

CAPITAL UNIVERSITY OF SCIENCE AND
TECHNOLOGY, ISLAMABAD



Metadata Extraction using Geometric and Layout Features from Research Publications

by

Raja Muhammad Waqas Ahmed

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the

Faculty of Computing

Department of Computer Science

2022

Metadata Extraction using Geometric and Layout Features from Research Publications

By

Raja Muhammad Waqas Ahmed
(DCS153002)

Dr. Faisal Qureshi, Professor
Ontario Tech University, Canada
(Foreign Evaluator 1)

Dr. Huiyu (Joe) Zhou, Professor
University of Leicester, UK
(Foreign Evaluator 2)

Dr. Nadeem Anjum
(Thesis Supervisor)

Dr. Abdul Basit Siddiqui
(Head, Department of Computer Science)

Dr. Muhammad Abdul Qadir
(Dean, Faculty of Computing)

DEPARTMENT OF COMPUTER SCIENCE
CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
ISLAMABAD

2022

Copyright © 2022 by Raja Muhammad Waqas Ahmed

All rights reserved. No part of this thesis may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, by any information storage and retrieval system without the prior written permission of the author.

To my parents and family



CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY ISLAMABAD

Expressway, Kahuta Road, Zone-V, Islamabad
Phone: +92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: <https://www.cust.edu.pk>

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the thesis, entitled “**Metadata Extraction using Geometric and Layout Features from Research Publications**” was conducted under the supervision of **Dr. Nadeem Anjum**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Computer Science, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Computer Science**. The open defence of the thesis was conducted on **November 18, 2022**.

Student Name : Raja Muhammad Waqas Ahmed
(DCS153002)

The Examination Committee unanimously agrees to award PhD degree in the mentioned field.

Examination Committee :

(a) External Examiner 1: Dr. Rabeeh Ayaz Abbasi
Associate Professor
QAU, Islamabad

(b) External Examiner 2: Dr. Hasan Ali Khattak
Associate Professor
SEECS, NUST, Islamabad

(c) Internal Examiner : Dr. Abdul Basit Siddiqui
Associate Professor
CUST, Islamabad

Supervisor Name : Dr. Nadeem Anjum
Associate Professor
CUST, Islamabad

Name of HoD : Dr. Abdul Basit Siddiqui
Associate Professor
CUST, Islamabad

Name of Dean : Dr. Muhammad Abdul Qadir
Professor
CUST, Islamabad

AUTHOR'S DECLARATION

I, **Raja Muhammad Waqas Ahmed (Registration No. DCS-153002)**, hereby state that my PhD thesis titled, '**Metadata Extraction using Geometric and Layout Features from Research Publications**' 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/ world.

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



(Raja Muhammad Waqas Ahmed)

Dated: 18 November, 2022

Registration No: DCS153002

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled “**Metadata Extraction using Geometric and Layout Features from Research Publications**” is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly 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 PhD Degree, the University reserves the right to withdraw/ revoke my PhD 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 thesis.



(Raja Muhammad Waqas Ahmed)

Dated: 18 November, 2022

Registration No: DCS153002

List of Publications

It is certified that following publication(s) have been made out of the research work that has been carried out for this thesis:-

1. **M. W. Ahmed** and M. T. Afzal, “FLAG-PDFe: Features Oriented Metadata Extraction Framework for Scientific Publications,” in *IEEE Access*, vol. 8, pp. 99458-99469, 2020, doi: 10.1109/ACCESS.2020.2997907.
2. **M.W. Ahmed**, and M.T. Afzal, “Comprehensive Analysis of Physical Layout Features, Logical Layout Features and Machine Learning Approaches for the Extraction of structural Components and Metadata of PDF-based Research Articles.,” *PhD Symposium at 16th International Conference on Frontiers of Information Technology (FIT'18)*, IEEE, 2018.

(Raja Muhammad Waqas Ahmed)

Registration No: DCS153002

Acknowledgement

First and foremost, all praises to the Almighty Allah, The most merciful and beneficent. Who blessed me with the opportunity, and resources to pursue the doctoral program, and it's the Almighty's blessing that I saw it through.

My Ph.D. supervisor, Dr. Nadeem Anjum's valuable guidance, scholarly input, and consistent encouragement made it possible for me to achieve my final goal. I am grateful to my initial supervisor Dr. Muhammad Tanvir Afzal whose guidance and wisdom always showed me a light at the end of the tunnel.

I am grateful to Dr. Muhammad Abdul Qadir, DEAN, Head of the Center for Distributed and Semantic Computing (CDSC), whose motivation and guidance made me work hard in the right direction. I am also thankful to all members of CDSC whose discussion and constructive criticism maintained an environment that was conducive to research.

I am extremely grateful to my parents, who persuaded, encouraged, supported, and helped me at every stage of my personal and academic life to see this achievement come true. I am very much thankful to my family, my wife, sons, and daughters, who supported me in every possible way to see the completion of this work.

Abstract

The unprecedented growth of the research publications in diversified domains has overwhelmed the research community. These research publications are linked and indexed in the digital libraries for researchers to find relevant literature in an efficient manner. Precise search on content and metadata is required by the researchers through renowned search engines, however, recall of such systems remains high, while the precision of such systems remains low. Semantic queries on research documents are only possible through the storage and indexing of the research paper's metadata and layout content information. In this context, document content extraction has attracted the research community in recent years and different approaches have been employed like heuristic-based, supervised, and unsupervised machine learning. Machine learning approaches produced optimum results for document structures with homogeneous nature and require a lot of tagged information. While rule-based approaches are not scalable as they require large rule files and domain knowledge for heterogeneous documents.

In this research, we have performed a comprehensive evaluation of the literature and identified research problems like (1) lack of complete analysis of the features that are useful to identify metadata (2) diversified datasets were not evaluated (3) previous approaches were not scalable for new publication styles (4) low performance on large datasets (5) no comprehensive evaluation of machine learning techniques were presented in previous approaches (6) a limited number of metadata elements were extracted. Therefore, a scalable and comprehensive approach is required to extract the metadata. Keeping in view of the above-mentioned issues, we have performed a comprehensive evaluation of physical and logical layout features and compared machine learning approaches to extract metadata and layout content from research articles like title, authors, affiliation, affiliation countries, figure captions, table description, heading and levels, body content, terms and keywords, articles publication metadata, and bibliographic section. The proposed

system consists of a four-stage process to extract metadata by transforming unstructured PDF documents to structure a layout-aware file. Textual and geometric information is important to evolve feature enrich text blocks. We have constructed a comprehensive annotated benchmark dataset from diversified domains of multiple publishers in order to evaluate and build a highly scalable approach and also identified the benchmark dataset. The system has utilized the best features describing the logical layout content of the research paper, to train and test different machine learning algorithms. To extract the logical layout structure, this thesis presents a detailed evaluation and comparison of classification models on the curated diversified dataset. The metadata has been extracted using natural language processing and heuristics. The finalized model is compared with state-of-the-art document structure analysis approaches like PDFX and CERMINE on curated dataset, and SectLabel on gold standards. The confusion matrix parameters are used to evaluate the experimental results. The proposed approach outperformed the state-of-the-art approaches on diversified datasets by achieving 16% performance gain. The Average F-score of the proposed approach “FLAG-PDFe” is 0.897 on CEUR dataset, 0.89 on our diversified curated dataset, and 0.938 on sectLabel dataset.

Contents

Author’s Declaration	vi
Plagiarism Undertaking	vii
List of Publications	viii
Acknowledgement	ix
Abstract	x
List of Figures	xvi
List of Tables	xviii
Abbreviations	xx
Symbols	xxi
1 Introduction	1
1.1 Background and Motivation	1
1.2 Research Challenges	8
1.2.1 Problem Statements	8
1.2.2 Research Questions	9
1.2.3 Research Objectives	10
1.2.3.1 Objectives	10
1.2.3.2 Significance	10
1.3 Proposed Methodology	11
1.3.1 Research Methodology	11
1.3.2 Identification of Datasets	14
1.3.3 Physical Layout Extraction	15
1.3.4 Features-set Identification	15
1.3.5 Logical Layout Structure (LLS) Extraction	16
1.3.6 Metadata Extraction	17
1.3.7 Applications	19

1.4	Thesis Outline	19
2	Literature Review	21
2.1	Rule-based Techniques	24
2.2	Machine-Learning Based Techniques	27
2.2.1	Classification	27
2.2.2	Clustering	28
2.2.3	Neural Network (Classification)	29
2.3	Graph-based Text Mining Techniques	30
2.4	Comparison of Approaches	32
2.5	Research Gaps	37
3	Extraction Algorithm	40
3.1	Dataset	42
3.1.1	CEUR Dataset	42
3.1.2	sectLabel Dataset	43
3.1.3	Construction of Diversified Dataset	44
3.1.3.1	Selection of Research Articles	45
3.1.3.2	Selection of Domain Experts	47
3.1.3.3	Tagging and Annotation of Dataset	48
3.2	Features Engineering	52
3.2.1	Physical Layout Extraction	54
3.2.2	Column Style Identification	55
3.2.3	Correct Reading Order	56
3.2.4	Physical and Logical Layout Evaluation	58
3.2.4.1	Font Properties	58
3.2.4.2	Text Location	59
3.2.4.3	Neighbor Distance	60
3.2.4.4	Font Typography	61
3.2.4.5	Lexical Properties	62
3.2.5	Features Set Extraction	63
3.2.5.1	Font Features	63
3.2.5.2	Text Location Based Features	65
3.2.5.3	Neighbor Distance based Features	65
3.2.5.4	Font Typography and Lexical Features	66
3.2.5.5	Feature Extraction Algorithm	66
3.2.6	Features Ranking	75
3.2.6.1	The Removal of Features having a Low Variance	76
3.2.6.2	Recursive Feature Elimination RFE	76
3.2.6.3	Feature Selection Using "SelectFromModel"	76
3.2.6.4	Feature Selection as Part of a Pipeline	77
3.2.6.5	Univariate Feature Selection	77
3.2.6.6	K-best Features	77
3.2.7	Summary	79

3.3	Logical Layout Structure (LLS) and Extraction	79
3.3.1	Title Section	82
3.3.2	Authors Section	82
3.3.3	Table Section	83
3.3.4	Figure Section	85
3.3.5	Headings and Levels	86
3.3.6	Headers and Footers	86
3.3.7	Acknowledgment Section	88
3.3.8	Body / Paragraphs	89
3.3.9	Machine Learning as LLS Extraction Method	92
3.3.9.1	Discussion	94
3.3.9.2	Theoretical Evaluation	94
3.3.10	Evaluation of Machine Learning Models	97
3.4	Metadata Extraction	99
3.4.1	Title of the Article	102
3.4.2	Authors and Affiliations	102
3.4.3	Country of Affiliation Extraction	102
3.4.4	Author's Email Extraction	103
3.4.5	Extraction of Publication and Acceptance Date	104
3.4.6	The Title of Journal	105
3.4.7	Issue and Volume Information Extraction	105
3.4.8	Page Number Extraction	106
3.4.9	Digital Object Identifier (DOI) Information	106
3.4.10	Headings and Levels	107
3.4.11	Abstract and Keywords Related Metadata Extraction	107
3.4.12	Table Caption Extraction	107
3.4.13	Figure Caption Extraction	108
3.4.14	References Section Extraction	108
3.5	Summary	109
4	Results and Evaluation	110
4.1	Results on CEUR Dataset	110
4.1.1	CEUR Challenge	111
4.1.2	Proposed Methodology	112
4.1.3	Experimental Setup	114
4.1.4	Metadata Extraction	115
4.1.4.1	Author and Author Affiliation	115
4.1.4.2	Country of Affiliation	116
4.1.4.3	Section Heading	116
4.1.4.4	Table Caption	116
4.1.4.5	Figure Caption	117
4.1.4.6	Supplementary Material	117
4.1.4.7	Funding Agency	117
4.1.4.8	Funded Projects	118

4.1.5	Results	118
4.1.5.1	Logical Layout Extraction	118
4.1.5.2	Metadata Information Extraction	120
4.1.6	SemPub Challenge Comparison	121
4.1.7	Summary	122
4.2	Results on Curated Dataset	123
4.2.1	Results	124
4.2.2	Comparison with CERMINE and PDFX on GOLD-Standard	125
4.2.2.1	XQuery for CERMINE Generated XML Files	125
4.2.2.2	XQuery for PDFX Generated XML Files	128
4.2.2.3	Evaluation	132
	FLAG-PDFe Comparison with Cermine	132
	FLAG-PDFe Comparison with PDFX	139
4.2.3	FLAG-PDFe Comparison with sectLabel	145
4.2.4	Summary	147
5	Conclusion and Future Work	149
5.1	Conclusion	149
5.2	Key Contributions	152
5.3	Limitations	153
5.4	Future Work	154
	Bibliography	156

List of Figures

1.1	Google Scholar options for advance search.	3
1.2	Google Scholar and Semantic Scholar number of results for the mentioned search query.	4
1.3	Example1: The research article’s metadata.	5
1.4	Example2: The research article’s table and figure captions.	6
1.5	Example3: The research article’s headings and sub-headings with levels.	6
1.6	The overview of task performed by the proposed methodology.	11
1.7	The proposed research methodology steps.	12
1.8	The proposed methodology to extract research paper sections in order to evaluate Logical layout structure LLS	16
1.9	The proposed methodology to extract metadata of research paper from logical layout sections	18
2.1	The correct reading order of an article.	22
2.2	Timeline of document’s metadata extraction models.	24
3.1	Overview of the proposed system methodology.	41
3.2	Research Article selection flow for dataset	45
3.3	Research Article’s metadata annotated file to identify paper metadata, keywords, and section headings.	50
3.4	Research Articles metadata annotated file to identify authors and affiliations, figures, and tables captions.	51
3.5	The methodology diagram to extract features set associated to text blocks used for extraction of the logical layout structures using machine learning.	54
3.6	The connected lines shows the reading order of PDF based sample research document when extracted by itext library.	57
3.7	Alignment feature of the text blocks within a column.	60
3.8	The text blocks share common text properties, but the figure caption distance from paragraph is more as compared to distance among paragraph lines.	61
3.9	The text blocks share common text properties, but the table caption distance from paragraph is more as compared to distance among paragraph lines.	62

3.10	The Ranking of the top K selected features from ds3.1.1 using chi-square (chi^2)	78
3.11	The hierarchical view of Logical Layout Structures, class labels, and their associated metadata.	80
3.12	The Snapshot of Logical Layout Structures in research article and annotated class labels in extracted content, and their associated metadata.	81
3.13	The title section example of different research articles.	82
3.14	The title section LLS and metadata relationship in research articles.	83
3.15	The authors section's examples and the formatting styles.	83
3.16	The relationship between authors section, class labels and the related metadata.	84
3.17	Example of tables presentation styles.	84
3.18	The relationship between table LLS and metadata.	85
3.19	The caption style of figures used by different publishers.	85
3.20	The relationship between the figure section and its metadata.	86
3.21	The heading styles adopted by different publishers.	87
3.22	The relationship between heading LLS and metadata.	88
3.23	Examples of header and footer of an article with related metadata.	89
3.24	The metadata present in header section and the class labels to identify LLS.	90
3.25	The footer section of an article with related metadata.	90
3.26	The caption style of figures used by different publishers.	91
3.27	The acknowledgement section and related metadata.	91
3.28	The column layout styles (single and double column) and the paragraphs.	92
3.29	The metadata and class labels of body/paragraph section.	93
3.30	PDF base research article's metadata extraction proposed methodology and comparison with state-of-the-art	101
3.31	The sequence of references under the title of reference section.	108
4.1	PDF base research article's metadata extraction proposed methodology flow diagram	113
4.2	The final result comparison of FLAG-PDFe with the SumPub2016 challenge participants	122
4.3	The final result comparison of FLAG-PDFe with the CERMINE	139
4.4	The final result comparison of FLAG-PDFe with the PDFX	145
4.5	The final metadata extraction F-Score comparison of FLAG-PDFe with sectlabel and PDFX	147

List of Tables

2.1	Metadata extracted by previous approaches.	31
2.2	The analysis of most representative techniques to extract metadata from research articles.	32
3.1	The list of selected Publishers for curated dataset.	46
3.2	The list of selected Journals for the dataset ds3.	47
3.3	Research Articles Logical Layout Structures annotations.	49
3.4	The list of text features adapted to train models for the extraction of metadata	53
3.5	Font names and style extracted by itext library for document font style.	59
3.6	The text blocks showing different font typographical styles adopted by the publishers to present headings and levels.	62
3.7	Features associated to text blocks in order to identify logical layout content of research paper	64
3.8	Time complexity of features extraction algorithms	73
3.9	Evaluation of the features extraction	74
3.10	The time complexity of models to extract each Logical Layout Structure.	97
3.11	The F-Score performance matrix is demonstrating the accuracy of machine learning algorithms for extracting the Logical Layout Structures for the curated training dataset.	98
3.12	The Logical Layout Structures and their associated metadata.	100
3.13	The regular expression to extract and identify different metadata elements, from sample research articles	104
4.1	The results of approaches to extract metadata in SemPub2016.	112
4.2	The performance matrix of models to extract each LLS component on training dataset	119
4.3	The final results and the Confusion matrix of extracted metadata by FLAG-PDFe using evaluation dataset	120
4.4	The Performance Matrix of FLAG-PDFe on the TD of SemPub2017 challenge.	121
4.5	The detail evaluation matrix of Flag-PDFe system to extract different metadata component from diversified publishers. Where R denotes Recall, P denotes precision, and F denotes F-measure	131

4.6	The comparison of CERMINE and FLAG-PDFe to extract meta- data from articles published by the AMC.	133
4.7	The comparison of CERMINE and FLAG-PDFe to extract meta- data from articles published in Elsevier	134
4.8	The comparison of CERMINE and FLAG-PDFe to extract meta- data from articles published by in IEEE	135
4.9	The comparison of CERMINE and FLAG-PDFe to extract meta- data from articles published by MDPI	136
4.10	The comparison of CERMINE and FLAG-PDFe to extract meta- data from articles published by the Springer.	137
4.11	The final metadata element wise comparison of CERMINE and FLAG-PDFx	138
4.12	The publisher wise comparison of CERMINE and FLAG-PDFx . .	139
4.13	The comparison of PDFX and FLAG-PDFe to extract metadata from articles published in the AMC.	140
4.14	The comparison of PDFX and FLAG-PDFe to extract metadata from articles published in Elsevier.	141
4.15	The comparison of PDFX and FLAG-PDFe to extract metadata from articles published by the IEEE.	142
4.16	The comparison of PDFX and FLAG-PDFe to extract metadata from articles published by the MDPI.	142
4.17	The comparison of PDFX and FLAG-PDFe to extract metadata from articles published by the Springer.	143
4.18	The overall metadata extraction comparison of PDFx and Flag-PDFe	144
4.19	The publisher wise comparison of PDFX and FLAG-PDFe	144
4.20	The evaluation of FLAG-FDFe to extract metadata using dataset published by the sectLabel approach.	146

Abbreviations

ACL	Association for Computational Linguistics
ACM	Association for Computing Machinery
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
DBLP	Digital Bibliography & Library Project
DL	Digital Libraries
ESWC	European Semantic Web Conference
GB	Graph Based
HAC	Hierarchical agglomerative clustering
HMM	Hidden Markov Model
IEEE	Institute of Electrical and Electronics Engineers
k-NN	K Nearest Neighbors
LLS	Logical Layout Structures
LSTM	long short-term memory
MDPI	Multidisciplinary Digital Publishing Institute
NLP	Natural Language Processing
OCR	Optical Character Recognition
PDF	Portable Document Format
RB	Rule Based
RFE	Recursive feature elimination
RNN	Recurrent Neural Network
SVM	Support Vector Machine
XML	Extensible Markup Language

Symbols

O	Big O Time Complexity
χ^2	chi-squared
$\exp(-\gamma \ x - x'\ ^2)$	Gaussian/RBF
$\langle x, x' \rangle$	Linear
$(\gamma \langle x, x' \rangle + r)^d$	Polynomial
$P(A B)$	Posterior probability
$O(nsvp)$	Prediction Complexity of SVM Gaussian Kernel function
$\tanh(\gamma \langle x, x' \rangle + r)$	Sigmoid
$O(n^2p)$	Time Complexity of SVM Gaussian Kernel function

Chapter 1

Introduction

This chapter is an overview of the research presented in the dissertation. Initially, this document states the background and motivation of the authors' work regarding metadata extraction from scholarly research articles. Based on the critical analysis of state-of-the-art, the authors formulated the problem statement and research objectives. Further followed by the adopted research methodology and important contributions made by the authors in the field of study. Finally, the chapter concludes with the applications of the proposed research and briefly presents the outline of the thesis structure.

1.1 Background and Motivation

Research plethora is rapidly increasing due to millions of annual online publications of research articles [1–4]. These cross-disciplinary publications are linked through online citation indexes for the research community to establish the relevance of literature. More often scholars cogitate queries based on complex scenarios to search and enlist their required research documents from this colossal scientific resource. Consider the need of the researcher to get the list of research articles that accomplish the following requirements.

- The papers must be published after the year 2016.
- It should compare the results based on time analysis of sorting algorithms available in table caption or figure caption.
- The reference of this caption should reside in sections with the heading as experiments.
- The research should be funded by the European Union research grant.

Another set of requirements of a researcher could be,

- The articles are published in journals, where the title contains the word "cardiology".
- The first author of the article is from the U.S.A.
- The article's title has phrases as "heart diseases" and "hospitalization".
- The article's section has a heading as experiments and has a table with the caption containing words like mortality, morbidity, and children.
- The paper must be published in the last two years.

The famous citation indexes like Google Scholar ¹ or Semantic Scholar², and renowned digital libraries like DBLP³ or ACM⁴ have gained popularity among researchers to search scholarly articles [5]. However, they have limitations when a precise search is required as they reproduce millions of surplus results based on citation indexes and keywords-based search, caused by lack of structural information.

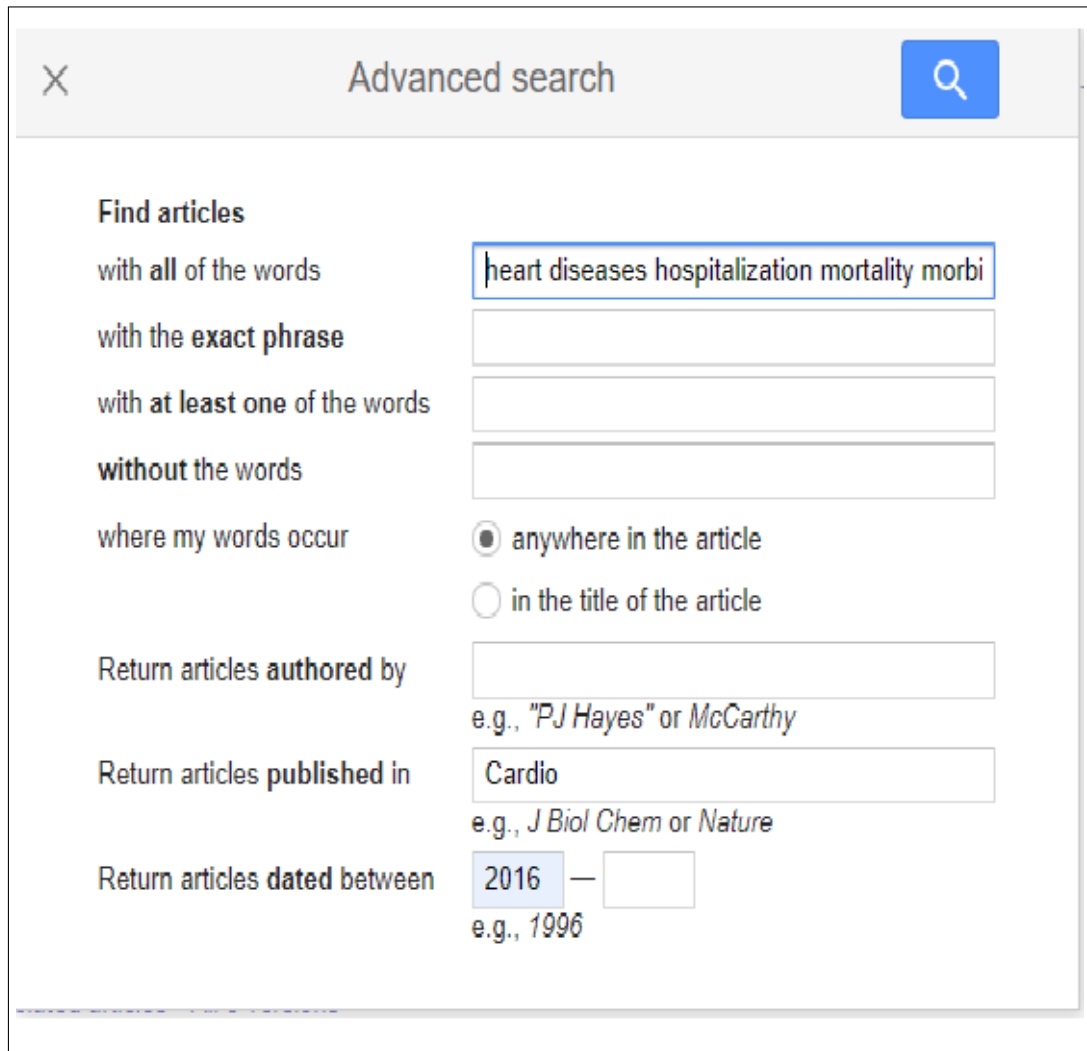
Figure 1.1 is the first example, where we have tried to pass the above-mentioned query to Google Scholar using its advance search option. However, this option has

¹<https://scholar.google.com>

²<https://www.semanticscholar.org/>

³<http://dblp.uni-trier.de/>

⁴<https://dl.acm.org/>



The image shows the 'Advanced search' interface in Google Scholar. At the top, there is a search bar with a magnifying glass icon and a close button (X). Below the search bar, the interface is organized into several sections:

- Find articles**
 - with all of the words**: A text input field containing the text "heart diseases hospitalization mortality morbi".
 - with the exact phrase**: An empty text input field.
 - with at least one of the words**: An empty text input field.
 - without the words**: An empty text input field.
 - where my words occur**: Two radio button options: "anywhere in the article" (selected) and "in the title of the article".
- Return articles authored by**: A text input field with the example text "e.g., 'PJ Hayes' or McCarthy".
- Return articles published in**: A text input field with the text "Cardio" and the example text "e.g., J Biol Chem or Nature".
- Return articles dated between**: Two date input fields. The first field contains "2016" and the second field is empty. Below the fields is the example text "e.g., 1996".

FIGURE 1.1: Google Scholar options for advance search.

limited search fields for a precise search on the metadata of the research article as mentioned below.

- There is no option to search the text available within the figure captions and the table captions.
- There is no option to search the title of the heading.
- There is no option to search the affiliation of the author or country of the affiliation institute or organization.
- The search requires the exact journal's title, however, the title of journal phrase can be searched like in our case 'cardiology'.

- No option to search for research funding or grant.

These options can be helpful for a user to perform a precise search on the metadata of the articles. As shown in Figure 1.2, lack of search options on metadata retrieved a lot of research articles, both by Google Scholar and Semantic Scholar. Which again requires a lot of manual effort by the user to find the desired article.

Therefore, human-understandable research document content (like title, author

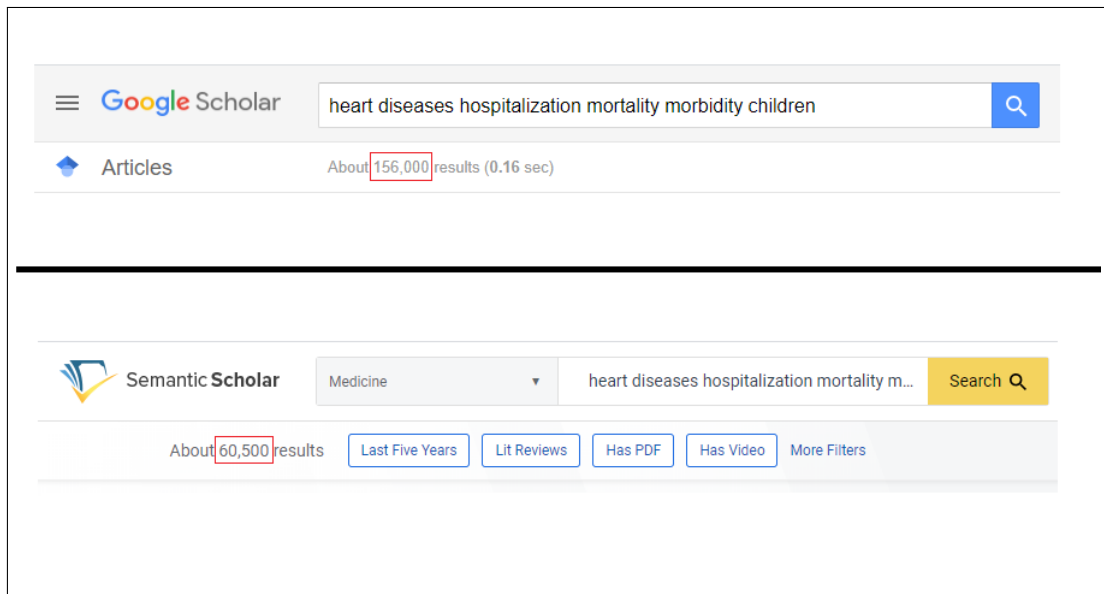


FIGURE 1.2: Google Scholar and Semantic Scholar number of results for the mentioned search query.

name, author's affiliation, affiliated country, author's email, section headings with levels, bibliography, funding agency, table caption, and figure caption) has to be indexed and stored in a machine comprehensible form to facilitate such complex queries.

Figure 1.3 illustrates an example of a research article that shows the metadata elements present at different geometric locations in a scholarly journal. The composition of metadata elements is in diverse textual formats. The highlighted areas in the example figure show the article's publication-related metadata, authors, and author's affiliation. This example shows only a formatting style of a single journal. However, the article's publication information in the different journals is composed

Journal Title ← Talanta 85 (2011) 681–686 → Page Numbers

Contents lists available at ScienceDirect



Talanta

journal homepage: www.elsevier.com/locate/talanta



Title of the article

Determination of cadmium and lead in beverages after leaching from pewter cups using graphite furnace atomic absorption spectrometry

Morgana B. Dessuy^a, Maria Goreti R. Vale^{a,b,*}, Bernhard Welz^{b,c}, Aline R. Borges^{a,b},
Márcia M. Silva^{a,b}, Patrícia B. Martelli^d

⇒ Authors

⇒ Affiliations

⇒ Authors

⇒ Affiliations

^a Instituto de Química, Universidade Federal do Rio Grande do Sul, Av. Bento Gonçalves 9500, 91501-970 Porto Alegre, RS, Brazil

⇒ Country of Affiliation

ARTICLE INFO
 Article history:
 Received 13 February 2011
 Received in revised form 16 April 2011
 Accepted 18 April 2011
 Available online 27 April 2011
 ⇒ Publication Date

ABSTRACT ⇒ Abstract Section

Two simple methods have been developed to determine cadmium and lead in different kinds of beverages and vinegar leached from pewter cups produced in Brazil. Leaching experiments have been carried out with different solutions: beer, sugar cane spirit, red and white wine, vinegar and a 3% acetic acid solution. The solutions were kept in cups with and without solder for 24 h. Lead and cadmium have been determined using graphite furnace atomic absorption spectrometry with deuterium background correction. The limits of detection were 0.05 and 1.4 µg L⁻¹, and the characteristic mass was 1.0 pg and 19 pg for Cd and Pb, respectively. With the developed methods it was possible to determine accurately cadmium and lead by direct analysis in these liquids and to evaluate the leaching of these metals from pewter cups. The results presented in this work show that pewter cups are not cadmium- and lead-free; this point goes against the manufacturers' declaration that their products are lead-free.

© 2011 Elsevier B.V. Open access under the [Elsevier OA license](#).

I. Introduction ⇒ Heading Level 1

Unalloyed tin is too soft to be used for kitchenware, tableware, drinking cups, etc.; hence a small percentage of hardening agents, such as copper, bismuth, antimony and lead are usually added, either alone or in combination, to make it more durable. These tin alloys are known as pewter. Today the composition of pewter for domestic use is rigidly controlled, based on British and European standards, specifying a minimum of 90% tin with the balance being made up of copper, antimony and perhaps some bismuth. The main difference from the seventeenth and eighteenth century ordinances is that the addition of lead is no longer permitted, with the maximum allowed level being 0.5% – in practice usually far less, so that lead should be present as a trace element only [1].

Pewter was part of everyday's life until the 19th century. Eating, drinking, celebrating, lighting rooms and taking communion all required long-lasting, affordable objects [1]. Nowadays pewter is enjoying a Renaissance worldwide, and its use is growing every year; both, consumers and craftsmen, have rediscovered the beauty

and practical function of fine pewter. Pewter production is a very important source of income for São João del-Rei, a Brazilian city situated in the state of Minas Gerais, which is one of the major production centers of pewter artifacts in Latin America [2].

It is well known that heavy metals, such as Cd and Pb, are toxic. The leaching of these elements from pewter cups hence could represent a health hazard. Krachler and Shoty [3] reported the leaching of relatively high concentrations of Tl and Sb from pocket flasks made of pewter. Considering the increasing use of pewter kitchenware in some regions of Brazil, it is of great importance to evaluate the leaching of potentially toxic metals from these utensils. In this work, leaching experiments have been carried out with pewter cups produced in Brazil nowadays, using beer, sugar cane spirit, red and white wine, vinegar and 3% acetic acid as test solutions.

Graphite furnace atomic absorption spectrometry (GF AAS) was the technique chosen for the determination of cadmium and lead, a technique with high sensitivity and tolerance to inorganic and organic matrices, which should make possible a direct determination of these analytes in the selected liquids. The literature is quite limited regarding the direct determination of metals in beverages by GF AAS. Most publications about the determination of metals in wine involve some kind of sample preparation. Mihaela et al. [4] used microwave-assisted mineralization of red and white wine for the determination of Pb by GF AAS; the authors reported an average

* Corresponding author at: Instituto de Química, Universidade Federal do Rio Grande do Sul, Av. Bento Gonçalves 9500, 91501-970 Porto Alegre, Rio Grande do Sul, Brazil. Fax: +55 51 33087304.
 E-mail address: mgrvale@ufrgs.br (M.G.R. Vale). ⇒ Author's Email

0039-9140 © 2011 Elsevier B.V. Open access under the [Elsevier OA license](#).
 doi:10.1016/j.talanta.2011.04.050 ⇒ DOI

FIGURE 1.3: Example1: The research article's metadata.

of diversified styles and layouts. Similarly, Figure 1.4 exhibits an example of the table captions and figure caption present in a research article. The composition of the figure and table caption is dissimilar to their references inside the body of

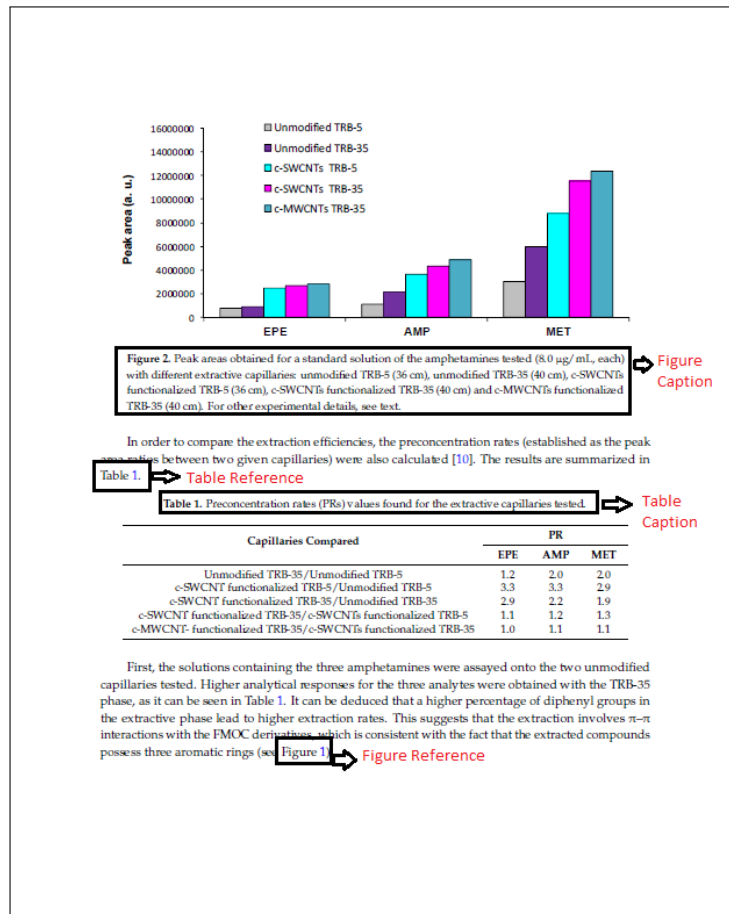


FIGURE 1.4: Example2: The research article's table and figure captions.

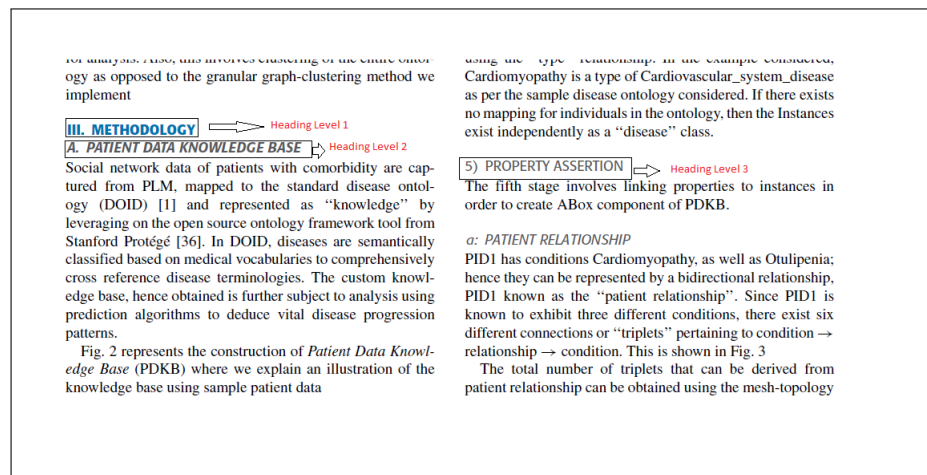


FIGURE 1.5: Example3: The research article's headings and sub-headings with levels.

a paragraph. Figure 1.5 demonstrates an example of the composition of headings and heading levels. The formatting styles of section headings help identify different levels of subheadings. These differences in the textual format and geometric

location of text bodies are helpful to identify different metadata elements of a research article. A similar visual aid can be employed in the metadata extraction technique. However, the composition and layout style of metadata elements is different for diversified publishers.

In this context, metadata extraction tools have gain popularity to extract and store machine recognizable research article's content to furnish precise semantic queries. Recently, research community has deemed metadata extraction as a challenge, and significant number of conferences has been organized (like SemPub⁵, CLSA⁶ OKE⁷, QALD⁸, RecSys⁹) to improve the quality of the linked data [6].

There are popular tools that are available online and offline to extract metadata from research articles. To extract metadata from full research article, rule base approach PDFX¹⁰ and machine learning based approach CERMINE¹¹ is available online, which stores extracted metadata from a PDF-based research article in the form of an XML file. Similarly, ParsCit¹² and Grobid¹³ are renowned techniques whose source is available online to extract bibliographic metadata. These techniques along with others are comprehensively presented and evaluated in our thesis.

Research document structure analysis and information extraction have been a well-researched area due to the increase of publications in diversified domains. Information extraction methods are essentially constructed upon machine learning and heuristic-based approaches. Machine learning technique relies on a group of fine-tuned parameters to learn good feature representations for structure extraction. These techniques are sub-categorized into supervised machine learning (Classifiers), unsupervised machine learning (clustering), Ensembled, and deep learning (neural networks), etc [7]. However, it requires a large tagged pre-trained

⁵<https://github.com/ceurws/lod/wiki/SemPub2017>

⁶<http://challenges.2014.eswc-conferences.org/index.php/SemSA>

⁷<https://project-hobbit.eu/open-challenges/oke-open-challenge/>

⁸<https://project-hobbit.eu/challenges/qald-8-challenge/>

⁹<http://2017.recsyschallenge.com/>

¹⁰<http://pdfx.cs.man.ac.uk/>

¹¹<http://cermine.ceon.pl/index.html>

¹²<https://github.com/SeerLabs/CiteSeerX/tree/master/src/perl/ParsCit>

¹³<https://grobid.readthedocs.io/en/latest/>

dataset; it has limited aspects of natural language processing and a limited performance guarantee. Initial work exhibits that heuristic-based approaches perform better because it is built on natural language processing and regular expression. This approach is constructed on a pre-defined set of rules and requires domain knowledge for diversified data. Yet, the rules have to be enhanced by every time once documents from a new publisher are extracted.

However, the document layout and elements are composed of geometric location and font properties of the text, which varies for different publishers. The text in a research document has different font attributes, which can uniquely identify a group of elements like Title, paragraph, authors and affiliation, headings, table captions and figure captions, etc., these distinct features are in discussed detail (in Section 3.2.4). An automated scalable approach is desired that can constantly and sufficiently perform whenever a document from a new publisher is added to the dataset. Current information extraction approaches utilize comprehensive updatable tags and rules to train the model, however textural features can be helpful to identify and classify the distinct group of the document elements on the fly. Therefore, we have proposed a methodology, which can effectively utilize the textural properties of the text to derive distinct text features to identify various research document elements with minimum prior training.

1.2 Research Challenges

This section will discuss the research challenges that are answered in the thesis.

1.2.1 Problem Statements

This thesis has formulated the following problem statements.

1. The previous metadata extraction approaches are mostly built for the research articles that are from a single publisher, hence they produce optimum

test results in a controlled environment, Their performance reduces in cases when articles from different publishers are tested.

2. The features being used in the state-of-the-art techniques are dependent on a particular dataset and those cannot be considered as generic features that can be used on the articles published by diversified publishers.
3. There are limited benchmark annotated datasets besides the evaluation tool to check the accuracy of the results of previous approaches on the diversified datasets.

1.2.2 Research Questions

The above problem statements have led us to explore the answers to the following research questions.

RQ1: Which method will perform efficiently and accurately to extract logical layout sections LLS for research papers with diversified publication styles? (This research question has been answered in chapter 3)

RQ2: How to devise a strategy to extract features from papers published by the diversified journals and how to select an optimum set of features from the identified features? (This research question has been answered in section 3.2)

RQ3: How to develop a comprehensive model to effectively extract metadata that is trained on research articles from multiple domains of different publishers? (This research question has been answered in sections 3.4 and 4.2)

Based on the comprehensive evaluation of all the research questions, we have finally proposed a comprehensive approach that will suggest the benchmark dataset, best features to identify metadata, machine learning approach to extract sections and structure of research paper, and the approach to extract metadata from a PDF-based research paper.

1.2.3 Research Objectives

1.2.3.1 Objectives

To construct a scalable approach that will automatically extract the metadata and the content information from the PDF-based research articles published online from diversified domains by the different research publishes.

1.2.3.2 Significance

More often scholars cogitate queries based on complex scenarios to search and enlist their required research documents from this colossal scientific resource, like:

“The articles in the biomedical literature that must contain leukemia in their title or in the sections heading and the first author has an affiliation with a cancer research institute in the U.S.A and the paper is published within last two years.”

“The articles that possess a phrase merge sort algorithm vs. insertion sort algorithm within a table or figure caption, and the reference must reside in the section with the heading experiment or result, and the research must be funded by a European Union grant.”

“Research papers published before June 2016 on the topic of technology impact that must have Brexit and Germany as the heading of the sections and one of the author’s affiliation country must be China.”

However, renowned digital libraries and search engines have limitations, when a precise list of relevant research papers is required, but millions of surplus results are reproduced caused by the lack of structural information. We have developed a system that extracts metadata and structural content of the research papers of diversified domains from different publishers. Which will provide the basis for

search engines to facilitate precise semantic queries on research document content for accurate retrieval of relevant literature.

1.3 Proposed Methodology

This section describes the overview methodology of our proposed technique, which focuses on metadata extraction from PDF-based research articles from diversified domains of different publishers. Figure 1.6 show the proposed methodology to perform the metadata extraction task, the chapter 3 provides the complete detailing of extraction algorithm. The initial part of this section explains that overall research adopted a renowned research methodology technique. Then discusses the identification of the benchmark datasets and the creation of a diversified curated dataset. The next part explains the proposed features that will be utilized for the extraction of the logical layout structures (LLS). The last part describes the extraction of metadata from the logical layout structures.

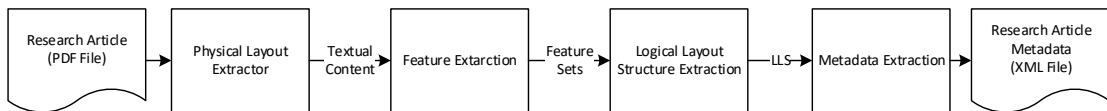


FIGURE 1.6: The overview of task performed by the proposed methodology.

1.3.1 Research Methodology

A three-phased and eight-step model has been adopted for the persuasion of the current research as it was suggested by Kumar [8], we modified it slightly as required accordingly. The maneuvers which were used while the conduction of this specific research, are mentioned below, along with the delineation between the current research and Kumar’s model in the form of a Figure 1.7.

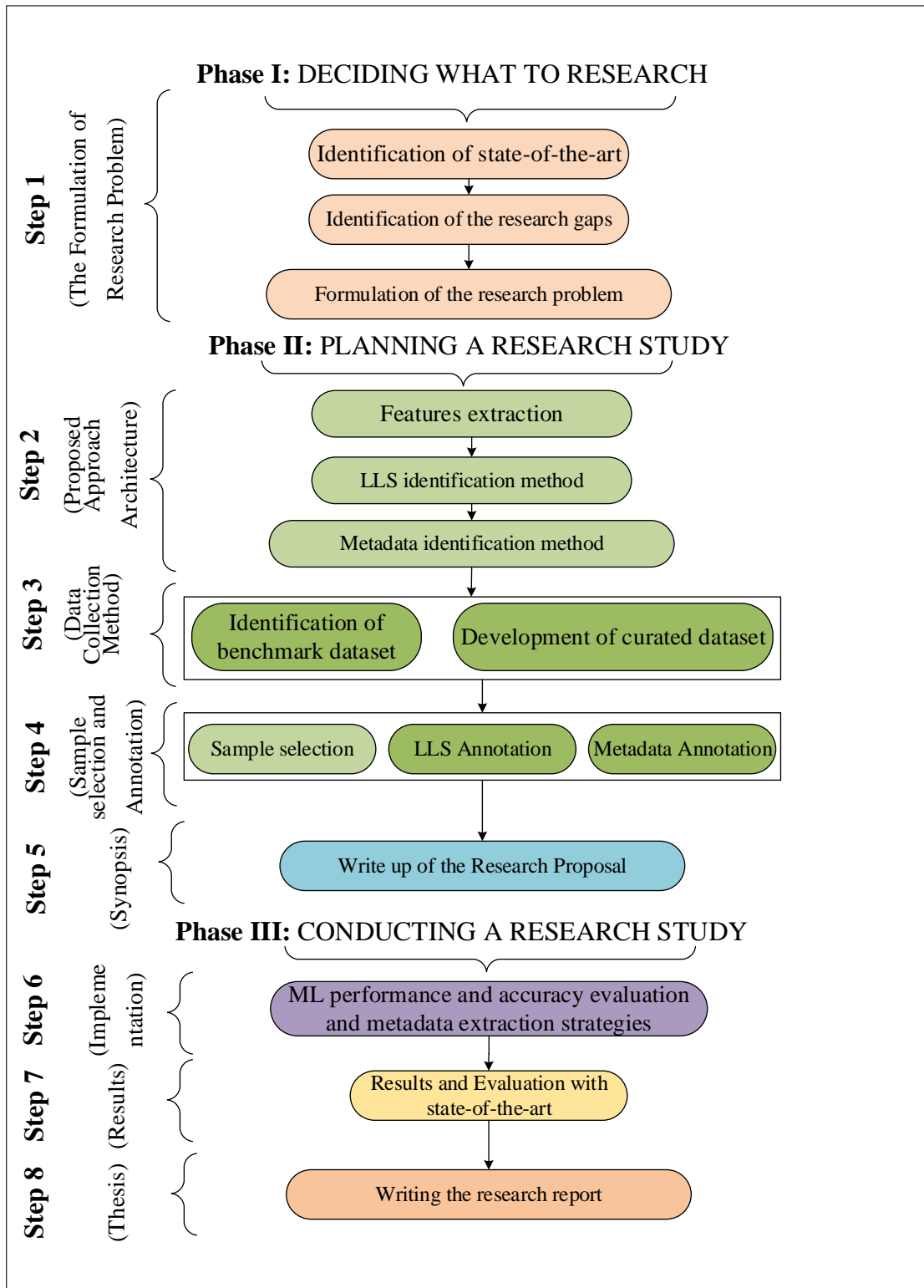


FIGURE 1.7: The proposed research methodology steps.

Phase I: DECIDING WHAT TO RESEARCH

Step 1: The Formulation of Research Problem:

At this step, the following tasks are performed (1) Identification of stat-of-the-art (2) Identification of the research gaps (3) Formulation of the research problem.

Phase II: PLANNING A RESEARCH STUDY

Step 2: Proposed Approach Architecture:

In this step, we divided the technique to solve the problem statement into sub-tasks. Initially, after performing the comprehensive analysis of the layout style of research articles with diversified publication styles, we have proposed a novel approach to extract generic features that can be used to identify different layout contents of a research article. The next task is to develop a strategy that can be used to identify the different logical layout structural components. And the final task is to extract desired metadata from research articles.

Step 3: Data Collection Method:

In step 3, the data collection was done using two methods, (1) Identification of benchmark dataset and gold standards based on literature review (2) Development of curated dataset with the help of domain experts.

Step 4: Sample Selection and Annotation:

In this step, we equally distributed the selected dataset after careful evaluation of the article's content. A team of annotators was hired to annotate the research article content at two stages (1) Annotation of logical layout content to train and test the machine learning algorithms (2) Annotation of desired metadata for evaluation purposes.

Step 5: Synopsis:

In this step, we developed the synopsis report after the initial implementation of the proposed architecture.

Phase III: CONDUCTING A RESEARCH STUDY

Step 6: Implementation: At this step, we evaluated machine learning techniques on the comprehensive datasets and feature to extract the LLS and define

the strategy to extract the metadata .

Step 7: Evaluation and Results: The results of each part presented in methodological steps are evaluated and discussed. Finally, The comparison with state-of-the-art approaches is conducted on the benchmark.

Step 8: Thesis:

As the final step, we produced a dissertation that includes the details of all the above-mentioned steps in a report form. Here, we discuss, analyze, and critically evaluate our proposed approach.

1.3.2 Identification of Datasets

To evaluate our proposed methodology regarding the logical content extraction of PDF-based research documents, we required a comprehensive diversified dataset. As mentioned earlier that previous techniques utilized datasets from a single domain or publisher, which will not be suitable for an extensive evaluation of our proposed technique. We selected the dataset of research articles from different renowned publishers with diversified publishing styles, these two datasets are presented in (sec 3.1.2 and 4.1.1). Hence, we have constructed a comprehensive and diversified dataset that covers a wide range of domains, journals, and publishers. Furthermore, to evaluate our technique with state-of-art approaches and compare our technique on gold standards, we have selected three benchmark datasets, published freely online by the authors of those techniques for justified comparison. The first benchmark dataset proposed by sectlabel (sec 3.1.2) is utilize by Constantin et al [9] and the technique name is PDFX, this dataset consists of research articles from Elsevier publisher, this dataset is described in (sec 4.1.1). The Second dataset is a benchmark published in “Semantic Publishing Challenge” by Sem-Pub2016 based on workshop papers of CEUR-WS.org held at ESWC conference, this dataset is described in (sec 3.1.1).

1.3.3 Physical Layout Extraction

A PDF file is composed of raw binary data without any associated metadata and logical structural information that identifies different layout categories of the content. Therefore, the first process is to extract textual information from the PDF file. At this stage, we used itext [10] open-source java library that provides a faster and reliable method to extract PDF files. Unlike other processing tools that extract text as text glyph or stream of characters, itext extracts the chunk of textual elements that reduces resources and computational cost. Further, itext implements an advanced strategy to extract structural components that are text chunks, font properties, geometric locations, raster images, page numbers, and vector graphics.

The text chunks are retrieved, encapsulated in boundary boxes that identify their geometric position in the form of (x, y) coordinates on the page along with height and width. The itext library returns font attributes like font name, font size, bold, italic, orientation, etc. We used these attributes to generate the font properties feature set.

1.3.4 Features-set Identification

Based on text content information extracted for a PDF file, this stage extracts and evaluates different features that identify logical content and metadata of a research article. Therefore, we have analyzed different formatting styles of publishers and established that these layout and formatting styles can be used to extract metadata from research articles. Since this important information and layout components require annotations, therefore, we have categorized the formatting styles into two types of structural components, one is the physical layout and the other is logical layout structure components. The physical layout is based on individual article's distinct features, which consist of textual properties, geometric boundaries,

paragraphs, column styles, floating objects, headers, and footers, etc. The logical layout structures (LLS) are generic formatting features to identify different parts, contents, and sections of an article that are required by the publisher. The details of the proposed architecture are explained in detail in the section 3.3.

1.3.5 Logical Layout Structure (LLS) Extraction

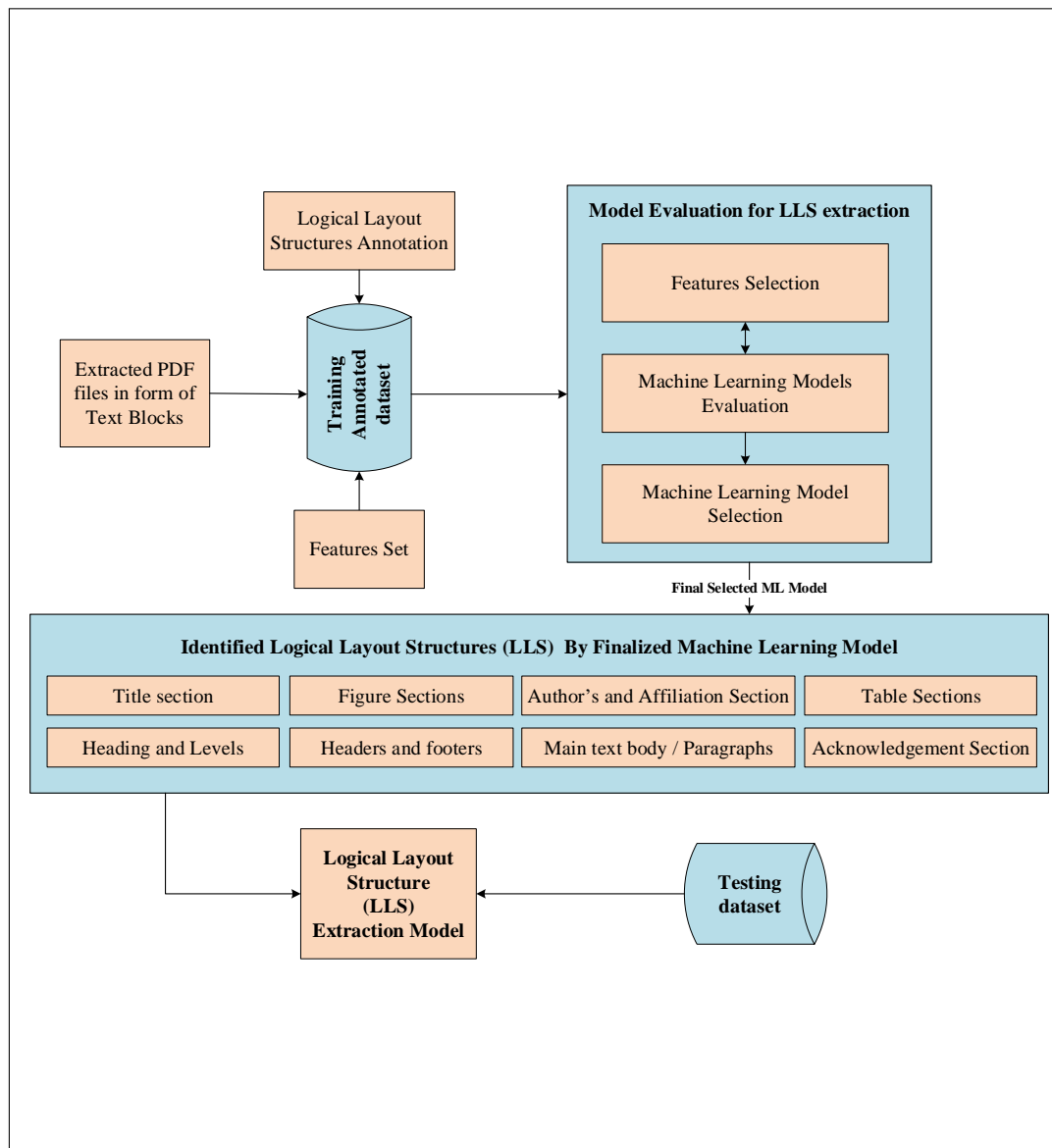


FIGURE 1.8: The proposed methodology to extract research paper sections in order to evaluate Logical layout structure LLS

A research paper structure is based on different sections, we proposed a methodology to extract the research paper sections as shown in Figure 1.8. The benchmark

dataset consists of section headings as class labels along with features set will be given as input to this process. In chapter 3, we shall evaluate different machine learning algorithms on the given dataset.

The classification techniques like Support vector machine, K-Nearest neighbor classifier, Bayesian classifier, Rule-based classifier, Decision tree induction, and Ensemble classifier will be evaluated both theoretically and experimentally, K-Cross Validation technique will be used to train the model to avoid overfitting and underfitting on the given input dataset. The Scores of all the techniques to identify different sections will be analyzed and best performing techniques will be reported and used.

The final output of this stage will be different logical layout structural (LLS) components of a research article. These components are the Title section, Author's section, table section, figure section, headings, header and footer section, and paragraphs. These sections are present in different formatting and layout styles. However, by using diversified feature sets and comprehensive machine learning techniques evaluation, we have been able to select a machine learning algorithm that can efficiently and accurately identifies different logical layout components for the articles with diversified publication styles.

1.3.6 Metadata Extraction

The metadata of the research paper is present both in structure layout and within the body of the sections. For this purpose, we have applied heuristics that are generalized in nature, so that they can extract metadata for research articles that has different formatting and publication styles. In a few cases, we used the aspects of natural language processing (NLP) and complex regular expression to find metadata from sections.

This stage takes logical layout structural components on a research article as input and further applies heuristics and regular expression to extract the metadata. This resulting metadata is stored in the form of an XML file and is evaluated on gold standard datasets as shown in the Figure 1.9.

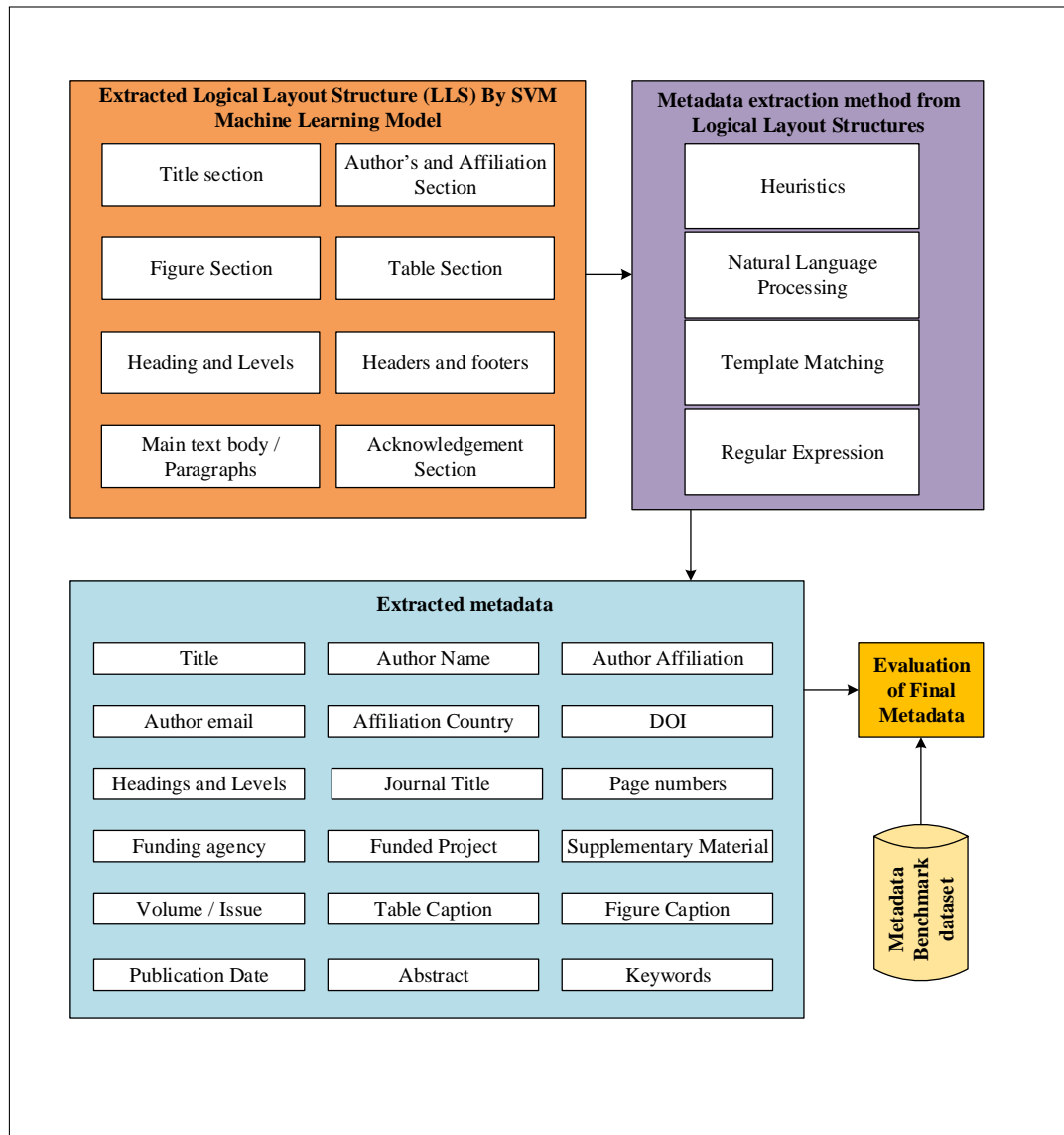


FIGURE 1.9: The proposed methodology to extract metadata of research paper from logical layout sections

This is the final stage of our proposed methodology, which identifies metadata components from different logical layout sections. The metadata identified at this stage is Title, authors, authors affiliation, country of affiliation, author email, abstract, keywords, journal name, publication date, pages, version and volume

number, headings and levels, figure captions, table captions, bibliographic count, funding agency, funded project, supplementary material, and endnotes. We will perform a comprehensive evaluation of this extracted metadata with state-of-the-art techniques on diversified datasets in the sections 3.4 and 4.2.

1.3.7 Applications

This research will facilitate text mining and document analysis for research articles from variety of domains. The huge collection of metadata form the articles of different publishers can be extracted and indexed. The following can be the use cases for numerous real-world applications in this context.

1. Digital Libraries (DL): provide the basis for DL for diversified domains of different publishers.
2. Scalability support: automatic content retrieval from diversified publishers.
3. Semantic queries: support precise search on research document's content.
4. Document metadata indexing: efficient management of metadata to store and retrieve research paper content.
5. Recommendation of literature: analysis of researcher's publication.
6. Supports linked data: content references in sections and linkage other to associated content.

1.4 Thesis Outline

The thesis is structured as follows. After the introductory chapter 1, in chapter 2 we shall describe the current state-of-the-art concerning scientific document analysis and automatic metadata extraction. Chapter 3 contains the overall methodology of the proposed research. Section 3.2 provides the details of features evaluation and extraction based on geometric and textual properties of extracted text.

In section 3.3.9, the machine learning algorithms, theoretical evaluation, and time complexities are discussed. Chapter 4 contains the results and evaluation of the proposed approach. In Section 4.1, we extracted metadata of articles published in the CEUR semantic conference challenge, and in section 4.2, we extracted metadata of the research article from the curated dataset with articles from diverse publishers. We evaluated our purposed techniques with state-of-the-art. Finally, the conclusion is drawn in chapter 5 of this thesis.

Chapter 2

Literature Review

The metadata and structure extraction from PDF-based documents is a well-explored research area since the emergence of the initial online search engines like CiteSeerx to find scholarly articles [11, 12]. There are many efforts in the literature that have used such metadata extraction as mentioned in [13–20].

A PDF file is stored in raw binary data form and lacks structure information tags or metadata that identifies different layout components of the document. Tools such as iText, PDFBox, and JPod extract text as the stream of characters from a raw file and require further processing, as extracted output has incorrect reading order and intercepting objects (like decorations, figures, and tables, etc.). Another prominent obstacle is the diversified nature of the document layout styles and textural features adopted by different scientific publishers. Numerous approaches developed to extract the structure layout of research documents are broadly categorized into heuristic-based techniques, machine-learning techniques, and graph-based techniques. Early document text extraction models used XY cut algorithm a rule-based technique [21, 22] to derive the reading order from image segmented pdf files [23]. Modern approaches utilize the bottom-up Docstrum algorithm to generate the correct reading order of text content due to its adaptation and simplicity by identifying near neighbors of text characters [24]. As presented in Figure 2.1, the reading order of a two-column article is from top to bottom

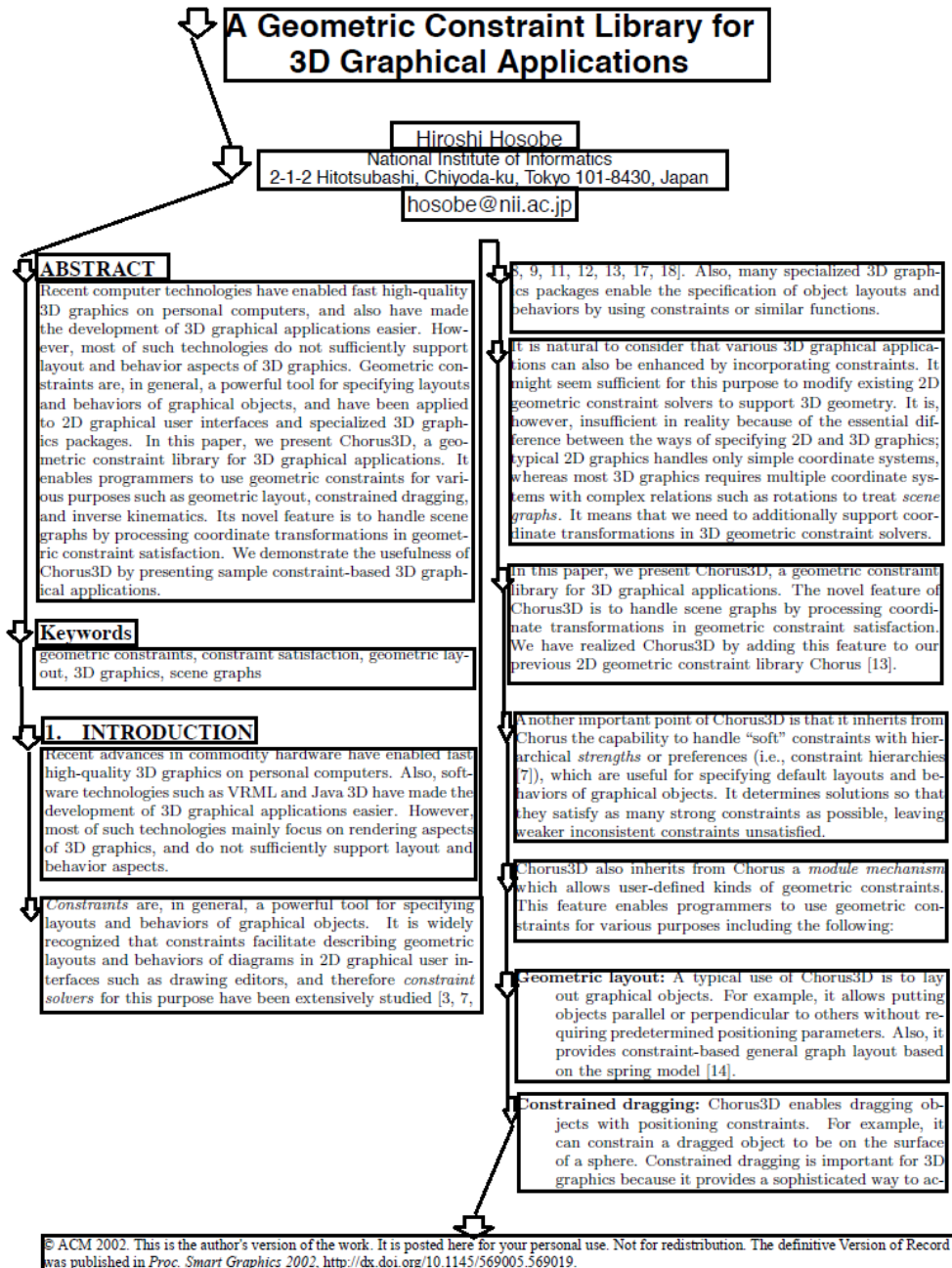


FIGURE 2.1: The correct reading order of an article.

and from the left column to the right column. Initially, document structure and content were extracted using template-based techniques. Researchers proposed supervised machine learning techniques and initially Hidden Markov Model (HMM)

[25] technique was used for text mining, however, later studies suggested that linear conditional random fields (CRF) [26] can produce optimum results [27] as CRFs performs well with sequences with dependent features. Bijari et al. [28] in their study introduced a hybrid algorithm based on heuristics and clustering, using BB-BC and k-means to improve k-means shortcomings in text mining.

ParsCit adopted conditional random fields (CRF) to extract bibliographic metadata from research documents using a package that parses the tokenized sequences for labelled bibliographic strings [29]. To find citation context, the approach employs heuristics on plain text to retrieve reference strings. Recently, ParsCit improved its technique by adopting long short-term memory (LSTM) [30, 31] that is a variant of the recurrent neural network [32]. The neuron layers are a linear chain of conditional random fields (CRF). This technique used the character-based and word embeddings instead of handcrafted features.

The CiteCeerX¹ team introduced PDFMEF that encapsulates artifacts of their existing approaches in a framework [33]. For full text extraction, the framework leverages two open-source java libraries Pdfbox [34] and pdflib TET [35]. It uses GROBID [36] to extract header metadata from research articles and ParsCit to extract references and citation context. The framework uses pseudocode detector [37] to extract and identify algorithms and PDFFigures [38] to identify and retrieve figures and table from scholarly research articles. Additionally, the document classification algorithm [39] is employed to improve the quality of indexing and storage of articles for effective retrieval from digital libraries.

Figure 2.2 presents a visualization regarding the evolution of structure and metadata extraction approaches on a timeline. In the following subsections of this chapter, we present a detailed literature review of different techniques used by the state-of-the-art to extract metadata from scientific research papers. These techniques are categorized into three parts that are rule-based, machine-learning-based, and graph-based approaches.

¹<http://csxstatic.ist.psu.edu/>

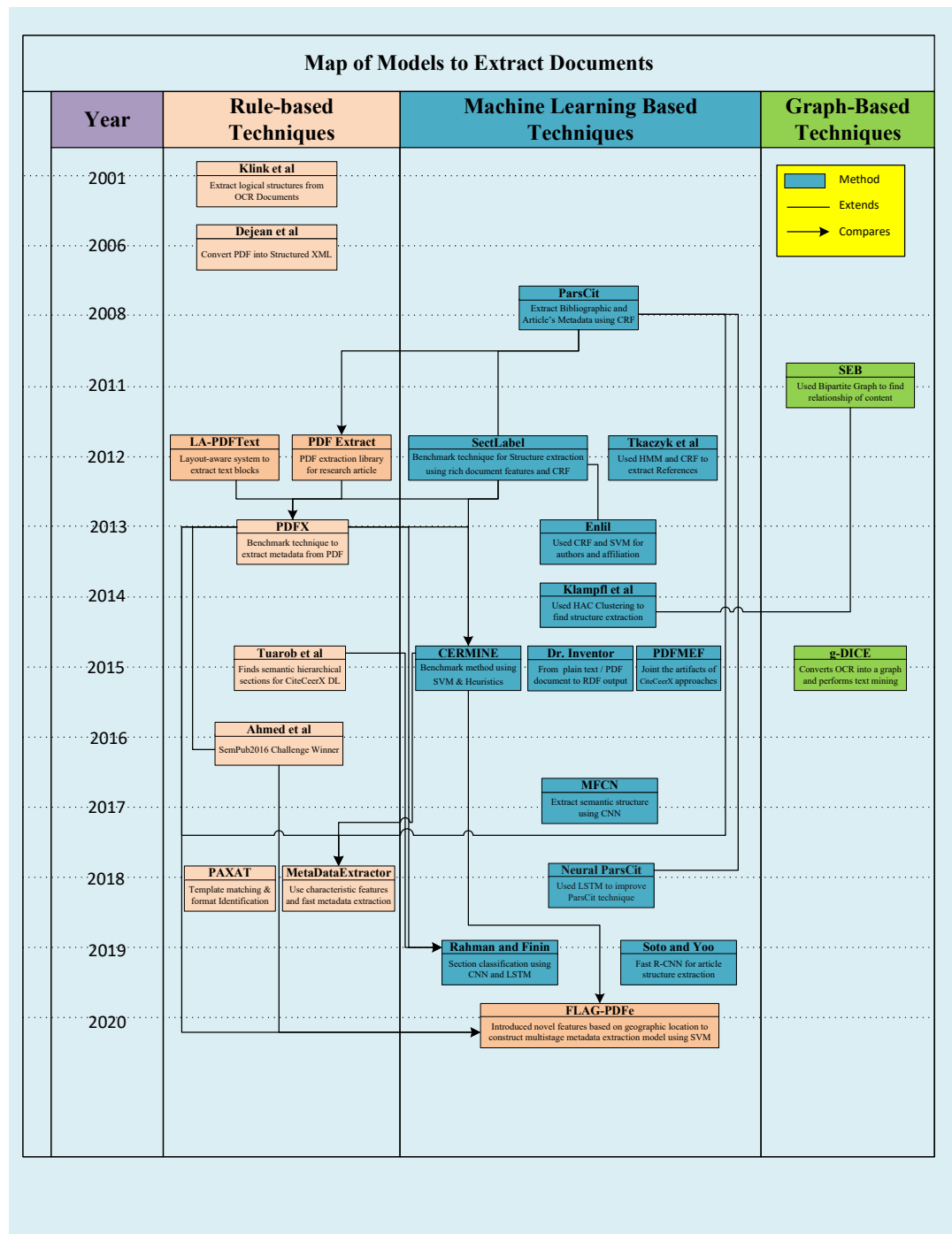


FIGURE 2.2: Timeline of document's metadata extraction models.

2.1 Rule-based Techniques

Rule-based approaches require a dataset to build a set of rules constructed upon natural language processing, regular expression, and domain knowledge.

Constantin et al. proposed a two-stage rule-based system (PDFX) using text feature and characteristics for conversion of PDF artifact documents into XML structure [9]. Rather than using the paradigm of template-matching, they constructed their approach on rules that are derived from font and layout parameters. We evaluated and compared our approach with PDFX on our selected dataset and benchmark dataset, detailed discussion is presented in section 4.2.

Klink and Kieninger proposed a rule-based approach to extract logical structure from OCR-based documents to facilitate office automation [40]. The rules are based on the fuzzy combination of textual and layout features. They evaluated their approach to extract physical and logical layouts on documents from two different domains; the first one is 89 documents consisting of business letters and the other consists of 979 journal pages from the University of Washington.

Similarly, Dejean and Meunier proposed an approach for transforming PDF legacy files into a structured XML file [41]. The system converts the raw stream of bitmap and text extracted from a PDF document into a logically structured XML document. The main objective of this system is to identify text components into words, line segments, and blocks by applying heuristics and distance based upon the geometric position of characters.

Ramakrishna et al. introduced LA-PDFText [42]. It's a layout-aware system that employs textual content to facilitate text mining in the biomedical domain research articles. The system detects adjoining text blocks and classifies them into the rhetorical categories using a rule-based method.

Azimjonov and Alikhanov presented a rule base approach "MetaDataExtractor" [43]. It only extracts the title, abstract, keywords, body text, conclusion, and reference in a faster way. They compared their approach with PDFX, GROBID, and

ParsCit on the selected dataset with better accuracy. Initially, the system uses characteristic features to identify the input PDF document as a research article or nonresearch document, then it extracts metadata, and finally, it sorts and stores the content in the form of an XML file.

Jiang et al. introduced their approach "*PDF Article eXtraction and Analyzer Tool*" (PAXAT) [44]. The approach is based on template matching and implicit format identification to extract metadata from digital research articles. The approach only extracts the title, authors, affiliations, and author-affiliation from articles published in ACM, ACL, Springer, IEEE, and arXiv. The approach is compared with Enlil on the selected dataset.

Recently, Ahmad et al. constructed a heuristics-based approach with an effective combination of tagged and plain text-based information extraction techniques [45]. The system is based on PDFX to extract structural metadata in XML format and further applies regular expression and heuristic to extract desired metadata. The approach outperformed approaches on the CEUR dataset in an open challenge Sempub2016. The system extracts metadata like 'Author', 'Sections', 'Figure', 'Table', 'Citations', and 'Supplementary materials'. We also compared our system with this approach on the same input and output parameters on the same dataset, which is discussed in detail in Chapter 4.

These approaches immensely rely on regular expressions and text pattern matching. Heuristics-based approaches require a predefined set of rules and text patterns to identify different elements of the research document. Hence, a huge set of rules has to be maintained for diversified datasets. Therefore, the underlying problem with these approaches makes them hard to manage any conflicts for the overlapping rules. Furthermore, domain-specified knowledge is required to apply them to a diverse dataset.

2.2 Machine-Learning Based Techniques

The previous approaches used classification, clustering, sequential, semi-supervised, and neural network-based machine learning models to extract metadata from research articles. The machine learning approach using classification models for metadata extraction requires a pretraining of the model is required by tagging data based on unique features. A limited number of unsupervised machine learning algorithms are used for metadata extraction as clustering algorithms are not well suited in such cases. Deep learning-based algorithms RNN and CNN are used for metadata extraction and OCR-based files' feature labeling. This subsection discusses these approaches in detail.

2.2.1 Classification

Tkaczyk et al. proposed their framework CERMINE [46], to compare its bibliographic metadata and layout metadata extraction technique with popular approaches like PDFX, GROBID, ParsCit, and PDFExtract [47]. The framework extracted metadata using SVM and heuristics and extracted bibliographic metadata using K-means clustering and CRF. We evaluated our approach with CERMINE on the selected dataset and sectLabel dataset that we have presented in section 4.2.

Granitzer et al. investigate the use of two-stage Support Vector Machines and Conditional Random Fields to remove noisy and imperfect data with a combination of heuristics on real-world systems like ParsCit, CRF, and the Mendeley Desktop, for automatic extraction of crowdsourced bibliographic metadata [48].

Tkaczyk et al. presented an adaptive modular workflow for extraction of metadata from born-digital and scanned scholarly articles [49]. The technique distributes the full content of the article into zones and it further uses 33 features to implement

reference tokens, The approach implemented a smoothing technique to implement HMM and CRF as a reference parser for bibliographic content.

Do et al. introduced Enlil that uses CRF to identify authors and author affiliations and SVM to discover the relationship of authors with their respective institutions [50]. The technique uses three datasets to evaluate their approach using ACM Digital Library, Association for Computational Linguistics's corpus (ACL), and selected journal articles from cross-disciplines. Finally, the study identifies the co-author network from extracted information.

2.2.2 Clustering

Kiss and Strunk proposed an effective system to detect sentence boundaries using unsupervised machine learning methods [51]. To detect language-independent sentence boundaries by using abbreviations, sentence starters, initials, and ordinal numbers on eleven different languages. They evaluated their approach with the state-of-the-art using a corpus of articles, newspapers, and periodicals.

Klampfl et al. proposed an unsupervised approach hierarchical agglomerative clustering (HAC) [52], a bottom-up approach to extract presentation optimized scientific documents without structural information [53]. The approach extracts adjacent text blocks from the PDF file by identifying the geometrical relationship and further classifies them to originate logical structures. The back draw of the approach is its time complexity $\mathcal{O}(n^3)$ and inability to extract metadata from articles with diversified publication styles.

Tsai et al. used an unsupervised bootstrapping algorithm for categorization and identification of the scientific research by transforming citation contexts into coherent concepts. The technique improved the concept of citation graphs by identifying

techniques, concepts, relations, trends, and applications [54].

Epp et al. proposed a grammar-based active wrapper and unsupervised aspect extraction technique to extract results that are reported in statistical form, from the scientific articles in the domain of Psychology [55]. The approach is evaluated on the CORD-19 dataset using the pipeline STEREO wrapper. Budhiraja et al. conclude that the decision tree has performed the best to extract the headings from digital research articles [56].

2.2.3 Neural Network (Classification)

Soto and Yoo presented an approach to visually segment important regions of scientific articles using object detection method with contextual characteristics [57]. The system adapts the technique Faster R-CNN [58] for document layout discovery to improve the accuracy of region detection. Initially, the system converts the PDF file into a JPEG and extracts the image features by applying the ResNet-101 convolutional network and then use Faster R-CNN for metadata identification.

Rahman and Finin proposed a technique to identify different sections and classify them to understand their meaning in the research document [59]. The model extracts the logical and semantic structure using deep learning techniques. Initially, the system identifies different sections or headings using top-level, sub-section, and sub-sub section headers by using "Recurrent Neural Network" (RNN) [60] and "Convolutional Neural Network" (CNN) [61]. Next, the system performs semantic identification of the sections using "Long short-term memory" (LSTM) and CNN. The comparison of metadata extraction is made with PDFx [9] and the classification of sections are made with Tuarob et al. [37] on two datasets.

Rizvi et al. used Mask R-CNN to develop an approach DeepBiRD inspired by the human visual perception and identifies references based on layout features in a scientific publication [62].

Madisetty et al. proposed a feature-based approach to extract inline mathematical expressions that uses a hybrid algorithm that combines Bidirectional Long Short Term Memory networks (Bi-LSTM) and Conditional Random Fields (CRF) [63].

Most recently, Boukhers et al. proposed an approach MexPub based on Mask R-CNN that is trained on a large COCO's PubLayNet dataset [64]. This approach extracts metadata from the initial page of a PDF-based research article, by converting it into a JPEG image. The approach works on the extraction of segments of images and classifies them into different metadata elements. Finally, the approach is evaluated with the GROBID technique.

2.3 Graph-based Text Mining Techniques

According to Washio and Motoda [65], subgraph categories, graph invariants, subgraph isomorphism, solution methods, and mining measures are the theoretical basis of graph-based approaches for text mining. Additionally, it also illustrated graph-theoretical approaches for text mining are based on inductive logic programming, greedy search, kernel function [66], inductive database, and mathematical graph theory.

Cao et al. proposed a graph-based framework to extract the logical layout structure from large documents of research articles [67]. The proposed hierarchical approach inserts the physical objects in sequential order in a tree and traverses them to extract the logical document hierarchy. The corpus is composed of Chinese financial, English financial, and arXiv datasets.

Gao et al. in their approach use a bipartite graph to extract metadata from PDF files and named it SEB [68]. The representation of a graph is utilized as a common structure to perform various tasks that include recovery of the reading order, association among figures and captions, and extraction of metadata. The

method of optimal matching (OM) is used to discover the global optima while extracting page level and document level structural components. The system finally stores and sorts a PDF file Table of content (TOC) and metadata into an XML file.

TABLE 2.1: Metadata extracted by previous approaches.

Models	Title	Author	Affiliation	Country	Email	H1	H2	H3	Funding	Table	Figure	Ref
Machine-Learning Based Techniques												
MexPub	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
MFCN	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✓
CERMINE	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓
Tuarob et al	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗
Klampfl et al	✓	✓	✓	✗	✗	✓	✓	✓	✗	✓	✓	✓
SectLabel	✓	✓	✓	✗	✓	5 ✓	✓	✓	✗	✓	✓	✓
ParsCit	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓
Dr. Inventor	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓
PDFMEF	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓
Enlil	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓
Tkaczyk et al	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓
Rules Based Techniques												
Ahmed et al	✗	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓	✗
PDFX	✓	✓	✗	✗	✓	✓	✓	✓	✗	✓	✓	✓
LA-PDFText	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓
PDF Extract	✓	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗
Déjean	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗
Graph-Based Techniques												
g-DICE	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✗	✗
SEB	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗

Santosh proposed a graph base text mining technique g-DICE using “*document information content exploitation*” [69]. The concept is based on three consecutive phases that are graph initialization, graph mining, and graph learning. It converted the content of the image-based digital document (OCR) to a graph and then evaluates the text fields based on the graph mining approach. The main objective of the system is to extract the content of the header, footer, vis, body,

and table present in real-world industrial documents. The results show 90.80% recall and 86.64% precision.

Finally, the list of metadata and content extracted by each model is presented to have a better overview and comparison of all the models, as shown in Table 2.1. The next section provides the comparison of previous approaches in structured form.

2.4 Comparison of Approaches

Table 2.2 shows a comprehensive overview and comparison of most representative models that extract metadata from digital documents. In some case, the information related to datasets and results is not available or clearly mentioned in the literature. The technique adapted (machine learning, heuristic, and graph-based) is mentioned against every model along with the dataset, type of documents extracted (like PDF-based research documents, OCR-based documents, and newspapers, etc) and the size of the dataset are shown after an in-depth analysis of the literature. The year of publication, along with consolidated results and result evaluation parameters is shown.

TABLE 2.2: The analysis of most representative techniques to extract metadata from research articles.

Models / Authors- Year	Size / Dataset	Evaluation Parameter / Result	Strengths	Limitations
Machine-Learning Based Techniques				
MexPub- 2021	100/ SSOAR Pages	Micro F1 / 0.613	Mask RCNN is ap- plied that is trained on a COCO's Pub- LayNet dataset	The dataset is of 100 pages and the docu- ment is initially con- verted to JPEG from PDF.

Continued on next page

Soto & Yoo-2019	822/ Journal Pages	Performance Improvement / 16%	The technique use R-CNN for document layout discovery. Extracts the image features by applying the ResNet-101 convolutional network and then uses Faster R-CNN for metadata identification	The document is converted from PDF to Image format (JPEG). The evaluation of each component is not present.
Rahman & Finin-2019	-	-	The technique has used RNN, CNN, and LSTM based neural networks to identify the section headings and subheadings of a research document.	The description of the training dataset is not present. The authors have not discussed the evaluation of the approach.
MFCN-2017	617/ IC-DAR2015, SectLabel, DSSE-200	IOU, F1 scores/ NA	The authors have categorized their approach as a framework of components, which extracts the metadata by utilizing the classification based algorithm.	The results are not present.
CERMINE-2015	1238/ PubMed, PMC	Avg F scores/ 77.50%	The CARMINE tool is available online. It is a state-of-the-art approach that utilizes SVM for structure recovery, and it applies the clustering algorithm for text mining.	The algorithm extracts a limited number of metadata elements from a research article. The training of the system was done on PubMed and PMC datasets. So this approach has limitations while evaluating diversified styled articles

Continued on next page

Tuarob et al-2015	117/ Cite-seer	F1 scores/ 0.92	The approach present solution for document classification, clustering of citation, metadata extraction, and indexing.	The approach only performs for the Cite-seer corpus.
Dr. Inventor-2015	40/ Au-thor's Selection	F1 scores/ 0.8	The authors have categorized this approach as a framework that is based on a machine learning algorithm	The features are not well defined and a few number of research articles are selected to evaluate the approach.
PDFMEF-2015	1000/ Cite-SeerX	Throughput/ NA	The framework is a collection of different metadata extraction approaches developed by CiteSeerX team.	The framework only works on CiteSeerX. The performance of the framework was measured by the processor throughput.
Klampfl et al-2014	1000/ PubMed	Macro F1/ 0.88	This technique has proposed an unsupervised approach algorithm. It adopts the hierarchical agglomerative clustering to extract text blocks by identifying their geometrical relationship and further classifying them into logical structures.	The approach has limitations while evaluating diversified styled research articles.
Enlil-2013	3143/ "ACL, ACM Author's Selection"	F1 Performance/ p i 0.01	This approach utilizes CRF to identify authors and author affiliations and use SVM to discover the relationship of authors with their institutions	The approach has limitations while evaluating diversified styled research articles and has extracted a few number of metadata elements.

Continued on next page

SectLabel-2012 [70]	39/ ACM	Micro F1/ 0.93	This technique is state-of-the-art and tested on a renowned gold standard dataset. It uses Conditional Random Fields for structure recovery and metadata extraction.	The approach works on plain text extracted from the PDF file.
Tkaczyk et al-2012	113/ CC-BY Licence	Accuracy/ 81.96	This approach presents an adaptive modular workflow by implementing the HMM and CRF as a reference parser for the bibliographic content	The approach has limitations while evaluating diversified styled research articles.
ParsCit-2008	700/ Field references of Cora, CiteSeerX, and FLUX-CiM	Avg F1 scores/ 0.916	This approach has adopted conditional random fields to extract bibliographic metadata from research documents using a package that parses the tokenized sequences for labelled bibliographic strings.	The approach relies on sectLabel to extract the metadata from the full text of a research article.

Rules Based Techniques

Jiang-2018	-	-	This approach apply template matching by implicit format identification for the extraction of metadata from research articles.	This technique extracts a limited number of metadata elements. And no evaluation of this approach has been performed.
------------	---	---	--------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------

Continued on next page

Ahmed et al-2016	40/ ACM, LNCS, IEEE	F1 scores/ 0.74	The approach is winner of semantic challenge SemPub2016	This approach is non-expandable as its based on PDFx.
Azimjonov-2018	-	-	This system uses characteristic features to identify the input PDF document as a research article or non research document, and further sorts and stores the extracts metadata in an XML file.	This technique extracts a limited number of metadata elements. And no evaluation of this approach has been performed.
PDFX-2013	50000/Elsevier 1943/ PMC	F1 scores 0.807 0.502	This technique is state-of-the-art and tested on a renowned gold standard dataset.	This approach has extracted a few metadata elements and does not perform well on diversified dataset.
LA-PDFText-2012	251/ ACM, PubMed	F1 scores/ 0.91	It's a layout-aware system that employs textual content for text mining in the biomedical articles.	This approach is non-expandable and it extracts a few metadata elements.
PDF Extract-2012	-/ ACL	-	This system applies heuristics to identify the layout structure of a PDF-based research document.	This approach extract limited number of metadata components.
Déjean-2006	13000/ OCR Pages	F1 scores/ 0.91	This system groups text components into line segments, and blocks by applying heuristics and distance based upon the geometric position of characters.	The features are not well defined and approach is non-extendable.

Continued on next page

Graph-Based Techniques

g-DICE-2015	3471/ OCR Document	F1 scores/ 0.92	This system converts the content of the image-based digital document (OCR) to a graph and then evaluates the text fields using the graph mining approach.	The approach has limited metadata extraction elements and does not work on diversified journal layout styles.
SEB-2011	300/ Book-pages	Avg F scores/ 0.97	It performs the recovery of the reading order, association among figures, captions and table of content.	The system is tested on limited dataset of same formatting style.

2.5 Research Gaps

The growth in the number of publications due to emerging fields has motivated researchers to constantly improve techniques for automatic extraction, indexing, and storage of metadata of research documents. We have performed an in-depth analysis of these techniques as mentioned in the literature review section and the overview of these techniques is present in Table 2.2. These approaches adopted machine learning and heuristic-based models to extract elements and content of the research papers like structure recovery, document metadata extraction, bibliographic metadata extraction, etc. After analysis of these techniques, a few research gaps are identified as mentioned.

1. Previous techniques utilized datasets of research papers from similar publications and layout styles from a single domain. As presented in the literature review that ACM and MedPub have been common choices by the state-of-the-art for metadata extraction. Mostly, journals from this corpus of articles have identical layouts and composition styles. Therefore, these models accurately extract metadata from the same publishers. However, these approaches have lesser accuracy when evaluated with dissimilar composition and layout styles.
2. Previous metadata extraction techniques showed accuracy when evaluated on research articles with similar publication styles. However, results showed less accuracy while measuring the metadata output of research papers from a different publisher. For example, the results showed a 0.807 f-score for the PDFX approach when trained on the Elsevier dataset. However, while testing the approach on PMC and sectLabel [70], the results were around a 0.5 f-score. Similarly, the Klamp et al. approach trained on the PubMed dataset gave a 0.88 f-score, and when tested on the CEUR dataset, the result was a 0.59 f-score. There is a need for robust metadata extraction technique.
3. A moderate size dataset of research articles is trained and tested by some of the approaches for metadata extraction. SectLabel, Dr. Inventor, and Ahmed et al. approaches have used a moderate dataset of approximately 40 research articles. Some techniques have utilized a few hundred research articles. Consequently, such models suffer from overfitting and do not perform well when a large number of articles are tested for metadata extraction.
4. The features used by the rule-based or machine learning-based techniques are not properly identified, and their extraction method is not well defined. Therefore, whether a feature has performed well for extracting a specific metadata component cannot be evaluated. Further, reproducing a feature to improve, evaluate, or compare its extraction method cannot be achieved.

5. The previous techniques extract metadata by directly applying rules to the metadata or tagging the metadata's set of features for the machine learning-based models. This method limits their ability to achieve scalability. Consequently, extending the existing approach for performing on the dissimilar formatting style becomes challenging. As new overlapping rules are required to be developed for heuristic-based approaches or retagging of features is required in the case of machine learning-based models.
6. The previous approaches extracted a few metadata elements to furnish specific requirements. Table 2.1 shows that previous techniques have extracted metadata related to journal publication information or table of contents. It's a challenge to extract a great number of metadata elements and ensure the accuracy of the extraction approach. As a result, the fewer metadata elements indexed by the publisher will provide limited options to the user to search and enlist the required research articles.

The background and motivation of the thesis have explained that researchers search and enlist scholarly articles using digital libraries and search engines. This search is through the queries applied to the metadata elements stored and indexed by these large corpora of scientific literature. Approaches developed using heuristic or machine learning models extract metadata from research articles composed in PDF format. In this chapter, the authors of the thesis have performed a comprehensive review of the state-of-the-art metadata extraction techniques. The research gaps suggest that previous methods use research articles of the same layout and composition style. These approaches selected a few research papers from a single publisher. The features used by these approaches are not well defined. They extracted a few metadata elements and have limited ability to achieve scalability. The next chapter 3 presents a detailed methodology to extract metadata from research articles composed in PDF format. The proposed technique manages the issues identified in the research gaps of this chapter.

Chapter 3

Extraction Algorithm

Note: The parts of this chapter have been published in the Journal.

This chapter describes the methodology for extracting the metadata from scholarly articles published in portable document format (PDF). This segment of the thesis performs a detailed analysis of document structures and specifies the architecture of the proposed approach. The algorithm extracts the metadata in several stages to achieve scalability and remove complexness. The objective of individual steps is to focus on every task with clearly determined parameters. The proposed modular approach allows the separate evaluation and enhancement of individual components without engaging the overall system.

The devised algorithm handles a variety of scientific publications from multiple domains with heterogeneous layouts and composition styles. Therefore, the approach is trained on research articles from diversified publishers. The parameters and functions of different stages used by the proposed technique are clearly defined. The algorithm extracts a considerable number of metadata elements identified by previous metadata extraction schemes.

Figure 3.1 shows the stage-wise process flow of the proposed technique in an illustration. The system automatically constructs the features that the machine learning algorithm utilizes to extract the logical layout section. The extracted

logical layout section provides the bases to extract the desired metadata. The approach is comprised of the following stages:

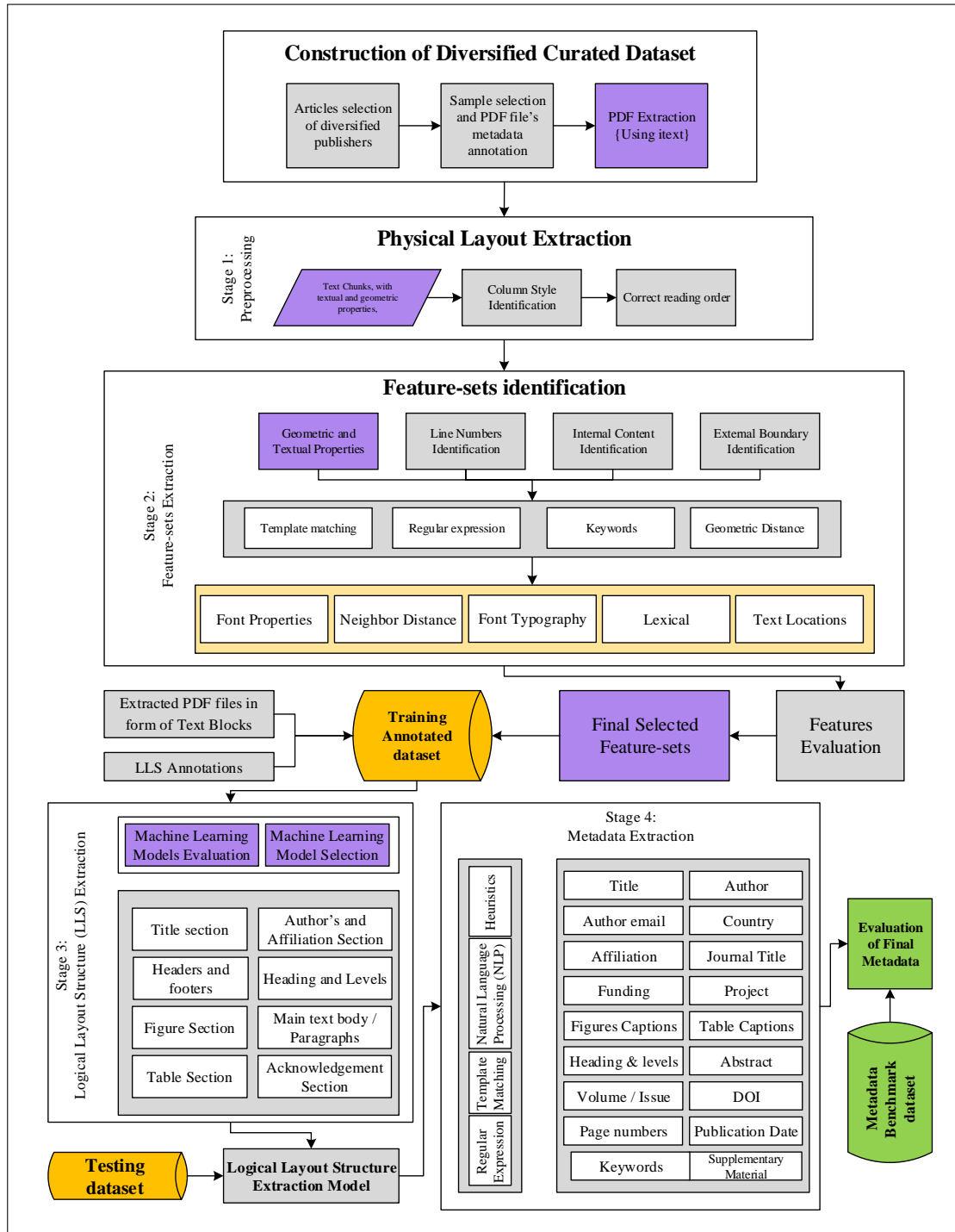


FIGURE 3.1: Overview of the proposed system methodology.

- Features Engineering (Section 3.2) :- The algorithm automatically generates features using textual properties and the geometric locations of the

text blocks. This stage initially calculates local level features, and further transforms them into generic features.

- Logical Layout Structure and Extraction (Section 3.3) :- The system tags the obtained features from the previous stage to associated text blocks. The SVM use these tagged chunks of textual blocks to extract logical layout sections.
- Metadata extraction (Section 3.4) :-The final stage processes the logical layout section to extract metadata in a structured output.

3.1 Dataset

This section presents the selection of benchmark datasets and the creation of the curated dataset to evaluate and construct the proposed technique that can extract metadata from research articles with diversified publishing styles.

3.1.1 CEUR Dataset

In order to develop a comprehensive model that can be used on diversified publishing styles, we chose ESWC 2016 challenge task 2 published dataset. Various gold standard datasets from the ESWC challenge are available at the link¹ along with an evaluation tool. This dataset consists of research articles having diversified formats and styles adopted from publishers like ACM, LNCS, and IEEE. The dataset has two parts, first is the training dataset (TD), which consists of 45 research articles, and the second part test dataset (ED) consists of 40 research articles. Initially, we used the training dataset (TD) of ESWC for model construction. The evaluation of the model was done on the test dataset (ED). The output of the ED contains 320 CVS formatted files. We evaluated the output of the proposed system at different stages on basis of comparison done with the gold

¹<https://github.com/angelobo/SemPubEvaluator>

standard dataset. We have referred to this dataset as *ds3*. ESWC 2017 challenge task 2² published a test dataset (TD) containing 40 research articles. Conference organizers have not published an evaluation dataset (ED) along with an evaluation of proposed techniques. However, we have also used the TD dataset to further evaluate the performance of our proposed model.

3.1.2 sectLabel Dataset

Luong et. al (2011) proposed their system to extract metadata from research articles and the dataset they have used, has gain popularity with the name of the approach "sectLabel". The dataset is a benchmark with 40 research articles from the computer science domain. The research papers are composed of different layout styles to identify the document body's logical layout. This corpus is composed of the "Association for Computing Machinery" (ACM) conferences and Proceedings of the "Association for Computational Linguistics" (ACL) from different years. Most popular techniques have used this dataset to evaluate their approach. This benchmark dataset is freely available online at the link ³ under research license by the publishers. We have referred this dataset as *ds2*.

Constantin et al. [9] evaluated the performance of their proposed system PDFX on the different datasets and compared sectLabel dataset to evaluate their approach with state-of-the-art. Another, renowned approach Klampfl et at. [53] used sectLabel to compare their approach with ParsCit [29]. Currently, ParsCit is using the sectLabel approach to extract the logical layout as an input for the system. Furthermore, Enlil [50] proposed by Do et al. built their technique based on sectLabel and used this dataset to evaluate their system. Similarly, other techniques have recognized this as a benchmark dataset.

We have used the dataset published by sectLabel to evaluate the metadata extraction of our approach and compared these results with the results of sectLabel

²https://github.com/ceurws/lod/wiki/SemPub17_Task2

³<https://parscit.comp.nus.edu.sg/sectLabelXML.tagged.txt>

approach that were published by its authors in their research article.

3.1.3 Construction of Diversified Dataset

In the previous section, we have described a benchmark dataset consisting of 40 articles from ACM and ACL proceedings, and as discussed earlier that renowned approaches had evaluated their techniques on this dataset. However, this corpus had research articles from three different conferences. The year of publication of these articles was before 2010, and also the research publications had logical layout formatting styles with limited diversification.

We require a diversified dataset that has research articles from diversified journals and has articles with different formatting and publishing styles to identify the logical layout structural components. This will enable us to comprehensively evaluate our approach to extract the metadata. In this regard, a diversified dataset is required that has research articles from different domains. It should consist of different publishers from multiple domains that are publishing in different scientific fields. The publishers must also contain different journals from different areas, which have articles with diversified publication styles. Within the journals, the papers must be published in different years, volumes, and issues to give the dataset more diversification.

On the basis of our requirements, regarding the dataset preparation, We have prepared a diversified dataset that has a collection of carefully selected diversified articles with different publishing and formatting styles. The articles are selected from different domains and are from renowned publishers. Where each publisher contains different journals that are related to different scientific areas. The journals have articles published in different years, volumes, and issues. In Figure 3.2, we have presented our overall methodology to construct the diversified dataset that

will be used to evaluate our metadata extraction approach and also use this dataset to compare the results of the state-of-the-art approaches, We have referred to this dataset as *ds3*.

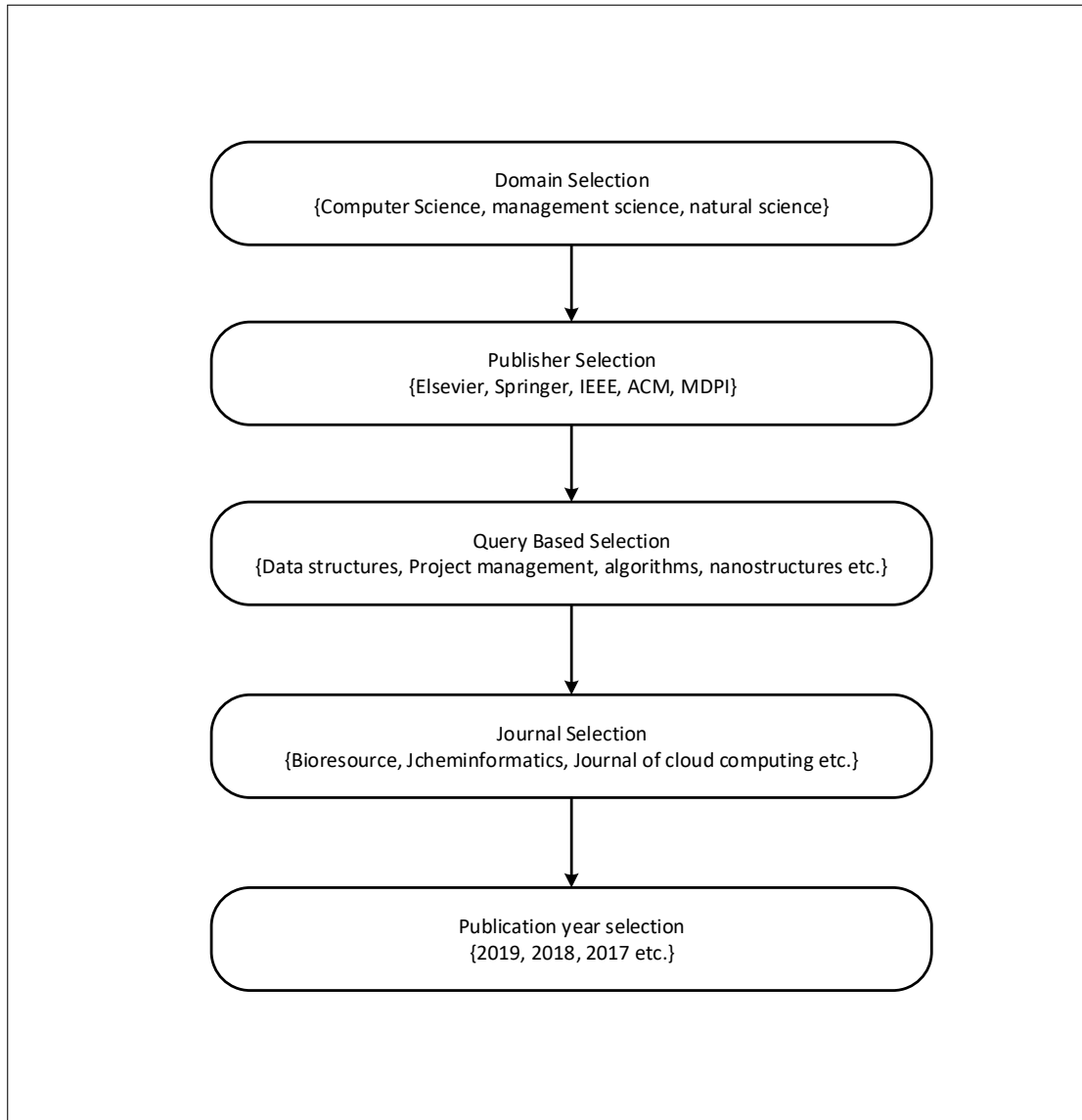


FIGURE 3.2: Research Article selection flow for dataset

3.1.3.1 Selection of Research Articles

The first step for the creation of the diversified dataset was the selection of the research articles. In this regard, initially, we selected domains from different areas of study. The fields we picked are from three popular domains that are computer science, natural sciences, and management sciences. As a recognized fact that

many publishers and journals are operating in these fields. Therefore, we can access a diversified corpus of the research articles. Another reason to opt for these domains is the availability of the team of researchers that we hired to perform metadata annotation of the research articles.

Based on the nominated domains, the next process was the selection of research publishers who publish the articles in these domains. Out of many renowned journals, we selected a few publishers that are popular in cross domains publications and also have diversified publishing styles. Table 3.1 presents the list of selected publishers.

TABLE 3.1: The list of selected Publishers for curated dataset.

Title of the Publisher
Springer
Elsevier
Institute of Electrical and Electronics Engineers (IEEE)
Association for Computing Machinery (ACM)
Multidisciplinary Digital Publishing Institute (MDPI)

The journal offers topic or query-wise search options to select the area of research. In the next process, by applying the topic search option and choosing the journals as an output, we obtained the list of journals for the publishers that associate with different areas of research from diversified domains. Table 3.2 shows the list of the selected journals to obtain the required articles. The reason to choose various journals from the same publisher is due to distinctive formatting and composition styles. The final process of dataset selection was to acquire research articles from different journals. The publication templates of journals changes over the period. Therefore, we downloaded articles from different years, volumes, and issues. In a few scenarios, the authors also do not observe the guidelines provided by the journal. We also included articles with these differences in the final dataset, to evaluate our approach over unseen problems.

We achieved the selection of research articles with diversified publication and formatting styles by following a comprehensive strategy. To avoid a biased evaluation

TABLE 3.2: The list of selected Journals for the dataset ds3.

Sr #	Title of the Journal
1.	Applied Petrochemical Research
2.	Bioresources and Bioprocessing
3.	Chemistry Central Journal
4.	Journal of Cheminformatics
5.	Journal of Nanostructure in Chemistry
6.	Journal of Magnetic Resonance
7.	Journal of Quantitative Spectroscopy and Radiative Transfer
8.	Photoacoustics
9.	Talanta
10.	Vibrational Spectroscopy
11.	Applied Informatics
12.	Human-centric Computing and Information Sciences
13.	IPSN Transactions on Computer Vision and Applications
14.	Journal of Big Data
15.	Journal of Cloud Computing
16.	Security Informatics
17.	Smart Learning Environments
18.	Artificial intelligence
19.	Neural Networks
20.	Arab Economic and Business Journal
21.	BRQ business research quarterly
22.	Burnout Research
23.	China Journal of Accounting Research
24.	Climate risk management
25.	European Journal of Family Business
26.	European Journal of Management and Business Economics
27.	European Research on Management and Business Economics
28.	Future business journal
29.	IIMB Management Review
30.	International Strategic Management Review
31.	Journal of Economics, Finance and Administrative Science
32.	She Ji The Journal of Design, Economics, and Innovation
33.	Asian Journal of Sustainability and Social Responsibility
34.	Business Research
35.	Frontiers of Business Research in China
36.	International Journal of Quality Innovation
37.	Journal of Global Entrepreneurship Research
38.	Journal of Innovation and Entrepreneurship
39.	Journal of Organization Design
40.	Journal of Shipping and Trade

of our proposed approach, we equally distributed the total number of research articles from selected publishers and journals in our final dataset.

3.1.3.2 Selection of Domain Experts

The second step for the creation of the ds3 dataset was the arrangement of the domain experts and annotators. We hired six experts from the Capital University

of Science and Technology, Islamabad. As in the previous subsection, we have described the initial step as the selection of diversified domains. Our domain experts are specialists in these domains and are Ph.D. students from different departments of our university. They are already performing the literature review by analyzing research articles from their field of research. Therefore, these experts have substantial knowledge regarding the disciplines or topics in their research domains. Also, they have a good understanding of the composition and structure of the research articles. The domain expert has the knowledge of renowned publishers and journals from their field of research. Based on the recommendation of our domain experts, we achieved the chosen list of research articles with diversified layouts and publication styles.

The domain experts performed three tasks for the creation of a diversified dataset ds3. The first task was the selection of research articles from publishers of diversified domains and journals from different areas of research in a domain. The second task was the line-wise tagging of logical layout structural components (LLS) of extracted content from PDF articles for eight different sections. The third task was the annotation of the metadata of research articles, as mentioned in Table 3.12. In the next subsection, we shall explain the process of annotation of research articles with the assistance of domain experts.

3.1.3.3 Tagging and Annotation of Dataset

The third step to create the diversified dataset ds3 was the annotation and tagging of logical layout structural components (LLS) and metadata of scientific research articles. We assigned individual tasks, to domain experts to perform the annotation of LLS and metadata. The first task to annotate LLS components requires line-wise tagging of the text blocks. We gave PDF-based research articles to the itext library, and the output was in the form of text blocks. We stored this output and converted it into excel format files for the annotation of LLS components. The team of annotators, assigned each line with an LLS tag. The range for the number of lines per article is between 500 to 5000 lines. The annotators tagged

all the lines with an LLS component by comparing them with original PDF files. Table 3.3 shows the line-wise tagging of logical layout structures performed by an annotator. We assign the same file to three different annotators and finally, we calculated the Inter-rater agreement (Kappa) and the majority vote was considered as an acceptable result for the annotation. We used these annotated files

TABLE 3.3: Research Articles Logical Layout Structures annotations.

Text	FontName	FontSize	IsBold	IsItalic	Top	Bottom	StartLine	EndLine	ClassLabel
A New Similarity Measure Based on Mean Measure of Divergence for Collaborative Filtering in Sparse Environment	Mea-Times-Roman	12.94666	0	0	583.42566	570.479	52	484	TITLE
Suryakant and Tripti Mahara IIT Roorkee, Roorkee, India	Times-Roman	9.90045	0	0	541.1633	531.2629	193	340	AUTHOR
Abstract	Times	6.90756	1	0	479.11856	472.211	46	77	ABSTRACT
Keywords:	Times	6.83771	0	0	347.13727	340.29956	46	82	KEYWORD
1. Introduction	Times	7.67502	1	0	297.8279	290.1529	46	109	H1
Corresponding author: Tel: +918894969853; Fax: +91 1792 245362.	Times-Roman	6.09256	0	0	117.37946	111.2869	59	266	AUTHOR
E-mail address: surya.dpt2015@iitr.ac.in	Times	6.07793	0	0	107.90033	101.8224	56	104	AUTHOR
2. Related Work	Times-Roman	6.09256	0	0	107.91496	101.8224	106	180	AUTHOR
3. Proposed Similarity Measure: CjacMD	Times	7.67502	1	0	392.56958	384.89456	46	114	H1
4. Experiments	Times	7.67502	1	0	277.54065	269.86563	46	216	H1
4.1 Data sets	Times	7.67505	1	0	679.1495	671.4744	46	109	H1
4.2 Evaluation measures	Times	7.59741	0	0	656.38544	648.788	46	98	H2
4.3 Experimental result and analysis	Times	7.59741	0	0	544.6949	537.0975	46	142	H2
Fig. 1. Performance Comparison of Different Approaches on ML-100 K Dataset: (a) MAE; and (b) RMSE.	Times	7.59741	0	0	361.80945	354.21204	46	187	H2
Fig. 2. Performance Comparison of Different Approaches on ML1M Dataset: (a) MAE; and (b) RMSE.	Times-Roman	6.09256	0	0	126.03496	119.9424	103	433	FIGURE
Fig. 3. Performance Comparison of Different Approaches on Each-Movie Dataset: (a) MAE; and (b) RMSE.	Times-Roman	6.09259	0	0	514.8738	508.7812	114	433	FIGURE
5. Conclusions	Times-Roman	6.09256	0	0	327.57175	321.4792	108	440	FIGURE
References	Times	7.67503	1	0	176.0966	168.42157	51	112	H1
	Times	7.67505	1	0	632.65076	624.9757	43	87	REF

for the implementation of the machine-learning algorithms. The machine learning models are trained on this dataset to identify different logical layout structural components. As we have already discussed in section 3.2, the stored text blocks along with features and the annotated files are used to select a machine-learning algorithm to identify different LLS.

The final task for the annotation team is to generate metadata files. We used these files to test and train the final output of our approach. We provided the annotators with a template of excel files. The template consists of excel files related to individual PDF-based files. The task is to physically examine the PDF files and

	A	B
1	title	automatic extraction of title from gernal document using mechine learning
2	journal	Information Processing and Management
3	accepted on	7 December, 2005
4	published on	2 February, 2006
5	pages	1276–1293
6	doi	doi:10.1016/j.ipm.2005.12.001
7	issue	5
8	vol	
9	year	2006
10	reference	27

	A	B	C	D	E	F
1	terms					
2	information extraction					
3	metadata extraction					
4	mechine learning					
5	search					
6						

	A	B	C	D	E	F
1	heading	level				
2	abstract	1				
3	introduction	1				
4	related work	1				
5	document metadata extraction	2				
6	information extraction	2				
7	search using title extraction	2				
8	Motivation and problem setting	1				
9	title extraction method	1				
10	outlines	2				
11	models	2				
12	feature	2				
13	format feature	2				
14	linguistic features	2				
15	Document retrieval method	1				
16	experimental results	1				
17	data set and evaluation measures	2				
18	cpmparison with baseline	2				
19	baseline	2				
20	accuracy of title is file properties	2				
21	comparsion between models	2				
22	domain adaptation	2				
23	language adaptation	2				
24	search with extract titles	2				
25	conclusion	1				
26	acknowledgments	1				
27						
28						
29						
30						

FIGURE 3.3: Research Article’s metadata annotated file to identify paper metadata, keywords, and section headings.

identify the metadata elements based on the publisher’s templates and then fill the templates of excel files with the identified metadata element. The Figures 3.3 and 3.4 show the metadata annotated excel files. We provided the same excel files and PDF research articles to three different annotators. We calculated the

	A	B	C	D	E
1	author	affiliation	country	email	corresponding
2	Yunhua HU	Computer Science Department, Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an, Shaanxi 710049, China	china	yunhuahu@mail.xjtu.edu.cn	Y
3	Hang Li	Microsoft Research Asia, SF Sigma Center, No. 49 Zhichun Road, Haidian, Beijing 100080, China	china	hangli@microsoft.com	
4	Junbo Gao	Microsoft Research Asia, SF Sigma Center, No. 49 Zhichun Road, Haidian, Beijing 100080, China	china	yucao@microsoft.com	
5	Liteng	Computer Science and Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China	china	leng@cse.cuhk.edu.hk	
6	Dmitry Meyerzon	Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA	USA	dmitrym@microsoft.com	
7	Qinghua Zheng	Computer Science Department, Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an, Shaanxi 710049, China	china	qzheng@mail.xjtu.edu.cn	

	A	B	C
1	figures		
2	Fig. 1. Distributions of file formats in internet and intranet		
3	Fig. 2. Title extraction from Word document.		
4	Fig. 3. Title extraction from PowerPoint document.		
5	Fig. 4. Title annotation tool.		
6	Fig. 5. Example of units.		
7	Fig. 6. Metadata extraction model.		
8	Fig. 7. Distributions of document genres.		
9	Fig. 8. Weights of some features in Perceptron models for Word.		
10	Fig. 9. An example Word document		
11	Fig. 10. An example PowerPoint document.		
12	Fig. 11. Search ranking results		
13			

	A
1	tables
2	Table 1 The portion of documents with title
3	Table 2 Contingence table with regard to title extraction
4	Table 3 Accuracies of titles in file propertie
5	Table 4 Accuracies of title extraction with Word
6	Table 5 Accuracies of title extraction with PowerPoint
7	Table 6 Sign test results
8	Table 7 Comparison between different learning models for title extraction with Word
9	Table 8 Comparison between different learning models for title extraction with PowerPoint
10	Table 9 Accuracies of title extraction with Word in DotGov
11	Table 10 Accuracies of title extraction with PowerPoint in DotGov
12	Table 11 Accuracies of title extraction with Word in DotCom
13	Table 12 Performance of PowerPoint document title extraction in DotCom
14	Table 13 Accuracies of title extraction with Word in Chinese
15	Table 14 Accuracies of title extraction with PowerPoint in Chines
16	Table 15 Accuracies of title extraction with Word in Japanese
17	Table 16 Accuracies of title extraction with PowerPoint in Japanese
18	Table 17 Accuracies of title extraction with Word in German
19	Table 18 Accuracies of title extraction with PowerPoint in Germa

FIGURE 3.4: Research Articles metadata annotated file to identify authors and affiliations, figures, and tables captions.

Inter-rater agreement (Kappa) regarding the metadata annotations. We selected the metadata elements with 100% Inter-rater agreement (kappa) and rechecked by ourselves for the elements with kappa above 60%. We used another excel-based template file to get the summary of annotations publisher-wise, which we have used for the comparison of metadata extraction evaluation

The flow of the selection of the articles has ensured that the PDF files are selected from diversified publishers by a group of experts. We achieved our goal to select a comprehensive dataset for diversified domains of different journals for renowned publishers. The total number of articles with diversified publication and layout styles is 500. For the training and the evaluation purpose, we have distributed the dataset into equal parts as already referred to as *ds3*.

3.2 Features Engineering

*"The answer of the **RQ2** is presented in this section".*

RQ2: How to devise a strategy to extract features from papers published by the diversified journals and how to select an optimum set of features from the identified features?

We have performed the comprehensive analysis of previous approaches that extract the metadata and layout components using rule-based and machine-learning-based techniques. In this regard, we also performed the comprehensive review of approaches to find the features that were used in their techniques. In Table 3.4, an evaluation matrix of features used by different approaches has been presented. However, in most scenarios, the mechanism to extract the features are not defined by the approaches [9, 43, 44, 46, 49, 57, 59]. The techniques have not shown which exact feature they have used and what metadata elements the feature has helped to extract? It is also unclear that either the mentioned features were extracted manually or they applied techniques like heuristics or template matching. Moreover, the details of extraction and evaluation are not presented by the previous approaches.

Therefore, we have to build a strategy to effectively extract the features that will be used by the machine learning approaches to extract logical layout structures in the next stage. The logical layout structures are different structural components of the research articles that are helpful for a reader to correctly differentiate the content of the research articles. We have categorized different structural components into eight different sections. These are the title section, authors section, headers and footers, acknowledgment section, table section, figure section, section and subsection headings, and body/paragraphs.

We shall extract these logical layout structures by using machine learning approaches. The machine learning approaches will be provided with LLS annotated text blocks and final feature sets extracted in this section to train and evaluate

TABLE 3.4: The list of text features adapted to train models for the extraction of metadata

	Ahmed et al	PDFX	LA- PDFText	Klampf et al	sect label	SEB	CERMINE	S.Tuarob et al	PDF Extract	Tkaczyk et al	Enlil	FlagPDFe
Approach	RB	RB	RB	ML	ML	GB	ML	ML, RB	RB	ML	ML	ML
Font Features												
Font Name	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Font Size	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓
Is Bold	✓	×	✓	✓	✓	×	×	×	✓	×	✓	✓
Is Italic	✓	×	✓	✓	✓	✓	×	×	×	×	✓	✓
Font Orientation	×	✓	×	×	✓	×	✓	×	×	✓	✓	✓
Line Number	×	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
Typological Features												
All Capital	✓	✓	×	✓	×	×	✓	✓	×	✓	✓	✓
Initial Capital	✓	✓	×	✓	×	×	×	×	×	✓	✓	✓
Initial Numeric	✓	✓	×	✓	×	×	✓	✓	✓	×	✓	✓
Special Characters	✓	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
Is Figure	✓	✓	✓	✓	×	✓	×	×	×	×	✓	✓
Is Table	✓	✓	✓	✓	×	×	×	×	×	×	×	✓
Keyword	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	✓
Last Character	×	×	×	✓	×	×	×	×	×	×	×	✓
Line Length	×	×	×	×	✓	✓	✓	×	✓	×	×	×
Under Line	×	×	×	×	✓	×	×	×	×	×	×	×
Paragraph Tag	×	×	×	×	✓	×	×	×	×	×	×	×
Page Number	×	×	×	×	×	✓	✓	×	✓	✓	✓	✓
Lower Case	×	×	×	×	✓	✓	×	×	✓	×	✓	×
Word Distince	×	×	×	✓	×	×	×	✓	×	×	×	×
First Character	×	×	×	×	×	×	✓	×	×	×	✓	×
Mix Case	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	×
Novel Features												
Column Number	×	×	×	×	×	×	×	×	×	×	×	✓
Alignment	×	×	×	×	×	×	×	×	×	×	×	✓
Start Indent	×	×	×	×	×	×	×	×	×	×	×	✓
End Indent	×	×	×	×	×	×	×	×	×	×	×	✓
Page Top Distance	×	×	×	×	×	×	×	×	×	×	×	✓
Page Bottom Distance	×	×	×	×	×	×	×	×	×	×	×	✓
Previous Line Distance	×	×	×	×	×	×	×	×	×	×	×	✓
Next Line Distance	×	×	×	×	×	×	×	×	×	×	×	✓

the models. The LLS are published in diversified layout and formatting styles, and the required metadata that needs to be extracted is placed in LLS. Therefore, to develop an approach that can extract LLS from diversified publishers, we need to develop a set of generic features that plays an important role in effectively to extract logical layout structures.

In the next stage, we shall process these extracted logical layout structures (LLS) to extract the metadata of the research articles. The accuracy of the metadata extraction dependent upon the correct extraction of logical layout structures. Metadata extraction is the final task of our proposed approach and we shall present it in section 3.4 and section 4.2 of this thesis.

To evaluate **RQ2**, we have proposed a three stage process to extract features of the text block as shown in Figure 3.5. In sub sections, we shall discuss the proposed methodology of each stage in detail.

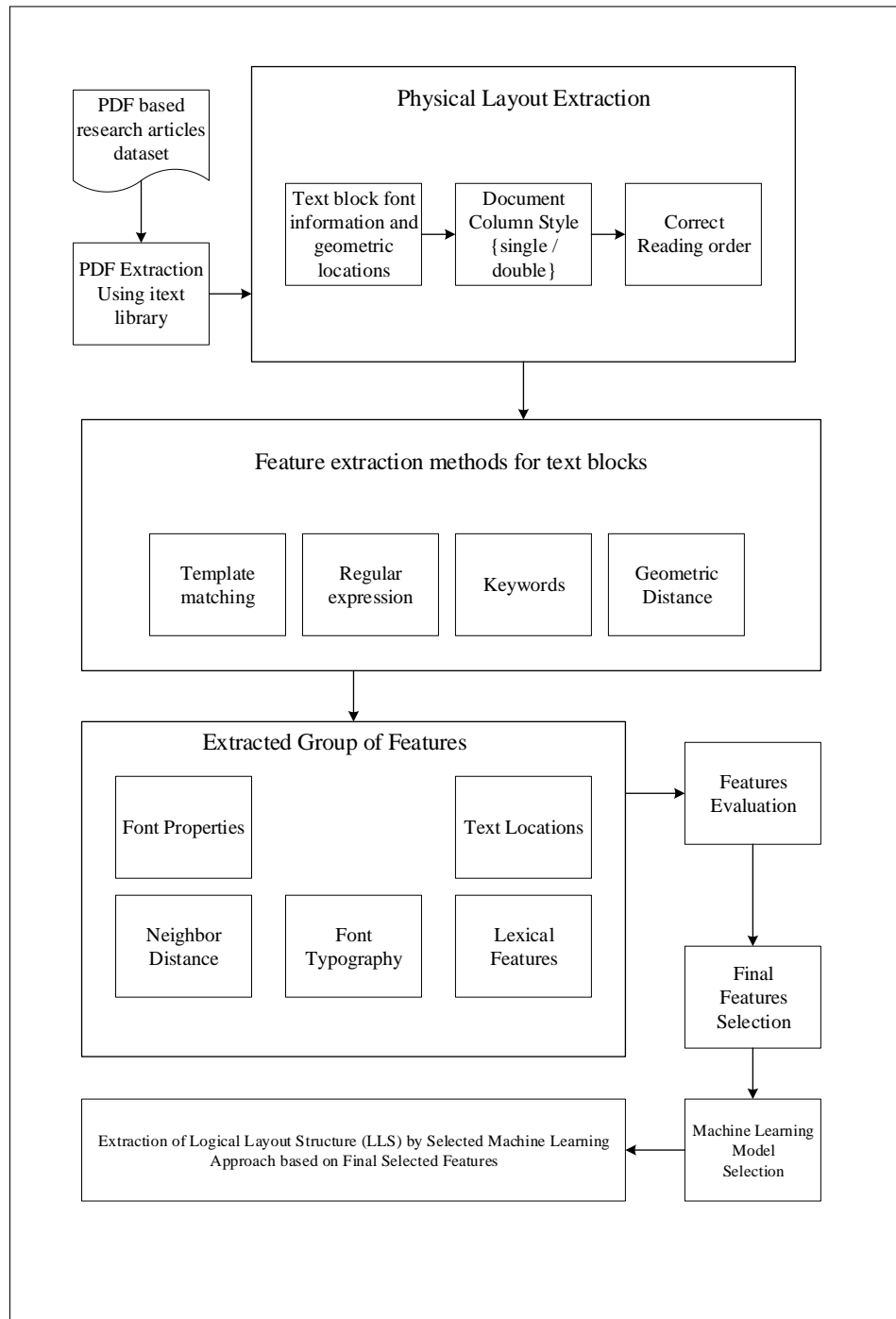


FIGURE 3.5: The methodology diagram to extract features set associated to text blocks used for extraction of the logical layout structures using machine learning.

3.2.1 Physical Layout Extraction

A PDF file is composed of raw binary data without any associated metadata and logical structural information that identifies different layout categories of the

content. Therefore, the first process is to extract the textual information from the PDF file. For this purpose many open source off line libraries are available (like PDFBox⁴, poppler⁵, JPod⁶ and itext⁷) to extract the textual content of the PDF file as the stream of characters. This stage we used itext [10] open source java library that provides faster and reliable method to extract PDF file. Unlike other processing tools that extract text as text glyph or stream of characters, itext extracts chunk of textual elements that reduces resources and computational cost. Further, itext implements advance strategy to extract structural components that are text chunks in sorted order, text font properties, geometric locations, raster images, page numbers, and vector graphics.

The text chunks are retrieved, encapsulated in boundary boxes that identifies their geometric position in form of (x, y) coordinates on the page along with height and width. Further, it constructs words from chunks of text and lines from words and finally text blocks. The itext library sorts the chunks of text in a reading order based on adjunct neighbors and reads in columns from left to right and top to bottom. The itext library returns font attributes like font name, font size, bold, italic, orientation etc. We used these attributes to generate font properties feature-set.

3.2.2 Column Style Identification

The research documents are composed in single or double column style. This process identifies the column style of the document in order to determine the boundary of the main text body. The column layout style further helps to identify the geometric position and layout properties of the text blocks. The process first calculates the right and left outermost margins of the page. The left outermost margin calculated by the MODE of minimum values of text blocks geometric start point, and the right outermost margin is calculated by the MODE of maximum

⁴<https://pdfbox.apache.org/>

⁵<https://poppler.freedesktop.org/>

⁶<https://www.openhub.net/p/jpodlib>

⁷<https://itextpdf.com/>

values of text blocks geometric end value. Thereafter the process calculates the number of columns present in the document. The process starts from left outermost margin and calculates MODE of maximum values of text blocks geometric end value. If the value is equal to right out most margin, the process stops, else process again computes the MODE of minimum values of text blocks geometric start point, till it finally reaches the left outer margin.

3.2.3 Correct Reading Order

Initial systems used X-Y cut algorithm that utilizes the geometric location of page segments in top-down approach. Another affective technique is Docstrum algorithm and its enhanced models that are bottom-up approach, which extract text blocks through K-nearest neighbors clustering. The output of the itext library is mostly in correct sorting and reading order; however, in few scenarios it can have slight irregularities while extracting the text reading order and font properties. The main cause of reading order irregularity is due to in-text citations, algorithms, tables content, vector graph based figures content, special characters and floating text objects.

Figure 3.6 demonstrates the reading order of itext library with the help of connected lines over text blocks. We slightly adjusted reading orders for text chunk comprising of in-text citations and special characters based on thresholds derived from layout of neighboring text chunks, adopted a similar approach. The process derives words from received scattered chunks of text and on basis of geometrical location and physical distance among them. The words grouped together to formulate lines while retaining the text features of individual text chunk. This process produces plain texts having no relationship between words and lines and paragraphs. Furthermore, the process computes page number and line numbers. line numbers originated by computing the reading order of text blocks, and rendering order of the content of text blocks. Text with same geometrical position and column had same line numbers. The process therefore placed the context in a symmetrical correct reading order.

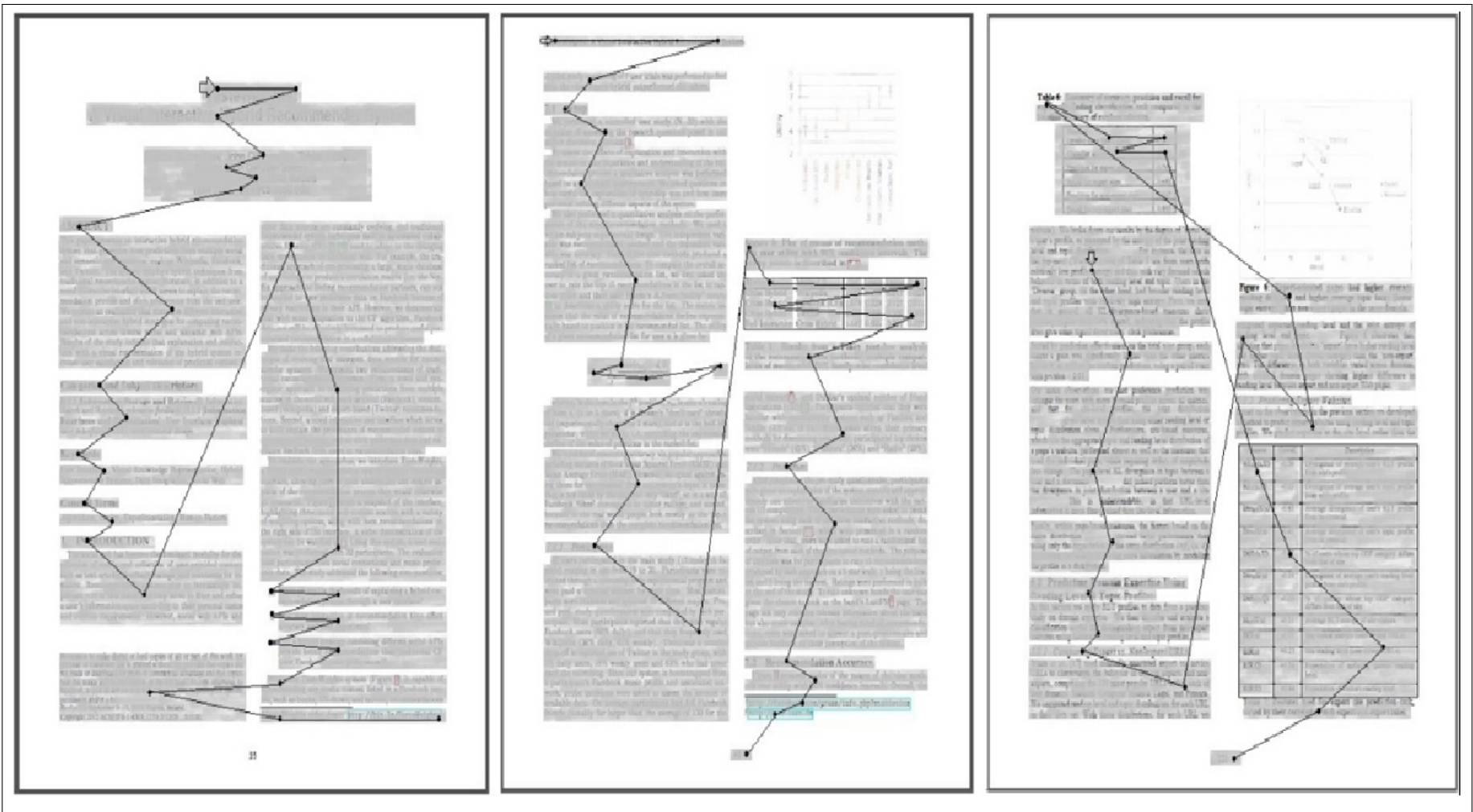


FIGURE 3.6: The connected lines shows the reading order of PDF based sample research document when extracted by itext library.

3.2.4 Physical and Logical Layout Evaluation

The process finally produce text blocks with geometric locations and reading order, page numbers, line numbers, font attributes, images with geometric location and floating objects. This vital information provides bases for next stage to extract physical layout of the document and further identifies different features of text blocks to extract logical structural layout.

The logical layout that determines the document’s layout categories comprised of title, authors and affiliation, figure and captions, tables and captions, heading and levels, paragraphs, bibliography etc. A PDF file most often lacks metadata tag associated to an individual logical layout category to support automatic retrieval or identification of required content. We have developed a framework, which address this core issue by extracting the logical structure categories of PDF-based research article, and finally generates layout aware RDF files in order to perform semantic queries.

3.2.4.1 Font Properties

The itext library extracts the font properties of the text characters or chunks from a PDF file. However, the font name contains all the font information having concatenated itext font code, font family name, bold or italic information, which requires further processing to extract individual font properties. Most often, individual categories of text blocks possess different font features, like section headings composed in bold or italic to mark as prominent. Based on font families there is variation in identification of the bold and italic properties. As Times, Arial, SegoeUI and Nimbus etc font families contain “bold” or “italic” keywords and Computer Modern “CM” fonts has BX for bold and TI for italic font style as represented in Table 3.5. The fore or background color can be identified but in case of research documents, this feature is of least significance. The itext library

TABLE 3.5: Font names and style extracted by itext library for document font style.

Output Font Name of itext library	Document font Style
DRPUQI+CMBX12	Introduction
SJXVJJ+Times-Bold	ABSTRACT
PIGZEP+CMBXTI10	<i>Relation Intersection.</i>
Times New Roman,BoldItalic	<i>Abstract</i>

also extracts the vertical and horizontal orientation of the text block.

3.2.4.2 Text Location

The column style helps determine the text location features of text blocks based on the presence of text block or line in a column. During the identification process of external boundaries the single or double columns styles were identified, in this stage the text blocks location in a column is defined. The documents with single column style have text blocks existing in column number one. However, with double column style a text block can exist in column number one or two, and text block that do not resides in any column is assigned with column number zero like title etc. The In Column feature has the information regarding column number of a text block. The align feature identifies the left, right or center alignment of text block with in a column. Figure 3.7 represent the identification of alignment of text blocks where the main section heading is center align within a column and starting line of each paragraph is right align and rest lines are left align. The

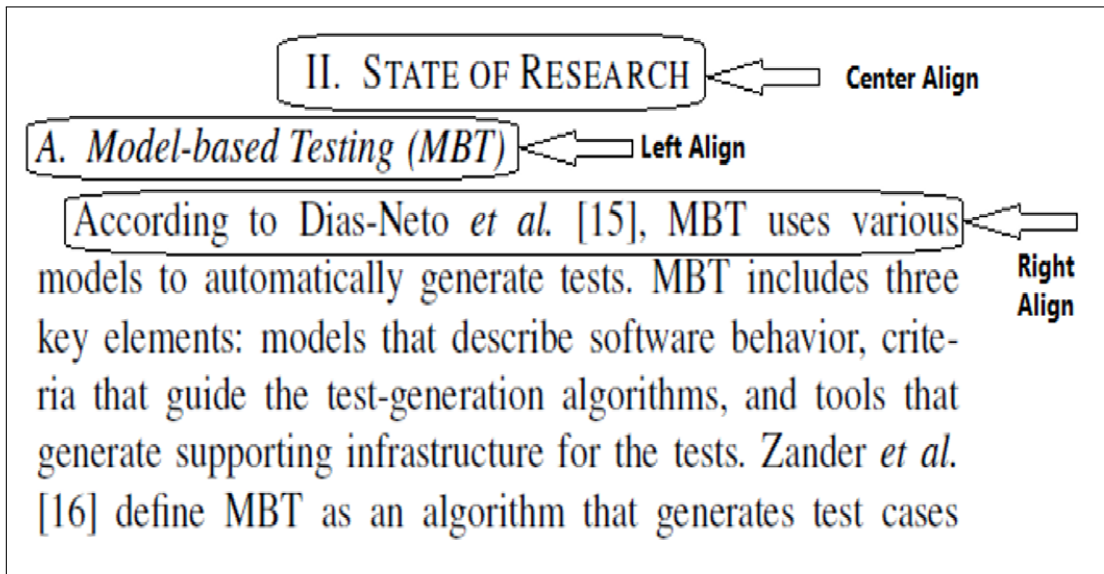


FIGURE 3.7: Alignment feature of the text blocks within a column.

distance of starting point of text line with reference to column start is present in start indent feature and ending point distance from column end is present in end indent feature.

3.2.4.3 Neighbor Distance

The reading orders helps to assign line numbers to individual text line. The line number enables the system to identify the sorted order of main body content, sections heading and bibliography. However, the sorting of table numbers and figure numbers cannot be guaranteed. The feature measuring distance of line from top and bottom of page calculates the sorting order of the content; also, this feature enables the system to identify text blocks that are composed close to the far boundaries of the page. The distance from adjacent lines helps to identify continuity among text block to form paragraphs. However, this parameter is conclusive when the Font properties are same between distinct text blocks. Like section heading, figure and table caption have same text size and font properties as compared to body text as illustrated in Figure 3.8 and Figure 3.9.

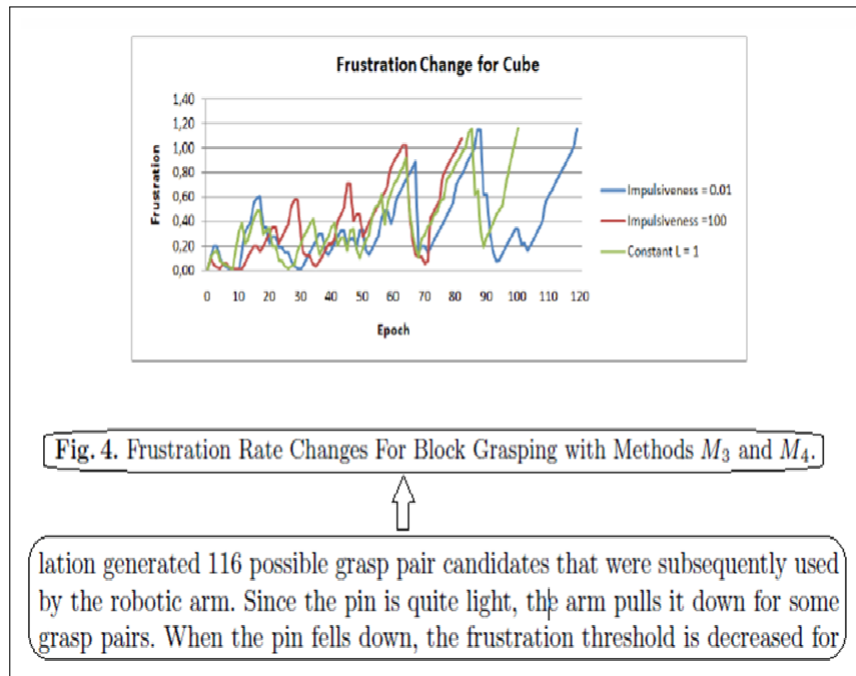


FIGURE 3.8: The text blocks share common text properties, but the figure caption distance from paragraph is more as compared to distance among paragraph lines.

3.2.4.4 Font Typography

The font typographical features facilities in the identification of Title, section headings and levels. Research articles have section heading in different typographies where heading has text in capital case or title case format, here the identification of initial capital words require some preprocessing as they may contain prepositions in small case letters. Therefore, we excluded the prepositions and then checked the initial capital phrases.

In Table 3.5, these text case features are found in the section heading or title of research article. Another, important typographic feature is the initial numeric values to define the heading number or heading level. The heading numbers are defined by either a numeric value or a roman value as shown in Table 3.5. The sub headings in such scenarios have outline numbering styles, the system counts the number of dots and eliminates if it is present at the end of the number hierarchy.

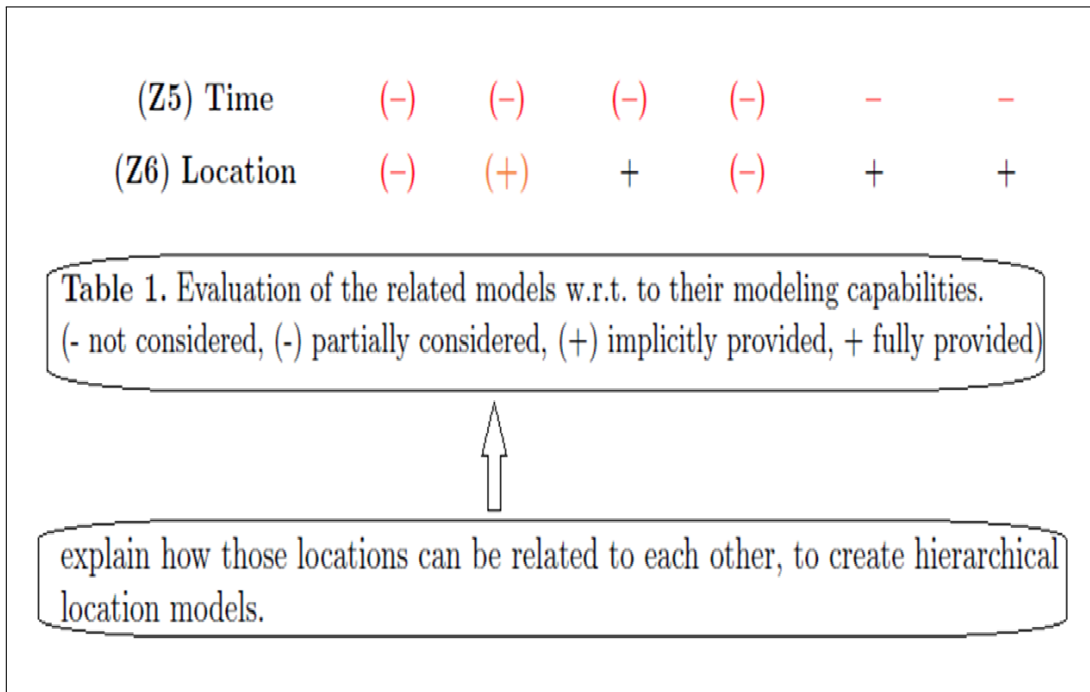


FIGURE 3.9: The text blocks share common text properties, but the table caption distance from paragraph is more as compared to distance among paragraph lines.

TABLE 3.6: The text blocks showing different font typographical styles adopted by the publishers to present headings and levels.

Features	Font Typographic Styles
Main-heading with Numeric Title Case	2 Related Work
Sub-heading with Numeric Title Case	2.1 NER on Social Media for English
Main-heading with Roman Capital Case	I. INTRODUCTION
Main-heading with Numeric Capital Case	1 INTRODUCTION
Sub-heading with Numeric Title Case	<i>2.1.2 Dataset Limitations</i>

3.2.4.5 Lexical Properties

The research documents has meaningful content that enables the system to identify the logical layout components. It has been observed that keywords based search

like “Abstract”, “Reference”, “Bibliography”, “General Terms”, “Keywords” and “Acknowledgments” etc can be an effective method to identify the relevant sections. Therefore, the content before “Abstract” most often contains the title section of the document. Similarly, the content after the “reference” heading will have bibliographic information. The Acknowledgment section contains the funding sponsors for the project, and we shall use it in later part to extract funding Agency.

The figure or table caption always starts with keywords like “Fig”, “Figure”, “Viz”, “Graph”, “Tab” and “Table” etc. However, such keywords may exist at start of a paragraph but combination of these keywords along other textual feature can be helpful to identify captions. Email’s always have “@” character and efficiently build regular expression on text lines with these special characters can detect correct emails addresses.

3.2.5 Features Set Extraction

After performing detailed analysis of logical and physical components, we trained the system to extract features using physical layout structure. Table 3.7 has the list of features along with the data type, on which machine learning algorithm with be trained and evaluated. In this subsection we shall discuss extraction strategy for each feature.

3.2.5.1 Font Features

The text chunks extracted by itext library have associated font name, coordinates of geometric boundary boxes and orientation. The font name is composed of four parts those are code, font family name, bold text and italic text. We have ignored the code part and concentrated on next three parts of the font name. We performed comprehensive analysis of all the font names extracted with text chunks of *ds1*. A regular expression is used to segregate font family, bold and italic property. However, this regular expression is based on complex scenarios as

TABLE 3.7: Features associated to text blocks in order to identify logical layout content of research paper

Feature Name	Description
Font Properties	
Font Name	Name of the font (e:g Times New Roman)
Font Size	Floor rounded size of font.(e:g 11)
Is Bold	The font style is bold
Is Italic	The font style is italic
Font Orientation	Horizontal or vertical orientation of font
Text Location	
Align	Text block aligned from column (left, center, justify or right)
Column Number	Text block exists inside column boundaries
Start indent	Intending space from start of column
End indent	End of line distance from end of column
Neighbor Distance	
Line Number	The line number of the text
Page Top Distance	The line distance from the top of page
Page Bottom Distance	The line distance from the bottom of the page
Previous line Distance	The geometric distance from the previous line
Next line Distance	The geometric distance from next line
Font Typography	
All Capital	The text has all capital letters
Initial Capital	The text has initial capital letters
Initial Numeric	The text start with numeric
Initial Roman	The text start with a roman value
Numeric dots	The number of dots in numeric values (e:g 1.1.1 = 2)
Lexical Features	
Special Characters	Contains special characters like (:, @ ; [] - etc)
Is Figure	Starts with keywords (Fig, Figure, viz etc)
Is Table	Starts with keywords (Tab, Table etc)
Keyword	keywords (Abstract, Reference, Bibliography etc)
Last Character	The last character of the line.

itext library do not have a standard way to represent the font text information, as shown in Table 3.5.

The font size cannot be accurately calculated, but the height of text can be used as a replacement with the size of the text. As already discussed the height can be measured with the help of distance between top and bottom coordinates of

boundary boxes. Finally, itext library output the text orientation as a binary value, where 0 is for horizontal and 1 is for vertical.

3.2.5.2 Text Location Based Features

The text location features are fully dependent on the column layout style. We have already discussed column identification in detail in external boundary identification section. The “column number” declares the position of text block. All the text blocks which are in the single column style papers are marked as 1. For double column style papers the text blocks in left column are marked as 1, the text in the right block is marked as 2 and rest text is marked as 0. However, all the text block must reside inside the external boundary. The start and end indent are the geometric distance of text block from the start and end position of the column it resides. The Alignment of text block is calculated on the bases of column they reside. If the start position of text block is equal to start position of the column and end position is less from end position of the column, then the text block is left align. If the start position of text block is greater than the start position of column and end position of text block is equal to end position of column, then the text block is right align. If both start and end position of text block is equal to start and end position of the column, then the text block alignment is justify. Finally the start position of text block is greater than start position of column and end position is less than end position of column, then text block is measured as center aligned.

3.2.5.3 Neighbor Distance based Features

The neighbor distance depends on the line and page level geometric positions. The line number of a text line is calculated after the identification of column number identification and text blocks are in correct sorting order. “Page Top Distance” feature is measured by calculating the geometric distance from top coordinates of text block from bottom coordinates of the header section. Similarly “Page Bottom

Distance” feature is measured by calculating the geometric distance from bottom coordinates of text block from top coordinates of the footer section. Finally the previous and next line distance is measured by calculating the distance of top coordinates of text block from bottom coordinates of previous text block and the distance of bottom coordinates of text block from top coordinates of next text block respectively.

3.2.5.4 Font Typography and Lexical Features

The regular expression to encode the ASCII character of text can be used to determine the typography features of the text block. If all the character of text block ASCII values are in uppercase then that text block is marked as “all capital”. If the first character of each word is in upper case and rest in lower case, then the text block is marked as title case, however preposition are not included. Similarly, regular expression on initial characters of text block can help identify the roman cases. Finally, the initial integer values and dot between them can be marked as initial numeric and numeric dot features of text blocks. The lexical feature set is identified by using special keywords and special characters present in text block.

3.2.5.5 Feature Extraction Algorithm

The initial step of the feature extraction algorithm is the identification of document column style. A single or double column identification helps correct the sorting order of text blocks and correct the reading order of the article. The column style identifies the boundary margin of each column that help in creating text location-based and neighbour distance-based features. The algorithm 1 detects the right and left outermost margins of the document and takes the extracted content of RAW pdf file as an input parameter ‘textPDF’. The algorithm stores the geometric location of the left most outer margin in Xmin and the rightmost outer margin in Ymax variables. The statistical mode function calculates the left and right most common geometric locations of all the text lines. These Xmin,

Ymax and textPDF variables are used in Algorithm 2 where document columns are identified. The number of columns and their respective margins are stored in Columninfo datatype. Finally, Algorithm 3 measures the header and footer area of the document based upon the Columninfo datatype.

Algorithm 1 Left Right Margins

```

1: function MAIN_MARGIN_DETECTION(TEXTPDF)
2:   X min  $\leftarrow$  MODE (  $\Sigma$  textPDFi MIN (LeftGeoLocation))
3:   Y max  $\leftarrow$  MODE (  $\Sigma$  textPDFi MAX (RightGeoLocation))

```

Algorithm 2 Column Margins Detection

```

1: function COLUMN_MARGIN_DETECTION(TEXTPDF, XMIN, YMAX)
2:   Main_Margin_Detection(textPDF)
3:   minCol  $\leftarrow$  X min
4:   maxCol  $\leftarrow$  nil
5:   colNum  $\leftarrow$  0
6:   while maxCol  $\leq$  Y max do
7:     ColNum ++
8:     maxCol  $\leftarrow$  MODE( $\Sigma$  MinCi MAX (RightGeoLocation))
9:     Columninfo [colNum, minCol, maxCol]
10:    if minCol < maxCol then
11:      minCol  $\leftarrow$  maxCol

```

Algorithm 3 Header Footer Margins

```

1: function HEADER_FOOTER_DETECTION(COLUMNINFO)
2:   X_Top  $\leftarrow$  MODE (  $\Sigma$  Ci  $\in$  Columninfo MAX (TopGeoLocation))
3:   Y_Bottom  $\leftarrow$  MODE (  $\Sigma$  Ci  $\in$  Columninfo MIN (BottomGeoLocation))

```

The second step creates font-based features, as presented in Algorithm 4. The “Font Features Extractor” takes an input of extracted pdf content by the third-party library (itext/ PDFExtract) as mentioned in the prior section of this chapter. The algorithm mention “Text PDF” as the data structure, which stores the output of extracted PDF file. This data structure is an array list that contain attributes “Full Font Name”, “Plain Text”, “Orientation”, “Page Number”, “Top Location”, “Bottom Location”, “Start Location”, and “End Location”. Algorithm 4 presents the scheme to generate features based on font properties. The input variable “textPDF” to the function is the extracted content of the PDF file. It calculates

Algorithm 4 Font Features Extractor

```

1: function FONT_FEATURE_EXTRACTOR(TEXTPDF)
2:   BoldRegExp ← `BOLD|CMBX|\ - B|TIMESB|\ .B'
3:   ItalicRegExp ← `ITAL|CMMI|\ - I|IT|TIMESI|\ .I'
4:   TrimFontCode ← `(? <= ^ .....)(.*)'
5:   i ← 0
6:   while i < textPDF.length do
7:     TextBlockId ← textPDF[i].TextBlockId
8:     FontSize ← round(textPDF[i].BottomLoc - textPDF[i].TopLoc)
9:     Width ← round(textPDF[i].EndLoc - textPDF[i].StartLoc)
10:    FontFullName ← textPDF[i].FullFontName
11:    FontOrientation ← round(textPDF[i].Orientation)
12:    if FontOrientation = 0 then
13:      if Width < FontSize then
14:        FontOrientation ← 1
15:      else
16:        FontOrientation ← 0
17:      RemoveCode ← replace(FontFullName, TrimFontCode, ``)
18:      Font ← replace(RemoveCode, BoldRegExp, ``)
19:      Font ← replace(Font, ItalicRegExp, ``)
20:      FontName ← Font.Map(FontNameList)
21:      if REGEX(FontFullName, BoldRegExp) then
22:        IsBold ← 1
23:      else
24:        IsBold ← 0
25:      if REGEX(FontFullName, ItalicRegExp) then
26:        IsItalic ← 1
27:      else
28:        IsItalic ← 0
29:      FontFeature.Push(TextBlockId, FontName, FontSize,
30:        IsBold, IsItalic, FontOrientation)
31:      i ← i+1
32:   Return textPDF.Append(FontFeature)

```

the font size by measuring the difference between the text block's bottom and top geometric locations. Similarly, the difference between the start and end geometric locations helps calculate the width of the text block. The orientation of the font is also present in the extracted file. The system further checks for the fonts published in a vertical style by comparing the height and width of the text block. The regular expression defined in BoldRegExp and ItalicRegExp variables checks for bold and italic keywords in the FullFontName variable. The algorithm generates the font name feature by removing the initial seven-figure code, bold, and italic keywords.

Algorithm 5 Text Location Features Extractor

```

1: function TEXT_LOCATION_FEATURE_EXTRACTOR
2: (TEXTPDF, COLUMNINFO)
3:   i ← 0
4:   while i < textPDF.length do
5:     TextBlockId ← textPDF[i].TextBlockId
6:     StartLoc ← textPDF[i].StartLoc
7:     EndLoc ← textPDF[i].EndLoc
8:     ColumnLocation ← find_column(Columninfo, StartLoc,
9:                                 EndLoc)
10:    ColumnNumber ← ColumnLocation.colNum
11:    LeftMargin ← ColumnLocation.minCol
12:    RightMargin ← ColumnLocation.maxCol
13:    if EndLoc > StartLoc then
14:      if LeftMargin <> NIL OR RightMargin <> NIL then
15:        StartIndent ← StartLoc -LeftMargin
16:        EndIndent ← RightMargin - EndLoc
17:        if StartLoc = LeftMargin & EndLoc = RightMargin then
18:          Align ← 1
19:        else if StartLoc > LeftMargin & EndLoc = RightMargin then
20:          Align ← 2
21:        else if StartLoc > LeftMargin & EndLoc < RightMargin then
22:          x ← (RightMargin/LeftMargin)/2
23:          y ← (EndLoc/StartLoc)/2
24:          d ← y * 0.1
25:          if (x-d) < y < (x+d) then
26:            Align ← 3
27:          else
28:            Align ← 5
29:        else if StartLoc = LeftMargin & EndLoc < RightMargin then
30:          Align ← 4
31:        else
32:          Align ← 5
33:          StartIndent,EndIndent ← NIL
34:        else
35:          ColumnNumber ← 0
36:          Align ← 0
37:          StartIndent, EndIndent ← NIL
38:        else
39:          ColumnNumber ← 0
40:          Align ← 0
41:          StartIndent, EndIndent ← NIL
42:        TextLocationFeature.Push(TextBlockId,ColumnNumber,Align,
43:                                StartIndent,EndIndent)
44:        i ← i+1
  Return textPDF.Append(TextLocationFeature)

```

Finally, the algorithm appends the font features stored in the FontFeature data structure with the original extracted file.

Algorithm 6 Find Column Number

```

1: function FIND_COLUMN
2: (COLUMNINFO, STARTLOCATION, ENDLLOCATION)
3:   i ← 0
4:   while i < Columninfo.length do
5:     minCol ← Columninfo[i].minCol
6:     maxCol ← Columninfo[i].maxCol
7:     if StartLocation ≥ minCol & EndLocation ≤ maxCol then
8:       return ColumnLocation = Columninfo[i]
9:     i ← i+1
10:  return ColumnLocation.Push(0,NIL,NIL)

```

The location-based textual features use extracted file content and global document column layout style, and column's external boundaries. Algorithm 5 initially checks for the column number of the received text block. For this purpose, the function `find_column` finds the location of the text block by utilizing the boundary values. The algorithm uses the boundary values of the column and the start and end geometric value of the text block to find the margin and alignment of the text block. In algorithm 5, the value of attribute `align` represents the style of the alignment.

Algorithm 7 Neighbor Distance Feature Extractor

```

1: function NEIGHBOR_DISTANCE_FEATURE_EXTRACTOR
2: (TEXTPDF, X_TOP, Y_BOTTOM)
3:   textPDF ← LineNumber_Extractor(textPDF)
4:   i ← 0
5:   while i < textPDF.length do
6:     TextBlockId ← textPDF[i].TextBlockId
7:     PageTopDistance ← textPDF[i].Top - X_Top
8:     PageBottomDistance ← Y_Bottom - textPDF[i].Bottom
9:     PreviousLineDistance ← textPDF[i].Top - textPDF[i-1].Bottom
10:    NextLineDistance ← textPDF[i].Bottom - textPDF[i+1].Top
11:    NeighborDistanceFeature(TextBlockId, PageTopDistance
12:    PageBottomDistance, PreviousLineDistance, NextLineDistance)
13:    i ← i+1
14:  Return textPDF.Append(NeighborDistanceFeature)

```

The value 1 means the block alignment is justified, the value 2 means the text is right-aligned, the value 3 means centre alignment, the value 4 means the left alignment, 5 represents no alignment, and value 6 represents out of alignment. Finally, the algorithm append these features with the original extracted file.

Algorithm 8 LineNumber Extractor

```

1: function LINENUMBER_EXTRACTOR
2: (TEXTPDF)
3:   textPDF_Page  $\leftarrow$  textPDF[TextBlockId, PageNumber,Top, ColumnNumber]
4:   MaxPageNumber  $\leftarrow$  textPDF_Page.Maximum(PageNumber)
5:   i  $\leftarrow$  0
6:   while i < MaxPageNumber do
7:     PageInfo  $\leftarrow$  textPDF_Page.Locate(PageNumber=i)
8:     Column1  $\leftarrow$  PageInfo.Locate(PageNumber<2)
9:     TextColumn1  $\leftarrow$  Column1.SortBy(Top)
10:    PageNumberBlock.Push(TextColumn1)
11:    Column2  $\leftarrow$  PageInfo.Locate(PageNumber=2)
12:    TextColumn2  $\leftarrow$  Column2.SortBy(Top)
13:    PageNumberBlock.Push(TextColumn2)
14:    i  $\leftarrow$  i+1
15:  j  $\leftarrow$  0
16:  while j < PageNumberBlock.Length do
17:    PageNumberBlock[LineNumber]  $\leftarrow$  j
18:    j  $\leftarrow$  j+1
19:  textPDF.Append(PageNumberBlock[LineNumber])
20:  return textPDF

```

The neighbour distance features use the top and bottom margins of the research article. Likewise, The line number of the article’s lines needs to be correctly identified to measure the line distance features. In algorithm 7, the number extractor function performs the task of line number identification. The Line number extractor function performs the task of identifying the line number of a text block. Line number identification is essential for the algorithm to measure the distance between adjacent lines.

The line number extractor takes the page number, the top geometric position of the text block, and the column number as an input parameter. Initially, the system extracts page number wise text blocks. Further, the algorithm 8 extracts

Algorithm 9 Font Typography and Lexical Features

```

1: function FONT TYPOGRAPHY LEXICAL FEATURES(TEXTPDF)
2:   TitleCaseExp ← ` \n\d \.? \s+([A-Z0-9]{1,2}[a-z : \.,-]*\s ) * [ \^x20-
3:   \7E]'
4:   AllCAPExp ← ` \n\d \.? \s*[- \p{Lu} : 0-9 \s&-]*'
5:   RomanExp ← `(n[IVX]*\s([\p{Lu}0-9\u2019\&-/*\s?])*)'
6:   NoOfDotsEXP ← ` \. {1,}'
7:   NumericExp ← ` ^ [1-9]'
8:   TableEXP ← `(TAB|TABLE|Tab|Table)(.\s) — \s [(1-9)\d{0,1}
9:   |MDCLXVI](.\s|\n)'
10:  FigureEXP ← `(FIG|FIGURE|Fig|Figure|Viz)(.\s) — \s [(1-9)\d{0,1}
11:  |MDCLXVI](.\s|\n)'
12:  SpecialCharEXP ← ` ([\:\@\| \|\;]{1,})
13:  DuplicateExp ← ` (.) \1' , LastCharExp ← ` .$'
14:  KeywordExp ← ` ^ ABSTRACT|Abstract|Keyword|KEYWORDS|ACK|
15:  Acknowledgement|Reference|Ref|Bibliography ` s + . * $'
16:  i ← 0
17:  while i < textPDF.length do
18:    TextBlockId ← textPDF[i].TextBlockId
19:    FullText ← textPDF[i].PlainText
20:    AllCapital ← REGEX(FullText,AllCAPExp)
21:    TitleCase ← REGEX(FullText,TitleCaseExp)
22:    InitialNumeric ← REGEX(FullText,NumericExp)
23:    InitialRoman ← REGEX(FullText,RomanExp)
24:    NumericDots ← REGEX(FullText,NoOfDotsEXP)
25:    SpecialCharacters ← Replace(REGEX(FullText,SpecialCharEXP),
26:    DuplicateExp,`)
27:    if REGEX(FullText,TableEXP) then
28:      IsTable ← 1
29:    else
30:      IsTable ← 0
31:    if REGEX(FullText,FigureEXP) then
32:      IsFigure ← 1
33:    else
34:      IsFigure ← 0
35:    Keywords ← REGEX(FullText,KeywordExp)
36:    LastCharacter ← REGEX(FullText,LastCharExp)
37:    FontTypographyLexicalFeature.Push(TextBlockId, AllCapital,
38:    TitleCase, InitialNumeric, InitialRoman,NumericDots,
39:    SpecialCharacters, IsTable, IsFigure, Keywords, LastCharacter)
40:    i ← i+1
41:  Return textPDF.Append(FontTypographyLexicalFeature)

```

text blocks of columns 0 and 1. Then it sorts the text blocks based on the top geometric position in ascending order. If text blocks of column 2 exist, then it

repeats the process. By applying this method, the reading order of text blocks gets corrected. Finally, algorithm 7 measures the distance between the current text block from the page-top, page-bottom, preceding text block and successive text block.

Algorithm 9 develops the font typography and lexical features by applying heuristics to the plain textual content. The specialized regular expression fetches the required textual content. The regular expression assigned to the TitleCaseExp variable recognise the title case line, the AllCAPExp variable identifies the capital case line, RomanExp variable determines that the initial characters of the line are roman number and NoofDotsExp variable identifies the number of points. Similarly, the regular expression present in TableExp, FigureExp, SpecialCharExp, LastCharExp, and KeywordExp creates the lexical features.

TABLE 3.8: Time complexity of features extraction algorithms

Algorithm #	Time Complexity
Algorithm 1 Left Right Margins	$O(n)$
Algorithm 2 Column Margins Detection	$O(n^2)$
Algorithm 3 Header Footer Margins	$O(n)$
Algorithm 4 Font Features Extractor	$O(n)$
Algorithm 5 Text Location Features Extractor	$O(n^2)$
Algorithm 6 Find Column Number	$O(n)$
Algorithm 7 Neighbor Distance Feature Extractor	$O(n)$
Algorithm 8 Line Number Extractor	$O(n) + O(n^2)$
Algorithm 9 Font Typography and Lexical Features	$O(n)$

In this section, we have proposed a scheme to automatically extract the features from the research article with diversified publication styles and layout formats. The feature set development algorithm utilized the curated dataset ds1 of documents with diversified layouts and formatting styles. The evaluators manually verified the extracted features by the visual observation of the original PDF for textual features and applied mathematical verification on geometric location-based characteristics. Table 3.8 shows the time complexity of all the algorithms used for feature extraction. Therefore, the over all time complexity of feature extraction method for the proposed methodology is $O(n^2)$.

TABLE 3.9: Evaluation of the features extraction

Feature Name	Recall	Precision	Fmeasure
Font Properties			
Font Name	0.923	0.970	0.946
Font Size	1.000	1.000	1.000
Is Bold	0.962	0.971	0.967
Is Italic	0.958	0.944	0.951
Font Orientation	1.000	1.000	1.000
Text Location			
Align	1.000	1.000	1.000
Column Number	0.996	0.900	0.946
Start indent	1.000	1.000	1.000
End indent	1.000	1.000	1.000
Neighbor Distance			
line Number	1.000	1.000	1.000
Page Top Distance	1.000	1.000	1.000
Page Bottom Distance	1.000	1.000	1.000
Previous line Distance	1.000	1.000	1.000
Next line Distance	1.000	1.000	1.000
Font Typography			
All Capital	1.000	1.000	1.000
Initial Capital	0.971	0.997	0.983
Initial Numeric	1.000	1.000	1.000
Initial Roman	1.000	1.000	1.000
Numeric dots	0.923	0.996	0.958
Lexical Features			
Special Characters	1.000	1.000	1.000
Is Figure	1.000	1.000	1.000
Is Table	1.000	1.000	1.000
Keyword	1.000	1.000	1.000
Last Character	1.000	1.000	1.000

Results using the confusion matrix parameters to evaluate the extracted features are present in Table 3.9. The training data of ds1 comprising of 250 articles were tested manually by the domain experts and these results were consider as the final

evaluated of feature extraction. This evaluation ensures that the features extraction algorithm has performed accurately. The results show that the algorithms have extracted the geometric based features accurately. In a few cases, the third-party pdf content extraction library did not correctly identify the font properties of the text blocks, which also affected the output of font features. This section explains the scheme to extract different textual and geometric location base features and evaluates the extraction results of these features from articles of diversified publishers.

3.2.6 Features Ranking

Feature selection and Data cleaning is the first and most important action for model designing is the data cleaning process and the selection of the features [71–74]. The section process of features by a manual or automatic method contributes to accurately predict the desired output or variable. The unnecessary features can reduce the accuracy of the predictive model by creating the model to learn based on irrelative features. The selection of features increases the model’s accuracy and performance over high-dimensional datasets. The following are the advantages of feature selection:

1. **Improvement of Accuracy:** The accuracy of the model improves when misleading data is removed.
2. **Reduction of Overfitting:** The redundant data has more chances that the model is built based on the noise.
3. **Improves the Training Time:** The lesser features reduce the time complexity of an algorithm, hence the training process gets more efficient.

The below listed are the different features selection methods:

3.2.6.1 The Removal of Features having a Low Variance

It's a baseline method to perform selection of features [75]. This method eliminates all those features whose variance does not meet a threshold. In equation 3.1 the variable $[X]$ represents the feature matrix and p denotes the threshold value like 0.8 to remove all the features with more than 80% of common values. This method as a default, removes all features with zero-variance, i.e. commoned valued features in all then samples.

$$Var[X] = p(1 - p) \quad (3.1)$$

3.2.6.2 Recursive Feature Elimination RFE

An external estimator gives weights to the features (as example, the linear models' coefficients), recursive feature elimination (RFE) [76, 77] performs the recursive selection of smaller set of the features. Initially, the estimator model is trained based on the initial set of features. Where for each feature the importance is obtained either using the "coefficient" or the "feature importances" attribute. After this, the prouning of lesser important features is performed from set of features currentlt held. The recursive procedure on prune set is repeated untill required number of features is finally obtained. For the optimal number of features the cross-validation loop is executed on RFE.

3.2.6.3 Feature Selection Using "SelectFromModel"

In this model, the features are removed or marked unimportant on bases of the threshold parameter provided for important feature values and corresponding coefficients. "SelectFromModel" is used with estimators as a meta-transformer after fitting along with "coefficient" or "feature importances" attribute. The threshold can be numerically assigned and their are in built heuristics for searching a threshold by applying a string parameter. These heuristics are "median", "mean", and "0.1 * mean" float multiples [78]. To set the limit on selected number of features, a max features parameter can also be used along with the threshold.

3.2.6.4 Feature Selection as Part of a Pipeline

A composite estimator is built using the combination of regressors, classifiers, transformers, and other estimators [79]. For this purpose, the Pipeline tool is used. It chains many estimators to as single and suites well while processing the data with a known sequence of processes.

3.2.6.5 Univariate Feature Selection

This type of selection works by performing univariate based statistical test to select the best features and works as preprocess for the estimator [80, 81]. Following are the transform methods that includes routines for feature selection as an objects:

1. **SelectKBest** Selects the k highest scoring features and remove others [82, 83].
2. **SelectPercentile** It works on user-specified percentage for features with highest scores and remove all others [84].

The "false positive rate" [85], "false discovery rate" [86], and "family wise error" [87] are univariate feature selection statistical tests.

GenericUnivariateSelect performs a configurable hyper-parameter strategy with a search estimator for univariate feature selection. This enable it to select best features [88, 89].

3.2.6.6 K-best Features

The authors have performed feature reduction by first converting categorical values to numeric values while excluding non-convertible values, and used chi-squared (χ^2) to select K-best features. Chi-squared [90–92] calculates the statistical score for every class and nonnegative feature.

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Where :

$c = \text{degrees of freedom}$

$O = \text{observed values}$

$E = \text{expected values}$

The score is used for the selection of n number of features for the highest values from X while performing the chi-squared statistical test. They are required to include only class corresponding nonnegative features like frequencies and booleans. The chi-square test by measuring the relationship between stochastic variables removes the independent features that are most probably not relevant to the classification.

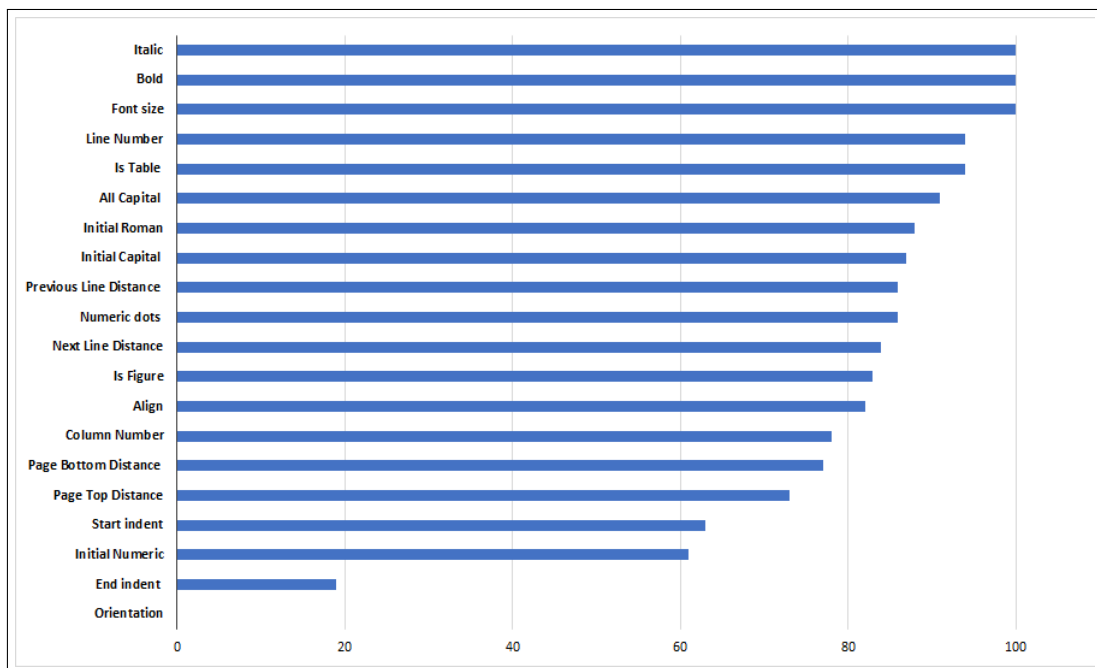


FIGURE 3.10: The Ranking of the top K selected features from ds3.1.1 using chi-square (χ^2)

As mentioned in Table 3.9, the feature extraction is evaluated on feature importance value for class labels from curated dataset 3.1.1. The Figure 3.10 shows final selected features based on the ranking of features by utilizing chi-squared method and top k selected features. Initially, the K value is set to 20 for each class label.

Then final features are manually selected based on the importance value above 20 for every each class label using K select method.

3.2.7 Summary

This section provide a comprehensive analysis of the features that will be used by the machine learning algorithm, which will be presented in the next section. The authors physically performed a detailed analysis of the layout and formatting styles of articles from curated dataset that includes 6 different diversified publishers and 40 journals. In the first section of this section explains the physical properties of individual papers. In the next section, a detail discussed is done regarding how to extract different features and their importance to identify different logical layout structures of the research articles. The features set extraction section explains the features extraction scheme and in the section, it also evaluated the accuracy of the technique to extract the features. Finally, the section discussed different feature's selection methods and selected the chi-squared method for feature selection and ranking of important features.

In the section "evaluation of the machine learning models" , the features extracted in current section will be provided to different machine learning algorithms. To select best performing model that accurately extracts the logical layout structure of the research articles.

3.3 Logical Layout Structure (LLS) and Extraction

*"The answer of the **RQ1** is presented in this section".*

RQ1: Which method will perform efficiently and accurately to extract logical layout sections LLS for research papers with diversified publication styles?

The authors of this thesis have performed a comprehensive styling and layout analysis by physical examination of the research articles from diversified publishers. On the basis of this analysis, we categorized the content of the research article into sections, which is known as the logical layout of a research article. These LLS components that we identified are the Title section, Author section, body and paragraphs, Heading and level, Table section, figures, header and footers, and acknowledgment section. The metadata of an article is mostly present within the

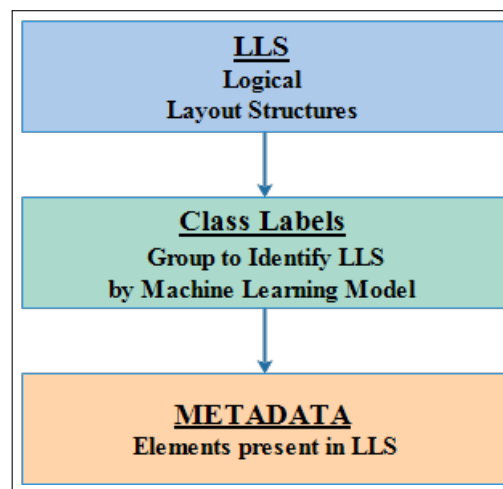


FIGURE 3.11: The hierarchical view of Logical Layout Structures, class labels, and their associated metadata.

body of these LLS components. As presented in Figure 3.11, we have described the flow for the identification of metadata with the help of class labels representing the logical layout structures. The LLS are the logical structures that are identified with the group of class labels and within each LSS component exists the metadata. Figure 3.12 explains in more detail the relationship between LLS, class labels, and metadata. The right-hand side is the snapshot of an example research article that shows the logical layout structures like header, title, author, and body/paragraph section. The left-hand side is the snapshot of the file that is the extracted content of the research article and is present in the form of text blocks. The distinct text blocks are annotated with class labels.

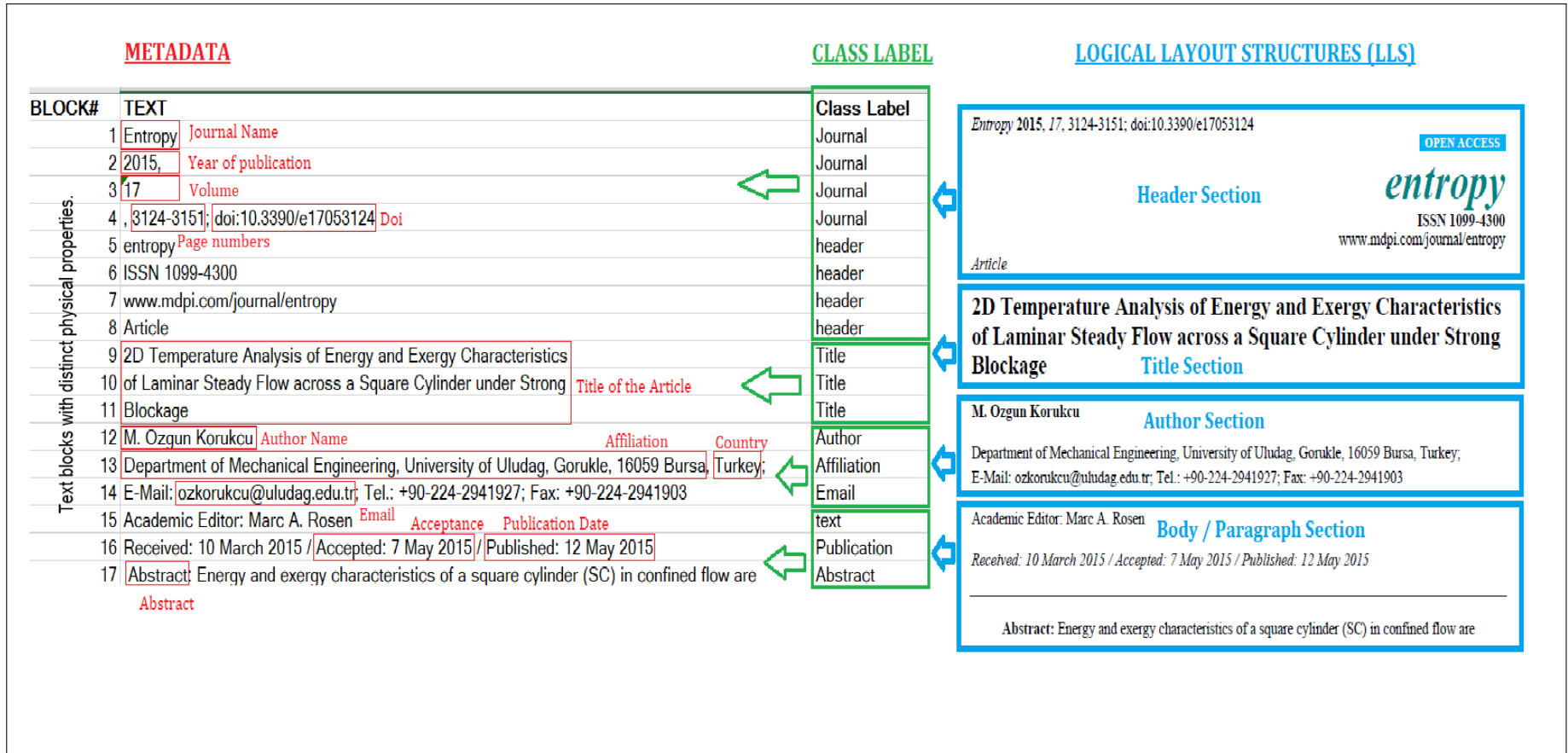


FIGURE 3.12: The Snapshot of Logical Layout Structures in research article and annotated class labels in extracted content, and their associated metadata.

These class labels are used by the machine learning algorithm to extract the LLS. Finally, we can see the metadata in text blocks and also the association with class labels, as each text block is tagged with a class label. In this section we will explain each logical layout structure and the metadata available in them.

3.3.1 Title Section

The title section is mostly present on the first page of a research article. The title is a unique identifier of a research article that very briefly explains the content present in the article. Figure 3.13 shows different layout styles in which a title is presented. We used the "title" class label to extract this LLS using a machine learning approach. The content of the title section is also the final desired metadata is presented in Figure 3.14.

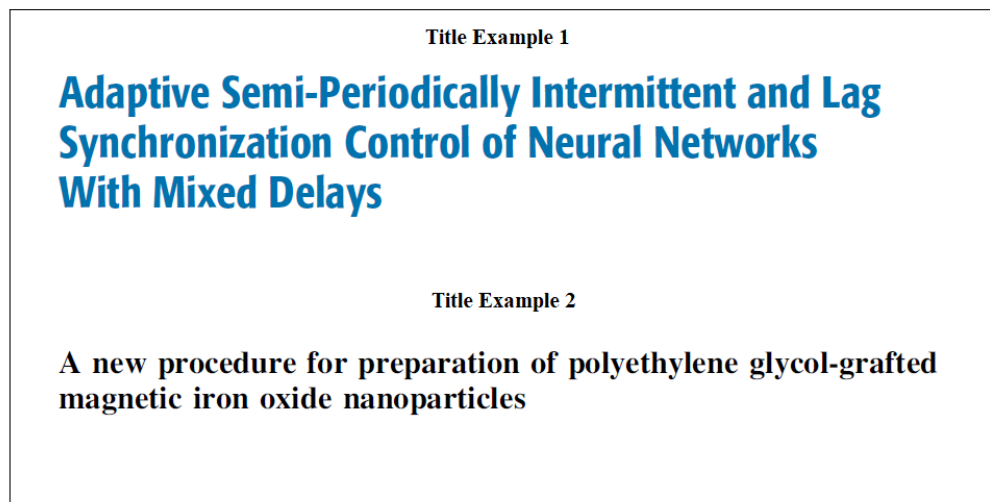


FIGURE 3.13: The title section example of different research articles.

3.3.2 Authors Section

The author section is mostly present on the first page of a research article after the title. The author section has four metadata components like author name, affiliation, the affiliation of the author, Country of the affiliation, email of the authors. Figure 3.15 shows the author section that is presented in different layout

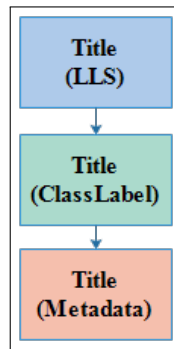


FIGURE 3.14: The title section LLS and metadata relationship in research articles.

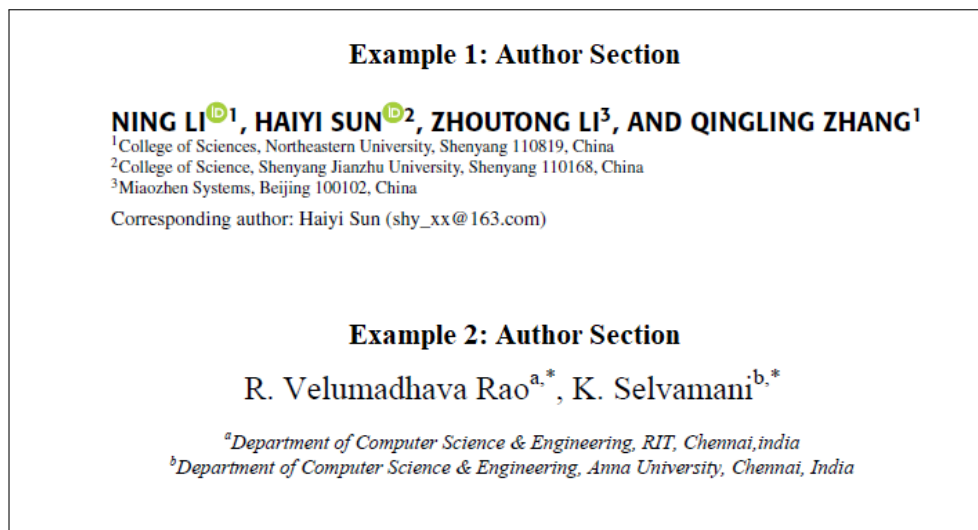


FIGURE 3.15: The authors section's examples and the formatting styles.

formats. Figure 3.16 presents the relationship between authors section LLS and metadata. The "author name", "affiliation", and "email" class labels are used to identify the content of the author's section LLS.

3.3.3 Table Section

The table section represents the valuable information present in their content regarding the research article's findings, results, or comparison. The table caption is the description present in the cells of the table, while the table sequence identifies the presentation order of the table. The table sequence can also be used to associate the reference within the body of a research article. The Figure 3.17 shows the formation styles of a table.

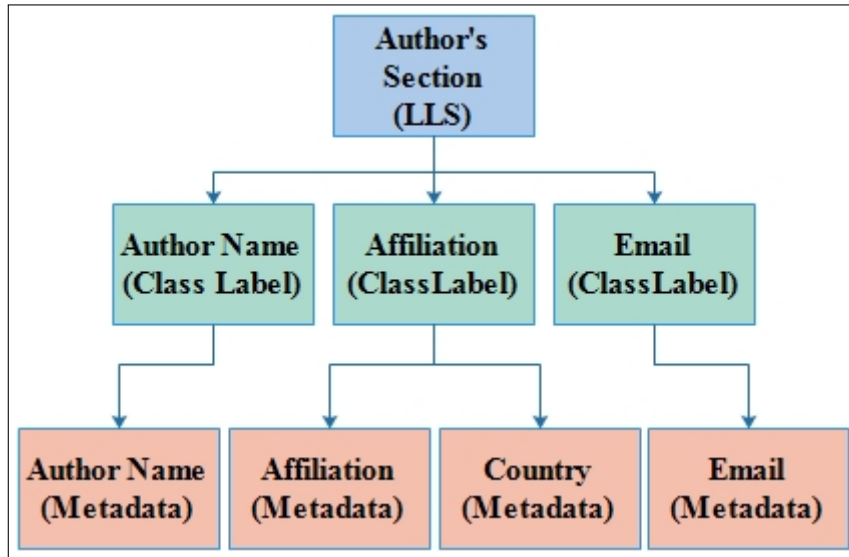


FIGURE 3.16: The relationship between authors section, class labels and the related metadata.

Table Section Examples

$t \in w$	$s(t)$	$r_1(t)$	$r_2(t)$	$r_3(t)$
1	16.1°	15.0°	15.9°	14.1°
2	15.8°	15.2°	15.7°	13.9°
3	15.9°	15.2°	15.8°	14.1°
4	16.2°	15.0°	15.9°	14.2°
5	16.5°	15.3°	15.7°	14.5°
6	16.1°	15.2°	16.0°	14.1°
7	?	15.0°	16.0°	14.3°

Table 1: Four time series in a window $w = [1,7]$.

TABLE 3. Differences between traditional e-learning and MES in mobile CPS.

	Traditional e-learning	MES in mobile CPS
Interaction Methods	Online education Mobile education	Mobile CPS based education Wearable Devices education Intelligent robot education
Location	Usually places with Internet access	Any places (wireless connection preferred)
Objects	Students of higher education	Students at all ages Underage children

FIGURE 3.17: Example of tables presentation styles.

The table section is identified by "table" class label and the table metadata like table number and captions are extracted from content classified as table section as represented in the Figure 3.18.

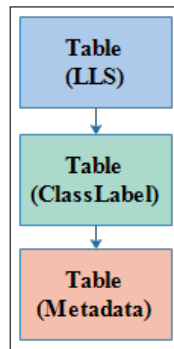


FIGURE 3.18: The relationship between table LLS and metadata.

3.3.4 Figure Section

The figure section also represents and explains the information present in the research article in a graphical form. The figure caption describes the content of the figure, The figure sequence is the presentation order and the figure sequence can be used to refer it in the body of a research paper. The Figure 3.19 shows different styles adopted by the publishers to present a figure.

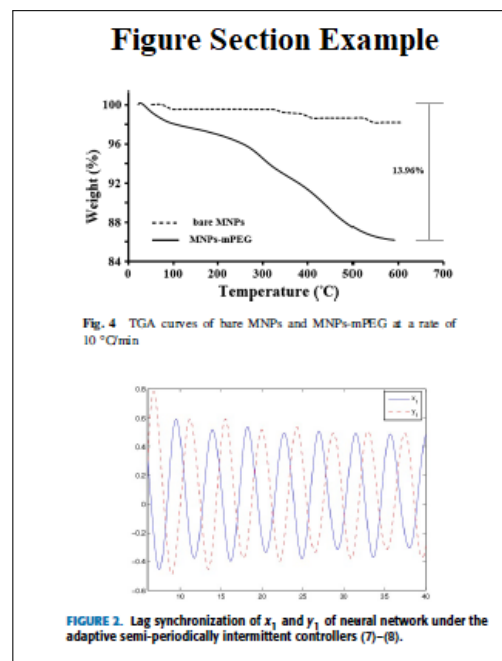


FIGURE 3.19: The caption style of figures used by different publishers.

The figure section is identified by the class label "figure" and the figure metadata like figure number and captions are extracted from content classified as figure section as represented in the Figure 3.20.

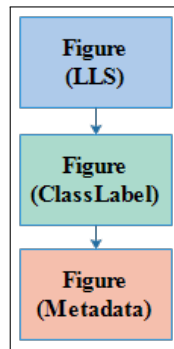


FIGURE 3.20: The relationship between the figure section and its metadata.

3.3.5 Headings and Levels

Another logical layout structure is The table of contents that are presented in the form of headings. The heading is present in levels for the identification of sections and sub-sections. The extraction of heading and their levels is a challenge, therefore many techniques have been described in previous literature. Different variants of the heading formats are shown in Figure 3.21 that are adopted by different publishers. The heading and their levels are composed in a different formatting style from the main text body, which makes them easier to be identified by a human. However, to extract the headings and sub-headings, a variety of features are required to be identified and annotated for a machine-learning algorithm to extract them in the correct manner.

We have used "headings" class label for all the levels of headings. This enables the system to identify all headings present in a research article. The headings are further processed in metadata extraction phase to extract the level of headings. The system extracts heading level 1 (section), heading level 2 (subsection), and heading level 3 (subsubsection) as presented in Figure 3.22.

3.3.6 Headers and Footers

The header and footer sections keep valuable information regarding the publication-related information of an article. Figure 3.23 presents only two header and footers

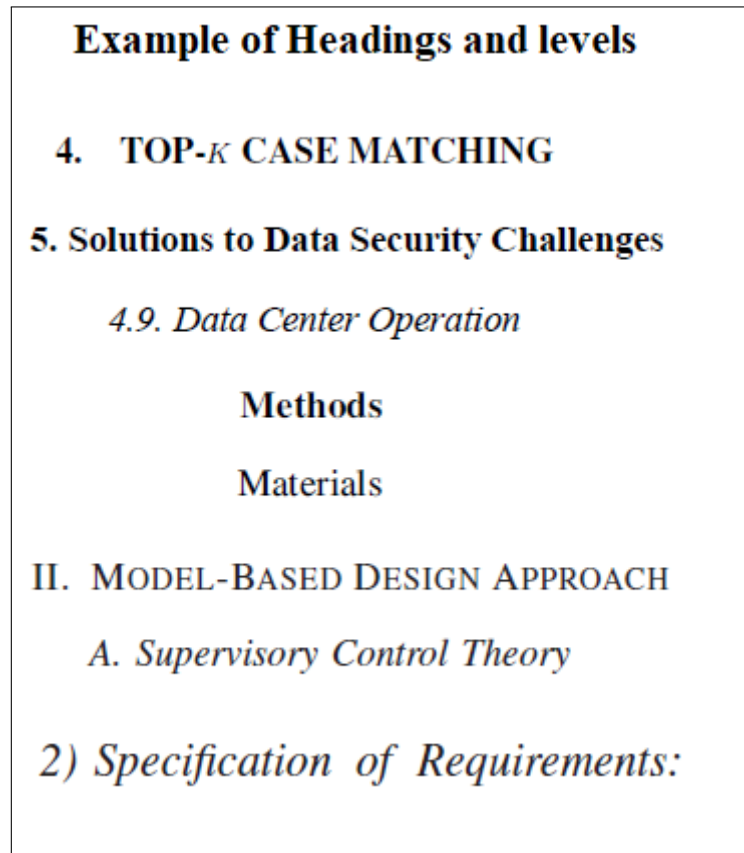


FIGURE 3.21: The heading styles adopted by different publishers.

examples, which are adopted by the publishers. As discussed, the metadata present in these layout components is related to the publication information like journal name, publication date, volume and issues, venues, pages, and doi, etc. Another important piece of information that is present in the footer section is footnotes and supplementary material etc. A lot of approaches that were previously defined have only extracted the publication information of the research articles, as it's a challenge to extract such metadata when the publication styles are diversified and valuable for the digital libraries (DL) to store the articles publication information.

We tagged the content of header section with "header" and "journal" class labels. The journal class label has metadata regarding journal title, doi, page numbers, and year. The header class label has metadata regarding page number and journal title. These relationship are presented in the Figure 3.24.

Different journals use the footer section to publish information related to acknowledgments, journal information, article's publication information. This content is

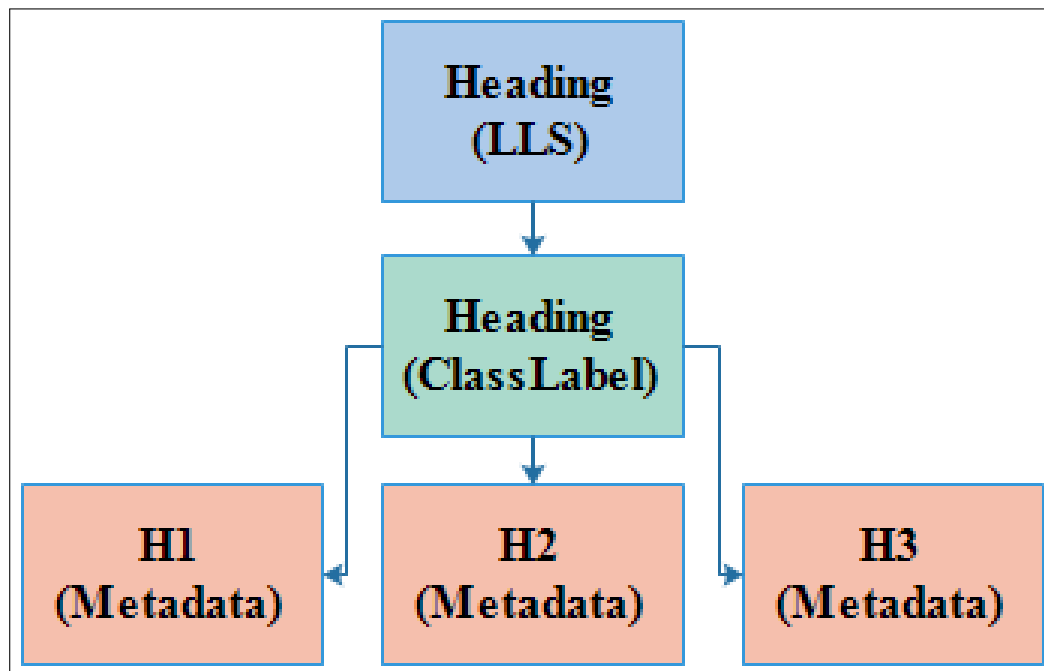


FIGURE 3.22: The relationship between heading LLS and metadata.

mostly published in a similar composition style. Therefore, we only gave footer class labels to these elements and later applied heuristics and NLP to identify the metadata. The information available in the footer section is presented in Figure 3.25.

3.3.7 Acknowledgment Section

The acknowledgment section has the information regarding the funding of the research work. This enables the reader to perceive the scope of the work and the usability of the research work in the form of the applications. Figure 3.26 shows different styles adopted in articles of different publishers to write the acknowledgments. The acknowledgment section in some cases has a heading like 'acknowledgments' or they can be written in the footer section. This section can be present at the initial page or may be presented at the end of the research article, mostly before the bibliography or reference section.

The Figure 3.27 shows the association of funded project and funding agency related metadata with the acknowledgement section.

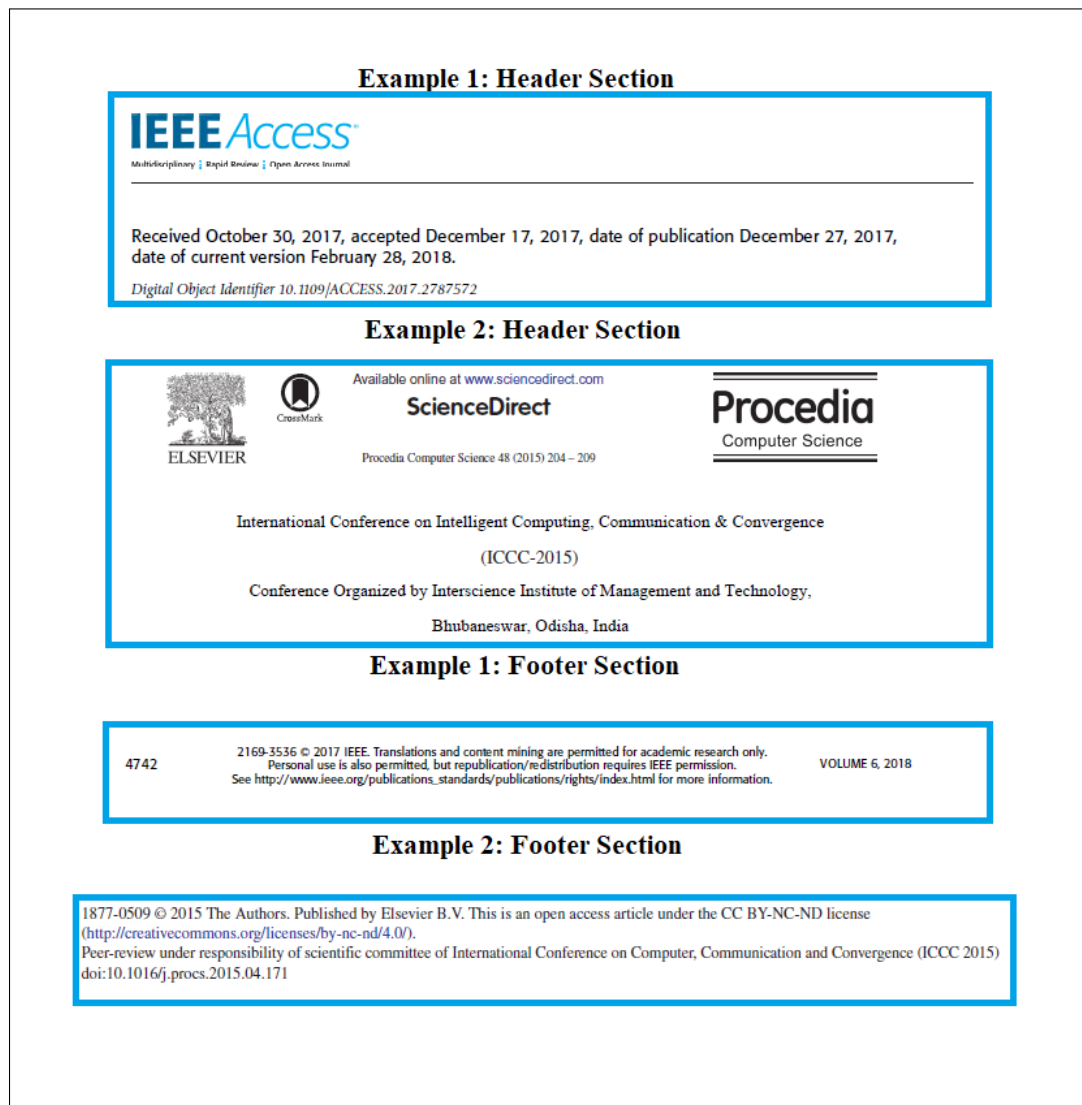


FIGURE 3.23: Examples of header and footer of an article with related meta-data.

3.3.8 Body / Paragraphs

The paragraphs and main body of an article present the actual content of the conducted research. The different layouts already mentioned above supports the literary work of the research publications. The dataset and the journals which we have selected to evaluate our approach were published in two forms of column layout styles. In the previous section, while extracting the physical properties of the research document, we have discussed in detail the column styles that are single column and double column layout styles. This style has a major impact on the forming of different logical layout structural components. The Figure 3.28

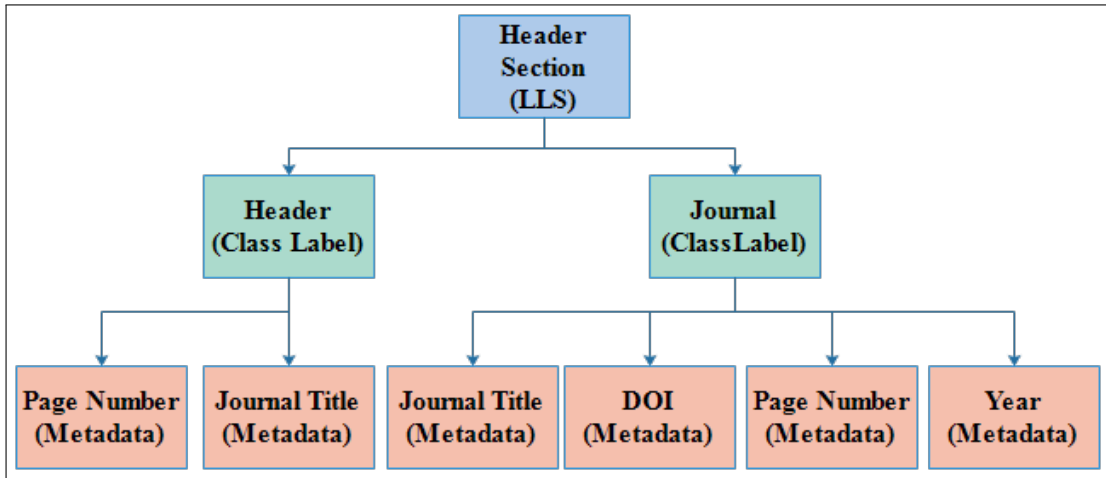


FIGURE 3.24: The metadata present in header section and the class labels to identify LLS.

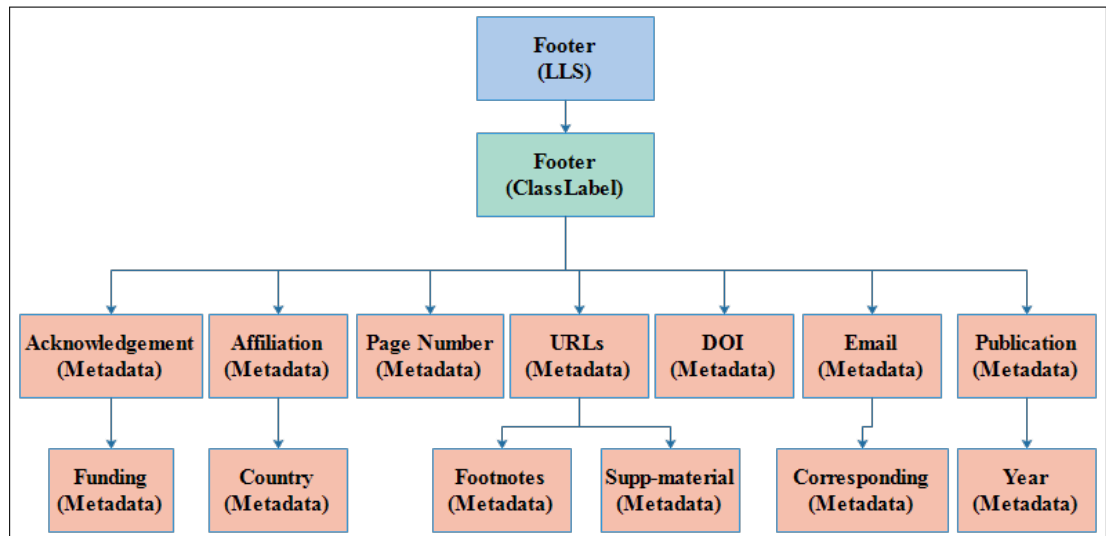


FIGURE 3.25: The footer section of an article with related metadata.

shows column styles adopted by the publishers.

We assigned four class label tags to the content of the body/ paragraphs section. These are "abstract", "keywords", "publication date", "text", and "references". The association of these class labels and metadata are presented in Figure 3.29 . We did not consider the output of "text" and "references" class labels in section 3.3.10, while evaluating the results of body and paragraph LLS.

In the next section, we shall discuss different machine learning models that we

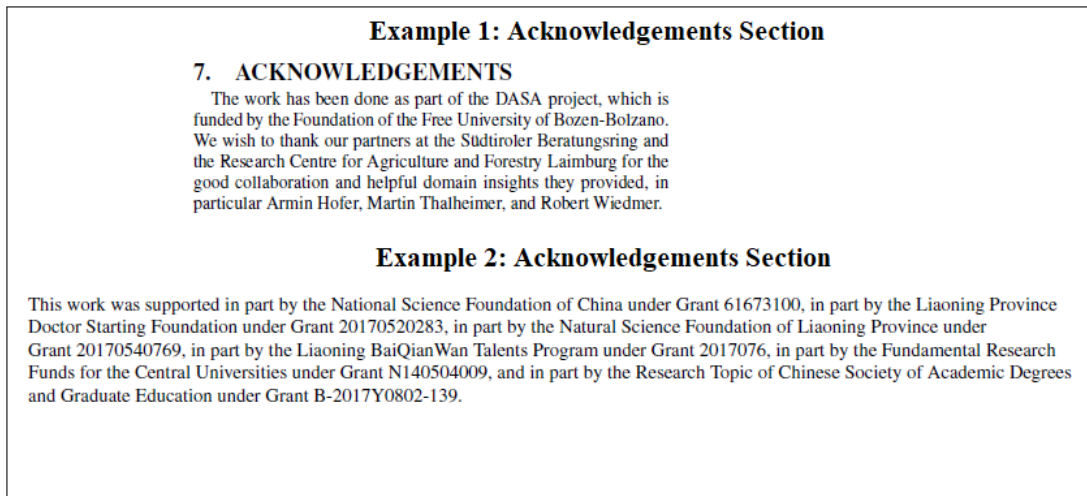


FIGURE 3.26: The caption style of figures used by different publishers.

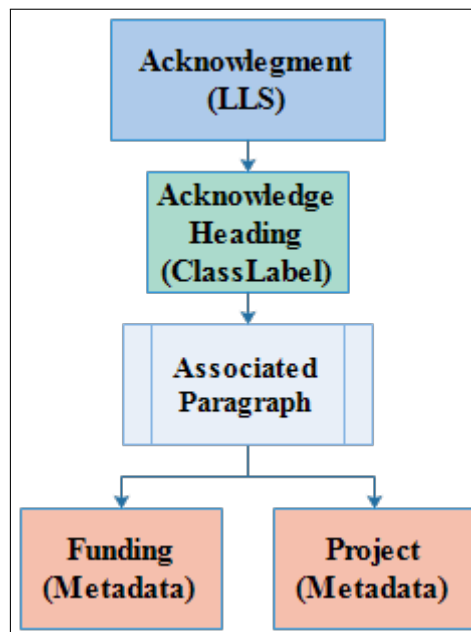


FIGURE 3.27: The acknowledgement section and related metadata.

have selected on the basis of features types, nature of the application, and literature review. We shall first perform the theoretical evaluation of the techniques and then we shall perform experiments to analyze the accuracy of different machine learning techniques to extract different logical layout structural components.

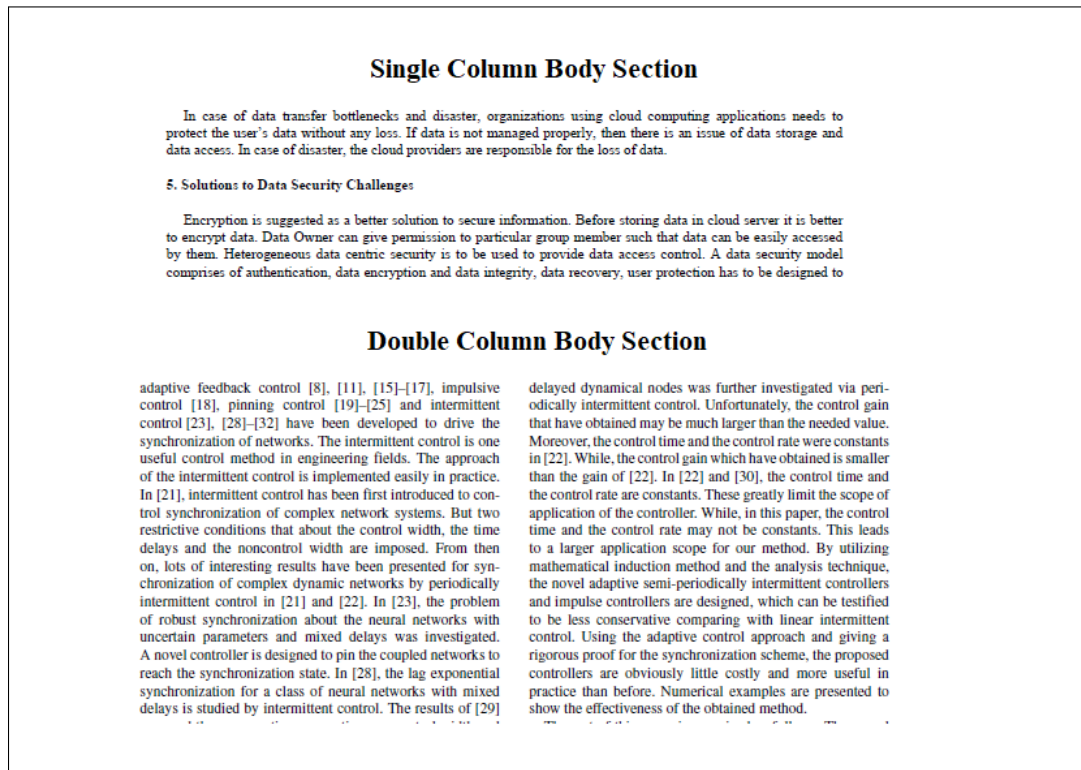


FIGURE 3.28: The column layout styles (single and double column) and the paragraphs.

3.3.9 Machine Learning as LLS Extraction Method

We have divided the problem at hand into sub-tasks to extract metadata from PDF-based research articles. Here, we are focused on the extraction of the logical layout structure. In this regard, we have used machine learning algorithms that are applied to the content of text blocks to identify their structural group. So we have engineered features to extract LLS that are presented in the section 3.

In this section, to answer the **RQ1**, we have performed a comprehensive review of supervised machine learning algorithms, to find logical layout structures LLS from research publications. We have performed the theoretical and experimental evaluation of selected machine learning models to find their efficiency and accuracy, for the selection of the models to be used in the final approach.

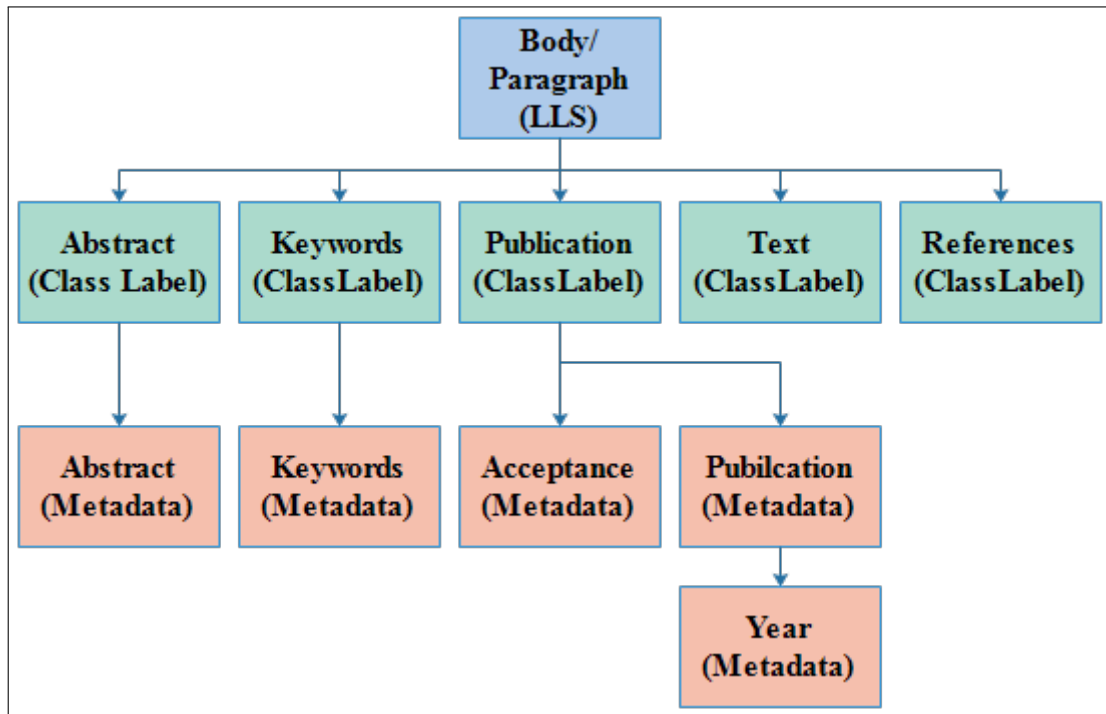


FIGURE 3.29: The metadata and class labels of body/paragraph section.

Machine learning is used to instruct machines on how to manage the data more efficiently. Most often, after manually inspecting the data, humans are unable to interpret the hidden patterns or extract relevant information from the given data. For this purpose, machine learning algorithms are applied [93]. Many industries applications apply machine learning algorithms to extract relevant information. The purpose of machine learning is to learn from the data. Several studies have been conducted to make machines learn by themselves [94, 95]. In this regard, many researchers and scientists apply various approaches to find an efficient algorithm for machine learning.

Ayon Dey [7] in his survey study categorized machine learning into eight types. According to him the types of learning are supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, multi-task learning, ensemble learning, neural network, and instance-based learning. Therefore, taking advantage of machine learning models we have developed the third stage to extract logical layout structures. So that our selected algorithm based on our generic

set of features can consistently perform to extract LLS for articles from unknown publication styles.

3.3.9.1 Discussion

Before the setup of a machine-learning algorithm to extract logical layout structural components from diversified layout styles. Theoretically, a few points are considered regarding the properties of the features and the nature of the problem. As earlier described the features are of different data types like numerical, nominal, and boolean. The problem and properties show a non-linear relationship between the features. It's a classification problem with multiple class labels. The number of features is lesser than the training data instances $n \gg p$. Based on the facts described, we only selected the machine learning algorithms for evaluation to prove our theoretical evidence that best fits for non-linear, distinct features and multi-class labels, on a large dataset. In the succeeding subsection, we shall present a brief overview of different machine learning algorithms that we evaluated for our proposed methodology, as comprehensive details and computational complexities are available [7, 96–99].

3.3.9.2 Theoretical Evaluation

The evaluation is based on n , representing the number of the training samples, where p is the number of the features. For tree base classification algorithms n_{trees} represents the number of the trees. Similarly, the number of the support vectors is denoted by n_{sv} , and finally, n_{li} is the number of neurons at a layer i in a neural network.

Naïve Bayes algorithm depends on the conditional probability based on Bayes theorem and generates a tree based on probability known as Bayesian Network.

Its characteristics are independent of each other. The time complexity is linear for both testing $\mathcal{O}(n * p)$ and prediction $\mathcal{O}(p)$ of the model. where n is the number of features and p is the number of class labels. The posterior probability is calculated by $P(A|B)$ where,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

k -nearest-neighbor are defined in the terms of distance of all instances that correspond to the point in n -D space. It searches the pattern space of close unknown tuple for k training and classifies it by a majority vote of its neighbors. The distance metrics, such as Euclidean distance, Manhattan, and Minkowski are used to define “closeness”. The time complexity can be reduced to a constant $\mathcal{O}(1)$, independent of the training dataset of $|D|$. The time complexity for KNN of the k nearest from n neighbors at a point in the plane with d data dimensionality is $\mathcal{O}(kdn)$ [100].

The decision tree classifier constructs the tree based on entropy and information gain by using the ID3 algorithm, unlike the standard deviation reduction method. The nodes represent the attributes needed to be classified, while the branches represent the allowed value. A full homogeneous sample achieves entropy equal to zero by dividing the sample into equal parts. The time complexity to train the classifier is $\mathcal{O}(p * n \log n)$, and for prediction is $\mathcal{O}(p)$ [96]. The ensemble classifiers use a combination of models to increase the accuracy [101]. Different methods can be applied, where improved model M^* is created with combine series of k learned models $\{M_1, M_2, M_3, \dots, M_k\}$ on data D with k learned sets, $\{D_1, D_2, D_3, \dots, D_k\}$. The bagging method considers majority vote by models to improve the accuracy and the term bagging origins from “bootstrap aggregation”. In Adaboost (Adaptive boosting), assigns weights to each classifier’s vote for each training tuple to boost the accuracy of the learned method. The weight is calculated on errors due to misclassification and the subsequent model focuses on classified tuples. Weight is calculated using $\log \frac{1 - \text{error}(M_i)}{\text{error}(M_i)}$. Stacking is a heterogeneous ensemble that consists of different models. The idea is to combine predictions of the base learners (level-0), do not just vote, and provide as an input to meta learner level-1 models. The Random forest ensemble the decision tree classifiers so that the collection

of classifiers is a “forest”. Each tree depends on the independently sampled values and all the trees have the same distribution in the forest. The accuracy is achieved using each tree’s vote and the most popular class is returned. The time complexity of bagging is $\mathcal{O}(T * t)$, where T is the number of iterations and t is the average time complexity of each model. The time complexity of Adaboost is $\mathcal{O}(T * f)$, where f is the complexity of the weak learner. The time complexity of stacking is $\mathcal{O}(E_1 + E_2 + E_3 + \dots + E_k)$. And the time complexity of random forest is $\mathcal{O}(ntree * p * d * n)$, where d is the depth of the tree.

The support vector machines classify both linear and nonlinear data. It transforms the training data into a new higher dimension by using nonlinear mapping and searches for linear hyperplanes. SVM uses support vectors to find this hyperplane. The tuples of different classes are separated using “decision boundary” or margins. The maximum distance between margins and classes is drawn. Finding maximum marginal hyperplane (MMH) and support vectors makes it a quadratic optimization problem. For linear data, linear SVM is employed and for nonlinear data, SVM provides a bag of $K(x, x')$ kernel tricks.

$$linear : \langle x, x' \rangle$$

$$Gaussian/RBF : \exp(-\gamma \| x - x' \|^2)$$

The Radial Basis Function (RBF) SVM kernel considers two parameters C and γ . C is the Regularization Parameter that helps SVM optimization to avoid miss classifying SVM for each instance. So a higher C value will choose a small margin hyperplane to minimize the miss classification. For a low C value, the margin will be big, which can have a high rate of miss classification. The γ parameter identifies the influence reach of a single training instance. The higher γ value considers instance values near the plausible hyperplane, and a low γ value will consider points at a far distance. So low γ will help find the correct hyperplane. In gaussian/RBF equation the γ is specified by keyword

gamma, that must be between 0 and 1. The overall training time complexity for kernel method is $\mathcal{O}(n^3p)$, and prediction time complexity is $\mathcal{O}(n_{sv}p)$ [102].

The Neural networks [103] are non-deterministic algorithms that generalize well but have the minimum mathematical foundation. They are learned in an incremental fashion, and nontrivial multilayer perceptrons are used to perform complex functions. Supervised, unsupervised, and reinforcement are three main types of artificial neural networks. The time complexity is $\mathcal{O}(n * e(\sum_{i=1}^{h-1} nl_i nl_{i+1}))$, where e is epochs and h is total number of layers in a neural network .

3.3.10 Evaluation of Machine Learning Models

We have selected the articles from diversified publishers like IEEE, ACM, Elsevier, MDPI, Springer, and Pensoft, and also used 3 gold standard datasets. These datasets have articles with diversified layouts and publishing styles, therefore it's a challenge to build a comprehensive approach that can extract the logical structures and metadata from such varying document layout structures. We provided text blocks along with associated features and annotation to machine learning classification algorithms for the logical layout structure to measure the accuracy of the algorithms. We have performed a theoretical evaluation of classification models using mathematical grounds to prove the efficiency in terms of time complexity, both in cases while training the model and testing the models on real data as shown in Table 3.10.

TABLE 3.10: The time complexity of models to extract each Logical Layout Structure.

Model Name	Training Complexity	Prediction Complexity
Naïve [104]	$\mathcal{O}(n * p)$	$\mathcal{O}(p)$
SVM [102]	$\mathcal{O}(n^2p + n^3)$	$\mathcal{O}(n_{sv}p)$
KNN [100]	$\mathcal{O}(kdn)$	$\mathcal{O}(n)$
Decision Tree [96]	$\mathcal{O}(p * n \log n)$	$\mathcal{O}(p)$.
Extra Tree [96]	$\mathcal{O}(p * n \log n)$	$\mathcal{O}(p)$.
Bagging [101]	$\mathcal{O}(T * t)$	$\mathcal{O}(T * t)$
Adaboost [101]	$\mathcal{O}(T * f)$	$\mathcal{O}(T * f)$
Stacking[101]	$\mathcal{O}(E_1 + E_2 + E_3 + \dots + E_k)$	$\mathcal{O}(E_1 + E_2 + E_3 + \dots + E_k)$
Random Forest [101]	$\mathcal{O}(ntree * p * d * n)$	$\mathcal{O}(ntree * p * d * n)$

TABLE 3.11: The F-Score performance matrix is demonstrating the accuracy of machine learning algorithms for extracting the Logical Layout Structures for the curated training dataset.

	Naïve	SVM	KNN	Decision	Extra	Bagging	Adaboost	Stacking	Random	Rule
	Bayes	RBF Kernel		Tree	Tree				Forest	(stratified)
Title Section	0.57	1.00	0.66	0.95	0.94	0.90	0.36	0.75	0.94	0.15
Author	0.68	0.98	0.79	0.91	0.91	0.92	0.35	0.22	0.93	0.15
Acknowledge	0.00	1.00	1.00	1.00	0.18	1.00	0.00	0.00	1.00	0.12
Affiliation	0.78	0.88	0.80	0.89	0.94	0.93	0.64	0.60	0.94	0.16
Email	0.52	0.90	0.65	0.75	0.77	0.84	0.00	0.04	0.81	0.07
Figure	0.13	0.96	0.70	0.80	0.80	0.83	0.52	0.74	0.84	0.07
Heading	0.45	0.95	0.88	0.95	0.95	0.96	0.69	0.82	0.95	0.08
Table	0.16	0.93	0.29	0.58	0.53	0.64	0.00	0.00	0.55	0.03
Header	0.45	0.89	0.61	0.44	0.86	0.56	0.26	0.54	0.82	0.21
Footer	0.75	0.92	0.86	0.62	0.58	0.67	0.35	0.00	0.00	0.11
Abstract	0.84	0.96	0.87	0.94	0.82	0.78	0.69	0.48	0.37	0.50
Keywords	0.67	0.92	0.74	0.83	0.66	0.65	0.43	0.25	0.57	0.01
Publication	0.78	1.00	0.89	0.75	0.87	0.77	0.54	0.37	0.85	0.24
Avg F-Measure	0.52	0.95	0.75	0.80	0.75	0.80	0.37	0.37	0.74	0.15
Avg Recall	0.53	0.93	0.73	0.79	0.78	0.74	0.35	0.36	0.78	0.17
Avg Precision	0.52	0.96	0.76	0.82	0.72	0.86	0.38	0.38	0.70	0.13

Table 3.11 shows the performance evaluation of classification models to extract different logical layout structural components. The properties of features and data suggest that the feature values are Non-linear and there is a non-linear relationship between the features. The test dataset used is of small size, and the features are lesser than the training data instances $n \gg p$. The annotation suggests that it is a multi-class label problem. We used Gaussian RBF as an SVM kernel, where the C value is set to 1 and the gamma value to 10. The K-Crossfold is used for data split for the training of all the algorithms. In our scenario, the SVM performed better than other machine learning models based on the properties of the data and features. The Naïve Bayes algorithm uses probability to estimate the class label. The decision tree perform better than the remaining algorithms as the non-linear relationships between parameters do not affect the model performance.

3.4 Metadata Extraction

"Answer of the RQ3 is presented in this section".

RQ3: How to develop a comprehensive model to effectively extract metadata that is trained on research articles from multiple domains of different publishers?

In this section, we have extended our approach to extract metadata from diversified publishers based on the logical layout structure in the previous section. In this regard, we have performed four additional tasks. At first, as discussed in the previous section, we use our curated dataset from different journals of diversified Publishers to evaluate our proposed approach. This dataset of articles has diversified layout and formatting styles and it has metadata elements that are composed in different formats. Secondly, based on the analysis performed on different approaches to extract the metadata, we have identified a superset of nineteen metadata elements from scientific research publications from state-of-the-art as shown in Table 3.12. We shall evaluate our approach to extract these elements. The third task is to evaluate our methodology to extract metadata on our proposed dataset (ds3) with state-of-the-art approaches. Finally, we evaluated our approach with the renowned approach sectLabel by using the benchmark sectLabel dataset (ds2).

The proposed methodology to extract metadata from research articles is split into four different steps. As discussed earlier, the first task is to extract textural and logical layout features for PDF-based research articles. The second step is to extract the logical layout features, we have already accomplished this task in section 3.2. We also used the same approach to extract the logical layout structural components using the machine learning algorithm that is identified by us in section 3.3.10. Here we have extracted eight logical structural components or the sections, which are title section, author section, header and footer, headings, figure, tables, acknowledgment section, and body/paragraph. In this section, we have extracted additional metadata elements that are presented in Table 3.12. In the next section, we shall explain each metadata element and define what strategy we have applied

TABLE 3.12: The Logical Layout Structures and their associated metadata.

Metadata Extracted	FLAG-PDFe	PDFX	CERMINE	sectLabel
Title	✓	✓	✓	✓
Author Name	✓	✓	✓	✓
Affiliation	✓	✗	✓	✓
Country	✓	✗	✗	✗
Emails	✓	✓	✓	✓
Journal Title	✓	✗	✓	✗
Publication Date	✓	✗	✓	✗
Acceptance Date	✓	✗	✗	✗
Pages	✓	✓	✓	✓
Version	✓	✗	✓	✗
Volume	✓	✗	✓	✗
doi	✓	✓	✓	✗
Figure Caption	✓	✓	✗	✓
Table Caption	✓	✓	✗	✓
Heading	✓	✓	✗	✓
Sub-Heading	✓	✓	✗	✓
SubSub-Heading	✓	✓	✗	✓
Abstract	✓	✗	✓	✗
Keywords	✓	✗	✓	✓
References	✓	✗	✓	✓

to the logical layout structure for the extraction of these metadata components.

The overall methodology is represented in a graphical form in Figure 3.30.

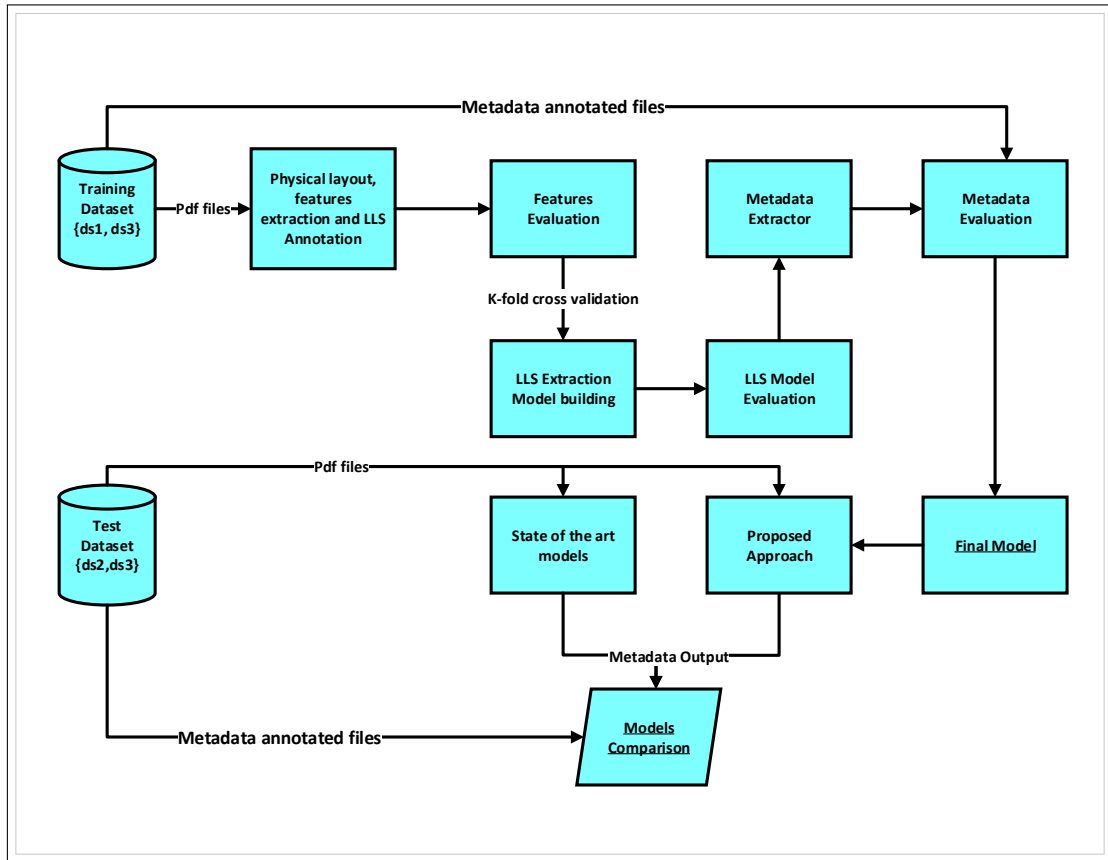


FIGURE 3.30: PDF base research article’s metadata extraction proposed methodology and comparison with state-of-the-art

The evaluation of our proposed system is conducted on two separate datasets. At first, we extracted the metadata elements of research articles using our constructed dataset (ds3). We used commercially available online state-of-the-art approaches like CERMINE and PDFX. We provided our dataset to these approaches to extract the metadata. They generated the output of extracted content in the form of XML files. We used xQuery on these files to locate the metadata elements. Finally, we compared the output of our approach with the state-of-the-art approach using metadata annotated files. The second dataset we used to evaluate our approach is of sectLabel (ds2), we compared the output of our approach to extract metadata with the output of sectLabel approach.

3.4.1 Title of the Article

As mentioned in the section 3.3, the title of the research article is the only metadata element that is present in the title section. As we have extracted it in logical layout structure as described in section 3.3.1 with the help of a machine learning algorithm. Therefore, we have not performed any further processing on this content to extract the desired metadata.

3.4.2 Authors and Affiliations

The metadata regarding the authors and their affiliation is available in the author section as identified in section 3.3.2. As previously discussed, the authors and affiliations are extracted by the SVM approach. As authors and affiliations were tagged with their respective class labels. We have also developed a strategy to extract the author's affiliation that is explained in the section 4.1.4.1. In some scenarios, it was observed that affiliations were present in the footer section of the article on the first page. The Sequence of author names with numeric or symbols separated by commas' or tab spacing. And Then in the footer section the related affiliations with numeric or symbol as referred against an author. This enhancement gave more precision to our strategy to extract the author's affiliation in more complex scenarios.

3.4.3 Country of Affiliation Extraction

The country of the affiliation is present within the address of the institute or organization. The algorithm 10 shows the selection of country from the author's affiliations. The list of affiliations (mentioned as data a structure Affiliation) is provided to the algorithm as an input that maintains the list of countries (in data structure country). The data structure country contains all the country names

and also have know short names and abbrivations for the countries like "UK" for the "United Kingdom". The algorithm applies n-gram [105] method on the words present in an element of affiliation list and the countries list. This returns the intersection of words present in both lists. There can be more then one countries in the list of affiliations, so the selectedCountry data structure has the final list of extracted countries. The equation $A \cap C = \{x : x \in A \text{ and } x \in C\}$ represents the intersection function, where A is the array of words in an affiliation and C is the list of countries. The intersection of both arrays returns the common country name, if present in both arrays.

Algorithm 10 Search country algorithm

```

1: function FINDCOUNTRY(AFFILIATION)
2:    $j \leftarrow 0$ 
3:    $i \leftarrow 0$ 
4:   while  $i \leq \text{Affiliation.length}$  do
5:      $country \leftarrow \text{intersection}(country, \text{Affiliation})$ 
6:     if  $country \neq \text{NULL}$  then
7:        $\text{selectedCountry}[j++] [0] \leftarrow country$ 
8:        $i \leftarrow i + 1$ 
9: return  $\text{selectedCountry}$ 

```

3.4.4 Author's Email Extraction

The email of the author is present within the author section or in the footer section. Mostly mentioned against the title as the corresponding author or author's reference. We used an email parser using the regular expression on these text blocks and used the same approach to extract email against the author that is used to extract affiliation against authors as mentioned in the current section 3.4.2. The regular expression $\wedge ([a-zA-Z0-9_-\.\.]) @ ([a-zA-Z0-9_-\.\.]) \wedge . ([a-zA-Z]{2,5}) \$$ is used to extract and valid the email.

TABLE 3.13: The regular expression to extract and identify different metadata elements, from sample research articles .

Regular Expression	Metadata	ID
Example 1: J Nanostruct Chem (2015) 5:227–236		
<code>\w+(\w+)*</code>	Publisher	1
<code>\s(\+(\s{1}[0-9]{4})\s(\))+</code>	(Year)	2
<code>(\s{1}[0-9]+)</code>	Volume	3
<code>\s(\:+)(\s{1}[0-9]+)\s(\-+)(\s{1}[0-9]+)</code>	Page range	4
Example 2: Bioresources and Bioprocessing (2014), 1:13		
<code>\w+(\w+)*</code>	Publisher	5
<code>\s(\+(\s{1}[0-9]{4})\s(\))+</code>	(Year)	6
<code>(\s{1}[0-9]+)</code>	Volume	7
<code>\s(\:+)(\s{1}[0-9]+)</code>	Issue	8
Example 3: Appl. Sci. 2017, 7, 890; doi:10.3390/app7090890		
<code>\w+(\w+)*</code>	Publisher	9
<code>(\s{1}[0-9]{4})(\s(\,))</code>	Year,	10
<code>(\s{1}[0-9]+)\s(\,+)</code>	Volume,	11
<code>(\s{1}[0-9]+)\s(\;+)</code>	Issue;	12
<code>\s(doi:10.(\\d)+/([^\s\\>\"\\<]))+</code>	DOI:	12
Example 4: European Research on Management and Business Economics 22 (2016) 124–130		
<code>\w+(\w+)*</code>	Publisher	14
<code>(\s{1}[0-9]+)</code>	Volume	15
<code>\s(\+(\s{1}[0-9]{4})\s(\))+</code>	(Year)	16
<code>(\s{1}[0-9]+)\s(\-+)(\s{1}[0-9]+)</code>	Page range	17
Example 5: ACM Transactions on Database Systems, Vol. 43, No. 1, Article 6. Publication date: March 2018.		
<code>\w+(\w+)*</code>	Publisher	18
<code>\s(\, Vol.)(\s{1}[0-9]+)</code>	Vol.	19
<code>(\, No.)(\s{1}[0-9]+)</code>	, Issues.	20
<code>(\, Article+)(\s{1}[0-9]+)</code>	, Article No	21
<code>(\ . Publication date:)([January Febuary March April May June July August September October November December]+)(\s{1}[0-9]{4})</code>	Publication date:	22

3.4.5 Extraction of Publication and Acceptance Date

The publication dates related information is present in the header or footer sections of the logical layout structure (LLS). We employed a two-step approach on LLS, first is the identification of the publication date and acceptance date text block. The publishers have used different styles to mention the date like Publication date, Pub. Date., Published, Accepted, Available online, date of publication and

Published online, etc. A regular expression is used to identify these keywords. It was also noted that in a few scenarios, the publication date is available in the same text block with the journal title or volume number is mention. In some cases, as shown in Figure 3.13 and id 22 the publication date is part of journal information. In the second step, we used a parser to extract the date that is available in different formats by using the

“(?:\d{1,2}[-\th|st|nd|rd\s]*)?(?:Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec|January|Feburay|March|April|May|June|July|August|September|October|November|December)?[a-z\s,.]*(?:\d{1,2}[-\th|st|nd|rd)\s,]*)?(?:\d{2,4})+” regular expression. Finally, we applied year, month, and date range validators using the calendar checks.

We also extracted the year of publication from the publication date and also used reference id 2, 6, 10, and 16 of Figure 3.13 to get this metadata.

3.4.6 The Title of Journal

The extraction of the title of the journal is a challenge as there are not specific keywords associated with this metadata. We analyzed the research publications from diversified publishers and observed that the journal’s titles are available in the header or footer logical layout structural component. Therefore, we developed an approach that performs Journal specific template-matching on the content of the header or footer section on the first page of the articles. We used different regular expressions to find and verify the journal-title string from the LLS components identified as ”Journal” and ”footer”. $\backslash\mathbf{w}+(\ \backslash\mathbf{w}+)*\backslash\mathbf{s}(\backslash(+)(\backslash\mathbf{s}\{1}[0-9]\{4})\backslash\mathbf{s}(\backslash+)(\backslash\mathbf{s}\{1}[0-9]+)\backslash\mathbf{s}(\backslash:+)(\backslash\mathbf{s}\{1}[0-9]+)\backslash\mathbf{s}(\backslash-+)(\backslash\mathbf{s}\{1}[0-9]+)$. The first part of the string was identified as journal’s title $\backslash\mathbf{w}+(\ \backslash\mathbf{w}+)*$.

3.4.7 Issue and Volume Information Extraction

The volume number of research articles from diversified publishers has different composition styles. The volume and issue numbers are present in the header or

footer logical layout structure. This information has diversified composition styles as they can have keywords like "Volume, Vol, No, and Issue", etc. In such cases, we used keywords-based parsers to extract this metadata. However, in some other instances, there can have no keywords mentioning this metadata. Therefore, for this scenario, we have applied the template-based approach that is based on regular expression present in journal information in header or footer section as shown in the Table 3.13.

3.4.8 Page Number Extraction

The page number information of the articles under an issue is located in the header or footer logical layout structure. This metadata also does not have any associated keywords; therefore, we employed a templated-matching approach to extract the page numbers. The templates use geometric layout information and phrase parser, and finally, we applied a numeric parser to fetch the exact page number. As presented in Table 3.13, the page numbers can also be extracted from journal information.

3.4.9 Digital Object Identifier (DOI) Information

The extraction of the Digital Object Identifier (DOI) of the articles is a challenge as recently identified by Kun Ma [106], who extracted DOI metadata from research articles. We applied regular expression on the content of the header and footer logical layout section. The publishers use keywords with DOI information like "doi, or Digital Object Identifier" and we further applied the regular expression `b(10[.] [0-9]{4,}(?:[.] [0-9]+)*/(?:(!["&\'<>])\S)+)\b` to extract the doi number.

3.4.10 Headings and Levels

The table of contents of the research articles is logically categorized as sections, subsections, and subsubsections. Each section has a heading and the levels identify the hierarchy of the heading. In this approach, we have identified the headings and their hierarchies down to three levels. The extraction of this metadata is already defined in the sections 3.3.5 and 4.1.4.3.

3.4.11 Abstract and Keywords Related Metadata Extraction

The abstract and keywords are present on the first page of the research articles after the title and authors section. They start with the keywords "abstract" and "keyword". After the evaluation of the logical layout structure, these metadata elements are present in the paragraph or body LLS of the research articles. However, the font properties of this content are different from the main body or paragraph of the sections. We identified the keywords by using a word separators parser consisting of "commas, hyphens, tab spaces, and line numbers", etc. We applied $(\backslash d+)(,|;|:|[\backslash t] |[\backslash n] \backslash s* \backslash d+)*$ to identify these text patterns that are associated to keywords terms.

3.4.12 Table Caption Extraction

In sections, 3.3.3 to identify the logical layout structure of table sections and 4.1.4.4 the metadata extraction approach is used in this section to extract the table caption from diversified research articles. $(\text{Tab}[a-zA-Z]*) (\backslash s (\backslash d (:|,| \backslash s)*)+ | \backslash d \backslash s | \text{and} \backslash s \backslash d | \backslash s \backslash d - \backslash d | \backslash s \backslash d)*$. Further, we applied the regular expression on the plain text to identify the text pattern of the figures to extract the sequence numbers and the caption body.

3.4.13 Figure Caption Extraction

The method to extract the figure captions has been mentioned in the sections 3.3.4 and 4.1.4.5. The stage to extract the logical layout structure identifies the figure section, which is further evaluated using a regular expression mentioned to identify the sequence numbers and the captions of the figures. $(\text{Tab}[a-zA-Z]^*|\text{Viz}[a-zA-Z]^*)(\backslash s(\backslash d(:|,|\backslash s)^*+|\backslash d\backslash s|\text{and}\backslash s\backslash d|\backslash s\backslash d-\backslash d|\backslash s\backslash d)^*$. This regular expression enables the system to correctly identify metadata of figures published in different composition styles.

3.4.14 References Section Extraction

We evaluate the body of the reference section which has the heading as 'references' or 'bibliography'. In this approach, we have summed the number of references used by the research articles. Each reference is formulated in the form of a text block. We evaluated these text blocks to find the references. For this purpose, we used some of the features described in section 3.2 that includes Text indentation, line numbers, initial numeric value, the first letter is a special character '[', then a numeric value and ends with another character ']'. The approach parses each text line of the reference section and applies the above-mentioned features on it to identify the reference text block as shown in Figure 3.31. The system assigns the reference sequence number when a new reference text block is located.

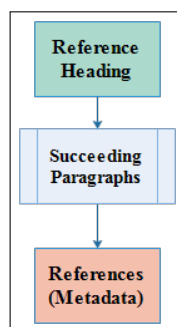


FIGURE 3.31: The sequence of references under the title of reference section.

In this section, we have discussed the strategy which has been adopted to extract different metadata elements from logical layout structural components of

research articles from the diversified publisher. We applied different techniques like heuristics, template-matching, regular expression, and natural language processing (NLP) to extract the metadata.

3.5 Summary

In this chapter, different machine learning algorithms are theoretically and experimentally evaluated to extract the logical layout structure of the research articles. The machine learning algorithms used features extracted in the previous stage associated with text blocks and lines to identify their structural relevance. By evaluating the efficiency and accuracy of different machine learning algorithms. We have selected the support vector classification algorithm as a final model to extract the logical layout structures for articles published by diversified publishers. Finally the logical layout structure are further evaluated on different gold standard datasets to extract the desired metadata and also compare it with state-of-the-art approaches.

Chapter 4

Results and Evaluation

Note: The parts of this chapter have been published in the Journal.

*”The answer of the **RQ3** is presented in this section”.*

RQ3: How to develop a comprehensive model to effectively extract metadata that is trained on research articles from multiple domains of different publishers?

4.1 Results on CEUR Dataset

We have divided the problem for metadata extraction from the research articles into sub-tasks. Initially, we extracted the generic features, and then applied them to the machine learning models in order to extract the logical layout structure (LLS). The logical layout structural components are the Title section, Author section, body and paragraphs, Headings, Tables, figures, header and footers, and acknowledgment section that we have explained how to extract in the previous chapter. In this chapter, we shall try to extract different metadata components that reside in the LLS.

This section is devoted to extracting the metadata from scholarly research articles. We have selected the CEUR gold standard dataset, which was published at the

ESWC conference in the form of a challenge to extract the metadata.

4.1.1 CEUR Challenge

European Semantic Web Conference (ESWC) is an A-rated conference that conducts Semantic Publishing (Sempub2016) challenge, to extract information regarding research articles that were published in CEUR-WS.org. The main objective of this challenge was to extract metadata and make it available as linked open data (LOD). The metadata should represent the structure and content of the research article and it should produce a better comprehension of the context of the research work.

The challenge requirement is to extract metadata from PDF-based research articles with the help of Natural Language Processing (NLP) and text pattern Recognition. In precise, metadata that is required to be extracted required to be answered in the form of the queries related to authors and affiliations, sections, table caption, figure caption, supplementary material, and the fundings sources.

The exact queries are as follows:

- **Q2.1 (Affiliations in a paper):** Identify the affiliations of the authors of the paper X.
- **Q2.2 (Countries in affiliations):** Identify the countries of the affiliations of the authors in paper X.
- **Q2.3 (Supplementary material):** Identify the supplementary material(s) for the paper X.
- **Q2.4 (Sections):** Identify the titles of the first-level sections of the paper X.

- **Q2.5 (Tables):** Identify the captions of the tables in the paper X.
- **Q2.6 (Figures):** Identify the captions of the figures in paper X.
- **Q2.7 (Funding agencies):** Identify the funding agencies that funded the research presented in paper X (or part of it).
- **Q2.8 (EU projects):** Identify the EU project(s) that supported the research presented in paper X (or part of it).

The participants of the challenge were from renowned research groups associated with the field of text mining and document structure analysis. The winner of the challenge was Ahmad et al. [45], a former Ph.D. student in our research group from The Capital University of Science and Technology, Islamabad Pakistan. Table 4.1 shows the results of the SemPub2016, published by the organizers of the conference.

TABLE 4.1: The results of approaches to extract metadata in SemPub2016.

Author Name	Recall	Precision	F-Measure
Ahmad et al	0.778	0.775	0.771
Sateli et al.	0.629	0.64	0.632
Klamp et al.	0.606	0.593	0.592
Nuzzolese et al.	0.416	0.416	0.416
Ramesh et al.	0.428	0.393	0.389

4.1.2 Proposed Methodology

We covered hypothetical, theoretical, and experimental aspects in our research methodology. Hypothetically, the content of the research documents is presented in different layouts and formatting styles which makes it easier for humans to comprehend different parts and sections of a document. Most of the documents share common formatting styles which makes them easily readable.

Theoretically, we have analyzed different formatting styles of publishers and established that these layout and formatting styles can be used to extract metadata

from research articles. Since this important information and layout components require annotations, therefore, we have categorized the formatting styles into two types of structural components, one is the physical layout and the other is logical layout structure components. The physical layout is based on individual article's distinct features, which consist of textual properties, geometric boundaries, paragraphs, column styles, floating objects, headers, and footers, etc. The logical layout structures (LLS) are generic formatting features to identify different parts, contents, and sections of an article that are required by the publisher. The system's proposed methodology flow diagram is shown in Figure 4.1.

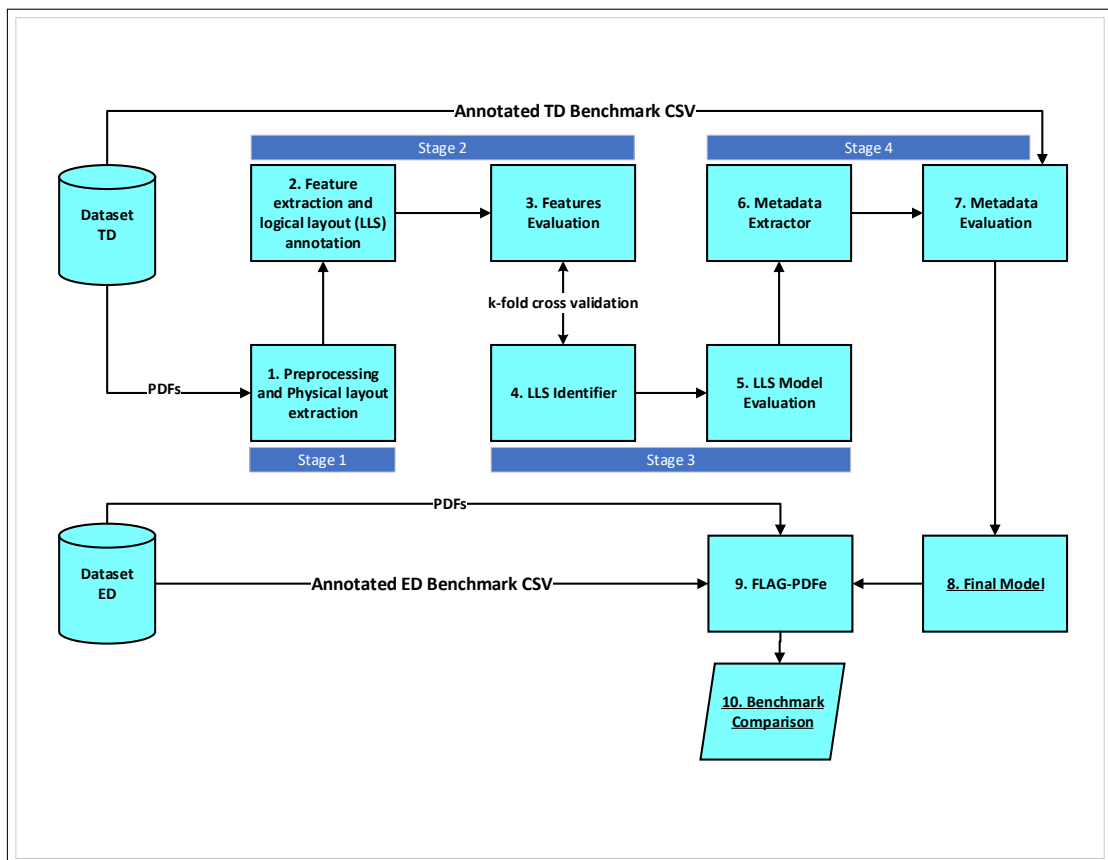


FIGURE 4.1: PDF base research article's metadata extraction proposed methodology flow diagram

FLAG-PDFe takes the research article as input in PDF format. The first stage extracts the physical layout of a PDF file and text chunks along with geometrical location and font property. These text chunks are processed and organized in the form of text blocks, with correct reading order and document formatting style

awareness. In the second stage, text blocks geometric and textual properties are used to create feature sets, which are used for classification algorithms to extract logical layout structure (LLS) components of the research articles in the third stage. Finally, heuristics are applied on LLS to get desired metadata, which is sorted and stored in CSV output form. In the preceding sections of the paper, every process is explained in chronological order, and in the next section, the formulation and extraction of the textural information are discussed.

4.1.3 Experimental Setup

The training of the models is performed by using the k-fold cross-validation technique, where $k = 10$ produced an optimum result. In order to improve the performance and efficiency of the model, we have performed feature ranking by first converting categorical values to numeric values while excluding non-convertible values, and used chi-squared (χ^2) to select K best features. We trained and tested the selected models described in the part of theoretical evaluation. The Euclidean distance method performed better to find k -NN, where $k = 5$ produced optimum results. We further evaluated ensemble classifiers bagging, Adaboost, and Stacking with the input of classification models used in current experiments.

We have followed the guidelines of [107] for the construction of the SVM model. We set different kernel functions like linear, polynomial, Gaussian-RBF, and sigmoid. In order to avoid the issue of overfitting, we choose the C value of 1 and γ value equal to 10, and by selecting Gaussian-RBF as kernel function, produced the optimal results among all the classification algorithms that we evaluated for our approach.

4.1.4 Metadata Extraction

This is the final stage to identify metadata and structural information of the research document. This desired metadata is extracted from different logical layout structural (LLS) components, as we have defined the LLS in section 3.3. This section applies heuristics on the content of LLS to extract metadata and stores them in machine comprehensible form in order to perform task specialized queries.

4.1.4.1 Author and Author Affiliation

In the previous section, we have used the machine learning approach to identify different elements of the author's section. It has been observed that this information is available in three style formats.

1. "Sequence of author names separated by commas' or tab spacing, then the sequence of affiliations".
2. "Sequence of author names with numeric or symbols separated by commas' or tab spacing. Then sequence of related affiliations with numeric or symbol".
3. "Group of an author's name, author's affiliation, and email address".

The itext library provided an edge here, as the output text rendering is in the sequence of the above-mentioned format styles. We applied a parser based on regular expression in order to separate authors and assigned them reference IDs. This id is based on the sequence of rendering, numeric, and symbols. Thereafter, the affiliations are assigned with authors id' based on the sequence of rendering, numeric, and symbols. The process generated a bipartite graph of authors and affiliations.

4.1.4.2 Country of Affiliation

A knowledge-based library is employed having country names, city names, and country domain names like de, uk, etc. After retrieval of the author's affiliation, the country name and city names are extracted based on a comparison made with the knowledge-based library. If that affiliation has missing country information, then we parse the email id domain name and compared it with the country domain name to extract the country of affiliation. Finally, a distant list of countries is stored.

4.1.4.3 Section Heading

Heading levels identification is a challenging task, different complex models proposed in the literature, efficiently identify the table of contents. To extract the table of content of a book, heuristics based on TOC identification methods using information present in the TOC section are employed. However, such approaches are not suitable for research articles. In the previous section, the level 1 headings were annotated along with level 2 and level 3 heading, and the output was based on the classification model. However, the ESWC challenge task is only to identify level1 heading, therefore no further processing is done on the output of the previous stage, and extracted heading is stored in ascending order.

4.1.4.4 Table Caption

In the previous section, the classification model identifies the start point of the table caption before the sequence number. At this stage, the remaining text chunk is analyzed. The process starts from the sequence number of the table and breaks when the next text line has different line spacing, by which multiple lines and different text properties do not break the complete caption sentence. The system further stores the caption of the table in an ascending order based on the sequence number.

4.1.4.5 Figure Caption

A similar approach has been adopted like the extraction of table caption for the extraction of figure captions. We used the keyword 'FIG' and 'FIGURE' to extract this metadata. The figures are sorted on the basis of sequence numbers that are present after the keyword.

4.1.4.6 Supplementary Material

The identification of supplementary material is part of the ESWC challenge and this information is present in the footnotes. The textual properties are different from main text body properties and start with a numeric or footnotes symbol identifier. The supplementary material is in the form of URLs. We have converted all text of footnotes in a single text block and then utilized a URL parser ¹ using a regular expression which extracts the complete URL from the descriptive part.

4.1.4.7 Funding Agency

The acknowledgments section contains the funding agencies and funded project information. We have used a task-specific knowledge-based approach to identify funding agency names and funded project names. The training dataset TD is analyzed and a regular expression is developed to extract funding agency by locating keywords starting with 'contri', 'support', 'fund', 'grant' and ends with 'from' or 'by' and the expression ends with "brackets", "quotation marks" or "punctuation marks". The Parser recognizes the funding agency name along with its acronyms and finally removes the preposition and punctuation around the funding agency information.

¹<https://docs.python.org/2/library/re.html>

4.1.4.8 Funded Projects

The final metadata extracted by our system is the list of funded projects. After manual analysis of the content, we observed that this information is also available in the acknowledgment section and placed after the funding agency name if available. The funding project name is placed between or after the keywords “the” and “project” like “by the EU FP7-ICT-2011-8 project”. The regular expression finally removes the keywords and parses them around the content.

4.1.5 Results

To evaluate the results, standard evaluation measures like recall, precision, and f-measure are mostly employed. These methods are based on classification parameters known as *true positive* (TP), *false positive* (FP), *true negative* (TN), or *false negative* (FN). Recall (sensitivity) is a statistical measure used to judge the relevant results produced by the model. Precision analyzes the quality of results. F-measure is the harmonic mean to measure test quality based on Recall and precision.

$$Recall = \frac{TP_i}{TP_i + FN_i} \quad , \quad (4.1)$$

$$Precision = \frac{TP_i}{TP_i + FP_i} \quad , \quad (4.2)$$

$$F - Measure = \frac{(2 * \rho_i * \pi_i)}{(\pi_i + \rho_i)} \quad (4.3)$$

4.1.5.1 Logical Layout Extraction

Table 4.2 illustrates the comparison of average recall, precision, and f-measure of all the classification models using TD. The results show the support vector Machine (SVM) Kernel trick classified correctly more relevant structural components. The Gaussian RBF SVM performed better than other machine learning models as the feature values are Non-linear and there is a non-linear relationship between the features.

TABLE 4.2: The performance matrix of models to extract each LLS component on training dataset

	Naïve	SVM	KNN	Decision	Extra	Bagging	Adaboost	Stacking	Random	Rule
	Bayes	RBF Kernel		Tree	Tree				Forest	(stratified)
Author Section	0.679	0.980	0.794	0.905	0.905	0.915	0.349	0.215	0.930	0.145
Acknowledge	0.000	1.000	1.000	1.000	0.176	1.000	0.000	0.000	1.000	0.120
Body and paragraphs	0.781	0.884	0.795	0.885	0.944	0.925	0.639	0.600	0.935	0.159
Header and footer	0.524	0.897	0.650	0.748	0.768	0.835	0.000	0.036	0.814	0.069
Figure	0.128	0.955	0.696	0.801	0.797	0.829	0.518	0.741	0.844	0.069
Table	0.156	0.934	0.294	0.578	0.533	0.643	0.000	0.000	0.552	0.030
Headings	0.449	0.945	0.883	0.945	0.949	0.960	0.689	0.820	0.954	0.475
Avg F-Measure	0.388	0.942	0.730	0.837	0.724	0.872	0.314	0.345	0.861	0.152
Avg Recall	0.429	0.921	0.723	0.839	0.707	0.874	0.353	0.351	0.844	0.153
Avg Precision	0.517	0.966	0.747	0.840	0.759	0.871	0.316	0.387	0.887	0.153

The Naïve Bayes algorithm uses probability to estimate the class label, the features are lesser than the training data instances $n \gg p$ and it is a multi-class label problem. The decision tree performed better than the remaining algorithms as the non-linear relationships between parameters do not affect the model performance. The output of this stage will be used to evaluate content present in related sections and final metadata will be generated. It also reveals that our generic feature set extraction approach has played a pivotal role to correctly identify the logical layout structure "on the fly".

4.1.5.2 Metadata Information Extraction

In this section, results of extracted metadata in the document are presented. We have evaluated authors and author's affiliation, country of affiliation, sections (heading level 1), Table and Figure Captions, supplementary material, funding agency and funded project. The recall, precision and f-measure are measure of each element and the mean value of these measuring methods are calculated against each metadata element. The final model results presented in Table 4.3 reveals that average recall = 0.877, precision = 0.928 and F-Measure = 0.897.

TABLE 4.3: The final results and the Confusion matrix of extracted metadata by FLAG-PDFe using evaluation dataset

	Author	Affiliation	Country	Supp-Material	Sections	Table	Figure	Funding	Projects
Authors	107	8	1	0	0	0	0	0	0
Affiliation	9	44	2	0	0	0	0	0	0
Country	0	2	41	0	0	0	0	0	0
Supp- material	0	0	0	12	0	0	0	0	0
Sections	0	0	0	0	269	0	0	0	0
Table	0	0	0	0	0	33	0	0	0
Figure	0	0	0	0	0	0	103	0	0
Funding	0	0	0	0	0	0	0	20	1
Projects	0	0	0	0	0	0	0	2	5
Actual	118	50	46	14	275	37	110	24	7
Recall	0.907	0.880	0.891	0.857	0.978	0.892	0.936	0.833	0.714
Precision	0.930	0.800	0.953	1.000	1.000	1.000	1.000	0.952	0.714
F-Measure	0.918	0.838	0.921	0.923	0.989	0.943	0.967	0.889	0.714

Avg Recall	Avg Precision	Avg F-Measure
0.877	0.928	0.897

4.1.6 SemPub Challenge Comparison

In this section, we have compared our results with start-of-the-art on gold standard [108]. The previous approaches and FLAG-PDFe used the same dataset and are evaluated for the same evaluation parameters. In Figure 4.2, our approach is compared with state-of-the-art, and results suggested that our approach showed significant improvement from previous approaches, and the results indicate that FLAG-PDFe has 16% performance gain on the SemPub2016 winner.

TABLE 4.4: The Performance Matrix of FLAG-PDFe on the TD of SemPub2017 challenge.

Metadata	Recall	Precision	F-Measure
Author	0.932	0.932	0.932
Affiliation	0.761	0.823	0.791
Country	0.837	0.953	0.891
Supp. Material	0.715	1.000	0.833
Sections	0.953	1.000	0.976
Table caption	0.923	0.923	0.923
Figure Caption	0.899	0.961	0.929
EU Projects	0.643	1.000	0.783
Average	0.833	0.949	0.860

Additionally, we evaluated the performance of our proposed framework on the TD consisting of 40 research papers from the SemPub2017 challenge. On the basis of our TD dataset from SemPub2016, we evaluated results on 7 parameters that our

technique extracts. The results presented in Table 4.4 reveals consistent performance of model that average recall = 0.833, precision = 0.949 and F-Measure = 0.860.

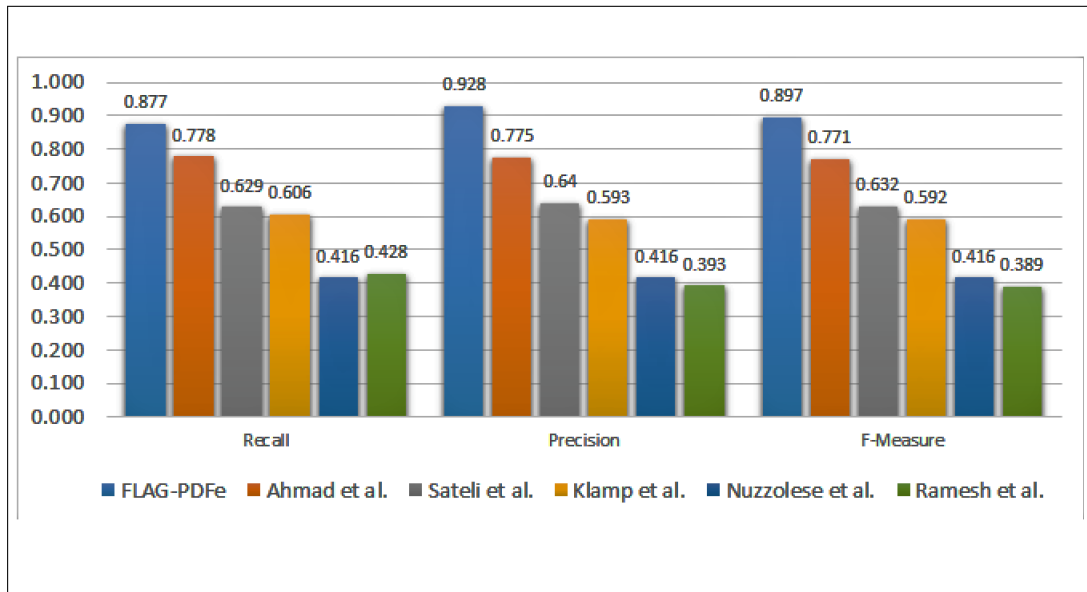


FIGURE 4.2: The final result comparison of FLAG-PDFe with the SumPub2016 challenge participants

4.1.7 Summary

In the previous chapter, we extracted the logical layout structure using the support vector Machine (SVM), a machine learning algorithm on the basis of the extracted feature set. Now in this section, we have extracted the metadata, and we have selected the dataset of research articles of CEURWS.org conference that is the world’s renowned metadata extraction challenge ESWC. We evaluated our approach with the results of the best approaches that were published by the organizers of the challenge. The winner of the challenge is the former Ph.D. student of our research group. We have proposed our novel approach based on the learning and training which I had acquired from the Ph.D. student of our research group. We trained and tested our approach on the SemPub2016 challenge and improved

the results by 16% from the winner of the challenge. We also evaluated our approach on the training dataset of SemPub2017.

We have used three datasets of articles published at CEURWS.org that are training data (TD), evaluation data (ED) from SemPub2016, and training data (TD) of SemPub2017. However, we require a more comprehensive dataset to check the comprehensiveness and diversification to prove the efficiency and completeness of our approach. Therefore, we have used benchmark datasets like sectLabel and curated dataset, which I created during my Ph.D. research duration which consists of articles from diversified publishers. In the next section, we shall evaluate our approach on these datasets and will further compare our approach's results with the results of the state-of-the-art.

4.2 Results on Curated Dataset

In previous section, we have used the CEUR dataset ² to evaluate our methodology to extract metadata from research articles. The European Semantic Web Conference (ESWC) organizes SemPub challenge ³, which had a dataset of research articles from CEURWS.org. These articles were selected from diversified publishers, however, the dataset was of small size, and few elements of metadata were extracted. For a comprehensive evaluation of our proposed methodology, we require a dataset that has research articles from diversified publishers. In this regard, we have performed experiments on two different datasets to further evaluate our methodology. We have also selected a second widely used benchmark dataset provided by the sectLabel approach and compare our approach with renowned

²<https://github.com/angelobo/SemPubEvaluator>

³https://github.com/ceurws/lod/wiki/SemPub17_Task2

commercially available tools and state-of-the-art approaches. We shall also evaluate additional metadata elements (as shown in the Table 3.12) and compare our approach with state-of-the-art approaches.

In the next sections, we shall discuss the datasets that we have selected and developed to evaluate the proposed approach for the extraction of metadata from research articles. The first dataset is a benchmark dataset that is proposed and developed by the sectLabel technique, and the state-of-the-art has used it to evaluate their proposed techniques. The second dataset of PDF-based research articles is a curated dataset selected from diversified publishers to evaluate the proposed approach and compare it with the state-of-the-art approaches like PDFX, CERMINE, and sectLabel.

In the next section, we shall evaluate our proposed methodology using gold standard datasets ds1 and ds2 and compare the approach with state-of-the-art approaches.

4.2.1 Results

This section presents the results of our proposed system and evaluates the performance on the gold-standard datasets with state-of-the-art. At first, we evaluated our system's performance with popular online available tools to extract metadata from research articles using the diversified curated dataset. The tools automatically annotate all the content of the research article and output them in different file formats. We used XML formate and applied xQuery to get our desired metadata generated by these tools for evaluation purposes. Secondly, we evaluated our system with a state-of-the-art system on the benchmark dataset.

4.2.2 Comparison with CERMINE and PDFX on GOLD-Standard

We evaluated the performance of our proposed system with popular online tools CERMINE and PDFX and used our curated diversified dataset. CERMINE and PDFX take a research article in pdf file format and annotate all the content of the research article. The output of these systems is in different file formats, we used XML format to evaluate the performance of these systems. We removed the undesired research article's content generated by these systems and extracted only relevant metadata by applying the XQuery on the XML files. The next subsections have the details of these XQueries used for each system.

4.2.2.1 XQuery for CERMINE Generated XML Files

We used the following XQueries to get the desired metadata from CERMINE system's XML file output.

Journal Name XQuery to find the name of the journal.

```
< journaltitle >
{data($articles/front/journal – meta/journal – title – group/journal – title)}
< /journaltitle >
```

Title XQuery to find the title of the article.

```
< title >
{data($articles/front/article – meta/title – group/article – title)}
< /title >
```

DOI XQuery to find the doi of the article.

```
< doi >
{data($articles/front/article – meta/article – id)}
< /doi >
```

Year XQuery to find the year of the article.

```
< pubyear >
{data($articles/front/article – meta/pub – date/year)}
< /pubyear >
```

Volume XQuery to find the volume of an article.

```
< volume >
{data($articles/front/article – meta/volume)}
< /volume >
```

Issue XQuery to find the issue number of the article.

```
< issue >
{data($articles/front/article – meta/issue)}
< /issue >
```

Pages XQuery to find the pages of the article.

```
< firstpage >
{data($articles/front/article – meta/fpage)}
< /firstpage >
```

```
< lastpage >
{data($articles/front/article – meta/lpage)}
< /lastpage >
```

Keywords XQuery to find the keywords of the article.

```
< keywords > {for$keywordat$jin
$articles/front/article – meta/kwd – group/kwd
return < keyword > {data($keyword)} < /keyword >}
< /keywords >
```

Authors XQuery to find the authors of the article.

```

< authors > {for$authorsat$jin
$articles/front/article – meta/contrib – group/contrib/string – name
return < author > {data($authors)} < /author >}
< /authors >

```

Affiliations XQuery to find the author affiliations of the article.

```

< affiliations >
{
for$affiliationat$jin
$articles/front/article – meta/contrib – group/aff
return < institution >
{data($affiliation/institution)} < /institution >
}
{
for$affiliationat$jin
$articles/front/article – meta/contrib – group/aff
return < country >
{data($affiliation/country)} < /country >
}
< /affiliations >

```

H1 XQuery to find the Heading level 1 of the article.

```

< section1 >
{
for$h1at$jin$articles/body/sec
return < h1 > {data($h1/title)} < /h1 >
}
< /section1 >

```

H2 XQuery to find the Heading level 2 of the article.

```

< section2 >
{

```

```

for$h2at$jin$articles/body/sec/sec
return < h2 > {data($h2/title)} < /h2 >
}
< /section2 >

```

H3 XQuery to find the Heading level 1 of the article.

```

< section3 >
{ for$h3at$jin$articles/body/sec/sec/sec
return < h3 > {data($h3/title)} < /h3 >
}
< /section3 >

```

References XQuery to find the number of references in the article.

```

< refcnt >
{
for$refat$jin$articles/back/ref-list
return < ref > {count($ref/ref)} < /ref >
}
< /refcnt >

```

Abstract XQuery to find the abstract of the article.

```

< abstract >
{count($articles/front/article-meta/abstract)}
< /abstract >

```

4.2.2.2 XQuery for PDFX Generated XML Files

We used the following XQueries to get the desired metadata from PDFX system's XML file output.

Title XQuery to find the title of the article.

```
< title >
{data($articles/article/front/title – group/article – title)}
< /title >
```

DOI XQuery to find the doi of the article.

```
< doi > {data($articles/meta/doi)} < /doi >
```

Abstract XQuery to find the abstract of the article.

```
< abstract >
{count($articles/article/front/abstract)}
< /abstract >
```

Authors XQuery to find the authors of the article.

```
< authors > {for$authorsat$jin
$articles/article/front/region/email
return < email > {data($authors)} < /email >}
< /authors >
```

H1 XQuery to find the Heading level 1 of the article.

```
< section1 >
{
for$h1at$jin$articles/article/body/./h1
return < h1 > {data($h1)} < /h1 >
}
< /section1 >
```

H2 XQuery to find the Heading level 2 of the article.

```
< section2 >
{
for$h2at$jin$articles/article/body/./h2
return < h2 > {data($h2)} < /h2 >
}
```

```
}
< /section2 >
```

H3 XQuery to find the Heading level 3 of the article.

```
< section3 >
{
for$h3at$jin$articles/article/body/./h3
return < h3 > {data($h3)} < /h3 >
}
< /section3 >
```

References XQuery to find the number of references in the article.

```
< ref >
{count($articles/article/body/section/./ref – list/ref)}
< /ref >
```

Table Caption XQuery to find the table captions in the article. < tables > {

```
for$captionsat$jin
$articles/./region[contains(@class,' DoCO : TableBox')]
return < table > {data($captions/caption)} < /table > } < /tables >
```

Figure Caption XQuery to find the Figure captions in the article.

```
< figures > {
for$captionsat$jin
$articles/./region[contains(@class,' DoCO : FigureBox')]
return < fig > {data($captions/caption)} < /fig >
} < /figures >
```

TABLE 4.5: The detail evaluation matrix of Flag-PDFe system to extract different metadata component from diversified publishers.
Where R denotes Recall, P denotes precision, and F denotes F-measure

Metadata	ACM			Elsevier			IEEE			MDPI			Springer			Overall		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Journal	0.98	1.00	0.99	0.96	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.84	1.00	0.91	0.96	1.00	0.98
Title	0.98	1.00	0.99	0.96	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.99
volume	0.94	0.90	0.92	0.94	0.96	0.95	1.00	1.00	1.00	1.00	1.00	1.00	0.94	1.00	0.97	0.96	0.97	0.97
Issue	0.94	1.00	0.97	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.16	0.16	0.16	0.82	0.83	0.83
Pubyear	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
pages	0.82	0.98	0.89	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.77	0.61	0.76	0.86	0.81	0.82	0.92	0.86
ref	0.97	0.99	0.98	0.82	0.98	0.89	0.93	1.00	0.96	0.97	0.99	0.98	0.9	0.99	0.94	0.92	0.99	0.95
keywords	0.51	0.62	0.56	0.86	0.96	0.9	0.83	0.94	0.88	0.94	0.93	0.93	0.84	0.96	0.9	0.8	0.88	0.84
h1	0.82	0.86	0.84	0.9	0.78	0.83	0.9	0.9	0.9	0.9	0.73	0.81	0.83	0.67	0.74	0.87	0.79	0.82
h2	0.71	0.79	0.75	0.8	0.79	0.79	0.77	0.81	0.79	0.8	0.84	0.82	0.25	0.33	0.29	0.67	0.71	0.69
h3	0.39	0.39	0.39	0.86	1.00	0.93	0.41	0.47	0.44	0.56	1.00	0.71	0.53	0.62	0.57	0.55	0.7	0.61
doi	0.84	1.00	0.91	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	1.00	0.94	0.94	1.00	0.97
author	0.86	0.87	0.86	0.89	0.95	0.92	0.97	0.98	0.98	0.93	0.99	0.96	0.94	1.00	0.97	0.92	0.96	0.94
affiliation	0.78	0.86	0.82	0.84	0.92	0.88	0.86	0.88	0.87	0.9	0.92	0.91	0.86	0.86	0.86	0.85	0.89	0.87
country	0.49	1.00	0.65	0.79	1.00	0.88	0.95	1.00	0.98	0.95	1.00	0.97	0.87	1.00	0.93	0.81	1.00	0.88
abstract	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
emails	0.86	1.00	0.93	0.93	1.00	0.96	0.82	1.00	0.9	0.87	1.00	0.93	0.92	1.00	0.96	0.88	1.00	0.94
tables	0.84	0.95	0.89	0.8	0.88	0.84	0.48	0.62	0.54	0.82	0.88	0.85	0.69	0.73	0.71	0.73	0.81	0.77
figures	0.82	0.99	0.90	0.91	0.98	0.95	0.78	0.97	0.86	0.69	0.95	0.8	0.73	0.93	0.82	0.79	0.96	0.87
accepted on	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
published on	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
corresponding	0.72	0.94	0.82	0.72	1.00	0.84	0.87	1.00	0.93	0.80	1.00	0.89	0.68	0.97	0.8	0.76	0.98	0.86
Total	0.83	0.92	0.87	0.91	0.96	0.93	0.89	0.93	0.91	0.89	0.96	0.92	0.80	0.87	0.83	0.86	0.93	0.89

4.2.2.3 Evaluation

To evaluate our proposed methodology and state-of-the-art systems, we used the recall, precision, and f-measure evaluation measures. We calculated these parameters using the below-mentioned equations. To measure the recall, the correctly classified number of elements is divided by the actual number of elements. Similarly, the precision is calculated by the correctly classified number of elements divided by the total number of elements classified by the system. The F-measure is the harmonic mean of recall and precision.

Initially, we evaluated the results of our proposed approach using these evaluation parameters. Table 4.5 shows the detailed output of our system to extract different metadata elements for journals of diversified publishers.

The results illustrate that our approach Flag-PDFe has constantly performed on different journals from diversified publishers. The approach accurately extracted the generic set of features from research articles with diversified layouts and publication styles. The final selected classification model SVM support vector machine retrieved the logical layout structure of research articles with different layout styles. Finally, the approach adopted to extract diverse metadata elements with diversified formatting styles worked effectively. The system average recall is 0.86, precision is 0.93, and f-measure is 0.89. Next, we shall compare these results with the results produced by the state-of-the-art approaches to extract the metadata element.

FLAG-PDFe Comparison with Cermine The CERMINE approach extracts different metadata elements like the name of the journal, the title of the article, author names, author affiliation, the country of affiliation, abstract, volume, and issue of the journal, the year of publication, DOI, pages, keywords, headings, and references. We performed a comparison of FLAG-PDFe with the CERMINE on these metadata elements. We have shown a detailed comparison of metadata

extraction results in this section. First, we have presented the journal-wise individual metadata elements extraction comparison of both approaches. Then we presented the summary of results publisher-wise. Finally, we have shown the overall performance of both approaches in the form of graphical representation.

TABLE 4.6: The comparison of CERMINE and FLAG-PDFe to extract metadata from articles published by the AMC.

Metadata	Cermine			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Journal	0.98	1.00	0.99	0.98	1.00	0.99
Title	0.98	0.98	0.98	0.98	1.00	0.99
volume	1.00	1.00	1.00	0.94	0.90	0.92
Issue	1.00	1.00	1.00	0.94	1.00	0.97
Pubyear	1.00	1.00	1.00	1.00	1.00	1.00
pages	0.06	0.06	0.06	0.82	0.98	0.89
ref	0.96	0.99	0.98	0.97	0.99	0.98
keywords	0.73	0.73	0.73	0.51	0.62	0.56
h1	0.67	0.32	0.43	0.82	0.86	0.84
h2	0.00	0.00	0.00	0.71	0.79	0.75
h3	0.01	0.01	0.01	0.39	0.39	0.39
doi	0.50	1.00	0.67	0.84	1.00	0.91
author	0.51	0.54	0.53	0.86	0.87	0.86
affiliation	0.10	0.67	0.18	0.78	0.86	0.82
country	0.17	1.00	0.29	0.49	1.00	0.65
abstract	1.00	1.00	1.00	1.00	1.00	1.00

Table 4.6 shows the comparison of FLAG-PDFe and CERMINE using the research article of Association for Computing Machinery (ACM). Our approach has outperformed CERMINE in many scenarios. Especially while extracting the headings and the subheadings. Our approach also produced better results to extract authors and affiliation metadata.

Table 4.7 shows the comparison of FLAG-PDFe and CERMINE using the dataset of research articles for Elsevier publisher. Our approach has again outperformed CERMINE in many cases. Especially while extracting the Journal publication-related information, headings, and the subheading.

TABLE 4.7: The comparison of CERMINE and FLAG-PDFe to extract meta-data from articles published in Elsevier

Metadata	Cermine			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Journal	0.94	1.00	0.97	0.96	1.00	0.98
Title	0.94	1.00	0.97	0.96	1.00	0.98
volume	1.00	1.00	1.00	0.94	0.96	0.95
Issue	1.00	0.29	0.44	1.00	1.00	1.00
Pubyear	1.00	1.00	1.00	1.00	1.00	1.00
pages	1.00	1.00	1.00	1.00	1.00	1.00
ref	0.90	1.00	0.94	0.82	0.98	0.89
keywords	0.87	0.98	0.92	0.86	0.96	0.90
h1	0.91	0.70	0.79	0.90	0.78	0.83
h2	0.84	0.65	0.73	0.80	0.79	0.79
h3	0.02	0.03	0.03	0.86	1.00	0.93
doi	0.88	1.00	0.94	1.00	1.00	1.00
author	0.93	0.99	0.96	0.89	0.95	0.92
affiliation	0.99	0.84	0.91	0.84	0.92	0.88
country	1.00	1.00	1.00	0.79	1.00	0.88
abstract	0.98	1.00	0.99	1.00	1.00	1.00

Table 4.8 shows the comparison of FLAG-PDFe and CERMINE using the research article from journals of the Institute of Electrical and Electronics Engineers (IEEE). Our approach has outperformed CERMINE in all scenarios. Especially

while extracting the authors, affiliations, journal’s publication information, headings, and subheadings. Our approach also produced better results due to the efficient use of metadata extraction methodology.

TABLE 4.8: The comparison of CERMINE and FLAG-PDFe to extract metadata from articles published by in IEEE

Metadata	Cermine			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Journal	0.24	0.86	0.38	1.00	1.00	1.00
Title	1.00	1.00	1.00	1.00	1.00	1.00
volume	0.96	0.96	0.96	1.00	1.00	1.00
Issue	0.63	0.38	0.47	1.00	1.00	1.00
Pubyear	1.00	1.00	1.00	1.00	1.00	1.00
pages	0.98	0.98	0.98	1.00	1.00	1.00
ref	0.99	0.99	0.99	0.93	1.00	0.96
keywords	0.38	1.00	0.55	0.83	0.94	0.88
h1	0.58	0.53	0.55	0.90	0.90	0.90
h2	0.46	0.53	0.49	0.77	0.81	0.79
h3	0.20	0.35	0.26	0.41	0.47	0.44
doi	0.02	1.00	0.04	1.00	1.00	1.00
author	0.92	0.99	0.95	0.97	0.98	0.98
affiliation	0.82	0.93	0.87	0.86	0.88	0.87
country	0.86	1.00	0.92	0.95	1.00	0.98
abstract	0.92	1.00	0.96	1.00	1.00	1.00

Table 4.9 shows the comparison of FLAG-PDFe and CERMINE using the dataset

of research articles for Multidisciplinary Digital Publishing Institute (MDPI) publisher. Our approach has again outperformed CERMINE in all cases. Especially while extracting the Journal publication-related information, authors, and affiliations.

TABLE 4.9: The comparison of CERMINE and FLAG-PDFe to extract metadata from articles published by MDPI

Metadata	Cermine			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Journal	1.00	1.00	1.00	1.00	1.00	1.00
Title	1.00	1.00	1.00	1.00	1.00	1.00
volume	1.00	1.00	1.00	1.00	1.00	1.00
Issue	0.13	0.02	0.03	1.00	1.00	1.00
Pubyear	1.00	1.00	1.00	1.00	1.00	1.00
pages	0.74	0.50	0.60	0.50	0.77	0.61
ref	0.97	0.99	0.98	0.97	0.99	0.98
keywords	1.00	0.99	1.00	0.94	0.93	0.93
h1	0.76	0.80	0.78	0.90	0.73	0.81
h2	0.63	0.37	0.46	0.80	0.84	0.82
h3	0.41	0.05	0.08	0.56	1.00	0.71
doi	1.00	1.00	1.00	1.00	1.00	1.00
author	0.96	0.99	0.98	0.93	0.99	0.96
affiliation	0.29	1.00	0.45	0.90	0.92	0.91
country	0.41	1.00	0.58	0.95	1.00	0.97
abstract	1.00	1.00	1.00	1.00	1.00	1.00

Table 4.10 shows the comparison of FLAG-PDFe and CERMINE for the research article from journals of Springer Publishers. The FLAG-PDFe has outperformed CERMINE in most scenarios. While extracting the authors, affiliations, headings,

TABLE 4.10: The comparison of CERMINE and FLAG-PDFe to extract meta-data from articles published by the Springer.

Metadata	Cermine			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Journal	0.74	1.00	0.85	0.84	1.00	0.91
Title	1.00	1.00	1.00	1.00	1.00	1.00
volume	1.00	1.00	1.00	0.94	1.00	0.97
Issue	0.16	0.16	0.16	0.16	0.16	0.16
Pubyear	1.00	1.00	1.00	1.00	1.00	1.00
pages	0.56	0.56	0.56	0.76	0.86	0.81
ref	0.99	0.99	0.99	0.90	0.99	0.94
keywords	0.92	0.98	0.95	0.84	0.96	0.90
h1	0.97	0.77	0.86	0.83	0.67	0.74
h2	0.04	0.05	0.05	0.25	0.33	0.29
h3	0.02	0.03	0.02	0.53	0.62	0.57
doi	0.86	1.00	0.92	0.88	1.00	0.94
author	0.94	1.00	0.97	0.94	1.00	0.97
affiliation	0.99	0.85	0.91	0.86	0.86	0.86
country	0.81	1.00	0.90	0.87	1.00	0.93
abstract	0.96	1.00	0.98	1.00	1.00	1.00

and subheadings, our approach produced better results. Although the CERMINE was tested for the same publisher.

Table 4.11 shows the final comparison of FLAG-PDFe and CERMINE showing the

TABLE 4.11: The final metadata element wise comparison of CERMINE and FLAG-PDF_x

Metadata	Cermine			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Journal	0.78	0.97	0.84	0.96	1.00	0.98
Title	0.98	1.00	0.99	0.99	1.00	0.99
volume	0.99	0.99	0.99	0.96	0.97	0.97
Issue	0.58	0.37	0.42	0.82	0.83	0.83
Pubyear	1.00	1.00	1.00	1.00	1.00	1.00
pages	0.67	0.62	0.64	0.82	0.92	0.86
ref	0.96	0.99	0.98	0.92	0.99	0.95
keywords	0.79	0.94	0.84	0.80	0.88	0.84
h1	0.78	0.62	0.68	0.87	0.79	0.82
h2	0.39	0.32	0.35	0.67	0.71	0.69
h3	0.13	0.09	0.08	0.55	0.70	0.61
doi	0.65	1.00	0.71	0.94	1.00	0.97
author	0.85	0.90	0.88	0.92	0.96	0.94
affiliation	0.64	0.86	0.66	0.85	0.89	0.87
country	0.65	1.00	0.74	0.81	1.00	0.88
abstract	0.97	1.00	0.99	1.00	1.00	1.00

aggregated results of all the publishers that we have previously explained. The FLAG-PDFe approach has outperformed CERMINE in all cases for aggregated results. Figure 4.3 shows the overall average aggregate results of FLAG-PDFe and CERMINE, where FLAG-PDFe has produced the best results.

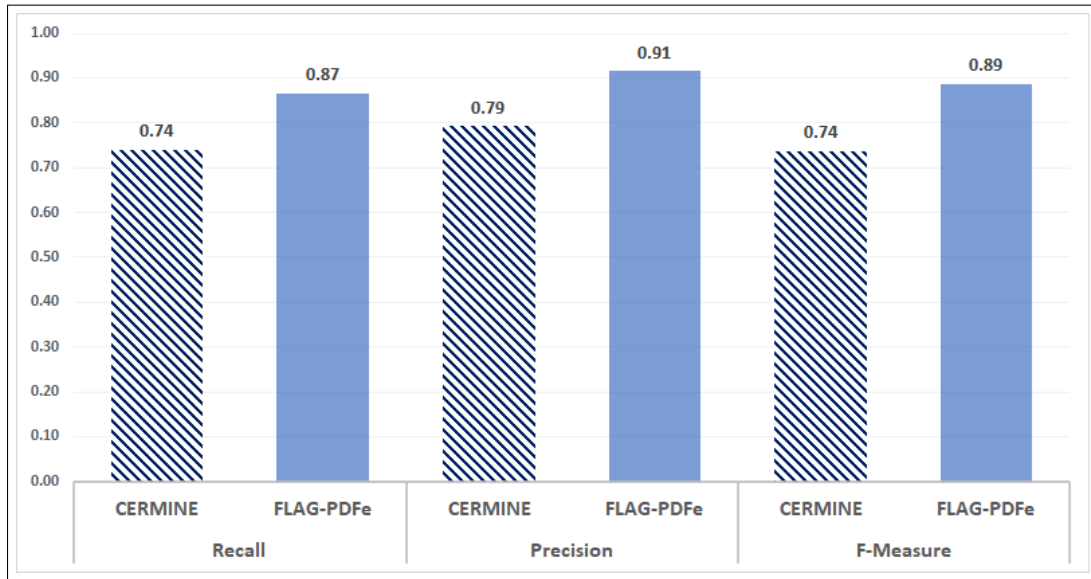


FIGURE 4.3: The final result comparison of FLAG-PDFe with the CERMINE

Table 4.12 shows the final comparison of FLAG-PDFe and CERMINE shows the publisher wise result summary.

TABLE 4.12: The publisher wise comparison of CERMINE and FLAG-PDFx

Publisher	Recall		Precision		F-Measure	
	CERMINE	FLAG-PDFe	CERMINE	FLAG-PDFe	CERMINE	FLAG-PDFe
ACM	0.61	0.81	0.71	0.89	0.62	0.85
Elsevier	0.89	0.91	0.84	0.96	0.85	0.93
IEEE	0.69	0.91	0.84	0.94	0.71	0.92
MDPI	0.77	0.9	0.79	0.95	0.75	0.92
Springer	0.75	0.79	0.77	0.84	0.76	0.81

FLAG-PDFe Comparison with PDFX The PDFX approach extracts different metadata elements like the title of the article, emails, tables, figures, headings, subheadings, sub subheadings, DOI, abstract, and references. We performed a comparison of FLAG-PDFe with the PDFX on these metadata elements. We have shown a detailed comparison of metadata extraction results in this section.

First, we have presented the journal-wise individual metadata elements extraction comparison of both approaches. Then we presented the summary of results publisher-wise. Finally, we have shown the overall performance of both approaches in the form of graphical representation.

TABLE 4.13: The comparison of PDFX and FLAG-PDFe to extract metadata from articles published in the AMC.

Metadata	PDFX			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Title	0.98	0.98	0.98	0.98	1.00	0.99
emails	0.23	1.00	0.38	0.86	1.00	0.93
tables	0.89	0.92	0.91	0.84	0.95	0.89
ref	1.00	0.99	0.99	0.97	0.99	0.98
figures	0.87	0.99	0.93	0.82	0.99	0.90
h1	0.86	0.59	0.70	0.82	0.86	0.84
h2	0.66	0.60	0.63	0.71	0.79	0.75
h3	0.02	1.00	0.04	0.39	0.39	0.39
doi	0.76	1.00	0.86	0.84	1.00	0.91
abstract	0.46	1.00	0.63	1.00	1.00	1.00

Table 4.13 shows the results of both approaches using a research article of Association for Computing Machinery (ACM) journal. Our approach outperformed PDFx in all cases, especially while extracting emails, heading level 1, and abstract.

Table 4.14 shows the comparison of results of our approach and PDFx for articles of Elsevier publisher. Our approach outperformed PDFx in all cases, especially while extracting emails, heading level 1, and heading level 3.

TABLE 4.14: The comparison of PDFX and FLAG-PDFe to extract metadata from articles published in Elsevier.

Metadata	PDFX			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Title	0.96	0.96	0.96	0.96	1.00	0.98
emails	0.11	1.00	0.20	0.93	1.00	0.96
tables	0.76	0.79	0.78	0.80	0.88	0.84
ref	0.71	0.91	0.80	0.82	0.98	0.89
figures	0.94	0.99	0.96	0.91	0.98	0.95
h1	0.89	0.50	0.64	0.90	0.78	0.83
h2	0.75	0.58	0.65	0.80	0.79	0.79
h3	0.64	1.00	0.78	0.86	1.00	0.93
doi	1.00	1.00	1.00	1.00	1.00	1.00
abstract	0.78	1.00	0.88	1.00	1.00	1.00

Table 4.15 represents the metadata extraction results by using the articles of the Institute of Electrical and Electronics Engineers (IEEE) publisher. The results shows that PDFx is unable to extract metadata from IEEE articles and our approach consistently showed better results. Table 4.16 shows the comparison of both the techniques using the Digital Publishing Institute (MDPI) publisher's research article. Our approach outperformed PDFx in all cases, especially while extracting emails, figure captions, and heading of all levels.

TABLE 4.15: The comparison of PDFX and FLAG-PDFe to extract metadata from articles published by the IEEE.

Metadata	PDFX			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Title	1.00	1.00	1.00	1.00	1.00	1.00
emails	0.54	1.00	0.70	0.82	1.00	0.90
tables	0.48	0.62	0.54	0.48	0.62	0.54
ref	0.99	0.91	0.95	0.93	1.00	0.96
figures	0.37	0.93	0.53	0.78	0.97	0.86
h1	0.73	0.46	0.57	0.90	0.90	0.90
h2	0.78	0.62	0.69	0.77	0.81	0.79
h3	0.01	1.00	0.01	0.41	0.47	0.44
doi	0.02	1.00	0.04	1.00	1.00	1.00
abstract	0.98	1.00	0.99	1.00	1.00	1.00

TABLE 4.16: The comparison of PDFX and FLAG-PDFe to extract metadata from articles published by the MDPI.

Metadata	PDFX			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Title	1.00	1.00	1.00	1.00	1.00	1.00
emails	0.37	1.00	0.54	0.87	1.00	0.93
tables	0.82	0.88	0.85	0.82	0.88	0.85
ref	0.84	0.99	0.91	0.97	0.99	0.98
figures	0.45	0.98	0.62	0.69	0.95	0.80
h1	0.82	0.37	0.51	0.90	0.73	0.81
h2	0.83	0.50	0.63	0.80	0.84	0.82
h3	0.19	1.00	0.31	0.56	1.00	0.71
doi	1.00	1.00	1.00	1.00	1.00	1.00
abstract	0.90	1.00	0.95	1.00	1.00	1.00

Similarly, Table 4.17 shows the results of metadata elements extraction from the articles of Springer Publisher. The results shows that PDFx extracted heading level 2 more accurately. However, the for other metadata element our approach outperformed PDFx, especially in case of emails, heading level 1, and heading level 3.

TABLE 4.17: The comparison of PDFX and FLAG-PDFe to extract metadata from articles published by the Springer.

Metadata	PDFX			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Title	1.00	1.00	1.00	1.00	1.00	1.00
emails	0.40	1.00	0.57	0.92	1.00	0.96
tables	0.69	0.73	0.71	0.69	0.73	0.71
ref	0.87	0.98	0.92	0.90	0.99	0.94
figures	0.70	0.96	0.81	0.73	0.93	0.82
h1	0.67	0.31	0.42	0.83	0.67	0.74
h2	0.82	0.47	0.59	0.25	0.33	0.29
h3	0.06	1.00	0.11	0.53	0.62	0.57
doi	0.86	1.00	0.92	0.88	1.00	0.94
abstract	0.96	1.00	0.98	1.00	1.00	1.00

Table 4.18 shows the overall comparison of results of FLAG-PDFe and PDFX for each metadata element. The results shows that our approach outperformed PDFx in all cases. For metadata elements including emails, heading level 1, heading level 3, doi, and abstract; our approach extracted metadata will better accuracy.

Table 4.19 shows the final comparison of FLAG-PDFe and PDFX showing the aggregated results of all the publishers that we have previously explained. The

TABLE 4.18: The overall metadata extraction comparison of PDFx and Flag-PDFe

Metadata	PDFX			Flag-PDFe		
	Recall	Precision	Fmeasure	Recall	Precision	Fmeasure
Title	0.99	0.99	0.99	0.99	1.00	0.99
emails	0.33	1.00	0.48	0.88	1.00	0.94
tables	0.73	0.79	0.76	0.73	0.81	0.77
ref	0.88	0.96	0.91	0.92	0.99	0.95
figures	0.67	0.97	0.77	0.79	0.96	0.87
h1	0.80	0.45	0.57	0.87	0.79	0.82
h2	0.77	0.55	0.64	0.67	0.71	0.69
h3	0.18	1.00	0.25	0.55	0.70	0.61
doi	0.73	1.00	0.77	0.94	1.00	0.97
abstract	0.82	1.00	0.88	1.00	1.00	1.00

FLAG-PDFe approach has outperformed PDFX in all cases for aggregated results.

TABLE 4.19: The publisher wise comparison of PDFX and FLAG-PDFe

Publisher	Recall		Precision		F-Measure	
	PDFX	FLAG-PDFe	PDFX	FLAG-PDFe	PDFX	FLAG-PDFe
ACM	0.67	0.83	0.91	0.92	0.7	0.87
Elsevier	0.75	0.91	0.87	0.96	0.76	0.93
IEEE	0.59	0.93	0.85	0.91	0.6	0.91
MDPI	0.72	0.89	0.87	0.96	0.73	0.92
SPRINGER	0.7	0.8	0.84	0.87	0.7	0.83

Figure 4.4 shows the overall average aggregate results of FLAG-PDFe and PDFX, where FLAG-PDFe has produced the best results.

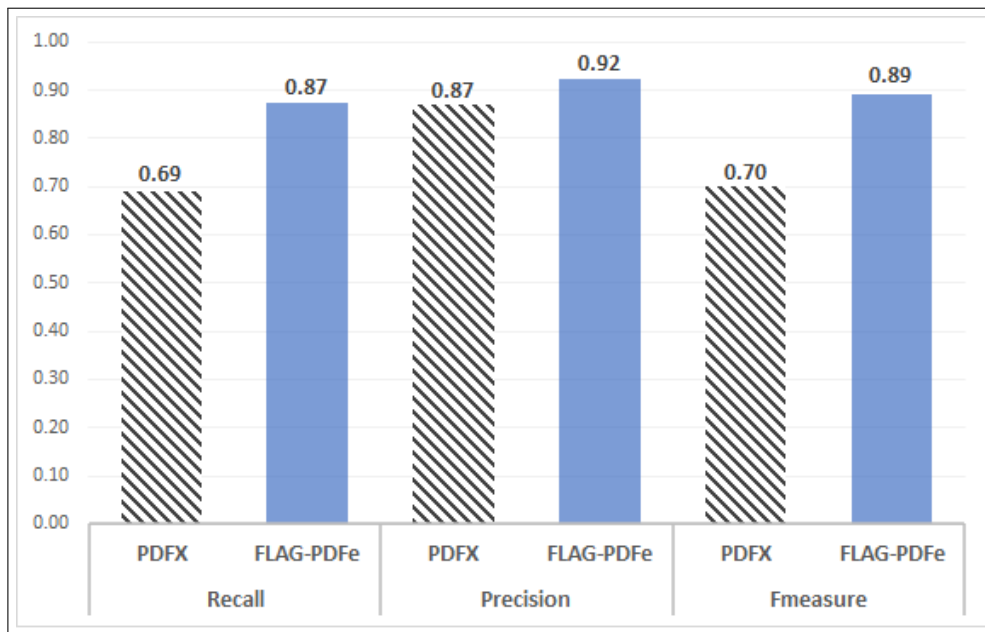


FIGURE 4.4: The final result comparison of FLAG-PDFe with the PDFX

4.2.3 FLAG-PDFe Comparison with sectLabel

In this section, results of extracted metadata in the document are presented. We have evaluated title, journals, acceptance and publication date, pages, DOI, year, references, authors, affiliations, country of affiliation, emails, corresponding emails, keywords, sections (heading level 1), and the Table and the Figure Captions. The recall, precision, and f-measure are a measure of each element and the mean value of these measuring methods are calculated against each metadata element. The final model results presented in Table 4.20 reveals that average recall = 0.922, precision = 0.954 and F-Measure = 0.938 of FLAG-PDFe approach while using the sectLabel dataset ds3.

Figure 4.5 shows the graphical representation of overall results of sectLabel, PDFX, and FLAG-PDFe while using the dataset of articles used by sectLabel approach.

TABLE 4.20: The evaluation of FLAG-FDFe to extract metadata using dataset published by the sectLabel approach.

Metadata	Flag-PDFe		
	Recall	Precision	F-Measure
Title	1.00	1.00	1.00
Journals	0.93	1.00	0.97
Accepted on	1.00	1.00	1.00
Published on	1.00	1.00	1.00
Pages	0.97	1.00	0.98
DOI	1.00	1.00	1.00
Year	1.00	1.00	1.00
References	0.97	0.99	0.98
Author	0.93	0.98	0.96
Affiliation	0.95	0.98	0.96
Country distinct	1.00	1.00	1.00
Email	0.97	0.99	0.98
corresponding	0.79	1.00	0.88
keywords	0.88	0.90	0.89
h1	0.96	0.95	0.96
h2	0.92	0.93	0.93
h3	0.57	0.44	0.50
figure cap	0.90	0.92	0.91
tables cap	0.87	0.95	0.91
tables	0.90	1.00	0.95
figures	0.87	1.00	0.93

The final results revealed the F-Score of sectLabel is 0.920, FDFX is 0.82, and

FLAG-PDFe is 0.92. Therefore, the final result shows the FLAG-PDFe has outperformed state-of-the-art approaches on the gold standard dataset ds3.

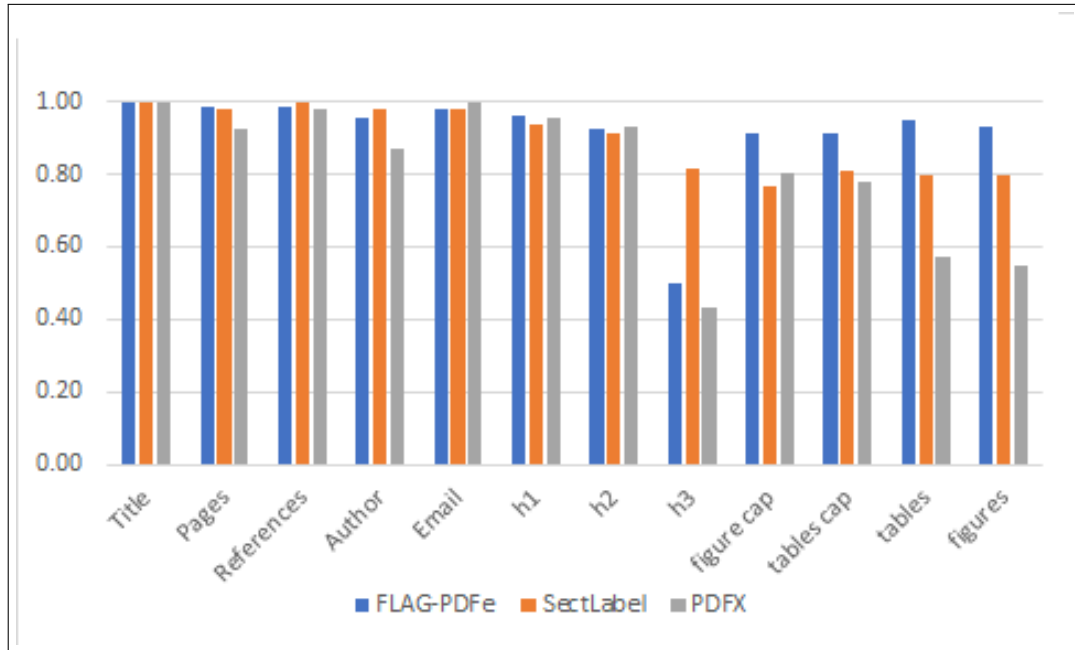


FIGURE 4.5: The final metadata extraction F-Score comparison of FLAG-PDFe with sectlabel and PDFX

4.2.4 Summary

Section 3.3 presents the logical layout structure using the support vector machine (SVM), a machine learning algorithm on the basis of the extracted feature set . Section 3.4 shows the metadata extraction technique from the research articles. The first dataset ds3 is a curated dataset that consists of 250 research articles from diversified journals, with the diversified publication, composition, and layout styles. The second dataset ds2 is a benchmark dataset, which consists of 40 research articles proposed by the sectLabel approach.

We comprehensively evaluated the output of our approach, FLAG-PDFe with the state-of-the-art approaches. First, we evaluated FLAG-PDFe with CERMINE and PDFX on the ds3, which consists of articles from diversified publishers. Then secondly, we evaluated the FLAG-PDFe with sectLabel and PDFX using sectLabel

dataset ds2. The results revealed that FLAG-PDFe has outperformed all previous approaches while using the dataset of articles with publishers having diversified publication styles.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In recent years, enormous growth has been witnessed in diversified domains in the form of scientific research publications that have contributed as an online plethora of scientific resources. Research communities use online tools to have quick access to these new scientific discoveries and inventions. However, there is a limitation when the precise search is required on the content of the metadata, which is due to the lack of structural information and metadata available to these scientific search engines. Hence different research communities and publishers have worked on techniques to extract useful metadata to enhance the capabilities of digital libraries, citation indexes, and search engines.

This thesis performs a critical evaluation of the state-of-the-art metadata extraction approach from scientific research publications. These approaches are classified into two different categories like heuristic-based approaches and machine learning-based approaches. We have performed a comprehensive review of previous literature after studying hundreds of research articles in the domain of metadata extraction and document layout analysis. Based on the evaluation, we have presented the strengths and deficiencies of these approaches which are highlighted in

Chapter 2. After performing the comprehensive evaluation of previous approaches, we have identified some research gaps that we shall try to resolve in our proposed approach. Previous approaches produced better results while extracting metadata from research articles from the same publisher. They produce better results in a controlled environment and the smaller size of the dataset was used in some cases. The feature extraction methodology is not comprehensively evaluated and their techniques are not properly identified. Previous systems use the direct approach on the content of the research article to extract metadata

Based on the identified research gaps, we have proposed a methodology that tries to address the problems in a modular approach. We have divided the problem into subtasks and then focus on each task to efficiently extract metadata from research articles of diversified publishers. We named the Framework “FLAGPDFe” that has four stages where each stage depends upon the output of its previous stage.

Initially, the system extracts the content of the PDF file using a Java-based library which provides text blocks along with textual and geometric information. The first stage extracts the physical layout structure based on the output of the PDF extraction library. Then the system identifies the column layout styles and finally, the correction of reading order is performed. The second stage performs the feature set identification that initially identifies the external boundaries, internal content, and generates line numbers. Different methods are applied to this identified content like template matching, regular expression, keywords, and geometric distance to create feature sets that are categorized as font properties, neighbor distance, font typography, lexical features, and text locations. The final sets of features were selected by using the method Chi-square for feature selection and indexing. In order to evaluate our extracted features, we created a training dataset that contains text blocks along with logical layout structural annotations and feature sets. In the third stage, we evaluated machine learning models to extract logical layout structure from the research article we have categorized these logical

layout structures into eight groups. These groups are the title section, authors and affiliation, header and footer, heading and levels, figure section, table section acknowledgment section, and main text body/paragraphs. We evaluated the logical layout structure extraction model using two testing datasets. In the fourth stage, the system extracts the metadata elements from the logical layout structure by applying different approaches like template matching, regular expression, natural language processing, and heuristics.

The benchmark datasets that are used to evaluate our proposed technique are ESWC dataset from CEURWS.org, sectLabel dataset, and curated data set from the diversified publishers. In section 4.1, we have presented the methodology to extract metadata from the ESWC dataset SemPub2016 challenge and compared our results with the winner of this challenge who is the former Ph.D. student of our research group. We outperformed the challenge winner approach by 16% and the evaluation was performed on the same input and output parameters. In section 4.2, we evaluated our approach on the diversified curated dataset that consists of five Publishers and forty journals. The article composition and layout styles are diversified and the metadata content has different presentation styles. On this dataset, we also evaluated state-of-the-art approaches like PDFX and CERMINE, both tools are freely available online. The system average recall is 0.86, precision is 0.93, and f-measure is 0.89 on the curated dataset. We again evaluated our approach and compare it with the state-of-the-art on the renowned benchmark dataset named "sectLabel". The results reveals that average recall = 0.922, precision = 0.954 and F-Measure = 0.938 of FLAG-PDFe approach while using the sectLabel dataset.

The feature sets that we constructed to find the logical layout structural components of the research article are of diversified nature and they effectively performed for all the data sets. Similarly, we used the support vector machine (SVM) machine learning algorithm to extract the logical layout structure by employing feature sets, again this approach effectively extracted the LLS from articles of diversified publishers. Finally, the approach was able to extract metadata from different logical

layout structures based on template matching, regular expression, natural language processing, and heuristics. The results show that our approach is scalable and it works effectively to extract many metadata elements from research articles of the diversified publisher and has outperformed the state-of-the-art approaches.

5.2 Key Contributions

The selected datasets to evaluate our proposed technique has diversified publishing style and unique metadata requirements. FLAGPDFe outperformed renowned techniques when evaluated on a selected dataset. We have made the following contributions in this regard:

1. After performing comprehensive analysis we divided the problem into parts and then focused on each part to solve it individually. Unlike previous approaches, which directly focus on metadata extraction either by applying rules or tagging metadata for machine learning.
2. Proposed technique generates well-defined features set identified at two levels; the first is the physical layout and textual properties of the individual research article, which are then used to develop the generic set of features. These features can be used to extract logical layout content of articles from publishers with diversified composition styles.
3. Our approach does not completely rely on a single feature set, as in few cases a feature set cannot be extracted correctly from a PDF file. In such scenarios, other feature sets reinforce the missing features and minimize the impact on the output of the system.

4. Our technique works on standard rules to extract similar metadata elements that are present in articles from diversified publication and layout styles.
5. The technique extracts unique metadata hidden in the content of the logical layout structure. Also, not all previous approaches extracted the complete metadata elements, rather they focus on a few as discussed in the chapter 2.
6. We have developed a comprehensive diversified curated dataset from diversified publishers with different layouts and publishing styles.
7. We have done a comprehensive evaluation of our approach with state-of-the-art approaches like CERMINE and PDFX on the benchmark dataset of sectLabel and the diversified curated dataset. Additionally, we evaluated our technique with the winner of the open challenge by leading the A-rated ESWC conference.

We have further explained our research contributions and findings in the relevant sections of this thesis in order to prove the novelty of the work.

5.3 Limitations

The proposed approach Flag-PDFe separates the structural components into two layers. The first is the logical layout structure (LLS), and the second is the metadata present in these LLS. The approach uses the SVM technique to identify the LLS based on textual and geometric features. The approach use heuristics and regular expression on LLS to extract the final metadata of the research article. There are a few limitations in underlying components of the proposed approach, which are mentioned below.

1. The input of the proposed technique is a research article. The approach extracts metadata from born-pdf research articles and will not work on OCR or image-based input research files.
2. The approach uses third-party pdf extraction libraries (like itext or pdfextract) to extract the content of a PDF file. However, there are a few scenarios in which these libraries do not correctly retrieve the textual features of the text blocks from the low-quality PDF files. This limitation of the libraries results in the incorrect identification of the font based features.
3. The authors of the research article should ensure the correct alignment of the paragraph columns. The system selects the most commonly published boundary marge. The incorrect alignment of paragraphs will affect the extraction of neighbor and text location-based features.
4. The features prepared in the proposed work have used the dataset of author selected, CEUR and sectLabel research documents. The research papers of these scientific resources are related to diversified domains. The machine learning technique SMV identifies the LLS and specialized heuristics to extract metadata composition patterns. There is a possibility that a few these methods may not correctly work for the research articles with untested layout and composition styles.

5.4 Future Work

The proposed work exhibits a comprehensive framework to extract metadata from different scalable stages. This dissertation has promoted further exploration and potential directions for future research in this field, which have been highlighted below:

1. This work can be extended in future to extract metadata from publishers that are also associated with the research domains other than computer

science, management sciences, and natural sciences domains. Which will allow researchers to evaluate more machine learning algorithms to identify logical layout structures and metadata elements.

2. A future enhancement of the metadata extraction technique can be achieved through the replacement of regular expression with natural language processing. That can be based on tokenized text mining algorithms like the Hidden Markov Model (HMM), Conditional Random Fields (CRFs), and Recurrent Neural Network (RNN) etc.
3. An intelligent approach can be developed to find and address anomalies in research articles where authors did not follow the publisher guidelines and published PDF files are of low quality.
4. In future, the proposed feature extraction algorithms can be extended by analyzing more diversified composition styles available in the scientific literature. Furthermore, the extracted text with Unicode merged words should be replaced with the correct characters to improve the overall accuracy of lexical and typographical features extraction.
5. Replacing third-party libraries (like iText, and PDFExtract) with a more robust tool to extract the PDF content can increase the efficiency of the font feature extractor. In case the pdf content extraction tool doesn't obtain the font properties, then an alternative method can be developed by measuring the font-weight based on the character boundary boxes.

Bibliography

- [1] A. E. Jinha, “Article 50 million: an estimate of the number of scholarly articles in existence,” *Learned Publishing*, vol. 23, no. 3, pp. 258–263, 2010.
- [2] L. Bornmann and R. Mutz, “Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references,” *Journal of the Association for Information Science and Technology*, vol. 66, no. 11, pp. 2215–2222, 2015.
- [3] F. Ronzano and H. Saggion, “Knowledge extraction and modeling from scientific publications,” in *International Workshop on Semantic, Analytics, Visualization*. Springer, 2016, pp. 11–25.
- [4] R. Johnson, A. Watkinson, and M. Mabe, “The stm report,” *An overview of scientific and scholarly publishing. 5th edition October*, 2018.
- [5] J. Kim, J. R. Trippas, M. Sanderson, Z. Bao, and W. B. Croft, “How do computer scientists use google scholar?: A survey of user interest in elements on serps and author profile pages.” in *BIR@ ECIR*, 2019, pp. 64–75.
- [6] A. Dimou, S. Vahdati, A. Di Iorio, C. Lange, R. Verborgh, and E. Mannens, “Challenges as enablers for high quality linked data: insights from the semantic publishing challenge,” *PeerJ Computer Science*, vol. 3, p. e105, 2017.
- [7] A. Dey, “Machine learning algorithms: a review,” *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1174–1179, 2016.

-
- [8] R. Kumar, *Research methodology: A step-by-step guide for beginners*. Sage Publications Limited, 2019.
- [9] A. Constantin, S. Pettifer, and A. Voronkov, “Pdfx: fully-automated pdf-to-xml conversion of scientific literature,” in *Proceedings of the 2013 ACM symposium on Document engineering*. ACM, 2013, pp. 177–180.
- [10] B. Lowagie, *iText in Action*. Manning, 2011.
- [11] J. Wu, K. M. Williams, H.-H. Chen, M. Khabsa, C. Caragea, S. Tuarob, A. G. Ororbia, D. Jordan, P. Mitra, and C. L. Giles, “Citeseerx: Ai in a digital library search engine,” *AI Magazine*, vol. 36, no. 3, pp. 35–48, 2015.
- [12] S. Mao, A. Rosenfeld, and T. Kanungo, “Document structure analysis algorithms: a literature survey,” in *Document Recognition and Retrieval X*, vol. 5010. International Society for Optics and Photonics, 2003, pp. 197–208.
- [13] M. Granitzer, M. Hristakeva, R. Knight, K. Jack, and R. Kern, “A comparison of layout based bibliographic metadata extraction techniques,” in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*. ACM, 2012, p. 19.
- [14] R. Habib and M. T. Afzal, “Sections-based bibliographic coupling for research paper recommendation,” *Scientometrics*, vol. 119, no. 2, pp. 643–656, 2019.
- [15] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, “A comprehensive survey of mostly textual document segmentation algorithms since 2008,” *Pattern Recognition*, vol. 64, pp. 1–14, 2017.
- [16] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. D. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko, “vispubdata. org: A metadata collection about ieee visualization (vis) publications,” *IEEE transactions on visualization and computer graphics*, vol. 23, no. 9, pp. 2199–2206, 2016.
- [17] S. Marinai and H. Fujisawa, *Machine learning in document analysis and recognition*. Springer, Berlin, Heidelberg, 2007, vol. 90.

- [18] C. Breitinger, B. Gipp, and S. Langer, “Research-paper recommender systems: a literature survey,” *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2016.
- [19] A. M. Khan, A. Shahid, M. T. Afzal, F. Nazar, F. S. Alotaibi, and K. H. Alyoubi, “Swics: Section-wise in-text citation score,” *IEEE Access*, vol. 7, pp. 137 090–137 102, 2019.
- [20] S. Ma, C. Zhang, and X. Liu, “A review of citation recommendation: from textual content to enriched context,” *Scientometrics*, pp. 1–28, 2020.
- [21] G. Nagy and S. Seth, “Hierarchical representation of optically scanned documents,” *CSE Conference and Workshop Papers. 292.*, vol. 1, 1984.
- [22] J. Ha, R. M. Haralick, and I. T. Phillips, “Recursive xy cut using bounding boxes of connected components,” in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 2. IEEE, 1995, pp. 952–955.
- [23] J.-L. Meunier, “Optimized xy-cut for determining a page reading order,” in *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*. IEEE, 2005, pp. 347–351.
- [24] L. O’Gorman, “The document spectrum for page layout analysis,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 11, pp. 1162–1173, 1993.
- [25] K. Seymore, A. McCallum, and R. Rosenfeld, “Learning hidden markov model structure for information extraction,” in *AAAI-99 workshop on machine learning for information extraction*, 1999, pp. 37–42.
- [26] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, p. 282–289.

- [27] F. Peng and A. McCallum, “Information extraction from research papers using conditional random fields,” *Information processing & management*, vol. 42, no. 4, pp. 963–979, 2006.
- [28] K. Bijari, H. Zare, H. Veisi, and H. Bobarshad, “Memory-enriched big bang–big crunch optimization algorithm for data clustering,” *Neural Computing and Applications*, vol. 29, no. 6, pp. 111–121, 2018.
- [29] I. G. Councill, C. L. Giles, and M.-Y. Kan, “Parscit: an open-source crf reference string parsing package.” in *LREC*, vol. 8, 2008, pp. 661–667.
- [30] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 1999.
- [31] M. Sundermeyer, R. Schlüter, and H. Ney, “Lstm neural networks for language modeling,” in *Thirteenth annual conference of the international speech communication association*, 2012.
- [32] A. Prasad, M. Kaur, and M.-Y. Kan, “Neural parscit: a deep learning-based reference string parser,” *International Journal on Digital Libraries*, vol. 19, no. 4, pp. 323–337, 2018.
- [33] J. Wu, J. Killian, H. Yang, K. Williams, S. R. Choudhury, S. Tuarob, C. Caragea, and C. L. Giles, “Pdfmef: A multi-entity knowledge extraction framework for scholarly documents and semantic search,” in *Proceedings of the 8th International Conference on Knowledge Capture*. ACM, 2015, p. 13.
- [34] A. PDFBox, “A java pdf library,” *Dostopno na: <https://pdfbox.apache.org/index.html>*, 2014.
- [35] H. Plochow-Besch, “Pdflib: A library of all available parton density functions of the nucleon, the pion and the photon and the corresponding α_s calculations,” *Computer Physics Communications*, vol. 75, no. 3, pp. 396–416, 1993.

- [36] P. Lopez, “Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications,” in *International conference on theory and practice of digital libraries*. Springer, 2009, pp. 473–474.
- [37] S. Tuarob, S. Bhatia, P. Mitra, and C. L. Giles, “Automatic detection of pseudocodes in scholarly documents using machine learning,” in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 738–742.
- [38] S. R. Choudhury, P. Mitra, A. Kirk, S. Szep, D. Pellegrino, S. Jones, and C. L. Giles, “Figure metadata extraction from digital documents,” in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 135–139.
- [39] C. Caragea, J. Wu, K. Williams *et al.*, “Automatic identification of research articles from crawled documents,” *WSDM Workshop on Web-Scale Classification: Classifying Big Data From the Web*. New York, NY.
- [40] S. Klink and T. Kieninger, “Rule-based document structure understanding with a fuzzy combination of layout and textual features,” *International Journal on Document Analysis and Recognition*, vol. 4, no. 1, pp. 18–26, 2001.
- [41] H. Déjean and J.-L. Meunier, “A system for converting pdf documents into structured xml format,” in *International Workshop on Document Analysis Systems*. Springer, 2006, pp. 129–140.
- [42] C. Ramakrishnan, A. Patnia, E. Hovy, and G. A. Burns, “Layout-aware text extraction from full-text pdf of scientific articles,” *Source code for biology and medicine*, vol. 7, no. 1, p. 7, 2012.
- [43] J. Azimjonov and J. Alikhanov, “Rule based metadata extraction framework from academic articles,” *CoRR*, vol. abs/1807.09009, 2018. [Online]. Available: <http://arxiv.org/abs/1807.09009>

- [44] C. Jiang, J. Liu, D. Ou, Y. Wang, and L. Yu, "Implicit semantics based metadata extraction and matching of scholarly documents," *Journal of Database Management (JDM)*, vol. 29, no. 2, pp. 1–22, 2018.
- [45] R. Ahmad, M. T. Afzal, and M. A. Qadir, "Information extraction from pdf sources based on rule-based system using integrated formats," in *Semantic Web Evaluation Challenge*. Springer, 2016, pp. 293–308.
- [46] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek, and Ł. Bolikowski, "Cermine: automatic extraction of structured metadata from scientific literature," *International Journal on Document Analysis and Recognition (IJ-DAR)*, vol. 18, no. 4, pp. 317–335, 2015.
- [47] Ø. R. Berg, S. Oepen, and J. Read, "Towards high-quality text stream extraction from pdf: technical background to the acl 2012 contributed task," in *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Association for Computational Linguistics, 2012, pp. 98–103.
- [48] M. Granitzer, M. Hristakeva, K. Jack, and R. Knight, "A comparison of metadata extraction techniques for crowdsourced bibliographic metadata management," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, 2012, pp. 962–964.
- [49] D. Tkaczyk, L. Bolikowski, A. Czczeko, and K. Rusek, "A modular metadata extraction system for born-digital articles," in *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*. IEEE, 2012, pp. 11–16.
- [50] H. H. N. Do, M. K. Chandrasekaran, P. S. Cho, and M. Y. Kan, "Extracting and matching authors and affiliations in scholarly documents," in *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2013, pp. 219–228.
- [51] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006.

- [52] P. Willett, "Recent trends in hierarchic document clustering: a critical review," *Information processing & management*, vol. 24, no. 5, pp. 577–597, 1988.
- [53] S. Klampfl, M. Granitzer, K. Jack, and R. Kern, "Unsupervised document structure analysis of digital scientific articles," *International Journal on Digital Libraries*, vol. 14, no. 3-4, pp. 83–99, 2014.
- [54] C.-T. Tsai, G. Kundu, and D. Roth, "Concept-based analysis of scientific literature," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013, pp. 1733–1738.
- [55] S. Epp, M. Hoffmann, N. Lell, M. Mohr, and A. Scherp, "A machine learning pipeline for automatic extraction of statistic reports and experimental conditions from scientific papers," *arXiv preprint arXiv:2103.14124*, 2021.
- [56] S. S. Budhiraja and V. Mago, "A supervised learning approach for heading detection," *Expert systems*, vol. 37, no. 4, p. e12520, 2020.
- [57] C. Soto and S. Yoo, "Visual detection with context for document layout analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3455–3461.
- [58] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [59] M. M. Rahman and T. Finin, "Unfolding the structure of a document using deep learning," *CoRR*, vol. abs/1910.03678, 2019. [Online]. Available: <http://arxiv.org/abs/1910.03678>
- [60] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

- [61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [62] S. T. R. Rizvi, A. Dengel, and S. Ahmed, "A hybrid approach and unified framework for bibliographic reference extraction," *IEEE Access*, vol. 8, pp. 217 231–217 245, 2020.
- [63] S. Madisetty, K. K. Maurya, A. Aizawa, and M. S. Desarkar, "A neural approach for detecting inline mathematical expressions from scientific documents," *Expert Systems*, vol. 38, no. 4, p. e12576, 2021.
- [64] Z. Boukhers, N. Beili, T. Hartmann, P. Goswami, and M. A. Zafar, "Mexpub: Deep transfer learning for metadata extraction from german publications," *arXiv preprint arXiv:2106.07359*, 2021.
- [65] T. Washio and H. Motoda, "State of the art of graph-based data mining," *Acm Sigkdd Explorations Newsletter*, vol. 5, no. 1, pp. 59–68, 2003.
- [66] D. Windridge and M. Bober, "A kernel-based framework for medical big-data analytics," in *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*. Springer, 2014, pp. 197–208.
- [67] R.-Y. Cao, Y.-X. Cao, G.-B. Zhou, and P. Luo, "Extracting variable-depth logical document hierarchy from long documents: method, evaluation, and application," *Journal of Computer Science and Technology*, vol. 37, no. 3, pp. 699–718, 2022.
- [68] L. Gao, Z. Tang, X. Lin, Y. Liu, R. Qiu, and Y. Wang, "Structure extraction from pdf-based book documents," in *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*. ACM, 2011, pp. 11–20.
- [69] K. Santosh, "g-dice: graph mining-based document information content exploitation," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 4, pp. 337–355, 2015.

- [70] M.-T. Luong, T. D. Nguyen, and M.-Y. Kan, “Logical structure recovery in scholarly articles with rich document features,” in *Multimedia Storage and Retrieval Innovations for Digital Library Systems*. IGI Global, 2012, pp. 270–292.
- [71] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.
- [72] D. Garrett, D. A. Peterson, C. W. Anderson, and M. H. Thaut, “Comparison of linear, nonlinear, and feature selection methods for eeg signal classification,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 141–144, 2003.
- [73] V. Bolón-Canedo, N. Sánchez-Marroño, and A. Alonso-Betanzos, “A review of feature selection methods on synthetic data,” *Knowledge and information systems*, vol. 34, no. 3, pp. 483–519, 2013.
- [74] J. R. Vergara and P. A. Estévez, “A review of feature selection methods based on mutual information,” *Neural computing and applications*, vol. 24, no. 1, pp. 175–186, 2014.
- [75] M. J. Alam, P. Kenny, and D. O’Shaughnessy, “A study of low-variance multi-taper features for distributed speech recognition,” in *International Conference on Nonlinear Speech Processing*. Springer, 2011, pp. 239–245.
- [76] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, “Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products,” *Chemometrics and Intelligent Laboratory Systems*, vol. 83, no. 2, pp. 83–90, 2006.
- [77] K. Yan and D. Zhang, “Feature selection and analysis on correlated gas sensor data with recursive feature elimination,” *Sensors and Actuators B: Chemical*, vol. 212, pp. 353–363, 2015.

- [78] K. K. Bharti and P. K. Singh, “Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering,” *Expert Systems with Applications*, vol. 42, no. 6, pp. 3105–3114, 2015.
- [79] V. Bolon-Canedo, D. Fernández-Francos, D. Peteiro-Barral, A. Alonso-Betanzos, B. Guijarro-Berdiñas, and N. Sánchez-Marroño, “A unified pipeline for online feature selection and classification,” *Expert Systems with Applications*, vol. 55, pp. 532–545, 2016.
- [80] A. Jović, K. Brkić, and N. Bogunović, “A review of feature selection methods with applications,” in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 1200–1205.
- [81] C. Lai, M. J. Reinders, and L. Wessels, “Random subspace method for multivariate feature selection,” *Pattern recognition letters*, vol. 27, no. 10, pp. 1067–1076, 2006.
- [82] C.-F. Tsai, “Feature selection in bankruptcy prediction,” *Knowledge-Based Systems*, vol. 22, no. 2, pp. 120–127, 2009.
- [83] P. Shi, S. Ray, Q. Zhu, and M. A. Kon, “Top scoring pairs for feature selection in machine learning and applications to cancer outcome prediction,” *BMC bioinformatics*, vol. 12, no. 1, p. 375, 2011.
- [84] R. S. Olson, N. Bartley, R. J. Urbanowicz, and J. H. Moore, “Evaluation of a tree-based pipeline optimization tool for automating data science,” in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, 2016, pp. 485–492.
- [85] S. Mukherjee and N. Sharma, “Intrusion detection using naive bayes classifier with feature reduction,” *Procedia Technology*, vol. 4, pp. 119–128, 2012.
- [86] S. B. Kim, V. C. Chen, Y. Park, T. R. Ziegler, and D. P. Jones, “Controlling the false discovery rate for feature selection in high-resolution nmr spectra,”

- Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 1, no. 2, pp. 57–66, 2008.
- [87] J. Läuter, E. Glimm, and M. Eszlinger, “Search for relevant sets of variables in a high-dimensional setup keeping the familywise error rate,” *Statistica Neerlandica*, vol. 59, no. 3, pp. 298–312, 2005.
- [88] L. Zhu, S. He, L. Wang, W. Zeng, and J. Yang, “Feature selection using an improved gravitational search algorithm,” *IEEE Access*, vol. 7, pp. 114 440–114 448, 2019.
- [89] X. Han, D. Li, P. Liu, and L. Wang, “Feature selection by recursive binary gravitational search algorithm optimization for cancer classification,” *Soft Computing*, vol. 24, no. 6, pp. 4407–4425, 2020.
- [90] A. W. Haryanto, E. K. Mawardi *et al.*, “Influence of word normalization and chi-squared feature selection on support vector machine (svm) text classification,” in *2018 International Seminar on Application for Technology of Information and Communication*. IEEE, 2018, pp. 229–233.
- [91] M. Makrehchi and M. S. Kamel, “Text classification using small number of features,” in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2005, pp. 580–589.
- [92] Y. Wahba, E. ElSalamouny, and G. ElTaweel, “Improving the performance of multi-class intrusion detection systems using feature reduction,” *arXiv preprint arXiv:1507.06692*, 2015.
- [93] W. Richert, *Building machine learning systems with Python*. Packt Publishing Ltd, 2013.
- [94] M. Welling, “A first encounter with machine learning,” *Irvine, CA.: University of California*, vol. 12, 2011.
- [95] M. Bowles, *Machine learning in Python: essential techniques for predictive analysis*. John Wiley & Sons, 2015.

-
- [96] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [97] R. Choudhary and H. K. Gianey, “Comprehensive review on supervised machine learning algorithms,” in *2017 International Conference on Machine Learning and Data Science (MLDS)*. IEEE, 2017, pp. 37–43.
- [98] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.
- [99] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, New York City, USA, 2009.
- [100] F. P. Preparata and M. I. Shamos, *Computational geometry: an introduction*. Springer Science & Business Media, 2012.
- [101] M. Abdar, U. R. Acharya, N. Sarrafzadegan, and V. Makarenkov, “Ne-nu-svc: A new nested ensemble clinical decision support system for effective diagnosis of coronary artery disease,” *IEEE Access*, vol. 7, pp. 167 605–167 620, 2019.
- [102] M. Claesen, F. D. Smet, J. A. K. Suykens, and B. D. Moor, “Fast prediction with svm models containing rbf kernels,” *ArXiv*, vol. abs/1403.0736, 2014.
- [103] S. Hiregoudar, K. Manjunath, and K. Patil, “A survey: research summary on neural networks,” *International Journal of Research in Engineering and Technology*, vol. 3, no. 15, pp. 385–389, 2014.
- [104] L. Jiang, H. Zhang, and Z. Cai, “A novel bayes model: Hidden naive bayes,” *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 10, pp. 1361–1371, 2008.
- [105] Y. Lin, J.-B. Michel, E. L. Aiden, J. Orwant, W. Brockman, and S. Petrov, “Syntactic annotations for the google books ngram corpus,” 2012.

-
- [106] K. Ma, “Automatic literature metadata extraction from datacite services,” *Recent Patents on Computer Science*, vol. 11, no. 1, pp. 25–31, 2018.
- [107] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, “A practical guide to support vector classification,” *National Taiwan University, Taiwan, Tech. Rep*, 2003.
- [108] A. Dimou, A. Di Iorio, C. Lange, and S. Vahdati, “Semantic publishing challenge—assessing the quality of scientific output in its ecosystem,” in *Semantic Web Evaluation Challenge*. Springer, 2016, pp. 243–254.