believe that performance of these systems can be enhanced if these papers are labeled to their respective domains. This extensive disorganization of research article has grabbed the attention of research community to classify the research article into its appropriate category/ies. The researchers have focused to classify the documents in such a way that guarantees to retrieve maximum relevant information [4]. Researchers have faced such a big challenge to classify the document into appropriate category/ies, due to presence of such huge amount of data present on the web.

For addressing the above problem, in 80's, document classification was based on human crafted rules to assign manually the predefined category to document. Afterwards in 90's machine learning approaches have performed extraordinary against manual systems, because these approaches automatically assign category via supervised learning. Until now, numerous ML approaches have performed document classification into its appropriate category[5][6][7]. From these approaches some of them have addressed the problem of research articles classification [4][8]. Every research article associates with one and more categories. The issue of mapping the research articles with associated categories can assist in multifarious aspects by helping scholars such as 1) Helping researchers to find the relevant documents to their topic 2) Finding relevant literature to narrate the background concept of the proposed study and so on. 3) Search engines and digital libraries returns relevant document for user queries. The research articles classification has mainly divided into two broad categories, 1) Content based approaches and 2) Meta data based approaches. Both are detailed explained in Chapter 2 of this thesis.

Normally the content based approaches produces good results as compared to metadata based approach due to richness of features[9][10][11]. However the main issues with content based approaches is that it is not freely available because the major journal publisher like ACM, IEEE etc. have not provided the access to overall content of the research article. In such scenarios some of the researchers have utilized a Meta data as an alternative way for classification of research articles [12][13][14]. So, one of the possible substitute of content based approach is

metadata based approach. Meta data is actually data about data. Meta data of the research articles contain tile, keywords, general term, authors, categories etc. and mostly these metadata are freely available online.

This thesis mainly focuses on research document classification in Computer Science domain by utilizing the freely available metadata of research articles. For addressing this issue, we have used these meta data, title, keywords, general terms individually as well as their combination, because each metadata parameter of research papers holds a significant potential and their collective contribution can beneficiate in improving the accuracy.

1.1 What is Classification

Classification is the process of sorting and categorizing data into various types, forms or any other distinct class which is already defined. The classification task is very well renowned in the area of machine learning. In various Machine Learning problems, the classification task is performed in the form of text classification, speech recognition etc [8][15][16]. From literature we have observed that, research article classification is very helpful in retrieving relevant documents against a posed query. Research article classification method is mainly composed of selecting the handy features from content which could help in assigning some suitable category to the research articles. Mainly, there are two types of classification. 1) Single label Classification (in which an item belongs to only one class, while there are two or more class's available), 2) Multi-label classification (in which an item belongs to more than one class). However, single research article could belongs to more than one category, this aspects diverts the attention of research community towards multi label classification of research articles. Most of the existing approaches produce low accuracy and authors have used a few numbers of classes. That why this is open research area to classify the document into multiple categories with high accuracy. This research work mainly focuses on multi label classification by using metadata features of research articles.

1.2 ACM Classification System

The Association for Computing Machinery (ACM), was founded in 1947 as the first scientific and educational society dedicated to computing. The first ACM classification system in the field of computer science was published in 1964. Later, in 1982, ACM published a completely new system. This was followed by new versions based on the 1982 system in 1983, 1987, 1991 and the latest in 1998 valid in 2007 [CCS 2008]. Afterword a new ACM system is developed in 2012 which were mapped on old schemes. The ACM Computing Classification System version 1998 (ACM) version 1998 or CCS98 classification system involves three Levels.

In Figure 1.1, first level contain topics from A (General Literature) to K (Computing Milieux) and it total count is 11. In second level, every topic of first level have subtopics. for example, first level topic A (General Literature)", there subtopic is A.0 (General), A.1 (Introductory and Survey), A.2 (Reference),.. . . .., A.m (Miscellaneous) topics and total count of topic in second level is 81. At third level every second level have further sub topics. For examples such as A.0.0 (Biographies / autobiographies), A.0.1 (Conference proceedings),. . . , A.0.2 (General literary works),.., C.2.m (Miscellaneous), and total count of third level topic is 400.

In computer science domain, in literature most of the conferences and journals have utilized the ACM classification hierarchy to categories their research articles into different categories. However, we have also utilized the root level categories of ACM hierarchy to categorize our research article into different categories of Computer Science domain. We have used 1998 version of ACM due to different reasons such as: 1) this version is utilized by many conferences and journal including ACM for categorizing their research articles into different predefined categories, and 2) One of another and big reason is that after evaluation of our proposed methodology we have compared our results with state-of-the-art approach which is proposed by with Ali and Asghar [17] which are also utilized the same version of ACM hierarchy to categorize their research article.

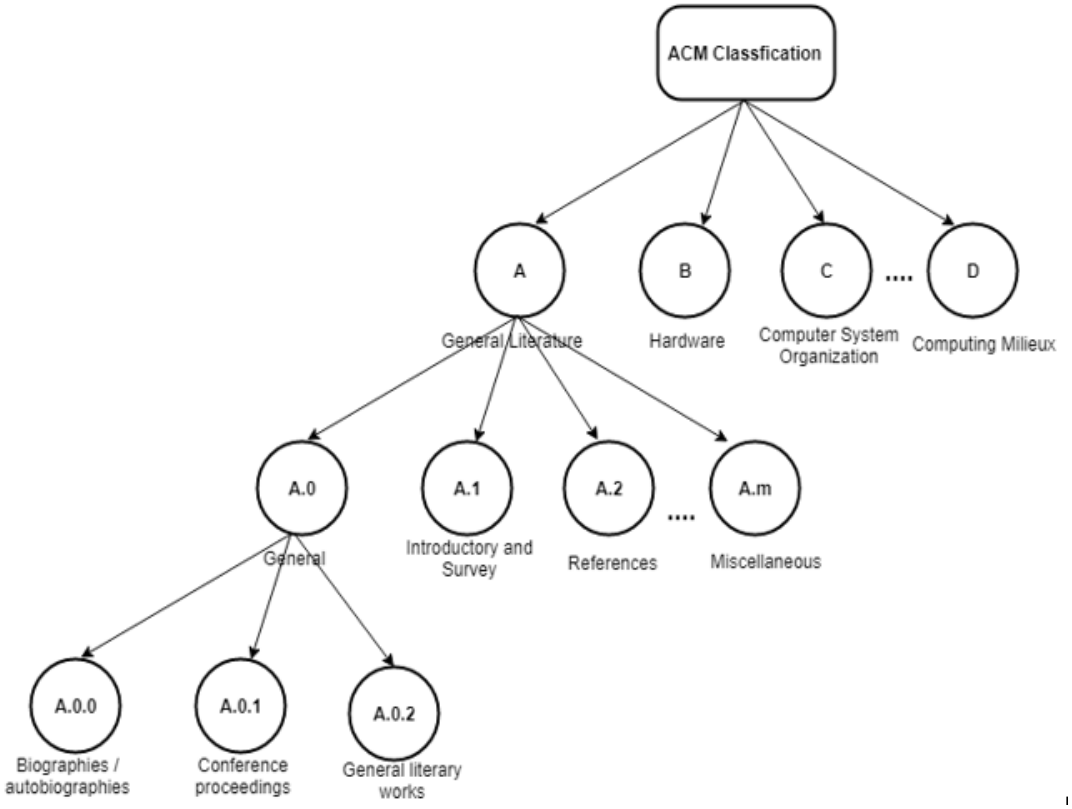

FIGURE 1.1: ACM Hierarchy.

1.3 Text Representation

Text representation is one of the fundamental problems in text mining and Information Retrieval (IR)[18]. It aims to numerically represent the unstructured text documents to make them mathematically computable. For the representation of a text document in numeric form, numerous techniques have been used in literature like Bag of Word (BOW), Term Frequency (TF), Term Frequency and Inverse Document frequency (TFIDF) etc. For capturing information these statistical measures totally rely on frequency of terms and ignored the semantic and context of term. The current state-of-the-art approaches[9][10][11][12][13][14] for research article classification have employed these conventional statistical measures like TF, BOW, and TFIDF etc. due to which they have ignored the semantic and contextual information of a terms and it might be assign a wrong categories to the research articles. This thesis have also focused on text representation, before

performing any mathematical operation like finding similarity between text document, the semantic and contextual information is considered in representation which is ignored by existing statistical measures.

To address the above mentioned issue we have to use some alternative technique for representation which considered the semantic and contextual information.In literature we have studied different semantic techniques for representation. One of the most well known technique which is used in different domains is word embedding[19][20][21]. Word embedding is used to represent document vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. For this, Word2Vec is one of the most popular technique to learn word embedding's using shallow neural network. It was developed by Mikolov et al. in 2013 at Google [22].These models are shallow, two-layer neural networks equipped to reconstruct linguistic contexts of words.Word2vec takes as its source a wide corpus of text and generates a vector space, usually several hundred dimensions, with a corresponding vector in space being allocated to each specific word in the corpus. Word vectors are placed in the space of the vector so that terms sharing common meanings in the corpus are located in space near to each other. Afterword, similarities score between identical words is greater than two different words which shows that these model also capture the semantic and context of a words. The detailed working of this model is explained in Chapter 3.

1.4 Similarity Threshold

In case of Multi label classification, for assigning multi labels to the documents we set some similarity threshold value which is the lower limit for identifying research article categories. The average similarity score of a test document with every category compared with similarity threshold value, all the categories having score higher than threshold value, selected as a final categories to the test document. In existing state-of-the-art [12][13][14], researchers have picked the method of selecting similarity threshold value either asking from domain experts or by choosing

some arbitrary values and then ensuring them on the basis of trial and error on dataset, which is a time consuming task. Dependence on domain experts or on some arbitrary value does not adequately serve the said purpose. The threshold value must be defined based on rigorous analysis of the dataset being employed. This thesis have also focused in this problem and proposed a method for defining threshold values based on a dataset which is explained in detail in Chapter 3.

1.5 Overview of State-of-the-Art Approaches

This section provides a brief overview of already proposed approaches that provide information about the current trends regarding research articles classification. These approaches are detail explained in Chapter 2. All these approaches are mainly divided into two categories, Content based approaches [9][10][11] and Metadata based approaches [17][13][14]. In Content based approaches the classification is mainly performed by exploiting the overall content of research articles. Usually the features of these approaches are in the form words, phrases etc. For features extraction and transformation different measures have been used by researches for identifying the important term in the text author have used the BOW, TF, TFIDF, for finding similarity between extracted word using cosine, Jaccard etc. The results obtained from these measures are than assign as an input to different ML algorithms, which predict the category of research article. In literatures some of the authors have used freely available metadata for research articles classification. The existing metadata based approaches classify research articles in respective categories with low accuracy. From the detailed study of literature review we have observed some aspects which led us to formulate our research problem.

- Most of the existing approaches perform classification using content based features for single label and multi label classification.
- There exists a very few number of approaches that have utilized metadata however these approaches have produced low accuracy.

- To the best of our knowledge that none of the existing study evaluated individual and collective contribution of metadata.
- Most of state-of-the-art approaches have used conventional statistical measures for text representation which only capture information based on frequency of term rather than semantic and context of a term.
- In case of multi label classification, state-of-the-art approaches have set threshold value either asking from domain expert or by choosing some arbitrary values which is ensured by trial and error.

1.6 Problem Statement

The current state-of-the-art delineates that the researches have proposed various techniques to automate the classification process of scientific research papers according to some predefined classes. We have performed an in-depth analysis of the literature and identified the following gaps:

1. To compute a similarity between textual documents most of the studies have utilized conventional statistical measures like TF, BOF, TFIDF etc. For textual representation, which only capture information based on frequency rather than semantics and context of terms used.
2. For Multi-label classification, in existing approaches, researchers have picked the method of selecting threshold value either asking from domain experts or by choosing some arbitrary values and then ensuring them on the basis of trial and error on dataset, which is a time consuming task. We argue that dependence on domain experts or on some arbitrary values does not adequately serve the said purpose.

1.7 Research Questions

We have formulated the following research questions relying on the problem statement describe above:

1. To what extent the semantic model can improve the accuracy of classification as compared to statistical measure on individual and combination of Meta data features?

2. How to define a threshold values for multi label classification?

1.8 Purpose

The purpose of this study is to identify whether a Semantic model (use for text representation) is helpful in classification of research papers into predefined classes or not. Moreover, there is a need to have an approach which identifies the threshold value by rigorous analysis of data for multi label classification.

1.9 Scope

This thesis focuses on mapping research articles belonging to the Computer Science domain into ACM category or categories at root level. The hierarchy of ACM is shown in the Figure 1.1. Moreover, for Computer Science research articles we have used JUCS[23] and ACM dataset[10]. The reason for the selection of these datasets (Details are shown in Table 3.1 and 3.2) is that it contains research publications from different conferences, Journals and the workshops of Computer Science domain.

1.10 Significance of Proposed Solution

This research will contribute for the classification of scientific documents (research papers) to the pre-defined ACM taxonomy. It is useful for number of systems such as: information retrieval, analyzing trends, finding experts, recommender systems, search engine and citation index. The accurate classification system can further assist authors of research papers in their paper submission process. Authors can find the category of their research contributions. Conference/Journal paper submission systems can assign categories to the research papers and assign reviewers to those papers. Researchers can easily find required papers, if there is an accurate taxonomy maintained for all research papers.

1.11 Definitions, Acronyms, and Abbreviations

- Association of Computing Machinery (ACM)
- Journal of Universal Computer Science (JUCS)
- Word2Vec (W2V)
- Term Frequency (TF)
- Bag of Word (BOW)
- Single Label Classification (SLC)
- Multi-Label Classification (MLC)

Chapter 2

Literature Review

Chapter 1 provides sufficient details on scenarios that guide us to define the problem statement. This chapter focuses on critical analysis of all the state-of-the-art approaches, as every research study is dependent on the previous study, that have already been performed in this field. The research community of document classification has proposed number of new ideas for document classification as the number of documents in digital form is increasing day by day. As earlier in 18th Century, when first document classification approach was proposed by the scientific community, the process started in different branches as a results attention of research community moved towards classification of specific type of documents such as: 1) Newspaper 2) Webpages etc[24][25][26]. Afterwards attention of research community moved towards research paper classification due to rapid invention in literature. The stat-of-the-art proposed approaches in literature can be divided in to two broad categories.

1. Content based approaches
2. Metadata based approaches

The Content based approaches have mostly focused on the overall content of the research article while the metadata based approaches have focused on metadata of the research article. Both are explained in detailed in below sections.

2.1 Content Based Approaches

Content based approaches mostly depend on content of research articles. This is due to the fact that content include abundance features. This section elaborates the state-of-the-art content-based approaches:

In 2016, Tang et al. [26] proposed Bayesian classification approach for text classification by analyzing different content-based features. In this approach they used specific features for each class instead of using global features for all classes. They built rules for classification by using Baggenstoss's PDF Project Theorem for each specific class features. This approach has been tested on two different datasets such as: 1) "20-NEWSGROUPS" 2) REUTERS. The outcome of the study revealed that on "20- NEWSGROUPS" their approach performed extraordinary; however, they have not mentioned exact evaluation results. They further suggested that when they selected more features, their approach achieved small improvement in results. They also proposed another framework for feature selection [27]. In this approach they have ranked all the selected features. They came up with a new divergence measure Jeffrey's-Multi-Hypothesis (JMH) to measure multi distribution divergence for multi-label classification. They tested their approach on three datasets such as: 1) 20-NEWSGROUPS 2) REUTERS 3) TOPIC DETECTION AND TRACKING (TDT2). By using JMH they developed two feature selection method, Maximum discrimination (MD) and MD X2 for text classification which have performed well by achieving 0.95 accuracy and 0.90 F-Measures.

In 2015, Le et al. [28] performed survey on all existing feature selection approaches for text classification. In this survey, they discussed all method of feature selection and feature reduction. They categorized all the method into two broad categories 1) wrapper 2) filter. The performance of filter method is significantly better than wrapper method because filter does not depend on classification algorithm. In literature mostly researchers used the filter technique for text classification.

In 2016, Zhou et al. [29] proposed a content-based approach using naïve Bayes and Logistic regression algorithms. They used two diversified datasets from computer science domain which have already annotated such as: 1) CiteSeerX, 2) arXiv.

Their proposed classifier works on different content-based features, one of them is bigram feature which performed extraordinary on both datasets. The approach achieved F1 Score on arXiv and CiteSeerX datasets are 0.95 and 0.75 respectively. They concluded the results of CiteSeerX dataset show that their annotation is not as much accurate as arXiv dataset.

In 2015, Zhong et al.[30] proposed a similar approach like above but their approach focused on semantic similarity on different features for classification of text. They performed an experiment on two different datasets such as 1) Routers-10 2) 20-Newsgroups. They conducted a series of experiment on these two datasets by applying Support Vector Machine algorithm (SVM) and achieved F-Score of 0.76 for 20-Newsgroups and 0.91 for Router datasets.

In 2012, Chekima et al. [31] proposed document categorizer agent based on Naïve Bayes Classifier. In this approach, they trained the categorizer agent by using research papers downloaded from ACM. In training, they mainly used two types of data, negative examples and positive examples. After performing multiple experiment, the categorizer agent proved that it performs remarkably well and achieved high accuracy in case of Computer Science papers. After performing experiment on 1000 Computer Science papers, 91 percent of the time the approach has correctly categorized. The outcome revealed that the proposed algorithm has to be used for the purpose of categorizing the document, however result would be improved if they will increase the training and testing data.

Another similar approach to Khalifa [31], in 2007, Wang and Desai [11] proposed hierarchical text classifier for the experimental CINDI Digital library. This study has been proposed to control selected categories, and inspect two different methods for ranking categories at same level. The system was evaluated on self-generated corpus from ACM digital library. After experiment it has been found that Naive Bayes based classifiers performed well as compared to Centroid based classifiers. Moreover, for ranking categories, addition scheme produces better results than multiplication scheme.

Similar hierarchical approach was proposed by Cai and Hofmann [32] to classify text documents on the basis of SVM classifier. The approach mainly integrates all

the previous knowledge about class relationship which is expressed in the form of hierarchy. The approach has been evaluated using WIPO-alpha Collection dataset. The evaluation results revealed that the proposed approach performed significantly better than previous ones.

Another approach of hierarchical multi-label text classification has been proposed by Baker and Korhonen [33], in which neural network model is defined. The results have been evaluated by using biomedical field data. The experiment has been performed on both sentence level as well as document level classification. After performing detailed experiments, it has been concluded that document level classification performs better than sentence level classification.

In 2008, Kannan and Ramaraj [9], developed a system for text classification based on similarity of text. The approach has been designed for limited set of datasets. In this approach feature selection framework has been presented in which Information Gain (IG) Score used for every word which is used for text classification. Authors have also presented the initial learning model; in which unlabeled document has been randomly selected which were than annotated by field experts. The approach has been tested on Reuter dataset which contains almost 21578 documents. After conducting extensive experiment, it was identified that on sample of 2000 document their approach attained improved value of F-Measure 0.90. Moreover, the outcome of the study also reported that by reducing vocabulary size, the rate of classification increases.

One of another approach for text classification has been presented by Goller et al [34]. In this approach, authors thoroughly evaluated different previous approaches for Germen text classification. The approach has been evaluated using different feature selection method and different classifiers. After comprehensive evaluation, it has been found that that feature selection and reduction of dimension is very important for classification task and also for reducing overfitting problem. Moreover, the outcome also signifies that SVM classifier is best for text classification.

Another technique for annotating research paper on the basis of key phrases proposed by Chernyak [35]. These key phrases were collected from ACM Computing Classification System. For finding relevancy, they used different phrase to text

relevance measure, in which most relevance phrase goes for annotation. Authors have employed three measures to evaluate the proposed approach: i) Cosine Relevance score ii) BM25 iii) CPAMF. The approach has been evaluated on the dataset collected from ACM digital library. After comprehensive evaluation the outcome yielded that CPAMF performed extraordinary as compared to Cosine Relevance and BM25.

For multi-label text classification Wang et al. [36], presented approach that combines the two different classifiers, Random forest and semantic core co-occurrence latent semantic vector space (CLSVSM). In this ensemble-based approach these two classifiers selected due to following reasons. Random forest solves binary classification problem, and CLSVSM show the semantic of the text. Both of these reasons are very helpful in classification task. The proposed approach has been evaluated by using yahoo dataset. The outcome of the study revealed that the proposed has attained significantly better results than existing state-of-the-art approaches.

Nanba et al. [37] presented a study that classifies research papers using citation links and contents-based features. Based on this study, authors have developed research papers classification tool named PRESRI. PRESRI tools mainly locates the citing areas and identifies the relationship, which was letter used for classification purpose. The tool employs authors' name or title words-based features. PRESRI'S current version takes the features into account and categorizes the papers based upon the cited paper mentioned in the bibliography of the query paper. For evaluation and experimental purpose, they used 395 papers collection.

Taheriyan [38], proposed an approach which performed subject classification of scientific articles based on analysis of their interrelationship. The study has employed citations, common author and common references-based parameters. To do this, a relationship graph has been formed where research papers have been presented by nodes and link among those nodes determine the relationship between papers. For evaluation, the research papers dataset has been collected from ACM digital library. The outcomes of the study revealed that better results are produced against dense and close-packed graphs.

Sajid et al. [7], have proposed an approach for research paper classification based on references section of research paper. The approach exploits references segment of a research paper to locate topics of the paper. The study follows an assumption that most of time, authors cite the papers belonging to the same domain or similar category. To validate the claim, authors have employed a data set from Journal of Universal Computer Science (JUCS). They selected this dataset because it covers all the Computer Science areas. In this approach the stored references in the database have been matched with the extracted references of the paper. After performing experiment, authors reported their accuracy up to 0.70.

One of another content-based approach was proposed by Santos and Rodrigues [10]. The approach comprised of two main steps, 1) Create a dataset of a document in the form of multi-label hierarchy, these documents were extracted from ACM digital library. 2) Developed a methodology for multi-label text classification by combining various classification algorithms. The approach has utilized title, abstract and keywords as a feature for multi-label document classification. The approach has utilized different classification algorithm, like Binary relevance, Label Power set, sequential minimal optimization and Naive Bayes Multinomial etc. After conducting comprehensive experiment, the results revealed that Binary relevance combined with Naive Bayes Multinomial perform extraordinary and achieve 0.88 f-measure as compared to other classifier they used individual as well as combined.

For the classification of research articles, Balys and Rudzkis in [39] proposed an automatic classification method. The method is based on statically analysis of probabilistic distributions of scientific terms in texts. In this approach comparison has been performed between different classifier like Knn, SVM, LLSF and IDC. After the evaluation results, revealed that the influence of IDC algorithm is more than any other classifier. Another similar work in which Cunningham [40] has focused on machine learning algorithms to develop subject classification rules for documents. The documents used in this approach are from different fields like neural network and machine learning. In most of the cases such like approaches

focuses on overall content of the paper which consumed a lot of time. As aforementioned, one of the strengths of this work is that it saves the time to process full text of papers.

Jindal and Shweta [41], have proposed a method for Efficient Multi-label Text categorization of research articles. The approach used the concept of lexical and semantics to solve the problem of multi-label categorization of text documents. In lexical analysis steps tokens have been identified from research articles on the basis of IEEE taxonomy. In Semantic Analysis step, relationships between the tokens are analyzed using the standard lexical database of words, i.e. WordNet. In next step approach perform classification, in which classes of tokens are determined using IEEE taxonomy. The approach has been evaluated on 150 papers of computer Science domain. The outcome of the study revealed that their approach achieved accuracy upto 0.75.

Scientific community has mainly focused on content-based approaches. The main reasons of their attention towards content-based approaches is that it contains a lot of features and produced such remarkable results. However for the application of these approaches we must have the availability of the content of the research articles which is not possible all the time because famous digital libraries like ACM, IEEE, and Springer provide subscription based services. To overcome this issue, research community has found some alternative ways to classify research document, when content of the document is not available. Those ways contain useful aspect of research paper which is metadata. Metadata of research paper includes author, title, keywords, general terms etc. until now the approaches which have used the metadata of research document for the classification purpose are very few in numbers. The following section delineates the existing metadata based approaches:

Metadata of research paper includes author, title, keywords, general terms etc. until now the approaches which have used the metadata of research document for the classification purpose are very few in numbers. The following section delineates the existing metadata based approaches: The following section delineates the existing metadata based approaches:

2.2 Metadata Based Approaches

The existing metadata based approaches uses metadata of research articles for classification of research document task. In Metadata of research document includes title, author, keywords, general terms, categories etc. This type of metadata is almost freely available, while the whole content of the data is not freely available online. So that is the big motivation for the research community to move from content to freely available metadata of the research documents. Now in this section, we have briefly highlighted some metadata-based approaches.

Yohan et al.[42] proposed a technique using natural language processing for finding name entities and classified them in their respective categories. The approach generate rules for the identifying name entities and its classification, specifically for Teluge language. For recognizing name entities and its classification, the approach exploits word, work lookup and contextual based features. The approach has comprehensively been evaluated using different Newspaper and Teluguwiki datasets. Moreover the evaluation has been perform using full sentences. The results show that their approach achieved precisions in range of 0.79 to 0.94.

For document classification, another metadata extraction approach was proposed by Flynn [13]. The approach proposed the "post hoc" system for categorizing the documents. This approach has been divided into two phases.1) First they have extracted metadata based on template, 2) Secondly, they have performed classification on the basis of these extracted metadata. For an evaluation, the approach selected the diversified dataset of defense technical information center (DTIC), which contain one million data like scientific articles, PHD thesis, research papers of conferences, journals, slides and law document etc. After performing extensive experiment on this dataset, the results revealed that this approach has 0.83 time correctly categorized the document.

In another study, Bayesian based approach has been presented by khor and Ting [43] to classify research papers. In this study, 400 research papers from education conference have been considered as a data set and mapped to four different classes including e-learning, cognition issues, teacher instruction and intelligent coaching

system. The researchers have contended that there are keywords traits that can be used for categorizing the papers. The approach used a features selection algorithm to extract the keywords related to each topic. The approach is solely based on keywords-based features.

For improvement in classification, Zhang [12] proposed a method based on structural and citation-based information. In this approach, they combine the structural information (title, abstract) with citation of research paper for some big achievement in document classification. They evaluated different similarity measures on basis of citation structure and structure content of the papers. The approach has been evaluated on the dataset collected from ACM digital library. This approach used genetic programing-based approach for the classification of research document in to ACM subject hierarchy. They have reported that their approach works better as compare to traditional content based SVM approach.

Sajid et al.[14] proposed fuzzy logic-based classifier for the classification of research paper in Computer Science domain. As the paper does not belong to only one class, therefore, they used fuzzy logic, and proposed fuzzy based rule merger algorithm to merge the generated rules and fuzzy classifier to classify the paper into respective single and more categories. For experimental purpose they have selected the JUCS datasets because its covers all areas of Computer Science domain. From this dataset they have extracted title and keywords, which were used as a feature for papers classification. After performing detailed evaluation of the approach, the results revealed that the approach achieved 0.93 precision and 0.96 F measure but they have used single label classification measures.

Ali and Asghar [17], proposed multi-label scientific Document Classification based on metadata features. The approach utilized two metadata features (title and keywords). For performing multi-label classification the approach first convert the data into single label classification by using four different conversion techniques (Min, Max, Ran, and Single). The approach also used different similarity measures for finding the relevancy between documents and labels. The approach have used PSO based classifier for the classification of documents. The approach has been evaluated on two different dataset of research articles (JUCS and ACM). The

outcome of the study revealed that their approach achieved accuracy up to 0.78. For the solution of document classification problem, the research community has mainly focused on two types of features:

1. Content based approaches
2. Metadata based approaches

The content-based features mainly focuses on overall content of research papers. For an implementation of these techniques, the paper content is very necessary ingredient. These technique mostly provides better results because it contains a great number of features, however mostly the whole content of the research paper is not freely available. On contrary metadata-based approaches used by a limited number of researchers because it does not contain a great number of feature and mostly produces low accuracy as compare to content-based approaches. However benefits of metadata are that it is freely available in many digital libraries like IEEE, ACM etc.

2.3 Evaluation Criteria

After comprehensively studying literature related to document classification, what we get from these? For the explanation of our finding from the literature review is presented on the basis of different parameter, like types of data, type of classification, dataset, algorithm, results and limitation. In type of data, we tell about the approaches that which data source is used for classification, either they have used content or metadata of documents. In type of classification, we tell about which type of classification they has been performed, single label or multi-label. In dataset, we tell about which dataset has been used by the approach and also presented the quantity of a dataset. In algorithm, we tell about which strategy/algorithm/methodology, has been used by the approach to classify the documents. In results, we have presented the results as reported by the researchers in the form

of different evaluation measures like accuracy, precision, recall, f-score and Hamming Loss. In limitation, we have described the main drawback of the approaches that have been observed based on comprehensive analysis literature.

2.3.1 Analysis of Literature Review based on Evaluation Criteria

After detailed analysis of all the above-mentioned approaches, it was examined that these approaches mainly used content based and metadata-based data sources. On the basis of this observation we divide the already proposed approaches into two broad categories 1) Content based approaches and 2) Metadata based approaches. The content based approaches uses the overall text of the document and classify the documents into respective single or multiple classes from which they belong while the Metadata based approaches uses the metadata of the research documents for classification.

2.3.1.1 Critical Review of Content Based Approaches

In Content based approaches the researchers utilized the overall content of the document for the purpose of classification of document into single class or multiple classes from the given number of classes. For this purpose, researchers proposed different algorithms and used different datasets. In many approaches 20-Newspapers and Routers dataset were used [26][30]. Many other datasets were also incorporated by different researchers [33][10][29]. These approaches have used different measures like accuracy, precision, recall and f-score for evaluation purpose. In case of single label classification, they reported accuracy up to 0.95, precision reported up to 0.8, recall reported up to 0.76 and f-score reported up to 0.94. On the other hand, in multi-label classification some new datasets have also been used like, JUCS, WIPO-alpha, Enzyme etc. They reported accuracy up to 0.94 and precision up to 0.84. One of technique document classification technique which was proposed by Santos [10] for multi-label classification. For evaluation

purpose, Santos approaches utilized ACM dataset, and classify the documents into root level of ACM hierarchy (11 Classes). They reported the accuracy up to 0.88. The results reported by these approaches were extraordinary. The main reason of such a tremendous result is that the content of documents contains a lot of features as compare to metadata.

2.3.1.2 Critical Review of Metadata Based Features

In literature, very few of the researchers used the metadata for the purpose of document classification [13][43][14]. After critical analysis of these approaches we have examined that different approaches have used the different number of classes' data for the evaluation. E.g Flynn [13] used 99 predefined classes are used and they reported 0.79, 0.81, 0.79 precision, recall and f-measure respectively. Khor [43] used 4 generic classes' data and performed document classification and reported accuracy of their proposed algorithm is 0.84. Ali [17] used 11 classes of root level of ACM hierarchy and performed multi-label classification of research article and reported results up to 0.78.Moreover, based on critical analysis of literature, it has also been revealed that most of the metadata based approaches have performed single label classification.

The brief overview of most related technique to our work is described in Table 2.1 below along with their results and limitations.

TABLE 2.1: Critical Analysis Table of Literature

Ref	**Dataset, Classification types, Types of Data**	**Representation Technique, Algorithm**	**Result**	**Limitation**
[9]	Content, Single label	Term Frequency, Cosine Similarity,	Accuracy 90%	Content dependent,
Continued on next page				

Table 2.1 – continued from previous page

Ref	Dataset, Classification types, Types of Data	Representation Technique, Algorithm	Result	Limitation
	2000 papers	Base Algorithm		Limited to Singlelabel Classification, Cannot capture the meaning and contextual information of the terms
[10]	Content, Multi-label 5000 and 10000 Papers	TFIDF, Binary Relavance, Naie bayes Multinomial, Multi-label KNN	Accuracy 88%	Content dependent, Cannot capture the meaning and contextual information of the terms
[7]	Content, Multi-label, 1460 papers	TFIDF, Static Threshold, Citation Based category identification	Accuracy 70%	Limited to reference section, Use Static Threshold, Cannot capture the meaning

Continued on next page

Table 2.1 – continued from previous page

Ref	Dataset, Classification types, Types of Data	Representation Technique, Algorithm	Result	Limitation
				and contextual information of the terms
[37]	Content, Single label, 395 papers	Statistical measures, BCCT-C PRESRI using Citations, Cosine Similarity	Precisions 83%	Limited dataset, Content dependent, Cannot capture the meaning and contextual information of the terms
[38]	Content, Single label, 400 papers	Statistical measures, Analysis of Interrelationships of authors, references and citations	Precisions 90%	Limited dataset, Limited to single label Classification, Cannot capture the meaning and contextual information of the terms
[43]	Metadata, Single label,	Term Frequency, Bayesian Network(BN),	Accuracy 83.75%	Limited dataset,

Continued on next page

Table 2.1 – continued from previous page

Ref	Dataset, Classification types, Types of Data	Representation Technique, Algorithm	Result	Limitation
	400 papers	Naïve Bayes (NB)		Limited to single label Classification, Cannot capture the meaning and contextual information of the terms
[13]	Metadata, Single label, 2000 papers	Statistical measures, Independent Document Model (IDM) Framework	Accuracy 81%	Limited to Single label Classification, Cannot capture the meaning and contextual information of the terms
[7]	Metadata, Multi-label, 1460 and 86116 Papers	TFIDF, static threshold, Similarity Measures, PSO based Classifier	Accuracy 78.79%, 77.86	Use Statistical features, static Threshold, Cannot capture the meaning and contextual

Continued on next page

Table 2.1 – continued from previous page

Ref	Dataset, Classification types, Types of Data	Representation Technique, Algorithm	Result	Limitation
				information of the terms
[41]	Content, Multi-label, 150 papers	Term Frequency, Static threshold, Lexical and Semantic Concepts,	Accuracy 75%	Limited dataset, Static Threshold, Cannot capture the meaning and contextual information of the terms

After the critical analysis of the document classification domain, it was observed that some authors utilized the metadata of the documents but most of the authors have employed the contents of the documents. Content-based technique mainly focuses on overall content of research papers like abstract, introduction, methodology etc. To implement these techniques, content of the papers plays a significant role. Content-based approaches mostly produce results with good accuracy because diversified features can be extracted from the content. However, most of the time content of the research papers is not freely available. A survey conducted in study [44], reported that the 50(%) of research articles were accessed openly. Journals of all major publishers like IEEE, ACM, Springer, Elsevier and IOS do not provide open access to their articles. There are financial, legal and technical

barriers to access content of the paper. On contrary, metadata-based approaches have been employed by very few number of studies because Metadata does not contain a rich number of feature and mostly produces low accuracy as compare to content-based approaches. However, some of the Metadata play a pivotal role in classifying research articles such as, title, keywords, general terms etc each of this metadata serve a specific purpose that can be exploited to classify research papers into different number of categories. Moreover, Metadata of the research articles is freely available in many digital libraries like IEEE, ACM etc.

Moreover, in various classification problems, representation of a text is a very crucial steps in finding similarity between text documents. The current state-of-the-art depict that most of the existing studies have employed conventional statistical measures like TF, BOF, TFIDF etc. These measures mostly capture information using the frequency of terms. We argue that before computing the similarity between textual documents, the semantics of a text must also be considered which is ignored by existing statistical measures.

Further, in the area of research papers classification most of the approaches perform single label classification and their exist a very few approaches that have performed Multi-label classification. Multi-label classification based approaches have employed a static threshold values. In existing state-of-the art, researchers have picked the method of selecting threshold value either asking from domain expert or by choosing some arbitrary values and then ensuring them on the basis of trial and error on dataset, which is a time consuming task. We argue that dependence on domain experts or on some arbitrary value does not adequately serve the said purpose. We claim that threshold value should be defined based on rigorous analysis of the data set being employed. In this thesis, our aim is to classify research articles using metadata as individual features as well as their possible combination. We used word2vec model to store information semantically and contextually in a text representation. We have to find a threshold for multi-label classification by conducting an in-depth analysis of the data. For comparing our technique we have used the approach of Khor and Ting[43] for SLC and Ali and Asghar[17] for MLC. Khor and Ting have reported 0.83 average accuracy using

meta data features while Ali and Asghar have performed multi-label classification using ACM hierarchy. Their research work is little bit similar to my work. They used ACM and JUCS datasets. They performed multi-label classification. Their approach scores up to 78(%) on JUCS dataset and 77(%) on ACM dataset. My focus is to use freely available Metadata individually as well as its combination for the classification of the research document and use a semantic model for text representation to achieve a good result as compare to Ali and Asghar [17]. We will be using the above mentioned approaches for comparisons.

Chapter 3

Research Methodology

The critical analysis of already proposed approaches in previous chapter delineates that the research article classification community has proposed different techniques to classify the research articles into single and multiple categories. The primary observations from literature review that motivated and signify our proposed framework are as follows: 1) To the best of our knowledge, there does not exist any study that has comprehensively evaluated the freely available metadata individually and its possible combination, 2) there does not exist any study that has used semantic model for text representation that consider context and semantic of a term, 3) In case of multi-label classification, no study exist which has identified the threshold value by rigorous analysis of data. These observations have led us to propose a technique to address the issues discuss above. The proposed framework performed single and multi-label classification of research articles into predefined ACM hierarchy by comprehensively evaluating the metadata features (individually as well as its combination). In this chapter, the proposed methodology is described for the classification of research articles. The figure 3.1 is a graphical representation of our proposed technique. Each step of proposed technique is described in the following sections.

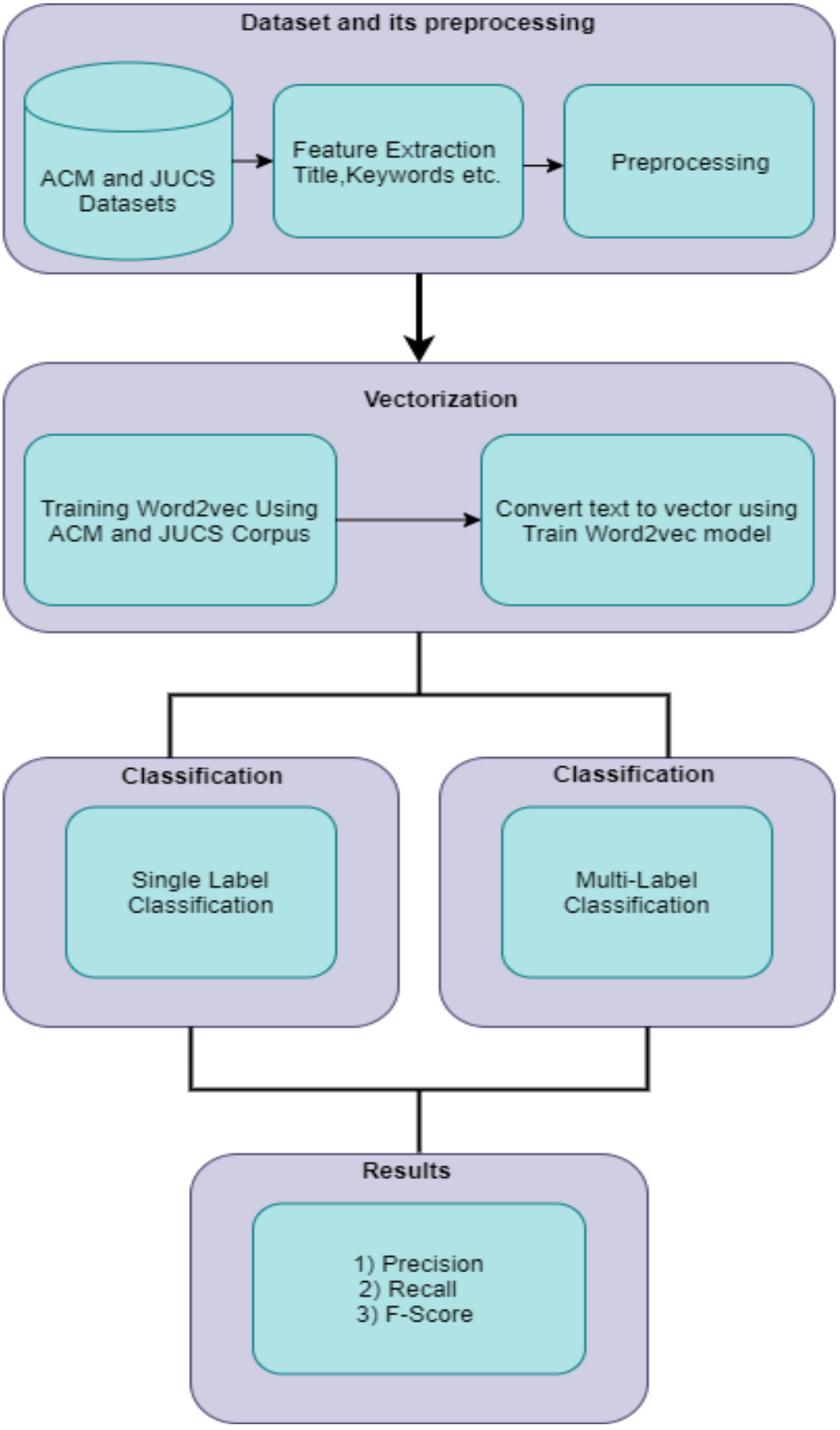

FIGURE 3.1: Context Diagram of Proposed System

3.1 Datasets

For comprehensive evaluation of our proposed system, the selection of datasets is a very crucial step. To evaluate our proposed technique, we have carefully selected two diversified datasets that contain research articles of Computer Science domain. The first one contains research articles from Journal of Universal Computer Science (JUCS)[23], and the second datasets, contains research articles from Association of Computing Machinery (ACM) which is developed by Santos [10] in 2009. The reason behind the selection of JUCS dataset is that, it contains papers from multiple areas of Computer Science domain which plays a significant role in comprehensively evaluation. Similarly, the reasons behind the selection of ACM dataset is that, it contains papers which belong from different journals, conferences and workshops. The detailed description about both datasets are given below:

3.1.1 JUCS Dataset

The dataset of JUCS contains almost 1460 papers of 13 different categories of Computer Science domain. JUCS has extended the categories of ACM from 11 to 13 by adding L (Science and Technology Learning) and M (Knowledge Management) categories. Therefore, its root level contains 13 categories. For the evaluation of our technique we have selected data of 11 categories from dataset because we have used root level categories of ACM hierarchy for classification of research articles. The 51% of data in dataset has been classified into single label which is used for single label classification and 49% of a data has been classified into multi-label data which is used for multi-label classification of research articles. The detailed statistics of dataset are presented in Table 3.1.

TABLE 3.1: JUCS Dataset Statistics

Features	Records
Total Number of Research Papers	1460
Total Number of Journals or Conferences or Workshops	1
Single-Label Research Papers Percentage (%)	51 (%)
Multi-Label Research Papers Percentage (%)	49 (%)
Total Number of Classes (Categories) at Root Level	13 (11 Selected)
Name of Metadata of research paper	Title, keywords, Categories
Number of records of A to K categories are:	A(35),B(50),C(123),D(311) E(55),F(302),G(110),H(380), I(235),J(86),K(149)

3.1.2 ACM Dataset

The ACM dataset is constructed by Santos and Rodrigues [10] in 2009. The dataset contains more than 86000 research articles. These papers have been published in different journals, conferences and workshops of diversified domains. The 54% of research articles in dataset has been classified into single label which is employed in single label classification and 46% of research articles has been classified into multi-label which is employed in multi-label classification of research articles. The dataset contains various metadata parameter of the research articles such as title, keywords, general terms etc. The detailed statistics of dataset are presented in Table 3.2.

TABLE 3.2: ACM Dataset Statistics

Features	Records
Total Number of Research Papers	86,116
Total Number of Journals or Conferences or Workshops	2,240
Single-Label Research Papers Percentage (%)	54 (%)
Multi-Label Research Papers Percentage (%)	46 (%)
Total Number of Classes (Categories) at Root Level	11
Name of Metadata of research paper	Title, keywords,General Terms Author Name and Categories
Number of records of A to K categories are:	A(644),B(5723),C(8735), D(17628),E(539),F(6257) ,G(3616),H(17845),I(15099) J(1343),K(9908)

3.2 Features Extraction and Combinations

The JUCS and ACM datasets contain the metadata of the research articles. From these datasets we have extracted the metadata of the research articles and made all the possible combinations of these selected metadata like Title, keywords, General Terms and categories. The selection of specific metadata's as a feature is based on the following reasons:

- The title of paper holds potential terms that can assist in determining the category of research article.
- Keywords and general terms are explicitly assigned by the actual authors of papers that are mostly from relevant areas.

From JUCS dataset we have selected two metadata such as: 1) Title and 2) Keywords due to free availability of these metadata in JUCS dataset. Similarly,

from ACM dataset we have selected three metadata such as: 1) Title 2) Keywords and 3) General Terms. The overview of extracted metadata parameters of JUCS and ACM is described in the Figure 3.2 and 3.3 respectively.

Afterwards to comprehensively evaluate all the metadata features we have formed all the possible combinations of these metadata features of both datasets by using the following algorithm presented in the Figure 3.4. The overview of possible combination of metadata parameters of JUCS and ACM is described in the Table 3.3:

The figure 3.4 represents procedure for extraction and making all possible combinations of metadata. In step 1, define an array in which we assign all the metadata

	A	B	C
1	**Title**	**Keywords**	**Categories**
2	Integration of Communities into Process-C	cooperative knowledge generation, kn	H
3	Small Groups Learning Synchronously Onli	professional,training,workplace,learni	HJ
4	Using weblogs for knowledge sharing and l	Experience,based,Information,System	ADHJK
5	Modelling and Implementing Pre-built Infc	modelling,method,introduction,methc	HIJ
6	Tube Map Visualization: Evaluation of a Nc	knowledge,visualization,information,\	H
7	Reconciling Knowledge Management and	workflow,knowledge,management	H
8	A Methodology and a Toolkit that Integrate	knowledge,management,knowledge,r	CI
9	KMDL - Capturing, Analysing and Improving	Process,oriented,Knowledge,Manager	DHI
10	The Role of Knowledge Management Solut	knowledge,management,business,pro	AH

FIGURE 3.2: Extracted Metadata of JUCS Dataset

	A	B	C	D
1	**Title**	**Keywords**	**General Terms**	**Categories**
2	Is a bot at the controls?:	online games cheat de	Security	C
3	Voronoi-based adaptive	peer-to-peer Voronoi	Performance Design	C
4	Wildlife net-gamekeepe	game modification na	Management Perfor	CK
5	ARMA 11 modeling of Q	online games simulati	Performance Theory	C
6	Adaptive Delta;-causalit	causality control consi	Performance Algorit	HK
7	An immersive voice ove	auditory scene creatic	Performance Algorit	CG
8	Synchronization mediun	communication abstra	Algorithms Design	D
9	Enhanced mirrored serv	peer-to-peer cheating	Performance Algorit	C
10	Performance analysis of	machine learning onli	Management Perfor	C

FIGURE 3.3: Extracted Metadata of ACM Dataset

TABLE 3.3: Metadata and its Combinations

Datasets	Uni Features	Bi Features	Tri Features
JUCS Dataset	1) Title 2) Keywords	1)Title and Keywords	No Tri Features
ACM Dataset	1)Title 2)Keywords 3) General Terms	1)Title and Keywords 2)Title and General Terms 3) Keywords and General Terms	3)Title, Keywords, and General Terms

Algorithm 1 : Extraction and making Combination of Metadata Procedure

Input: Raw Dataset of Research Paper Metadata
Output: All Possible Combination of Metadata

1: $Attribute \leftarrow Assign\ Metadata\ Names \qquad (Title, Keywordsetc....)$
2: $Records \leftarrow Retrived\ Reocrds\ from\ dataset\ of\ Attribute$
3: **for** $i = l$ to $len(Attribute)$ **do**
4: $\quad Combs \leftarrow Combinations(Records, i)$
5: $\quad$ **for all** $comb$ in $Combs$ **do**
6: $\quad\quad File \leftarrow Concatenate(comb, Dataset['Label'])$
7: $\quad$ **end for**
8: **end for**

FIGURE 3.4: Extraction and making Combination of Metadata Procedure

name (in our case names is: Title, Keywords and General Terms). In step 2, we have retrieved record against every metadata and stored in the list. In step 3, iterate the attribute array upto its length (in our case length is 3). In step 4, the combination function form all the possible combinations depending on the value of i, if the value i is 1 than the algorithm form all possible combination using one metadata parameter at a time and save it in a file using step 5 and 6 and map them with their respective label, if the value is 2 than the algorithm form all possible combination using two metadata parameters.

3.3 Pre-Processing

Preprocessing is a data mining technique that involves transforming dataset into own understandable format. Generally, datasets are incomplete: lacking attribute values (Missing Value), containing noisy data (meaningless data) etc. For pre-processing, we have performed different steps such as 1) Tokenization 2) Noise Removal 3) Stop word's Removal 3) Stemming. Let us discuss these steps one by one:

3.3.1 Tokenization

Tokenization is the first step of preprocessing. In this process, text can be divided into a set of meaningful pieces. These pieces are called tokens. For example, we can divide a chunk of text into words, or we can divide it into sentences. Depending on the task at hand, we can define our own conditions to divide the input text into meaningful tokens. In our scenario, we have divided the sentences into words. For this, we have used the Natural Language ToolKit (NLTK[1]), which is the best known and most used Natural language processing (NLP) library[45].

3.3.2 Noise Removal

Removing of noise from data is important because it can adversely affects the accuracy. Generally, the datasets contain noise such as: 1) Null values 2) Unnecessary punctuation's[2].Their exists various methods to remove noise such as: 1) Ignoring the missing record, 2) Filling the missing values manually, 3) Filling using computed values. We have very limited number of instances which contain missing values however, we have ignored these instances as it is the simplest and efficient method for handling the missing data. After tokenization, the some of the

[1]https://www.nltk.org/
[2]https://www.geeksforgeeks.org/python-string-isalpha-application/

punctuations can be considered as tokens which can be unnecessary (not meaningful) and can misguide us. Therefore, we have removed all of these unnecessary punctuations by using NLTK library.

3.3.3 Stop Word's Removal

Stop words are the most common words in a language such as: top 25 stop words are (a, an, and, are, as, at, be, by, for, from, has, he, in, is, its, of, on, that, the, to, was, were, will, with)[3]. These words do not carry important meaning so they must be excluded from the document to achieve accurate measurement. To remove stop words from all metadata parameters of a datasets, we have used NLTK library because it contains list of stop words. NLTK matches its own list of stop words with the tokenized list and then performed stop word removal from the corpus.

3.3.4 Stemming

Stemming is the process of reducing the words to their base or root words. The advantage of stemming is that it reduces the size of vocabulary. For example all these words, "consult", "consultant", "consulting", "consultative", "consultants" and "consulting", are stemmed into their root word "consult". We have performed stemming by using porter stemmer algorithm (Porter, 1980)[4], which converts all the terms of a text into their root terms. The stemming algorithm is applied on all the metadata of both datasets.

3.4 Vectorization

Most of the similarity measures and machine learning algorithms often take numeric vector as an input. However, before performing any operation on a text,

[3]https://gist.github.com/sebleier/554280
[4]http://snowball.tartarus.org/algorithms/porter/stemmer.html

we need a way to convert each document into a numeric vectors. This is one of the fundamental problem in data mining, which aims to numerically represent the unstructured text documents to make them mathematically computable. For this, numerous techniques have been presented in literature. These technique are mainly divided into two broad Category such as Count based approaches and Semantic based approaches. Some of the widely used count based techniques in research articles classification approaches are: 1) One Hot Encoding 2) Bag of Word (BOW) or Term Frequency (TF), 3) Term Frequency and Inverse Document frequency (TFIDF) etc, and semantic based approaches are: 1) Glove 2)FastText and 3)Word2Vec. Let us discuss their merits and demertis to be able to select the best approach for the implementation of the proposed model:

3.4.1 Count Based Approaches

3.4.1.1 One Hot Encoding

One-hot encoding is the most common and a most basic way to turn a texts into a vector. In this strategy, each word is converted into a binary value 1 or 0, which indicate the word appear in a document or not. Let us consider an example to understand the working of this technique. We have two documents that contain terms such as: 1) I want to play a cricket and the second document contains a word 2) I do not want to play a football. First we have to formed dictionary of unique words from these documents such as: (I, want, to, play, a, cricket, do, not, football). Then to make the vector of the first document, the terms of the document are matched with dictionary words. If term matched technique placed 1 in that index if not then placed 0 example represented in the Table 3.4.

The D1 Vec and D2 Vec represent the vectors of document 1 and document 2 respectively. Although this is very simple strategy to implement but it has some disadvantages such as: 1) this method does not consider the position of a terms therefore it become difficult to examine the context of a word 2) does not consider

TABLE 3.4: One Hot Encoding Strategy

Dictionary	I	want	to	play	a	cricket	do	not	Football
D1 Vec	1	1	1	1	1	1	1	0	0
D2 Vec	1	1	1	1	1	0	1	1	1

the frequency information of a terms 3) Vector representation size grows as the vocabulary size grows.

3.4.1.2 Bag of Word (BOW) or Term Frequency (TF)

The idea behind this strategy is straightforward, and simple and they solve frequency issue of one hot encoding. In BOW first we have defined a fixed length vector where each entry corresponds to a word in our pre-defined dictionary of words. The size of the vector equivalent to the size of the dictionary. Thereafter, to represent a document using this vector, we count how many times each word of our dictionary appears in the document and we put this number in the corresponding vector entry. For example: we consider two documents such as: 1) Ali likes to watch movies. Sana also likes movies too, 2) Sana also likes to watch cricket games. Based on these two documents the dictionary is: (Ali, likes, to, watch, movies, Sana, also, too, cricket, games). Now on the basis of this dictionary by using BOW strategy the vector of D1 and D2 are presented in Table 3.5. However, this strategy solves the frequency issue of one hot encoding but some time terms with highest frequency did not show their importance. For example the word "computer" is a general word from Computer Science domain so it's definitely repeated more times than any other words in a computer domain research

TABLE 3.5: Bag of Word or Term Frequency Strategy

Dictionary	Ali	like	to	watch	movies	sana	also	too	cricket	games
D1 Vec	1	2	1	1	2	1	1	1	0	0
D2 Vec	0	1	1	1	0	1	1	0	1	1

articles. Moreover, this strategy does not consider order of words and semantic and context of a words.

3.4.1.3 Term Frequency and Inverse Document Frequency

TFIDF tells us about the importance of a terms in a document. It contain two concept Term Frequency (TF) and Inverse Document Frequency (IDF).
Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (for normalization). The term frequency can be defined as:

$$TF = \frac{No of time the word appear in a document}{Total No of Word in a document} \tag{3.1}$$

The Inverse document frequency (IDF) measures that how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is","of", and "that", may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scaling up the rare ones, by computing idf score by the following formula:

$$IDF = log_{10}(\frac{Number of Document}{Number of Document in which word Appear}) \tag{3.2}$$

Consider an example of finding TFIDF Score: a document containing 100 words wherein the word cat appears 3 times. The term frequency for cat is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in one thousands of these. Then, the inverse document frequency is calculated as log (10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12. However, the TFIDF perform well in different scenarios and it has solved many issues of previous techniques but they lack Semantic and context of the terms.
After detailed analysis of some of the famous text representation technique which have used in literature of research article classification, it has been observed that

while capturing information these technique mostly rely on frequency of terms and ignored the semantic and context of terms. The current state-of-the-art approaches[9][10][11][12][13][14] for research article classification and have employed these conventional statistical measures like one hot Encoding, BOW, and TFIDF etc. Due to which they have not considered the semantic and context of the terms so that's why they might assign a wrong category to the research articles. However, we have focused on text representation in this thesis. Before performing any mathematical operation like finding similarity between text document, the semantic and context of a text is considered as representation which is ignored by existing statistical measures. So now in next section we have discussed an alternative of the above mentioned strategies which can capture the semantic and contextual information of a terms and it is widely used in different domains and producing good accuracy[19][46][21].

3.4.2 Semantic Based Techniques

3.4.2.1 Word Embedding Using Word2Vec

As one of the famous quote of English professor J.R. Faith is "You shall know a word by the company its keeps". To represent a word we must know the semantic and context of a term in which the term is used because the meaning of a term vary in different context. For example Let us consider a word 'bank'. One meanings of a term is financial institution and another one is land alongside a body of water. If in a sentence, bank occurs with neighboring words such as: money, government, treasury, interest rates etc. we can understand it is the former meaning. Contrarily, if neighboring words are water, shore, river, land etc. the case is latter. After performing an in depth study we have identified one of the technique known as word Embedding which is used in different fields such as 1) Image processing 2) NLP Tasks 3) Biosciences etc, to represent a text by using different models. Word embedding produces good results in different fields which is shown in table 3.6.

TABLE 3.6: Word2Vec Results in Other fields

Fields	**Results**
Image Processing [19]	90 (%) accuracy
NLP Tasks [20]	More than 90 (%) accuracy
Recommendation Tasks [21]	Up to 95(%) accuracy
Biosciences [47]	More than 90 (%) accuracy
Semantics Task [40]	More than 90 (%) accuracy
Malware Detection Tasks [46]	Up to 99 (%) accuracy

Word embedding is one of the renowned method that is employed to represent the text of the document. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. For this, Word2vec technique/model is used to produce word embedding for better word representation. It captures a large number of precise syntactic and semantic word relationship. This was developed by Tomas Mikolov in 2013 at Google [22]. The Word2vec represents words in a vector space. The Words are represented in the form of vectors and placement is done in such a way that similar meaning words appear together and dissimilar words are located far away. This is also termed as a semantic relationship. Word2vec reconstructs the linguistic context of words. Before going further, let us understand, what is linguistic context means? In general life scenario when we speak or write to communicate, other audience try to figure out what is the objective of the sentence. For example, "What is the temperature of Pakistan", here the context is the speaker wants to know "temperature of Pakistan" which is context. In short, the main objective of a sentence is context. Word or sentence surrounding spoken or written language (disclosure) helps in determining the meaning of context. Word2vec learns vector representation of words through the contexts. By using this model we have found some results and it looks too good that when we generate the vector for the words like "dirty" and "smelly" and then we found the cosine similarity between these word it gives score up to 0.76. Similarly when we find the similarity score between the word dirty and clean, the model produces 0.25 score. These experiment clearly shows that word2vec model save the meaning and context of a terms. Moreover, it allows us

to use vector arithmetic's to work with analogies. For example the famous example of this model is which look like a magic is: when they subtract the vector of a man with king and add the women vector it produces queen vector as a result (king - man + woman = queen).

The word2vec model exist in two forms such as: 1) Already Train Algorithm (Like Google News model), which is provided by Google and 2) Self Training (Train algorithm by its self on our own data). We have picked the second option because the words used in news articles have different meaning and context as compared to the words used in research articles.

There are two main Word2Vec architectures that are used to produce a distributed representation of words:

In CBOW, the current word is predicted using the window of surrounding context windows. For example, if w(t-1), w(t-2), w(t+1), w(t+2) are given words or context, this model find the target word using context word. Skip-Gram performs opposite of CBOW which implies that it predicts the given sequence or context from the word. You can reverse the example to understand it. If wi is given, this will predict the context or w(t-1), w(t-2), w(t+1), w(t+2). The general overview of this model is represented in the Figure 3.5 [48]:

(a) Continuous bag-of-words (CBOW)

(b) Skip gram

Before going further, let us discuss why these architectures or models are important from word representation point of view. Learning word representation is essentially unsupervised, but targets/labels are required to train the model. Skip-gram and CBOW convert unsupervised representation to supervised form for model training.

For training word2vec model, we have used the CBOW model because it is far faster than skip gram and with a slightly better accuracy for the frequent word.

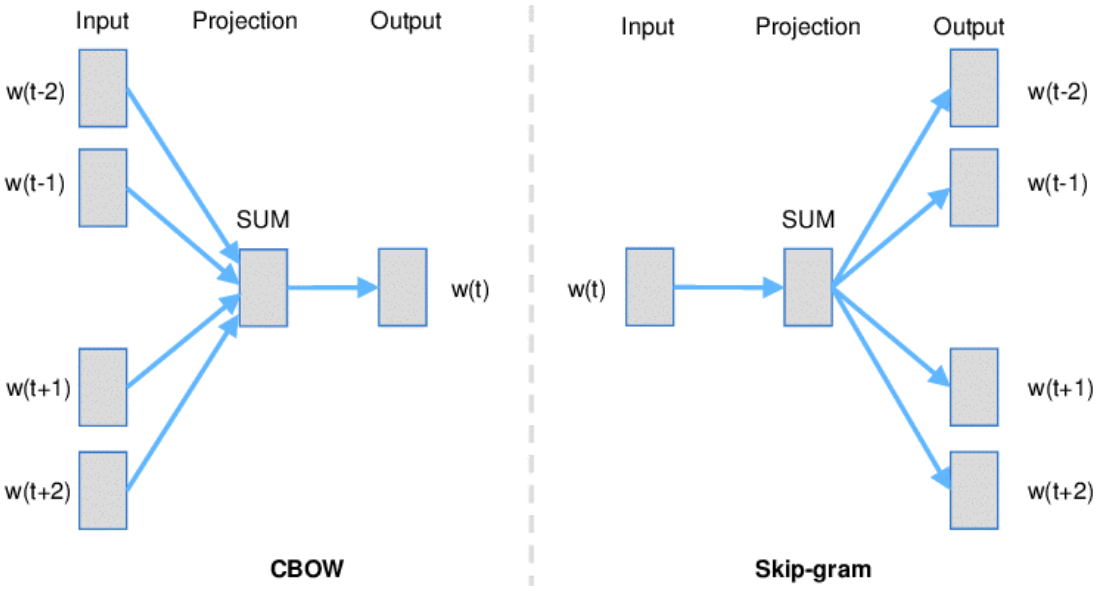

FIGURE 3.5: Word2Vec Models

3.4.2.2 Glove

Glove is one of another semantic based technique proposed by Pennington et al.[49] in 2014 with Stanford University. This techniques is come after word2vec one year letter. To understand what this variation attempts to do, we need to briefly talk about a less obvious aspect of Word2Vec. Word2Vec learns embeddings by relating target words to their context. However, it ignores whether some context words appear more often than others. For Word2Vec, a frequent co-occurrence of words creates more training examples, but it carries no additional information.In contrast, GloVe stresses that the frequency of co-occurrences is vital information and should not be "wasted" as additional training examples. Instead, GloVe builds word embeddings in a way that a combination of word vectors relates directly to the probability of these words' co-occurrence in the corpus. The main issue of Glove technique is that it focuses on words co-occurrences over the whole corpus and it behaving like count based approaches.

3.4.2.3 FastText

In 2016, artificial neural nets had gained quite some traction, and Word2Vec has proven its usefulness in many areas of NLP. However, there was one unsolved

problem: generalization to unknown words. FastText, a development by Facebook released in 2016 promised to overcome this obstacle. The idea is very similar to Word2Vec but with a major twist. Instead of using words to build word embeddings, FastText goes one level deeper. This deeper level consists of part of words and characters. In a sense, a word becomes its context. The building stones are therefore characters instead of words. The word embeddings outputted by FastText look very similar to the ones provided by Word2Vec. However, they are not calculated directly. Instead, they are a combination of lower-level embeddings. There are two main advantages to this approach. First, generalization is possible as long as new words have the same characters as known ones. Second, less training data is needed since much more information can be extracted from each piece of text. There are some drawback of this model such as 1)High memory requirement and 2) More focus on Syntactic of word rather than semantic.
After detailed analysis of count and semantic based technique we have concluded that Word2Vec model is best for capturing the semantic (More than 55%) and contextual information of a terms. So we have used Word2Vec model for vectorization of a text. First we have trained this model then we have used trained model to convert our text.

3.4.3 Word2Vec Training

During training phase the algorithm takes as an input a large corpus of text and produces a vector space, typically of several hundred dimensions, in which each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that every words that share common contexts in the corpus are located close to one another in the space. To train the word2vec model several steps have been performed which are described below:

(a) **Data Preparation**

In reality, text data are unstructured and can be dirty. Cleaning them will

involve steps such as removing stop words, punctuations, convert text to lowercase, remove numeric values, tokenization etc. After performing all the steps of preprocessing at the end the data will be converted in list of words.

(b) **Word2Vec Parameter**

Before training, the word2vec model required several parameters that affect both training speed and quality.

(i) **SIZE**

This parameter is used for the dimension of the word embedding and it typically ranges from 100 to 300. Large size dimension capture more information as compare to smaller sizes but it require more time to process.

(ii) **WINDOW**

This parameter is used to difine the window size for the model to train on text. When the window size to be 2, its means that words that are 2 to the left and right of the target words are considered as a context words. These four context words are used to find the target word.

(iii) **SG**

This parameter define the architecture of the model used in training. When value of this parameter is 1 algorithm considered skip-gram and if the value is 0 then algorithm considered CBOW architecture.

(iv) **ALPHA**

This parameter define the learning rate for the model. The learning rate controls the amount of adjustment made to the weights with respect to the loss gradient.

(v) **CBOW_MEAN**

This parameter is used only in case when you have used CBOW model. The value of this parameter is 0, so the algorithm takes the sum of context word vector, if the value is 1 than the algorithm take the mean of context word vector.

(vi) **MAX_VOCAB_SIZE**

This parameter is used to limits the RAM during vocabulary building; if there are more unique words than this, then prune the infrequent ones. Every 10 million word types need about 1GB of RAM. Set to None for no limit.

(vii) **ITER**

This parameter define the iteration of a neural network for finding the target word. In each iteration, the algorithm go through all training samples.

(viii) **HS**

This parameter define the activation fuction for the neural network.If the value is 1, hierarchical softmax will be used for model training. If 0, and negative is non-zero, negative sampling will be used.

We have performed various experiments to determine the optimium values of these parameters. The procedure of our experiments are briefly explained below:

- Fist, we take a dataset sample of research articles to trained the word2vec model.
- Next, we remove a noise from the dataset and trained a word2vec model by randomly selecting the parameters value.
- After that, we have used this trained model to transform our text into vectors.
- Afterwords, we have used this vectors for classification of research articles and computed average accuracy.
- In the last, we have just compared the results and performed the above steps if required.

The above mentioned procedure is applied for all the parameters which is required to trained model. However, some of the parameters have fixed value such as: 1) CBOW_MEAN (if we have used the CBOW architecture for

training, so its definitely the value of parameter is 1, 2) MAX_VOCAB_SIZE (if you have no issue regarding memory you have just set this parameter value to None) etc. Moreover, some of the parameter have binary value 1 & 0 such as 1) HS and 2) SG, we have checked both values of these parameters and select best of them. Furthermore, some of the parameters have continuous values such as: 1) SIZE, 2) WINDOW, 3) ALPHA and 4) ITER. We have started from some random value for these parameters, which is mostly used by the scientific community for experiments, after which the value is slightly changed to find results again and compared with previous results.We have repeatedly carried out experiments until improvement occurred. We stopped it when the result started to decline, and select the best value for that parameters.In Chapter 4, we have presented the optimum values which are found against the above experiments.

(c) **Generate or Transform Training Data**

The main objective of this step is to build vocabulary and then turn this vocabulary into a one-hot encoded representation for every word for the Word2Vec model to train. For example if one instance of a training example is "Hope can set you free". Now first build vocabulary, which contain 5 words. Afterwards change the word representation into one hot encoding like one hot encoding of Hope is (1, 0, 0, 0, 0) and for can is (0, 1, 0, 0, 0) and so on.

(d) **Model Training**

In model training phase, we start the training of word2vec model by using data of our own research articles. We have explained the internal mechanism of word2vec model training using CBOW by giving some example. Let us consider an example which contain a text this, "Hope can set you free". The number of words in this corpus are only 5. We have initially encode each word as one hot vector. For generating a vector we train a word2vec with that corpus. For simplicity we just assume that corpus contain only 5 words. Now we need to select the window size for iterating in a sentence, let us consider to be 1 in that case. As we discuss earlier the CBOW model

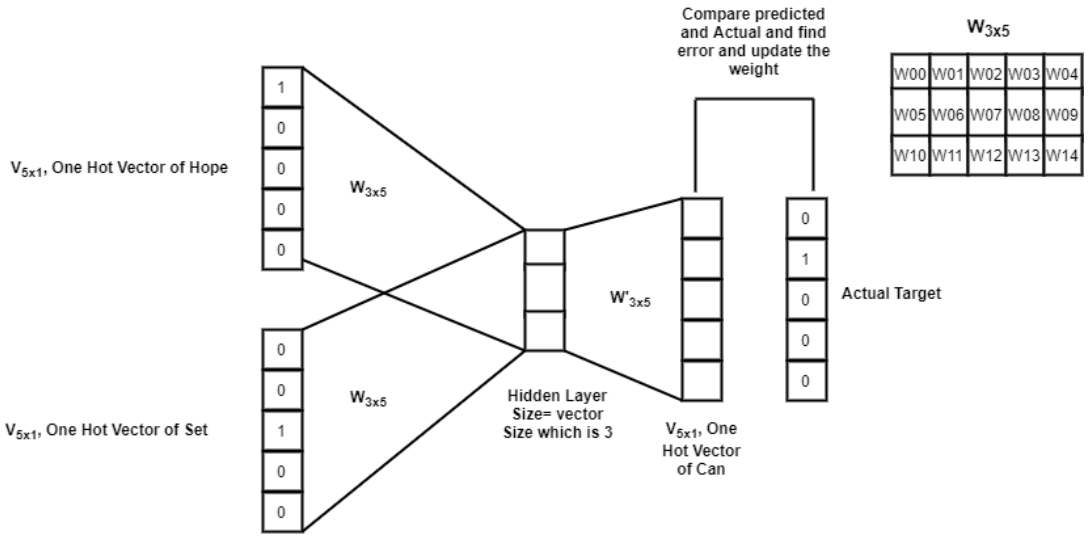

FIGURE 3.6: Neural Network

predict center word on the basis of contextual words, so our window size is 1 and we take first three word and the center word is identified by the two surrounding words. For this we employed a simple neural network, in which we give a two context words *hope* and *set* as an input and neural network will predict the word can. The one thing that we need to choose is the size of vector. Let us say the size of vector is 3 so the hidden layer in neural network size is also 3 shown in the Figure 3.6:

One important thing to note here is W5x3 input matrix is shared by context word. The output of the hidden layer pass to a softmax function, get the probabilities and compare with actual word vector. If the vector does not match with actual vector than find the error and update the weight by using back propagation method of a neural network. This step is repeated until a fixed number of iteration. Now slide the window on text and repeat the steps again for next prediction. In this way the weights are updated, once done we take the weight matrix and these weights are the set of vectors and this is how the continuous back of word trained on corpus. Therefore, the algorithm updates the weight matrix based on word and context. Once the neural network has been trained, the learned linear transformation in the hidden layer is taken as the word representation. Therefore, we pick weight matrix and multiply with one hot vector of every word to obtain its vector.

Now this word vector is word embedding. The above procedure has followed word2vec model to train on a corpus and generate the vector space in which words with similar meaning exist close to each other and words with different meaning far away from each other. The training of word2vec is performed by following the procedure presented in Figure 3.7.

The Figure 3.7 represents a procedure for training Word2Vec Model on research articles data. In step 1, list is define and assign all the records of a dataset to it. In step 2, we have iterate all the records of a dataset from a list. In step 3 and 4 we have split the sentence into words and removed the stop words and noise from the dataset records. In step 5, we have appended all the updated records into list. In step 7 we have defined a word2vec model with all its parameter. In step 8, model first builds a vocabulary for training from the list which contains records of the dataset. In step 9, the model train on that dataset according to parameter which is already defined in step 7. In last step we have saved the model in a .model extension file.

Algorithm 2 : Procedure of Word2Vec Model Training

Input: Research Article Dataset
Output: Trained Word2Vec Model

1: $Dataset\_R \leftarrow Read\ Dataset\ Record$
2: **for all** $row \leftarrow$ in $Dataset\_R$ **do**
3: $\quad Records \leftarrow row.split(" ")$ $\quad (Split\ records\ into\ words)$
4: $\quad Updated\_Records \leftarrow Remove\_Noise(Records)$
5: $\quad All\_Records \leftarrow Append\_Records(Updated\_Records)$
6: **end for**
7: $Model \leftarrow Word2Vec(SIZE = 300, WINDOW = 5, SG = 0, CBOW\_MEAN = 1, ALPHA = 0.1, MAX\_VOCAB\_SIZE = None, ITER = 10, HS = 1)$
8: $Model \leftarrow Model.builtvocab(All\_Records)$
9: $Model \leftarrow Model.train(All\_Records)$
10: $Model \leftarrow Model.Save('Research\_Paper.model')$

FIGURE 3.7: Procedure of Word2Vec Model Training

3.4.3.1 Text Conversion

After training the word2vec model, now we change the text corpus of our dataset using trained word2vec algorithm (as explained in previous section). Trained word2vec model has generated a vector of 75 * 4 lengths which consists of 300 elements. Each instance of record consists of random number of words, which is then combined to a single vector by taking mean of all word vectors.The conversion of text into vector is performed by following the procedure presented in Figure 3.8.

In step 1 of this procedure, list is define and assign all the records of a dataset to it. In Step 2, we have loaded the traind word2vec model. In step 3, we have iterated all the records of the dataset from the list. In step 4 we have iterate an individual record words. In step 5, we have passed that word to the trained word2vec model. This model generate a vector of 75 * 4 length, which was added with previous word vectors if that exists. In step 7, we take an average of individual record vectors to represent a fixed lenght vector. In step 8 the generated vector is written in a CSV file with their respective label.

Algorithm 3 : Procedure of Text Conversion to Vector

Input: Textual Dataset
Output: Vector Dataset

1: $Dataset\_R \leftarrow Read\ Dataset\ Record$
2: $Model \leftarrow Load\ already\ trained\ Word2vec\ model$
3: **for all** $row \leftarrow$ in $Dataset\_R$ **do**
4: **for all** $word \leftarrow$ in row **do**
5: $Vector\_Sum \leftarrow Vector\_Sum + Model[word]$
6: **end for**
7: $Average\_Vector \leftarrow Average(Vector\_Sum)/len(row)$
8: $File \leftarrow Write\ Vector\ in\ File\ with\ row\ Label$
9: **end for**

FIGURE 3.8: Procedure of Text Conversion to Vector

3.5 Similarity Measure

When all of the our research documents are represented as vectors, now for finding the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between two vectors, that is, the so-called cosine similarity. In machine learning, cosine similarity between two documents is calculated to examine how much the content of the two documents is similar. Mathematically, it measures the cosine of the angle between two document vectors projected in a multi-dimensional space. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity and higher the angle lower the cosine similarity. From the literature we have observed that the cosine similarity is one of the most popular similarity measures which has been applied in literature for finding similarity between text documents, such as in numerous information retrieval applications and data mining application. The standard formula for cosine similarity is given below:

$$CosineSimilarity(D_n, D_m) = \frac{\sum_{i=1}^{n} D_{n_i}, D_{m_i}}{\sqrt{\sum_{i=1}^{n} D_{n_i}^2}\sqrt{\sum_{i=1}^{n} D_{m_i}^2}} \tag{3.3}$$

In the above formula the D_n,D_m represent document 1 and document 2 vectors respectively. For understanding purpose, we have find the cosine similarity between two documents. Let's us consider an example for finding cosine similarity between two vectors. For this first we take two vector which represent two documents. For finding similarity we have performed all the required steps. The complete example shown in the given Table 3.7

TABLE 3.7: Finding Cosine Similarity Example

Document 1 Vector:	(3.4,0.1,0.4,3.1,2.3,3.5,4.3,0.5,4.5,0.6)
Document 2 Vector:	(3.2,1.5,0.4,3.5,2.6,5.5,3.3,0.5,4.6,0.8)
$\sum_{i=1}^{10} D_{1_i}, D_{2_i}$	=(3.4*3.2)+(0.1*1.5)+(0.4*0.4)+(3.1*3.5)+(2.3*2.6) +(3.5*5.5)+(4.3*3.3)+(0.5*0.5)+(4.5*4.6)+(0.6*0.8) =10.88+0.15+0.16+10.85+5.98+19.25+14.19+0.25 +20.7+0.48=82.89
$\sqrt{\sum_{i=1}^{n} D_{n_i}^2}$	$= \sqrt{(3.4)^2 + (0.1)^2 + (0.4)^2 + (3.1)^2 + (2.3)^2}$ $+(3.5)^2 + (4.3)^2 + (0.5)^2 + (4.5)^2 + (0.8)^2$ $= \sqrt{11.56 + 0.01 + 0.16 + 9.61 + 5.29 + 12.25}$ $+18.49 + 0.25 + 20.25 + 0.36$ $= \sqrt{78.23}$ $= 8.845$
$\sqrt{\sum_{i=1}^{n} D_{m_i}^2}$	$= \sqrt{(3.2)^2 + (1.5)^2 + (0.4)^2 + (3.5)^2 + (2.6)^2}$ $+(5.5)^2 + (3.3)^2 + (0.5)^2 + (4.6)^2 + (0.8)^2$ $= \sqrt{10.25 + 2.25 + 0.16 + 12.25 + 6.76}$ $+30.25 + 10.89 + 0.25 + 21.16 + 0.64$ $= \sqrt{90.85}$ $= 9.53$
$= \frac{\sum_{i=1}^{n} D_{n_i}, D_{m_i}}{\sqrt{\sum_{i=1}^{n} D_{n_i}^2}\sqrt{\sum_{i=1}^{n} D_{m_i}^2}}$	$= \frac{82.89}{8.845*9.53}$ $= \frac{82.89}{84.31}$ Cosine Similarity of D1 and D2=0.97

3.6 Single label Classification

The proposed approach is evaluated on both datasets for single-label document classification. In case of single label classification, test document is given to the system as an input, the system extract the metadata features from the test document. Thereafter, system transform these textual feature into numerical form by using semantic based train word2vec model. Afterwards, the system calculates the similarity score of a test document with every individual category papers. The

system has calculated an average of calculated score of a test document with the score of individual category papers. The average score represent the individual category similarity score. At the end system has select the category which have highest average similarity score. The equation 3.4 shows the behavior of a system:

$$AS_c = \frac{1}{n}\sum_{i=1}^{n} SS_{c_i}(T_p, P_{c_i}) \tag{3.4}$$

Where AS_c represents the average similarity score of individual category (A, B, C..... K), T_p represent test paper, P_{c_i} represents individual category papers. The system compute average similarity score for each category, and then select the max score category as predicted category like shown in equation below which are given below:

$$PredictedCategory = Max(AS_a, AS_b, AS_c.....AS_k) \tag{3.5}$$

The single label classification is performed by following the procedure presented in Figure 3.9.

In SLC procedure, the step 1 and step 2 just read the training and testing dataset and saved them in the list. In Step 3, we have defined the list for label names. In step 4 and 5 we have read training record one by one and taken the label of this record and placed in the variable (test_label). In step 6 and 7, we have read label from Label list one by one and extracted records against that label from training dataset and saved in the variable(training_records). In step 8 and 9, we have read record from training records one by one and computed similarity with a test sample. In step 11, we have calculate the average similarity score of Cosine Similarity score of individual category papers with test sample and saved in the list with their respective categories. In step 13, we have picked the highest average similarity score label as a predicted category of a test sample. From step 16 to 19 we have computed the results based on step 13 and step 14.

Algorithm 4 : Procedure of Single Label Classification

Input: Dataset
Output: Single Label Classification

$Training\_dataset \leftarrow Read\,Training\,Dataset\,Records$
$Testing\_dataset \leftarrow Read\,Testing\,Dataset\,Records$
$Labels \leftarrow Assign\,all\,Labels\,Name \qquad (A, B, Cetc)$
for all $test\_sample$ in $Testing\_dataset$ **do**
  $test\_label \leftarrow test\_sample['Label']$
  for all $label$ in $Labels$ **do**
    $training\_records \leftarrow Training\_dataset[Label]$
    for all $train\_sample$ in $training\_records$ **do**
      $Similarity\_Score \leftarrow Cosine\_Similarity(test\_sample, train\_sample)$
    end for
    $Ave\_Similarity\_Score\_List \leftarrow Similarity\_Score/len(training\_records)$
  end for
  $Predicted\_Label\_List \leftarrow MAX(Ave\_Similarity\_Score\_List).Label$
  $Actual\_Label\_List \leftarrow test\_label$
end for
$Average\_Accuracy \leftarrow Find\_Accuracy(Predicted\_Label\_List, Actual\_Label\_List)$
$Average\_Precision \leftarrow Find\_Precision(Predicted\_Label\_List, Actual\_Label\_List)$
$Average\_Recall \leftarrow Find\_Recall(Predicted\_Label\_List, Actual\_Label\_List)$
$Average\_F-Score \leftarrow Find\_F-Score(Predicted\_Label\_List, Actual\_Label\_List)$

FIGURE 3.9: Procedure of Single Label Classification

3.7 Multi-label Classification

In multi label classification, most of the approaches assign the relevant categories after comparing average value with some static threshold value like shown in equation.

$$AS_c = \frac{1}{n} \sum_{i=1}^{n} SS_{c_i}(T_p, P_{c_i}) \tag{3.6}$$

$$PredictedCategory = (AS_a, AS_b, AS_c.....AS_k) > Th \tag{3.7}$$

Th represent threshold values, so the categories score higher than threshold will be selected as final categories for test documents. In existing state-of-the-art, researchers have picked the method of selecting threshold value either asking from domain expert or by choosing some arbitrary values and then ensuring them on the basis of trial and error on dataset, which is a time consuming task. We argue that

dependence on domain experts or on some arbitrary value does not adequately serve the said purpose. We claim that threshold value should be defined based on rigorous analysis of the dataset being employed. Therefore, we have focused on designing a scheme which can help in finding threshold values based on rigorous analysis of the dataset.

3.7.1 Threshold Finding Scheme

To find threshold value, we have proposed a threshold finding scheme on the basis of dataset. In this scheme we find the correlation matrix of a dataset. The correlation matrix is defined in between the categories of a research articles. Each value in a correlation matrix is the average similarities score between two categories of research articles. The matrix of a dataset is shown in following matrix.

$$Correlationmatrix(Dn) = \begin{bmatrix} SS_{C1C1} & SS_{C1C1} & ..... & SS_{C1Cm} \\ SS_{C2C1} & SS_{C2C2} & ..... & SS_{C2Cm} \\ \vdots & \vdots & & \vdots \\ SS_{CnC1} & SS_{CnC2} & ..... & SS_{CnCm} \end{bmatrix} \tag{3.8}$$

$$SS_{C_nC_m} = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} SS(P_{C_{n_i}}, P_{C_{m_j}}) \tag{3.9}$$

The D_n represents the dataset chunks, $SS_{C_nC_m}$ represent the average similarity score between category n with category m where as m represents the rows and n represents the column. SS represent similarity score, $P_{C_{n_i}}, P_{C_{m_j}}$ represent the paper of category n and category m respectively. From correlation matrix of a dataset we have picked the diagonal value like as shown in equation 7.

$$M_T(D_n) = (SS_{C1C1}, SS_{C2C2}, SS_{C3C3}..., SS_{CnCm}) \tag{3.10}$$

The $M_T(D_n)$ represents the list of diagonal values from a correlation matrix of an individual dataset. These diagonal values are basically the threshold values which we have picked for every individual category data. For example SS_{C1C1} is a

Algorithm 5 : Finding Thresholds from datasets

Input: Datasets
Output: Threshold for each Category

1: $Dataset\_R \leftarrow Read\ Dataset\ Record$
2: $Labels \leftarrow Assign\ all\ Labels\ Name \qquad (A, B, C\ etc...)$
3: **for** $i \leftarrow 1$ to $len(Labels)$ **do**
4: **for** $j \leftarrow 1$ to $len(Labels)$ **do**
5: $Records\_1 \leftarrow Dataset\_R[Labels[i]]$
6: $Records\_2 \leftarrow Dataset\_R[Labels[j]]$
7: **for** $k \leftarrow 1$ to $len(Records\_1)$ **do**
8: **for** $m \leftarrow 1$ to $len(Records\_2)$ **do**
9: **if** $Labels[i] = Labels[j]$ **then**
10: **if** $k < m$ **then**
11: $A\_S\_Score \leftarrow Cosine\_Similarity(Records\_1[k], Records\_2[m])$
12: **end if**
13: **else**
14: $A\_S\_Score \leftarrow Cosine\_Similarity(Records\_1[k], Records\_2[m])$
15: **end if**
16: **end for**
17: **end for**
18: $Correlation\_matrix[i][j] \leftarrow mean(A\_S\_Score)$
19: **end for**
20: **end for**
21: $Thresholds \leftarrow Extract\_Diagonal\_Values(Correlation\_matrix)$

FIGURE 3.10: Finding Threshold Algorithm

threshold value for category A,SS_{C2C2} is a threshold value for category B and so on. We have selected these values because these are the average similarity score of every individual category with its own category papers. So if a test paper related to this category is more than 90 chances that their average similarity score with that category is higher than this selected threshold value for that category, so that's why we have selected a separate threshold value for every category. Finding threshold value from data by using following algorithm presented in the Figure 3.10.

In this algorithm, in step 1 we have read the dataset records and saved in the list. In Step 2, we have defined one list for label names. In step 3 and 4 we have read labels from Labels list and extract records against those labels. From step 7 to step 18, the algorithm find the average similarity score between extracted records

Algorithm 6 : Multi Label Classification Procedure

Input: Dataset
Output: Multi Label Classification

```
1: Training_dataset ← Read Training Dataset Records
2: Testing_dataset ← Read Testing Dataset Records
3: Labels ← Assign all Labels Name        (A, B, Cetc)
4: Thresholds ← Assign threshold Values against datasets
5: for all test_sample in Testing_dataset do
6:    test_label ← test_sample['Label']
7:    for all label in Labels do
8:       training_records ← Training_dataset[Label]
9:       for all train_sample in training_records do
10:         Similarity_Score ← Cosine_Similarity(test_sample, train_sample)
11:      end for
12:      Ave_Similarity_Score_List ← Similarity_Score/len(training_records)
13:   end for
14:   for all i ← 1 to len(Ave_Similarity_Score_List do
15:      if Ave_Similarity_Score_List[i] >= Thresholds[i] then
16:         Sample_Predicted ← Ave_Similarity_Score_List[i].Label
17:      end if
18:   end for
19:   Predicted_Label_List ← Sample_Predicted
20:   Actual_Label_List ← test_label
21: end for
22: Average_Accuracy ← Find_Accuracy(Predicted_Label_List, Actual_Label_List)
23: Average_Precision ← Find_Precision(Predicted_Label_List, Actual_Label_List)
24: Average_Recall ← Find_Recall(Predicted_Label_List, Actual_Label_List)
25: Average_F−Score ← Find_F−Score(Predicted_Label_List, Actual_Label_List)
```

FIGURE 3.11: Multi Label Classification Procedure

of both labels and saved in a correlation matrix on a specific index. At the end, in the last step diagonal values of a correlation matrix are extracted which represent the threshold value for different labels which are assign in Labels list.

After finding threshold values, we have just compared the average similarity score of a test paper with every individual category with their respective threshold value. If the category score satisfy the threshold value, these categories is selected as a final list of predicted categories. The multi-label classification performed by following procedure presented in the Figure 3.11.

In this procedure, the step 1 and step 2 just read the training and testing datasets

and saved it in the list. In Step 3 and 4, we have defined two list one for label names and second one for threshold values for that dataset. In step 5 and 6 we have read testing record one by one and taken the label of this record and placed in the variable (test_label). In step 7 and 8, we have read label from Label list one by one and extract record against that label from training dataset and saved in the variable(training_records). In step 9 and 10, we have read record from training record one by one and find a similarity with the test sample. In step 11, we have calculated the average similarity score of Cosine Similarity score of individual category papers with test sample and saved them in the list with their respective category. In step 14 and 15, we have iterate average similarity score list and compared them with their respective threshold, if the average similarity score of a category is equal to or greater than their respective threshold value than that category was placed in predicted list against test sample. From step 22 to 25 we have found the results based on step 19 and step 20.

3.8 Evaluation and Comparisons

To evaluate the results of our proposed technique, the standard formula of Precision, Recall and F-measure is calculated. The formula of these measures is somehow changed for single label and multi-label classification, because in multi-label classification the partially correct concept is used in these formulas.

3.8.1 Single label Classification Measures

The proposed approach have evaluated on both datasets for single-label classification. The evaluation parameter used for single label classification are given below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.11}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.12}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.13}$$

$$F - Measure = \frac{2(Precision)(Recall)}{Precision + Recall} \tag{3.14}$$

3.8.2 Multi-label Classification Measures

The proposed approach have evaluated on both datasets for multi-label classification. The following evaluation parameters for multi-label document classification proposed by Godbole and Sarawagi [12]. These formulas are described below:

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\frac{|\ Predicted \bigcap Actual\ |}{|\ Predicted \bigcup Actual\ |} \tag{3.15}$$

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|\ Predicted \bigcap Actual\ |}{|\ Predicted\ |} \tag{3.16}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|\ Predicted \bigcap Actual\ |}{|\ Actual\ |} \tag{3.17}$$

$$F - Measure = \frac{1}{n}\sum_{i=1}^{n}(\frac{2(Precision)(Recall)}{Precision + Recall}) \tag{3.18}$$

The results of our proposed technique will be compared with the results of (Ali at el 2018), as we have used the same dataset and same parameter with different proposed technique.

Chapter 4

Result and Evaluation

In the previous chapter we have explained the in-depth details of the proposed methodology. This chapter presents the details about the results that have been obtained by applying the proposed methodology.

4.1 Dataset Collection

Evaluation of proposed methodology based on two dataset which are already explained in chapter 3. The first dataset contains research articles, collected from JUCS. This dataset contain almost 1460 research articles. After analyzing it has been observed that 335 research articles did not contain their respective categories, therefore we removed these records from datasets. The research articles contain in this dataset belong to 13 categories (A to M) of computer science domain, however we have picked only that categories from JUCS which are similar to ACM root level categories (explained in last chapter 3). The categories that have been removed from JUCS dataset are L and M (which contain about 22 research articles). The experiments are performed on remaining 1103 papers. The states of remaining 1103 papers is illustrated in the Table 4.1.

TABLE 4.1: JUCS Dataset Records

Features	**Records**
Total Research Articles	1460
Remaining Articles for experiment	1103
Single Label Data	576
Multi-label Data	527

The second dataset is developed by Santos in 2009.This dataset contains 86,116 research publications from 2,240 different workshops, conferences and journals. From these 86,116 research publications there are only 54,994 research publications which contain authors provided categories or classes. All the research article belong to 11 categories of ACM hierarchy. The experiment is done on these 54,994 research publications. The details of remaining 54,994 papers is described in the Table 4.2.

TABLE 4.2: ACM Dataset Records

Features	**Records**
Total Research Articles	86,116
Remaining Articles for experiment	54,994
Single Label Data	29,697
Multi-label Data	25,297

4.2 Metadata Extraction and Combinations

The next step of our methodology is extraction of metadata features from these selected datasets and formed their possible combination. There are two ways to extract the features and form their possible combination: 1) *manual* and 2) *machine oriented*. We have followed machine oriented approach and write an algorithm which just take the parameter name as an input and it extract that parameters records from a dataset and formed its possible combination. This algorithm is presented in chapter 3. We have extracted the title and keywords from JUCS dataset

as a features. As the General terms are not present in JUCS dataset, therefore from JUCS dataset we have only two features title and keywords. The metadata parameter presence percentage is described in the Table 4.3

Table 4.3: JUCS Dataset Records Presence Percentage

Metadata Parameter	**Metadata Parameter Presence Percentage**
Title	100(%)
Keywords	100(%)

Similarly, the Table 4.4 described the parameter presence percentage of ACM dataset.

Table 4.4: ACM Dataset Records Presence Percentage

Metadata Parameter	**Metadata Parameter Presence Percentage**
Title	100(%)
Keywords	43(%)
General Terms	92(%)

In ACM, all the titles of research articles were present. Almost 57% of the papers do not contains keywords and 8% of the papers do not contain the General terms. After successfully extracting metadata the algorithm formed all possible combination of these metadata of JUCS and ACM datasets which is describe in the Table 4.5, and 4.6 respectively.

Table 4.5: JUCS Dataset Total No of Records

JUCS

Metadata Features	**Records**
Title	1103
Keywords	1103
Title and Keywrods	1103

Table 4.6: ACM Dataset Total No of Records

ACM	
Metadata Features	**Records**
Title	54994
Keywords	23645
General Terms	50890
Title and Keywrods	23645
Title and General Terms	50890
Keywords General Terms	23416
Title, Keywords and General Terms	23645

4.3 Pre-Processing

After extraction of all metadata features from both datasets, all of these metadata features needed to be cleaned such as titles, keywords and General Terms. The following steps are involved in pre-processing:

1. Tokenize the text of all metadata features on basis of space. The Tokenization has been performed by using NLTK Library (tokenize).
2. Removal of noise from all metadata (Remove duplicates, punctuation, digits etc.).
3. Remove of stop words from all the metadata features using NLTK Stop words List.
4. Conversion of all the metadata features terms into their root terms by using porter stemmer algorithm (Porter, 1980).

We have performed all the pre-processing steps 100(%) successfully.

4.4 Vectorization

After performing all the pre-processing steps, the data is ready for vectorization process. Afterwords, vectorization of text was performed by semantic word2vec model. The vectorization process contains two steps which are given below:

4.4.1 Training

The training of word2vec model was performed by using algorithm presented in chapter 3. For training, we have used the JUCS and ACM datasets as a corpus. We just taken the corpus as an input to the algorithm, and set some parameters value of word2vec model. The word2vec model performed training on that corpus using CBOW architecture. The word2vec parameters value have found after performing several round of experiments. The optimum values found for these parameter after performing several experiment according to methodology presented in chapter 3 are illustrated in table 4.7:

TABLE 4.7: Word2Vec Training Parameter Values

Parameter Name	**Value**	**Parameter Name**	**Value**
SIZE	300	ALPHA	0.1
WINDOW	5	MAX_VOVAB_SIZE	None
SG	0	ITER	10
CBOW_MEAN	1	HS	1

4.4.2 Conversion

After training the word2vec model, the algorithm generate a vector space in which the words having similar meaning lie close to each other and words having different meaning lie far away from each other in a vector space. Now we have used this already trained word2vec algorithm and convert our text of our dataset vectors. We have just given a word to this algorithm as an input, the algorithm generate a vector for that word. As there are multiple words in a single documents,

therefore to generate a vector of fixed length, the average of all words vector is considered.Conversion of text into vector was performed by algorithm which is presented in chapter 3. Using that algorithm we have convert all the metadata features of both datasets. The states of successful conversion of a metadata and combination of metadata of research article of JUCS and ACM datasets are described in the Table 4.8 and 4.11 respectively.

TABLE 4.8: JUCS Dataset Successful Record Conversion Percentage

JUCS

Metadata Features	**Successful Conversion Percentage**
Title	99.7(%)
Keywords	99.7(%)
Title and Keywrods	99.9(%)

TABLE 4.9: ACM Dataset Successful Record Conversion Percentage

ACM

Metadata Features	**Successful Conversion Percentage**
Title	98.7(%)
Keywords	98.2(%)
General Terms	95(%)
Title and Keywrods	100(%)
Title and General Terms	100(%)
Keywords General Terms	99.9(%)
Title, Keywords and General Terms	99.9(%)

Some of metadata feature combination converted successfully like title and keywords etc, while some of the metadata features and its combination instances contains words which were not present in trained word2vec model vocabulary, due to this reasons those records were discarded.

4.5 Single Label Classification

As discussed in chapter 3 that, the proposed technique was also intended to perform single label classification. To perform SLC same two datasets (JUCS and ACM) were utilized. For single label Classification, the algorithm (presented in chapter 3) predict only that category which have highest average similarity score with test paper. If the two categories have same average similarity score, one of the category was considered as predicted category. For the evaluation of our proposed techniques, we have collected the single label instances of (H, I, D, F and K) categories and (H, D and I) categories from JUCS and ACM datasets respectively.The reasons of choosing these categories of both datasets is that these categories cover maximum amount of record as compare to other categories. Moreover, in single label classification, these is no need of a threshold value as it is binary classification that whether a research paper is correctly classified or not. For this, a well-known binary evaluation measures were used which is described in chapter 3.
To analyze the contribution of each metadata individually and by making their different combinations, we performed a post-hoc analysis, where we evaluated variants of our model containing single to multiple metadata combination. While building all possible combinations, we have considered only metadata of those papers whose all parameters in the combinations are available to avoid biasness. The detailed description and analysis of experiments and their results of single label classification on both datasets are as given below:

4.5.1 Single Metadata Parameters

The classification based on individual metadata features helps a lot in finding which metadata feature individually contributed more in achieving good results. For every individual metadata feature, accuracy, precision, recall and f-measure score was calculated for all the categories, and average accuracy, precision, recall and f-measure score is obtained by calculating the average of all the categories. In case of JUCS datasets, title metadata achieved the highest average Accuracy 0.80,

Precision of 0.78, Recall of 0.82 and F-measure of 0.78, then Keywords metadata as shown in the Figure 4.1 . In Case of ACM datasets, the similar results are achieved in case of title, as title metadata in ACM outperformed other metadata with average Accuracy 0.79, Precision of 0.78, Recall of 0.77, and F1-measure 0.77,

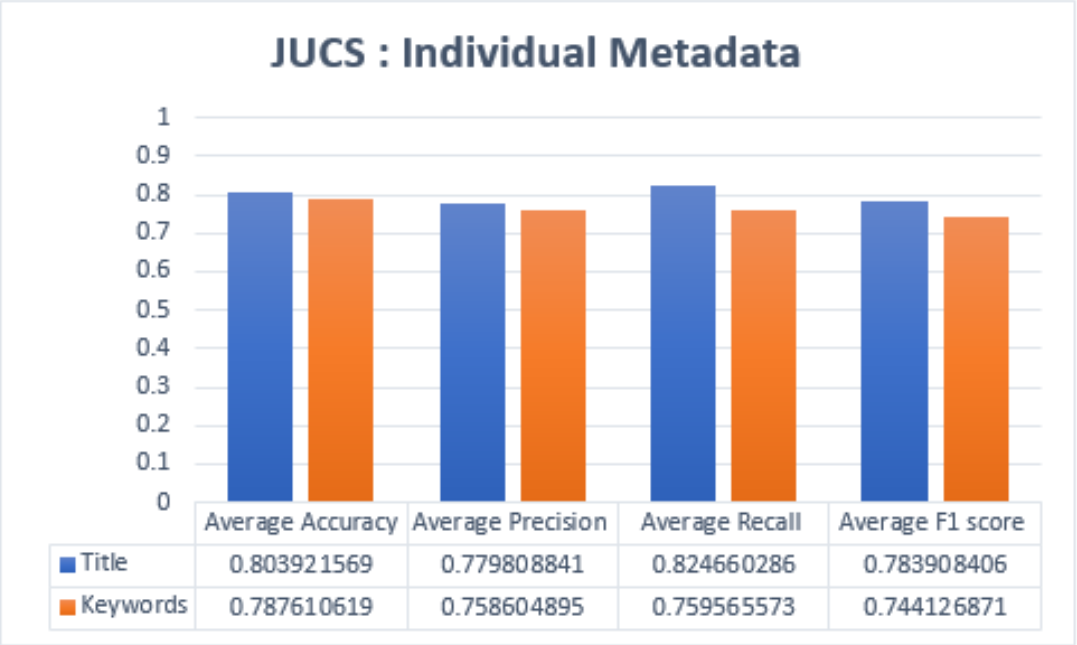

FIGURE 4.1: Results of Individual Metadata of JUCS Dataset

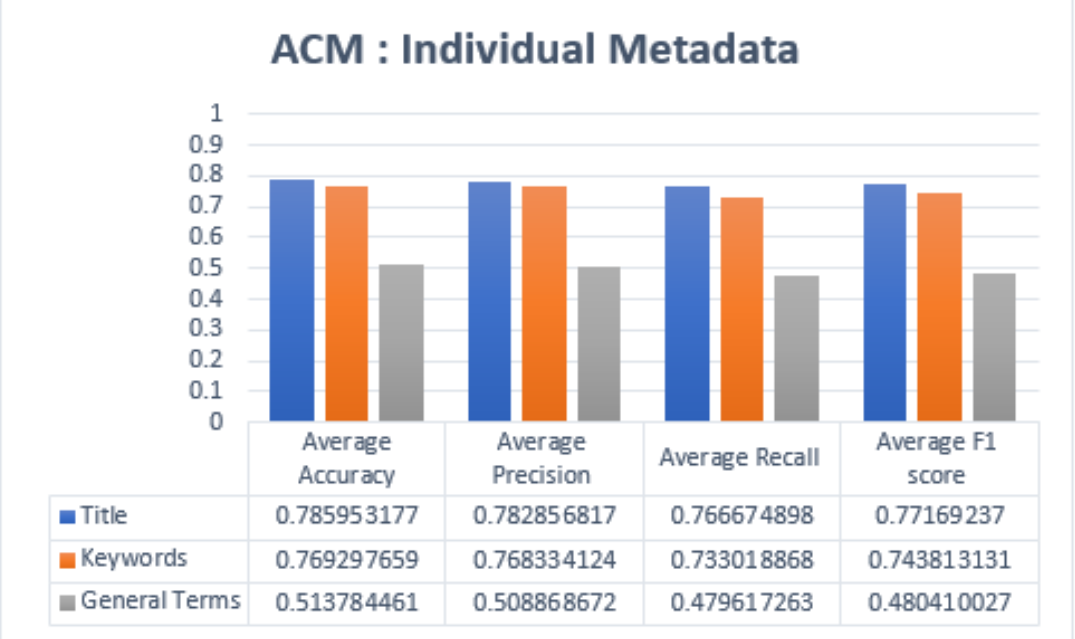

FIGURE 4.2: Results of Individual Metadata of ACM Dataset

followed by Keywords and General Terms parameter as shown in the Figure and 4.2. Similar behavior of title metadata in both datasets shows that title hold a strong potential in case of single label classification. As the title represent main idea of a research work so it contain such like words which specifically denote the particular subject. However, the articles belong to same category are more similar as compare to different categories articles. So that's why, when a test document is given to the system as an input, it will have high chances that its similarity score will be high with their actual paper categories as compared to other categories.

4.5.2 Double Metadata Parameters

In double metadata parameter every possible combination of two metadata parameters is exploited to obtain average accuracy, precision, recall and f-measures scores. In case of JUCS dataset there is only one combination of two metadata features "Title + Keywords" which obtained average Accuracy of 0.87, average Precision of 0.84, average Recall of 0.87 and average F1-measure is 0.81 which is shown in the Figure 4.3. In case of ACM datasets there are three double metadata parameter combinations "Title + Keywords", "Title + General Terms" and "Keywords + General Terms". The "Title + Keywords" combination outperformed other combination with the average Accuracy of 0.85, average precision of 0.80, average Recall of 0.82 and average F1-measure 0.81. The second top scored combination is "Title + General Terms" and the third one is "Keywords + General Terms" shown in the Figure 4.4. In case of Bi metadata features combination, while adding the keywords metadata with title metadata can improved the results of single label classification of research articles. The basic reason of improvement of classification is that, while adding keywords metadata it provide some more specific words which represent the subject of the paper. These words combine with Title words and classify the research article more accurately as compare to individual title words. The abbreviation of metadata parameters presented in the Figures 4.3, 4.4 are 1) Ti: Title, 2) Ke: Keywords 3) GT: General Terms.

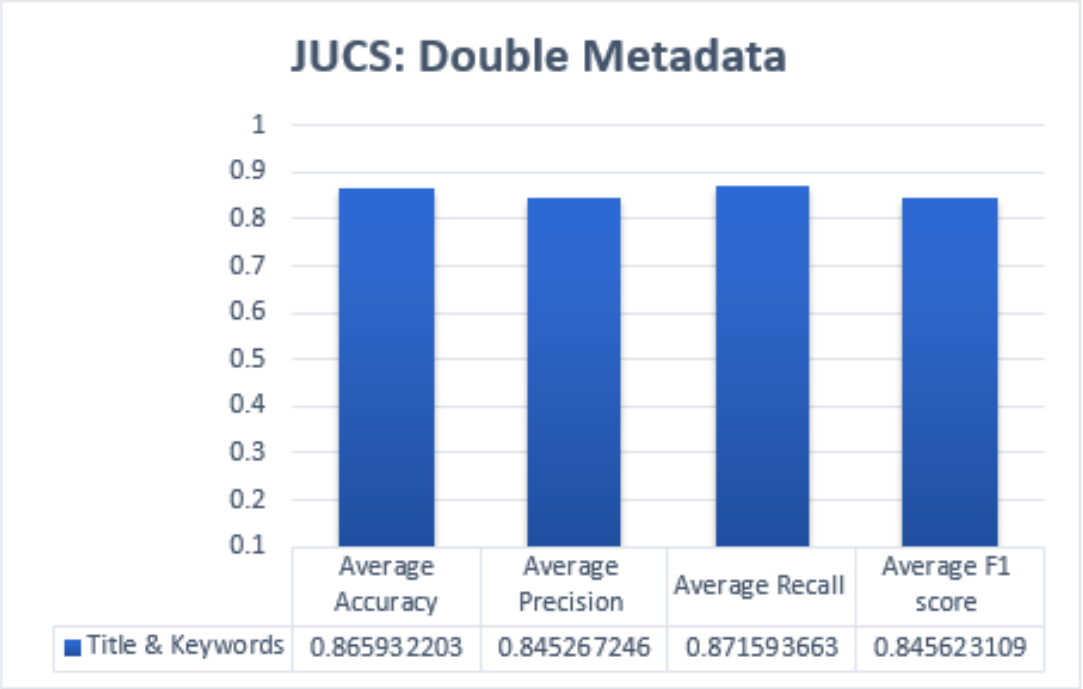

FIGURE 4.3: Results of Double Metadata Combination of JUCS Dataset

To further explain the above result, confusion matrixes of the experiments is given below.the following confusion matrixes represent Title and keywords metadata combination results on JUCS and ACM dataset respectively. In Table 4.10,

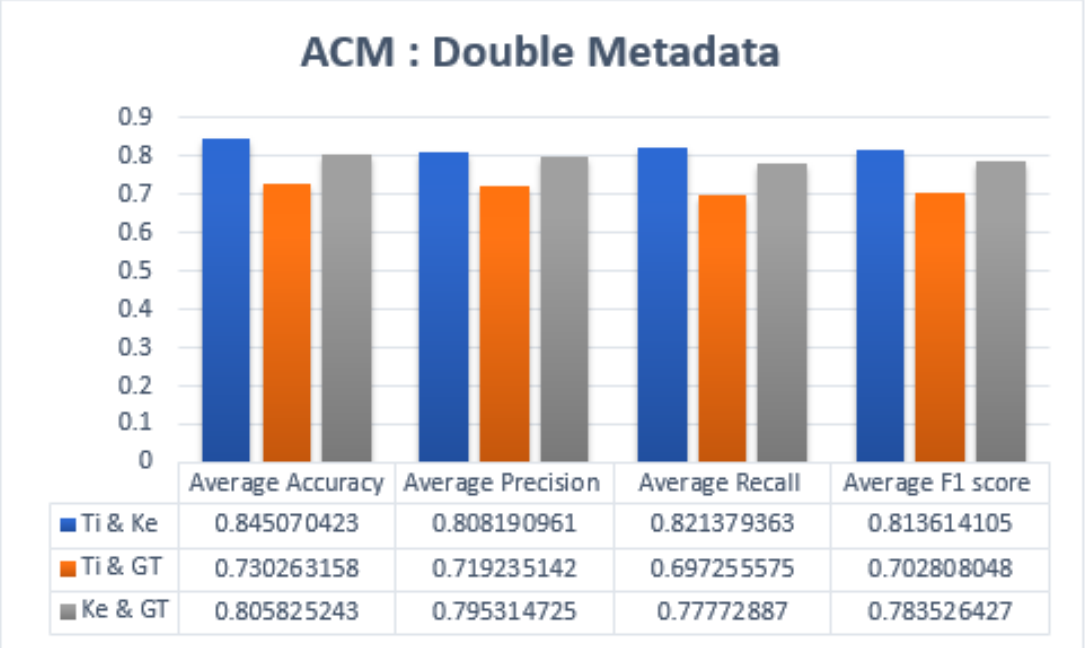

FIGURE 4.4: Results of Double Metadata Combination of ACM Dataset

in case of category H the diagonal value 32 represent true positive value. Except diagonal, the remaining value in row H represent false negative, column H represent false positive and the remaining represent Ture negative. In the same way values for other categories are extracted and then performance metices are calculated. The same procedure can be applied in case of ACM confusion matrix.

TABLE 4.10: Confusion matrix on JUCS dataset

Categories	D	F	H	I	K
D	23	0	4	1	1
F	1	26	1	0	2
H	0	2	32	0	3
I	0	0	1	10	1
K	0	0	0	0	10

TABLE 4.11: Confusion matrix for ACM dataset

Categories	D	H	I
D	90	4	2
H	10	118	14
I	5	9	32

4.5.3 Triple Metadata Parameters

In triple metadata parameter every possible combination of three metadata parameters is exploited to obtain average accuracy,precision, recall and f-measures scores. In case of JUCS dataset, there is no triple metadata combination while in case ACM dataset there is only one triple metadata combination which was "Title + Keywords + General Terms". The results obtained by this combination was lower as compare to "Title + Keywords" combination. Similarity of general term records are high as compared to title and keywords in different categories. Addition of general term with title and keywords negatively affected the classification results, due to decrease in diversification of records in the dataset. The results obtained by this combination are given in figure 4.5. The abbreviation of

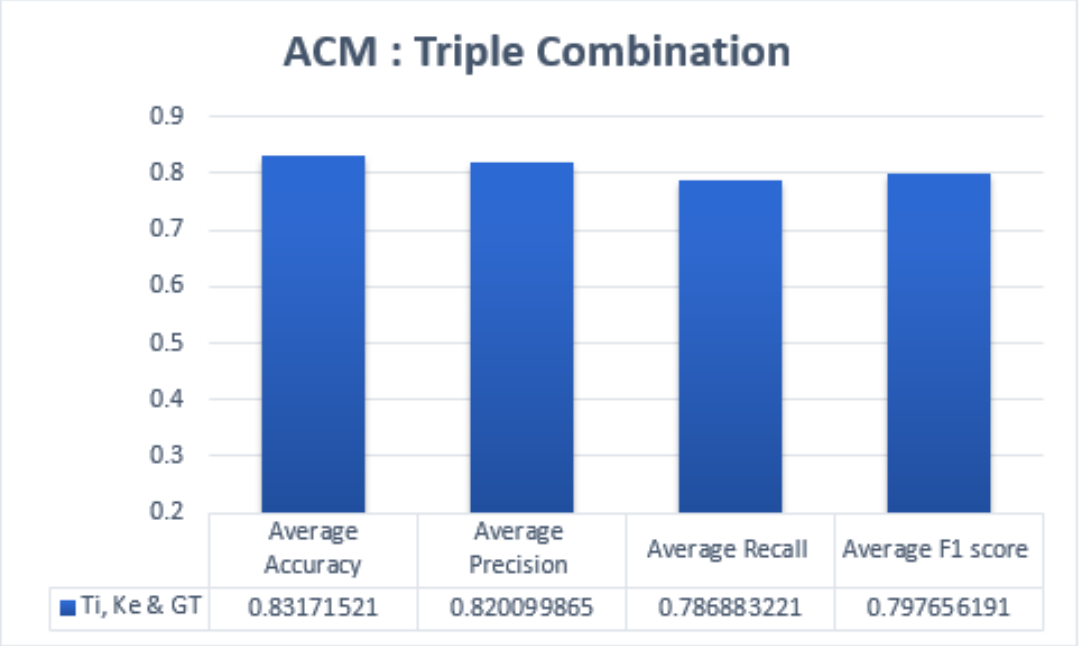

FIGURE 4.5: Results of Triple Metadata Combination of ACM Dataset

metadata parameters presented in the Figures 4.5 1) Ti: Title, 2) Ke: Keywords 3) GT: General Terms.

4.6 Multi-Label Classification

Before evaluating our proposed approach for performing multi-label classification, we have first defined a threshold values. In this thesis we have proposed an algorithm (Presented in Chapter 3) for finding a threshold values based on rigorous analysis of the datasets. However, by using that algorithm we have first defined the threshold values for all possible metadata combinations of both datasets. The algorithm defined the threshold value for every category of a single datasets. The threshold values of different combination of a JUCS and ACM datasets are illustrated in the Tables 4.12 and 4.13 respectively:

TABLE 4.12: Threshold Values for JUCS Dataset

Datasets	**Combinations**	**D**	**F**	**H**	**I**	**K**
JUCS	Title	0.37	0.35	0.39	0.34	0.31
	Keywords	0.4	0.39	0.4	0.36	0.45
	Title and Keywords	0.4	0.41	0.45	0.39	0.49

TABLE 4.13: Threshold Values for ACM Dataset

Datasets	**Combinations**	**D**	**H**	**I**
ACM	Title	0.13	0.12	0.10
	Keywords	0.16	0.13	0.13
	General Terms	0.46	0.52	0.51
	Title and Keywords	0.19	0.21	0.15
	Title and General Terms	0.28	0.30	0.23
	Keywords and General Terms	0.28	0.30	0.26
	Title,Keywords and General Terms	0.33	0.24	0.31

After defining all threshold value, now we have performed multi-label classification by using multi-label classification algorithm (Presented in Chapter 2). The algorithm finds average similarity score of a test document with every individual category papers. These average similarity score of each category was compared with their respective threshold. The category score which have met their threshold value is selected as a predicted categories. For experiments, the multi- label instances of (H, I, D, F and K) categories and (H, D and I) categories from JUCS and ACM datasets respectively. The reasons of choosing these categories of both datasets is that these categories cover maximum amount of record and another major reason is that we intend to compare our outcomes to one similar state-of-the-art study which has picked these categories. Since comparison results are justified when major factors among the studies have been contemplated on the basis of same grounds.

Similar to single label classification, we have analyzed the contribution of each metadata individually and collectively.While building all possible combinations,

we have considered only metadata of those papers whose all parameters in the combinations are available to avoid biasness. The detailed description and analysis of experiments and their results of multi-label classification on both datasets are as given below:

4.6.1 Single Metadata Parameters

Similar to single label classification we have also evaluated individual metadata features which helps us in finding some metadata features who's individually contributed more in achieving good results. For every individual metadata feature, accuracy, precision, recall and f-measure score was calculated for all the categories, and average accuracy, precision, recall and f-measure score was obtained by calculating the average of all the categories. In case of JUCS datasets, Keywords metadata achieved the highest average Accuracy of 0.77, average Precision of 0.87, Recall of 0.84 and F-measure of 0.84, then title metadata as shown in the Figure 4.6. In Case of ACM datasets, the similar results were achieved in case of Keywords metadata, as Keywords metadata in ACM outperformed other metadata with average Accuracy of 0.76, average precision of 0.82, average recall of 0.81, and average F1-measure 0.82, then Keywords, General Terms respectively achieved best score as shown in the Figure 4.7 . Similar behavior of Keywords metadata in both datasets shows that Keywords hold a strong potential in case of MLC. As we have noticed some contradictory behaviors of individual metadata in SLC and MLC. In SLC the title metadata is better than Keywords while in MLC Keywords is better than title. The reason of effectiveness of keywords in MLC is that, keywords contains such like words which represent different domains. However, these words are helpful in MLC as compare to title metadata which is better for SLC. I think that it might be possible due the reason that title metadata does not adequately covers all the domain keywords as compare to Keywords metadata. So the title metadata efficiently represent a single domain so it is better in SLC while Keywords contain such words which is mostly used in other domain areas so their similarity increases with other categories paper and its correct prediction

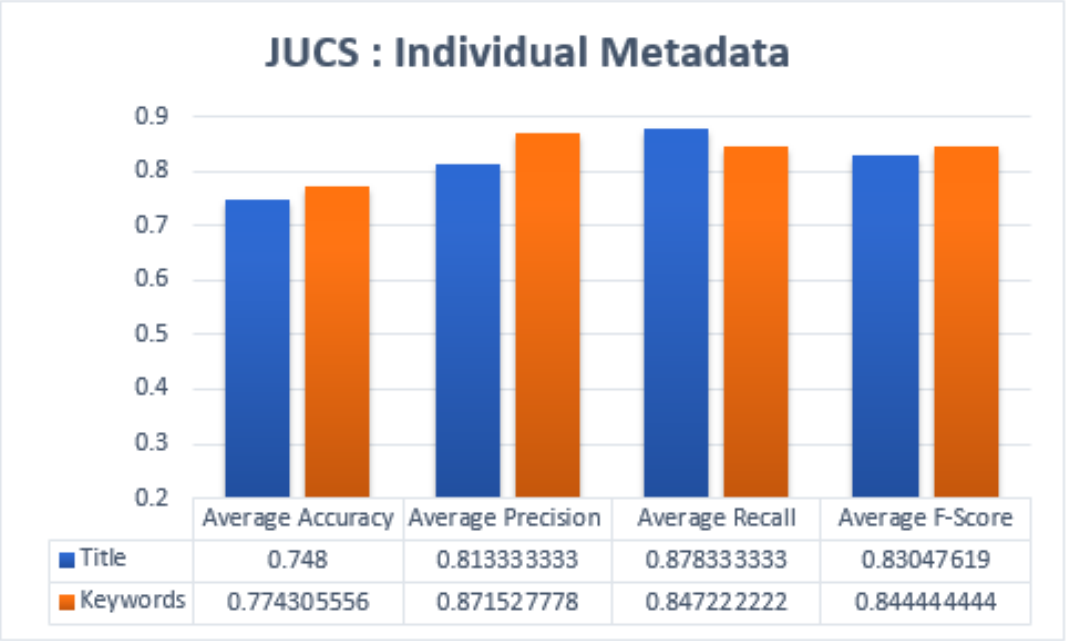

FIGURE 4.6: Results of Individual Metadata of JUCS Dataset

rate have also increase. In case of MLC, the document belong to more than one category, so we have used title to predict only one category effectively while in keywords metadata the different domain keywords help a lot to predict multiple correct categories.

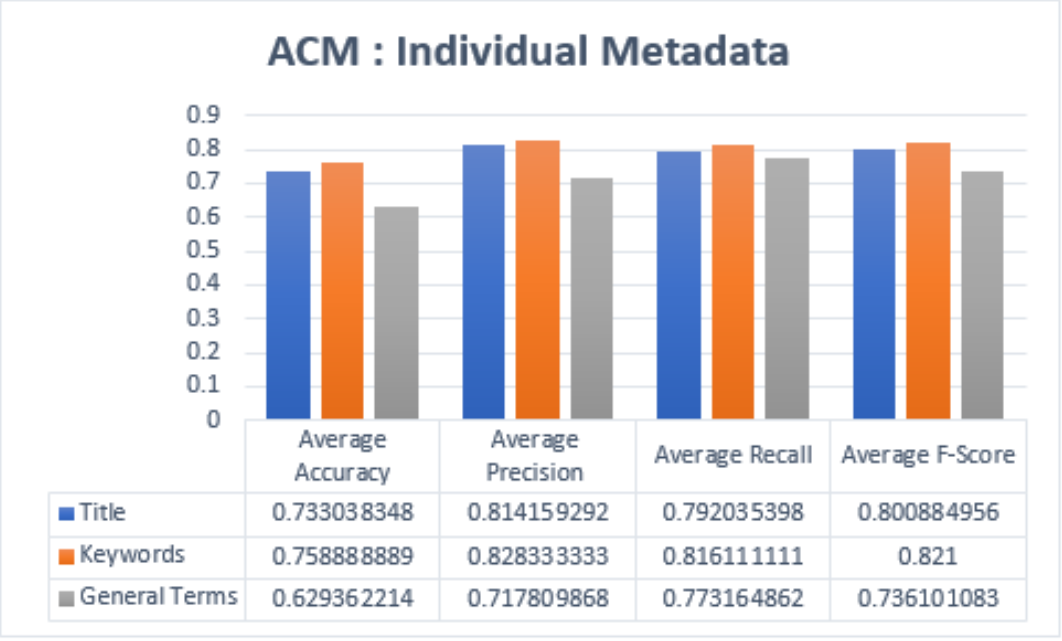

FIGURE 4.7: Results of Individual Metadata of ACM Dataset

4.6.2 Double Metadata Parameters

In double metadata parameter every possible combination of two metadata parameters is exploited to obtain average accuracy, Precision, Recall and F-measures scores. In case of JUCS dataset there is only one combination of two metadata features "Title + Keywords" which obtained average Accuracy of 0.83 average precision of 0.88, average recall of 0.91 and average F1-measure is 0.88 which is shown in figure 9. In case of ACM datasets there are three double metadata parameter combinations "Title + Keywords", "Title + General Terms" and "Keywords + General Terms". The "Title + Keywords" combination outperformed other combination with the average accuracy of 0.81, average precision of 0.86, average Recall of 0.86 and average F1-measure 0.86. The second top scored combination was "Title + General Terms" and the third one was "Keywords + General Terms" shown in the Figure 4.8. In case of Bi metadata features combination, while adding the title metadata with keywords metadata can improved the results of MLC of research articles. Some time the keywords metadata contains such like words which are generic in nature which mostly occur in different categories articles so its distract classification algorithm to classify the research article. In these scenarios, by adding title with keywords metadata at least one of the subject would be correctly classified. So that's why the accuracy of multi-label classification has been increased by adding title with keywords metadata.

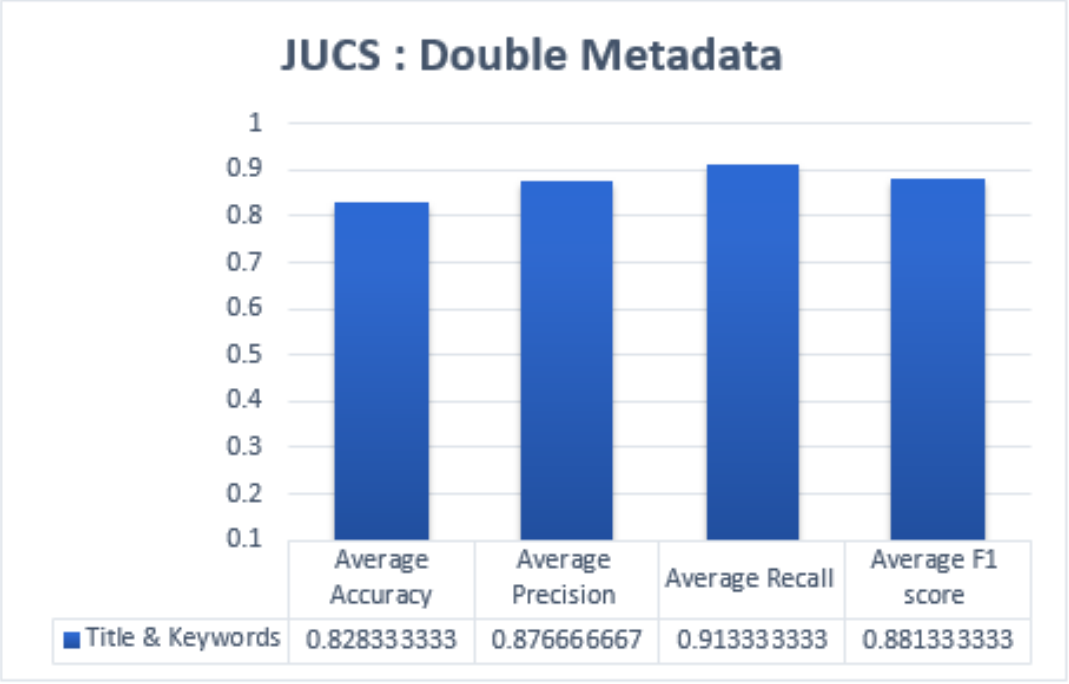

FIGURE 4.8: Results of Double Metadata Combination of JUCS Dataset

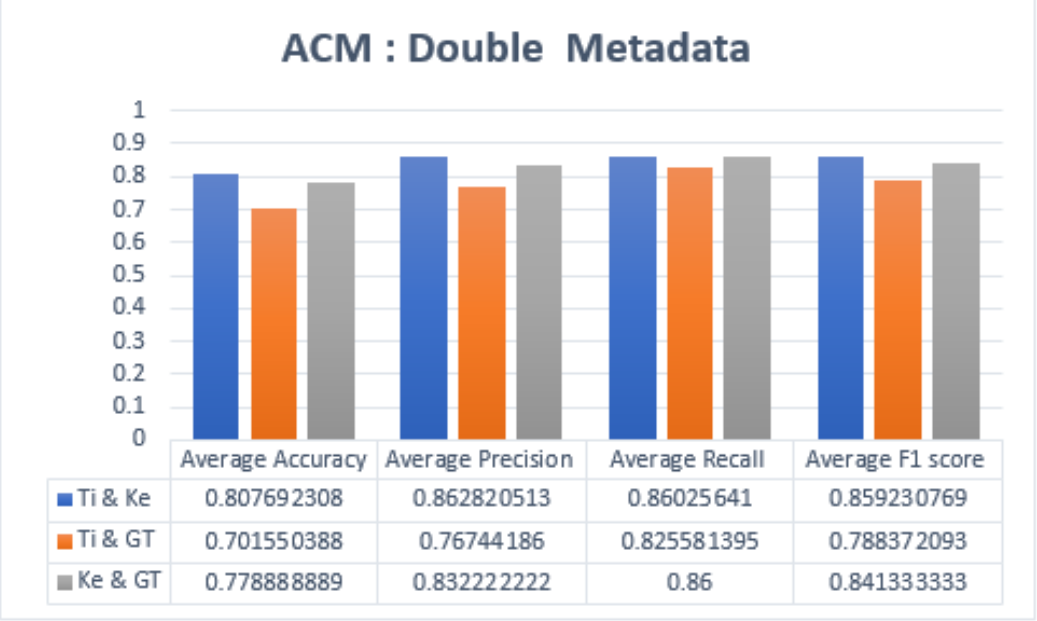

FIGURE 4.9: Results of Double Metadata Combination of ACM Dataset

4.6.3 Triple Metadata Parameters

In triple metadata parameter every possible combination of three metadata parameters is exploited to obtain average accuracy, precision, Recall and F-measures scores. In case of JUCS dataset, there is no triple metadata combination while in case ACM dataset there is only one triple metadata combination which is "Title + Keywords + General Terms". Similar to SLC, in MLC the results obtained by this combination is lower as compare to simple "Title + Keywords" combination. I think it might be due to reasons that the similarity between general term records are high as compared to title and keywords in different categories. Addition of general term with title and keywords negatively affect the classification results, due to decrease in diversification of records in the dataset. The results obtained by this combination are given in figure 10. The abbreviation of metadata parameters presented in the Figures 4.10 are 1) Ti: Title, 2) Ke: Keywords 3) GT: General Terms.

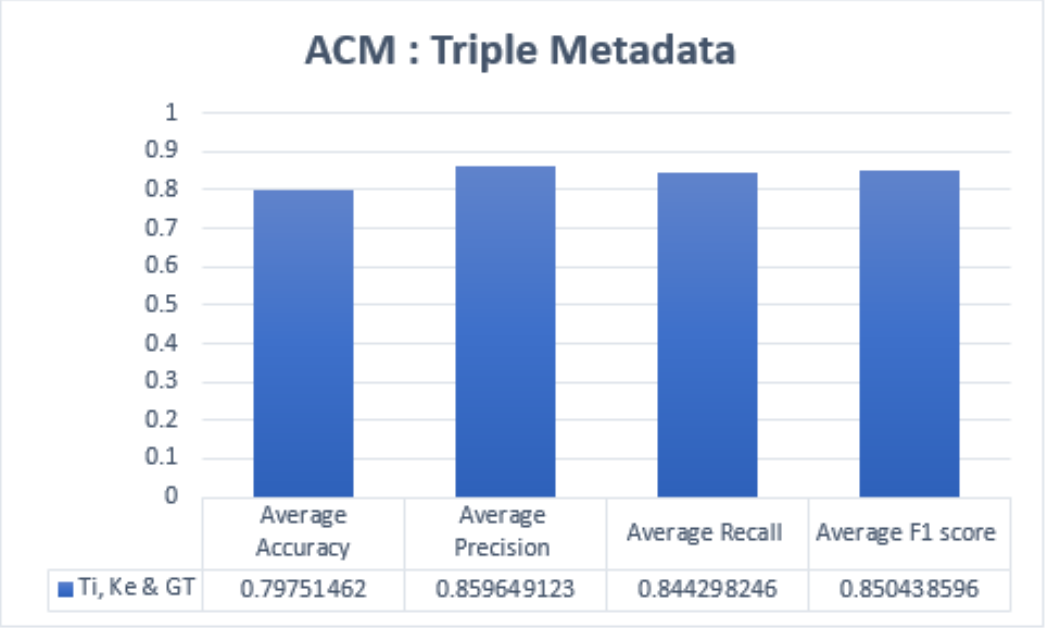

FIGURE 4.10: Results of Triple Metadata Combination of ACM Dataset

4.7 Comparisons

The document classification community has proposed multiple approaches for performing SLC and MLC. Most of these approaches have utilized the overall content of the research articles while some have prefer to harness metadata parameters due to unavailability of content. In this thesis we have also utilized the freely available such as 1) Title, 2) Keywords and 3) General Terms, for performing SLC as well as MLC. In case of SLC, we have compared our approach with the approach, proposed by Khor and Tang [43] which also utilizes the metadata of the research articles. For evaluation purpose the approach has used 400 educational conference's papers are collected and performed SLC into four topics such as "Intelligent Tutoring System", "Cognition", "E-Learning" and "Teacher Education". This approach have used different classifier for classification and achieved average accuracy up to 0.83. Moreover, this approach does not provide their dataset and in-depth detail of their methodology. So that's why we have not implemented it to reproduce the results. But due to unavailability of any other technique which has performed SLC by utilizing metadata of research articles, we have compared our results directly with khor and tang results.The comparison results are shown in the Figure 4.11: The approach proposed by Khor and Tang have considered a very few numbers of papers for the SLC. However, our datasets contain huge amount of research articles. In case of MLC, we have compared our results of our approach with the results of approach proposed by Ali and asghar in 2018. They have also utilized the metadata of research articles of both JUCS and ACM datasets. We have also utilized JUCS and ACM datasets for experimentation so it would be justified to make possible comparisons with their approach. The comparison results are shown in the Figure 4.12: From the 4.12 it can be seen clearly that in both datasets, our proposed technique achieved good results as compare to Ali and Asghar [17] for performing Multi-label Classification.

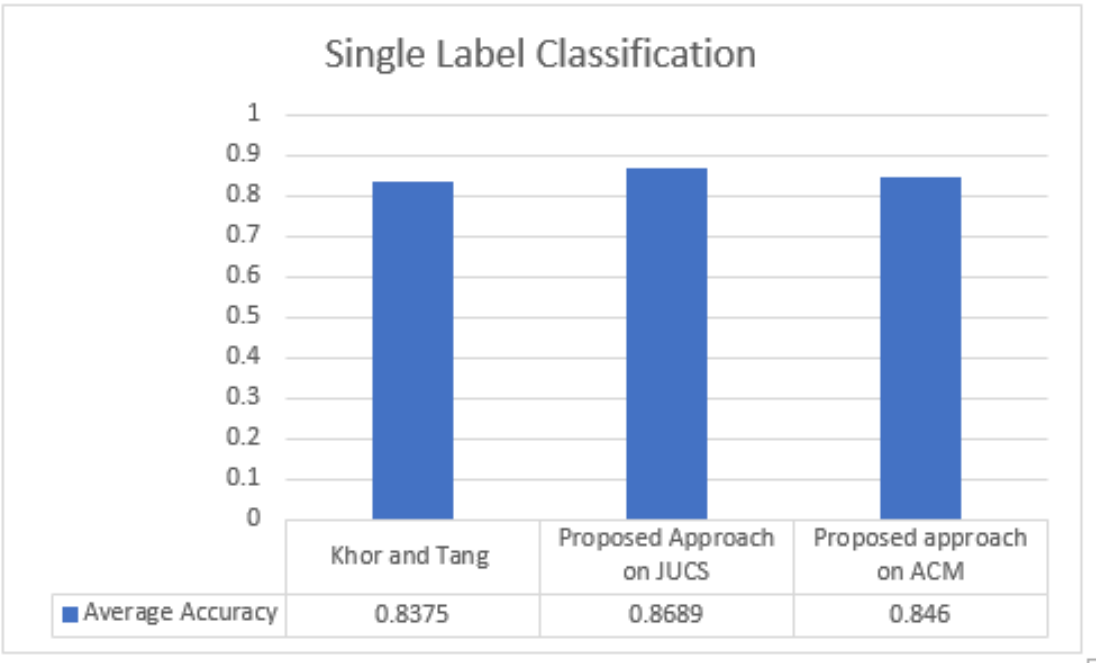

FIGURE 4.11: Single Label Classification Comparisons

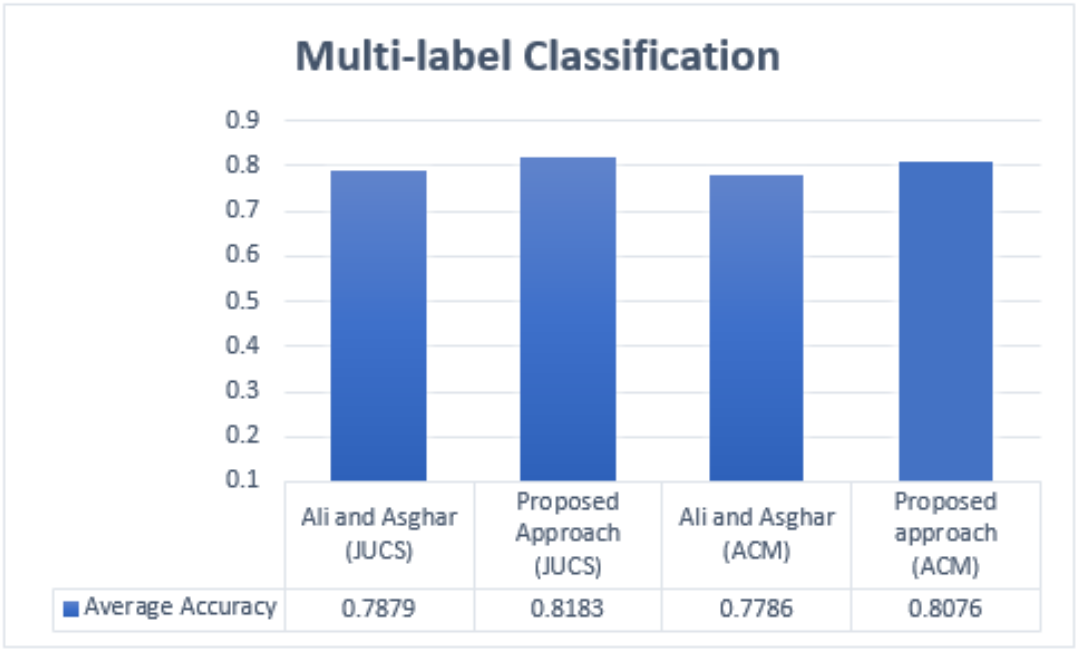

FIGURE 4.12: Multi-Label Classification Comparisons

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Classification of research articles into already predefined category/ies is a big research challenge. This research challenge has many applications such as 1) Indexing, 2) Retrieval of relevant papers, 3) paper submission and many more. A lot of approaches have been proposed by research community to categories the papers. After critical analysis of literature, we have observed that these approaches have been divided into two categories such as 1) Content based approaches, and 2) Metadata based approaches. Most of the these approaches utilized content based parameters due to richness of features. However, most of the time the content of the paper is not freely available, so in that scenarios scope of content based approaches gets limited. As an alternative, very few researchers have utilized the freely available metadata to categories the paper due to less number features and that's why these these approaches are unable to produced promissing results. Moreover, in classification of research articles, representation of a text is a very crucial steps in finding similarity or performing some statistical operation between text documents. The current state-of-the-art depict that most of the existing approaches have employed conventional statistical measures like TF, BOF, TFIDF etc. These measures mostly capture information using the frequency of terms.

We argue that before computing the similarity between textual documents, the semantics of a text must also be considered which is ignored by existing statistical measures. Moreover, multi-label classification based approaches have employed a static threshold values. In existing state-of-the art, researchers have picked the method of selecting threshold value either by asking from domain expert or by choosing some arbitrary values and then ensuring them on the basis of trial and error on dataset, which is a time consuming task. We argue that dependence on domain experts or on some arbitrary value does not adequately serve the said purpose. We claim that threshold value should be defined based on rigorous analysis of the data set being employed. These issues led us to our problem statement and their solution.

In this thesis, we have utilized the freely available metadata as a features for the classification of research articles. We have evaluated these metadata individually as well as its possible combination. For evaluation two benchmark dataset have been used for experiments. The first one is JUCS dataset which contain all the paper of JUCS journal. The second one is taken from ACM journal which is prepared by SANTOS in 2009. First we extracted metadata from these datasets. From first we have extracted title and keywords and from second we have extracted title, keywords and General Terms. Afterword's we have made all possible combination of these features. Further pre-processing is performed on both datasets in which first tokenize all the text into words, then performed stemming to stem all word into its root words and removed all the stop words and noise. Moreover, for the representation of a text we have used semantic model instead frequency based technique. The word2vec model capture the semantic as well as the context of a term used in a text. We have first trained this model by using the corpus of research articles. This model generate a vector space in which every word of a corpus represent by a unique vector. The vector lies in a vector space such like the similar word vector lies close to each other and dissimilar word lies in a vector space for away from each other. Afterword's, we have used this trained model and convert our text of both datasets records in to vector form.

When the datasets is ready for experiment we have first performed a single label classification. In SLC we just take input the test paper to the system, the system calculate the average similarity scores of a test document with every individual category papers. For similarity we have used the standard formula of cosine similarity. After finding all the categories score we have just the maximum average similarity score category as a predicted category against the test documents. In case of Multi-label classification, we have proposed the technique for finding threshold value for every category on the basis of in-depth analysis of datasets. Afterword's we have performed multi-label classification by using multi-label classification algorithm (Presented in Chapter 3). The algorithm find the average similarity score of a test document with every individual category papers. These average similarity score of every category compared with their respective threshold which is already defined. The category score which have met their threshold value is selected as a predicted categories.

In SLC and MLC we have evaluated the metadata individually as well as its combinations. In SLC, when we evaluated the metadata individually, in both dataset the title metadata achieved higher average accuracy of 0.80 and 0.79 for JUCS and ACM datasets respectively. In case of double metadata, the title and keywords combination performed extraordinary by achieving average accuracy of 0.87 and 0.85 for JUCS and ACM datasets respectively. As there is no triple metadata combination in JUCS dataset, while in ACM title, keywords and general terms combination achieved average accuracy 0.83.

In MLC, when we evaluated the metadata individually, in both dataset the keywords metadata achieved higher average accuracy of 0.77 and 0.76 for JUCS and ACM datasets respectively. In case of double metadata, similar to SLC the title and keywords combination performed extraordinary by achieving average f-score of 0.82 and 0.80 for JUCS and ACM datasets respectively. As there is no triple metadata combination in JUCS dataset, while in ACM title, keywords and general terms combination achieved average f-score 0.79.

We have compared our results with two state-of-art approaches. The first comparison is of SLC and the second one is MLC. The single label classification results

compared with khor or tang approach. This approached used metadata as a feature and achieved average accuracy up to 0.83. Our approach also utilized the metadata and achieved average accuracy up to 0.87 on JUCS and 0.85 on ACM datasets. The multi-label classification results were compared with Ali and asghar results. Their approach also used metadata as a features and achieved results up to 0.78 and 0.77 on JUCS and ACM datasets respectively, while our approach achieved 0.82 and 0.80 on JUCS and ACM datasets respectively. The overall finding of this thesis are, 1) in the scenarios when content is not available, we have used a metadata as a replacement which can achieved good results up to certain extent, 2) we have used a semantic model for text representation which performed better than conventional statistical features, and the last one 3) The proposed approach reduces the cognitive effort of defining a threshold value where in domain expertise were required. The main limitation of our approach is that it is a lazy learning method because at every time we have find the average similarity of a test paper with each individual category papers.

5.2 Future Work:

We have identified some of potential directions for future research in this area which are described below:

1. We will extend the classification of research articles into second and third level categories of ACM hierarchy.
2. Evolutionary approaches can be used to exploit metadata for multi-label document classification.
3. Evaluation of proposed approaches in domains other than Computer Science

Bibliography

[1] P. Larsen and M. Von Ins, "The rate of growth in scientific publication and the decline in coverage provided by science citation index," *Scientometrics*, vol. 84, no. 3, pp. 575–603, 2010.

[2] A. Hodgson and L. Schlager, "Closing the pdf gap: Readcube's experiments in reader-focused design." *Learned Publishing*, vol. 30, no. 1, pp. 875–880, 2017.

[3] M. Ware and M. Mabe, "The stm report: An overview of scientific and scholarly journal publishing," *International Association of Scientific, Technical and Medical Publishers*, vol. 4, no. 3, pp. 1175–1356, 2015.

[4] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words," *Stanford InfoLab*, vol. 2, no. 1, pp. 175–182, 1997.

[5] I. Kononenko, "Comparison of inductive and naive bayesian learning approaches to automatic knowledge adquisition," *B. Wielinga Editors, Current trends in Knowledge Adquisition*, vol. 5, no. 4, pp. 190–197, 1990.

[6] M. F. Porter, "An algorithm for suffix stripping, readings in information retrieval," *San Francisco, CA*, vol. 3, no. 5, pp. 22–36, 1997.

[7] N. A. Sajid, T. Ali, M. T. Afzal, M. Ahmad, and M. A. Qadir, "Exploiting reference section to classify paper's topics," *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, vol. 5, no. 1, pp. 220–225, 2011.

[8] F. Jelinek, "Statistical methods for speech recognition," *MIT University*, vol. 1, no. 1, pp. 1–279, 1997.

[9] S. S. Karman and N. Ramaraj, "Similarity-based techniques for text document classification," *Int. J. SoftComput*, vol. 3, no. 1, pp. 58–62, 2008.

[10] A. P. Santos and F. Rodrigues, "Multi-label hierarchical text classification using the acm taxonomy," *14th Portuguese Conference on Artificial Intelligence (EPIA)*, vol. 5, no. 5, pp. 553–564, 2009.

[11] T. Wang and B. C. Desai, "Document classification with acm subject hierarchy," *2007 Canadian Conference on Electrical and Computer Engineering*, vol. 2, no. 3, pp. 792–795, 2007.

[12] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," *Pacific-Asia conference on knowledge discovery and data mining*, vol. 6, no. 1, pp. 22–30, 2004.

[13] P. K. Flynn, "Document classification in support of automated metadata extraction form heterogeneous collections," *Digital Commons*, vol. 2, no. 3, pp. 1–189, 2014.

[14] N. Sajid, M. Afzal, and M. Qadir, "Multi-label classification of computer science documents using fuzzy logic," *Journal of the National Science Foundation of Sri Lanka*, vol. 44, no. 2, pp. 228–239, 2016.

[15] C. Apté, F. Damerau, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Transactions on Information Systems (TOIS)*, vol. 12, no. 3, pp. 233–251, 1994.

[16] I. Dagan, Y. Karov, and D. Roth, "Mistake-driven learning in text categorization," *arXiv preprint cmp-lg/9706006*, vol. 5, no. 3, pp. 331–330, 1997.

[17] T. Ali and S. Asghar, "Multi-label scientific document classification," *Journal of Internet Technology*, vol. 19, no. 6, pp. 1707–1716, 2018.

[18] J. Yan and J. Hu, "Text semantic representation," *Encyclopedia of Database Systems*, vol. 5, no. 1, pp. 3075–3078, 2009.

[19] A. U. Dey, S. K. Ghosh, and E. Valveny, "Beyond visual semantics: Exploring the role of scene text in image understanding," *arXiv preprint arXiv:1905.10622*, vol. 12, no. 3, pp. 71–74, 2019.

[20] L. Xiao, G. Wang, and Y. Zuo, "Research on patent text classification based on word2vec and lstm," *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1, no. 1, pp. 81–84, 2018.

[21] Q. Pan, H. Dong, Y. Wang, Z. Cai, and L. Zhang, "Recommendation of crowdsourcing tasks based on word2vec semantic tags," *Wireless Communications and Mobile Computing*, vol. 19, no. 1, 2019.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, vol. 2, no. 5, pp. 707–719, 2013.

[23] M. T. Afzal, N. Kulathuramaiyer, H. A. Maurer, and W. Balke, "Creating links into the future." *J. UCS*, vol. 13, no. 9, pp. 1234–1245, 2007.

[24] T. Li, S. Zhu, and M. Ogihara, "Hierarchical document classification using automatically generated hierarchy," *Journal of Intelligent Information Systems*, vol. 29, no. 2, pp. 211–230, 2007.

[25] S. Hingmire, S. Chougule, G. K. Palshikar, and S. Chakraborti, "Document classification by topic labeling," *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, vol. 29, no. 2, pp. 877–880, 2013.

[26] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A bayesian classification approach using class-specific features for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1602–1606, 2016.

[27] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in naive bayes for text categorization," *IEEE transactions on knowledge and data engineering*, vol. 28, no. 9, pp. 2508–2521, 2016.

[28] N. H. N. Le and B. Q. Ho, "A comprehensive filter feature selection for improving document classification," *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, vol. 22, no. 5, pp. 169–177, 2015.

[29] T. Zhou, "Automated identification of computer science research papers," *University of windsor*, vol. 1, no. 1, pp. 1–120, 2016.

[30] W. Zong, F. Wu, L.-K. Chu, and D. Sculli, "A discriminative and semantic feature selection method for text categorization," *International Journal of Production Economics*, vol. 165, no. 1, pp. 215–222, 2015.

[31] K. Chekima, C. K. On, R. Alfred, G. K. Soon, and P. Anthony, "Document categorizer agent based on acm hierarchy," *2012 IEEE International Conference on Control System, Computing and Engineering*, vol. 21, no. 3, pp. 386–391, 2012.

[32] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, vol. 44, no. 1, pp. 78–87, 2004.

[33] S. Baker and A. Korhonen, "Initializing neural networks for hierarchical multi-label text classification," *BioNLP 2017*, vol. 13, no. 3, pp. 307–315, 2017.

[34] C. Goller, J. Löning, T. Will, and W. Wolff, "Automatic document classification-a thorough evaluation of various methods." *ISI*, vol. 2000, no. 2, pp. 145–162, 2000.

[35] E. Chernyak, "An approach to the problem of annotation of research publications," *Proceedings of the eighth ACM international conference on web search and data mining*, vol. 76, no. 2, pp. 429–434, 2015.

[36] R. Wang, G. Chen, and X. Sui, "Multi label text classification method based on co-occurrence latent semantic vector space," *Procedia computer science*, vol. 131, no. 3, pp. 756–764, 2018.

[37] H. Nanba, N. Kando, and M. Okumura, "Classification of research papers using citation links and citation types: Towards automatic review article

generation." *Advances in Classification Research Online*, vol. 11, no. 1, pp. 117–134, 2000.

[38] M. Taheriyan, "Subject classification of research papers based on interrelationships analysis," *Proceedings of the 2011 workshop on Knowledge discovery, modeling and simulation*, vol. 4, no. 1, pp. 39–44, 2011.

[39] V. Balys and R. Rudzkis, "Statistical classification of scientific publications," *Informatica*, vol. 21, no. 4, pp. 471–486, 2010.

[40] A. Khatua, A. Khatua, and E. Cambria, "A tale of two epidemics: Contextual word2vec for classifying twitter streams during outbreaks," *Information Processing & Management*, vol. 56, no. 1, pp. 247–257, 2019.

[41] R. Jindal *et al.*, "A novel method for efficient multi-label text categorization of research articles," *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, vol. 21, no. 3, pp. 333–336, 2018.

[42] P. Yohan, B. Sasidhar, S. A. H. Basha, and A. Govardhan, "Automatic named entity identification and classification using heuristic based approach for telugu," *International Journal of Computer Science Issues (IJCSI)*, vol. 11, no. 1, p. 173, 2014.

[43] K.-C. Khor and C.-Y. Ting, "A bayesian approach to classify conference papers," *Mexican International Conference on Artificial Intelligence*, vol. 2, no. 4, pp. 1027–1036, 2006.

[44] É. Archambault, D. Amyot, P. Deschamps, A. Nicol, F. Provencher, L. Rebout, and G. Roberge, "Proportion of open access papers published in peer-reviewed journals at the european and world levels—1996–2013," *digital commons*, vol. 1, no. 1, pp. 100–156, 2014.

[45] E. Loper and S. Bird, "Nltk: the natural language toolkit," *CoRR*, vol. 28, no. 3, pp. 228–236, 2002.

[46] T. Chen, Q. Mao, M. Lv, H. Cheng, and Y. Li, "Droidvecdeep: Android malware detection based on word2vec and deep belief network." *TIIS*, vol. 13, no. 4, pp. 2180–2197, 2019.

[47] I. M. Soriano, J. L. C. Peña, J. T. F. Breis, I. San Román, A. A. Barriuso, and D. G. Baraza, "Snomed2vec: representation of snomed ct terms with word2vec," *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, vol. 56, no. 5, pp. 678–683, 2019.

[48] J. Landthaler, B. Waltl, D. Huth, D. Braun, C. Stocker, T. Geiger, and F. Matthes, "Extending thesauri using word embeddings and the intersection method." *ASAIL@ ICAIL*, vol. 8, no. 1, pp. 112–119, 2017.

[49] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, vol. 22, no. 7, pp. 1532–1543, 2014.