Analyzing and Navigating Electronic Theses and Dissertations

Aman Ahuja

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Edward A. Fox, Chair
Chris North
Lifu Huang
Eugenia Rho
Wei Wei

June 12, 2023
Blacksburg, Virginia

Keywords: Electronic Theses and Dissertations (ETDs), Topic Modeling, Object Detection

Analyzing and Navigating Electronic Theses and Dissertations

Aman Ahuja

(ABSTRACT)

Electronic Theses and Dissertations (ETDs) contain valuable scholarly information that can be of immense value to the scholarly community. Millions of ETDs are now publicly available online, often through one of many digital libraries. However, since a majority of these digital libraries are institutional repositories with the objective being content archiving, they often lack end-user services needed to make this valuable data useful for the scholarly community. To effectively utilize such data to address the information needs of users, digital libraries should support various end-user services such as document search and browsing, document recommendation, as well as services to make navigation of long PDF documents easier. In recent years, with advances in the field of machine learning for text data, several techniques have been proposed to support such end-user services. However, limited research has been conducted towards integrating such techniques with digital libraries.

This research is aimed at building tools and techniques for discovering and accessing the knowledge buried in ETDs, as well as to support end-user services for digital libraries, such as document browsing and long document navigation. First, we review several machine learning models that can be used to support such services. Next, to support a comprehensive evaluation of different models, as well as to train models that are tailored to the ETD data, we introduce several new datasets from the ETD domain. To minimize the resources required to develop high quality training datasets required for supervised training, a novel AI-aided annotation method is also discussed. Finally, we propose techniques and frameworks to

support the various digital library services such as search, browsing, and recommendation. The key contributions of this research are as follows:

- A system to help with parsing long scholarly documents such as ETDs by means of object-detection methods trained to extract digital objects from long documents. The parsed documents can be used for further downstream tasks such as long document navigation, figure and/or table search, etc.
- Datasets to support supervised training of object detection models on scholarly documents of multiple types, such as born-digital and scanned. In addition to manually annotated datasets, a framework (along with the resulting dataset) for AI-aided annotation also is proposed.
- A web-based system for information extraction from long PDF theses and dissertations, into a structured format such as XML, aimed at making scholarly literature more accessible to users with disabilities.
- A topic-modeling based framework to support exploration tasks such as searching and/or browsing documents (and document portions, e.g., chapters) by topic, document recommendation, topic recommendation, and describing temporal topic trends.

Analyzing and Navigating Electronic Theses and Dissertations

Aman Ahuja

(GENERAL AUDIENCE ABSTRACT)

Electronic Theses and Dissertations (ETDs) contain valuable scholarly information that can be of immense value to the research community. Millions of ETDs are now publicly available online, often through one of many online digital libraries. However, since a majority of these digital libraries are institutional repositories with the objective being content archiving, they often lack end-user services needed to make this valuable data useful for the scholarly community. To effectively utilize such data to address the information needs of users, digital libraries should support various end-user services such as document search and browsing, document recommendation, as well as services to make navigation of long PDF documents easier and accessible. Several advances in the field of machine learning for text data in recent years have led to the development of techniques that can serve as the backbone of such end-user services. However, limited research has been conducted towards integrating such techniques with digital libraries. This research is aimed at building tools and techniques for discovering and accessing the knowledge buried in ETDs, by parsing the information contained in the long PDF documents that make up ETDs, into a more compute-friendly format. This would enable researchers and developers to build end-user services for digital libraries. We also propose a framework to support document browsing and long document navigation, which are some of the important end-user services required in digital libraries.

Acknowledgments

I would like to express my heartfelt gratitude and appreciation to the following individuals and organizations who have played a pivotal role in the completion of this thesis:

First and foremost, I am deeply indebted to my advisor, Dr. Edward A. Fox, for his unwavering guidance, invaluable insights, and continuous support throughout the research process. His expertise and dedication have been instrumental in shaping the direction and quality of this work.

I extend my sincere thanks to the members of my thesis committee, Dr. Chris North, Dr. Lifu Huang, Dr. Eugenia Rho, and Dr. Wei Wei, for their valuable feedback, constructive criticism, and scholarly contributions. Their expertise and scholarly perspectives have greatly enriched the content of this thesis.

I am grateful to the undergraduate students, Alan Devera, Kecheng Zhu, Jiangyue Li, Zachary Gager, You Peng, Shelby Neal, Andrew Leavitt, Annie Tran, Brian Dinh, Kevin Dinh, Jiayue Lin, Kevin Liu, Mingkai Pang, Theodore Gunn, Zehua Zhang, Luke Wevley, Michael Nader, Elizabeth Keegan, and Gabrielle Nguyen, as well as graduate students, Chenyu Mao and Nirmal Amirthalingam, who have actively participated in this research. Their dedication, hard work, and insightful discussions have significantly contributed to the success of this thesis.

I would also like to express my thanks to William A. Ingram and the University Libraries for their invaluable support throughout the research process. Their resources, access to materials, and assistance in navigating academic databases have been indispensable.

A special mention goes to the members of the Digital Library Research Laboratory for their technical support and collaboration. Their expertise, assistance, and helpful discussions have

contributed to the development and refinement of the research methodology and implementation.

I am deeply grateful to my family and friends for their unwavering support, encouragement, and understanding throughout this journey. Their belief in my abilities and constant motivation have been crucial in overcoming challenges and staying focused on the thesis.

While the list of individuals mentioned here is not exhaustive, each and every one of them has played a significant role in the completion of this thesis. I am truly grateful for their support and contributions.

Contents

List of Figures

List of Tables

[LE,RO]1 [RE] [LO] [C]

Chapter 1

Introduction

1.1 Background and Motivation

Scholarly documents like ETDs contain important research findings, which are of value to a diverse group of users from the scholarly community. Examples of such users include students and researchers who want to review work related to their research area, as well as librarians and university administrators who want an overview of recent research in their institutions. With the vast amount of research being conducted across a variety of domains, millions of ETDs are now publicly available online. However, digital library services for ETDs have not evolved past simple search and browse at the metadata level, thus rendering the vast amount of information from these documents underutilized.

In recent years, advances have been made in NLP-based techniques such as topic modeling, question-answering and text summarization, which might be incorporated to make ETDs more accessible. However, a majority of these documents exist as PDF files, and are often long and filled with highly specialized details. While some tools can work with these files, the results we have observed have been poor; other tools require data in a structured format such as XML. Accordingly, there is a need to build electronic infrastructure that can leverage the rich scholarly information contained within ETDs and make it accessible to the wider community.

1.2 Problem Statement

This thesis aims to develop methodologies that can support making the knowledge contained in ETDs more accessible for digital library users. Although a comprehensive digital library system should ideally support multiple end-user services, like browsing, search, retrieval, and question-answering, the foremost requirement for any such service is to have data available in a structured, machine-friendly format such as XML. Hence, a major contribution of this thesis would be a framework to parse ETDs in PDF to structured formats like XML. The parsed document can then be employed for training models for supporting end-user services. It can also be helpful in making long documents more accessible by breaking them down into multiple smaller components like chapters and sections. Moreover, structured representations such as XML can be used to develop web-based systems, which have better compatibility with accessibility tools such as on-screen readers, thus allowing those with disabilities to access this information. Given the recent success of object detection models in document layout analysis, we will take the object detection approach for this work. We will develop methodologies that can address several challenges that arise in the process of parsing long PDF documents. These include limited availability of training data, heterogeneity in document types, the imbalanced number of elements in the various classes, and the resource-intensive nature of dataset annotation. Unfortunately, there is no mechanism for parsing extracted objects to determine relationships among them, and converting them into a structured format to make them accessible to users with special needs.

We also will investigate how this parsed information can be used for downstream tasks. Regarding the scope of this work, we will focus on techniques that can be helpful in navigating and browsing documents from a digital library. For example, consider that the most intuitive way of making a browsing system is to group items by categories. Users can select a cate-

gory of their preference and browse the respective documents. However, in case of scholarly documents, grouping documents by research areas is a non-intuitive task, since many documents only contain subject/department level information, which is often very high level. It is hard to classify documents based on pre-defined categories, due to the absence of a unified list of categories and the datasets essential for training such models. Hence, we will study unsupervised methods such as topic modeling for this task. The resultant topics or categories, as well as their respective documents, can then be used for supporting document browsing by research area in a digital library.

1.3 Research Hypotheses

The central hypotheses of this research are listed below:

- **H1:** Object detection based document layout analysis methods for long scholarly documents, trained on high quality domain-specific labeled data, perform better than those trained on a larger dataset originating from other related domains, such as research papers.
- **H2:** Pre-training on other scholarly datasets, albeit from a different domain such as research papers, improves the performance of document layout analysis methods on long scholarly documents such as ETDs.
- **H3:** Training on derived datasets, such as augmented versions of the original training data, can significantly improve the performance of layout analysis models.
- **H4:** To perform well on other document types, such as scanned documents, models trained on a specific type of documents, such as born-digital ones, require additional training using techniques, like augmentation, that help bridge the distribution gap.
- **H5:** AI-aided annotation methods, such as using models trained on existing smaller

datasets to extract weak labels for unlabeled data, reduce the resources required for annotating additional data.

- **H6:** Models trained on datasets with skewed distributions in terms of class labels achieve better performance on minority classes when trained on additional data from those classes, such as from AI-aided annotation methods.
- **H7:** Combining the predictive power of AI models with rules formulated based on domain expertise possessed by humans reduces errors in predictive tasks such as document structure parsing.
- **H8:** Neural topic models can outperform other traditional topic models, such as LDA, while doing topic modeling on scholarly documents such as ETDs and their chapters.

1.4 Research Questions

Based on the hypotheses listed above, the work proposed in this thesis will focus on the following research questions:

- **R1:** What are the different elements that are important in an ETD that can be helpful for training machine learning models for downstream tasks like searching, browsing, question-answering, etc.? How can we develop a dataset that can support training supervised machine learning models to extract these elements from an ETD?
- **R2:** Are datasets from other related domains, such as research papers, sufficient to train layout analysis methods for ETDs? How can these datasets benefit layout analysis methods for ETDs, when used in conjunction with domain specific datasets?
- **R3:** What type of augmentation strategies can be used to derive more training data for object detection models? How can we use augmented datasets to improve the performance of object detection models?

- **R4:** Can document analysis methods trained on documents of one type, such as digital PDF documents, facilitate document analysis on other types of documents, such as scanned documents?
- **R5:** How can annotation methods utilize the power of models trained on existing datasets, to reduce the resources required in the annotation process?
- **R6:** How can we improve the performance of machine learning models, especially on minority classes, using datasets developed using AI-aided annotation?
- **R7:** How can domain expertise, such as a set of rules about syntax and structure that are known to domain experts, be used to develop a set of post-processing rules, which when used in combination with machine learning methods, improve the process of document layout analysis?
- **R8:** Can neural topic models outperform traditional topic models such as LDA, on commonly used topic evaluation metrics, such as coherence and topic diversity?

1.5 Overview of Chapters

Figure 1.1 gives a high-level overview of different chapters proposed in this thesis, along with their respective contributions. The rest of this research is organized as follows:

- Chapter 2 outlines some of the important techniques and datasets related to the work proposed in this thesis.
- Chapter 3 introduces a list of important elements commonly found in ETDs, and a new dataset for training document layout analysis models on ETDs. It also describes training for object detection, and an evaluation of related models.
- Chapter 4 proposes an augmentation-based training approach for object detection models. Experimental results showing how co-training on augmented data alongside orig-

inal data can improve the performance of object detection models on layout analysis, are also presented.

- Chapter 5 introduces an AI-assisted framework for annotating object detection data to improve the performance of layout analysis methods on minority classes. A new dataset to support layout analysis of scanned ETDs, as well as to improve extraction of low-frequency elements such as metadata and algorithms, is also presented.
- Chapter 6 proposes a parsing framework to generate structured representations of long scholarly documents using the set of objects derived from object detection models.
- Chapter 7 introduces a framework for utilizing the elements extracted from document layout parsing, for downstream tasks such as browsing and recommendation, by means of topic modeling.

1.6 Author's Prior Work and Publications

During the course of the doctoral program, the author of this proposal has published several peer-reviewed papers in the domains of retrieval and ranking, question-answering, and topic modeling. These are listed below. Entries 1-4 relate closely to this dissertation, and the contributions of the co-authors thereof are hereby acknowledged.

1. Satvik Chekuri, Prashant Chandrasekar, Bipasha Banerjee, Sung Hee Park, Nila Masrourisaadat, **Aman Ahuja**, William A. Ingram and Edward A. Fox, "Integrated Digital Library System for Long Documents and their Elements." *In Proceedings of the 23rd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2023).*
2. **Aman Ahuja**, Kevin Dinh, Brian Dinh, William A. Ingram, and Edward A. Fox. "A New Annotation Method and Dataset for Layout Analysis of Long Documents." *In Companion Proceedings of the ACM Web Conference 2023, pp. 834-842. 2023*,

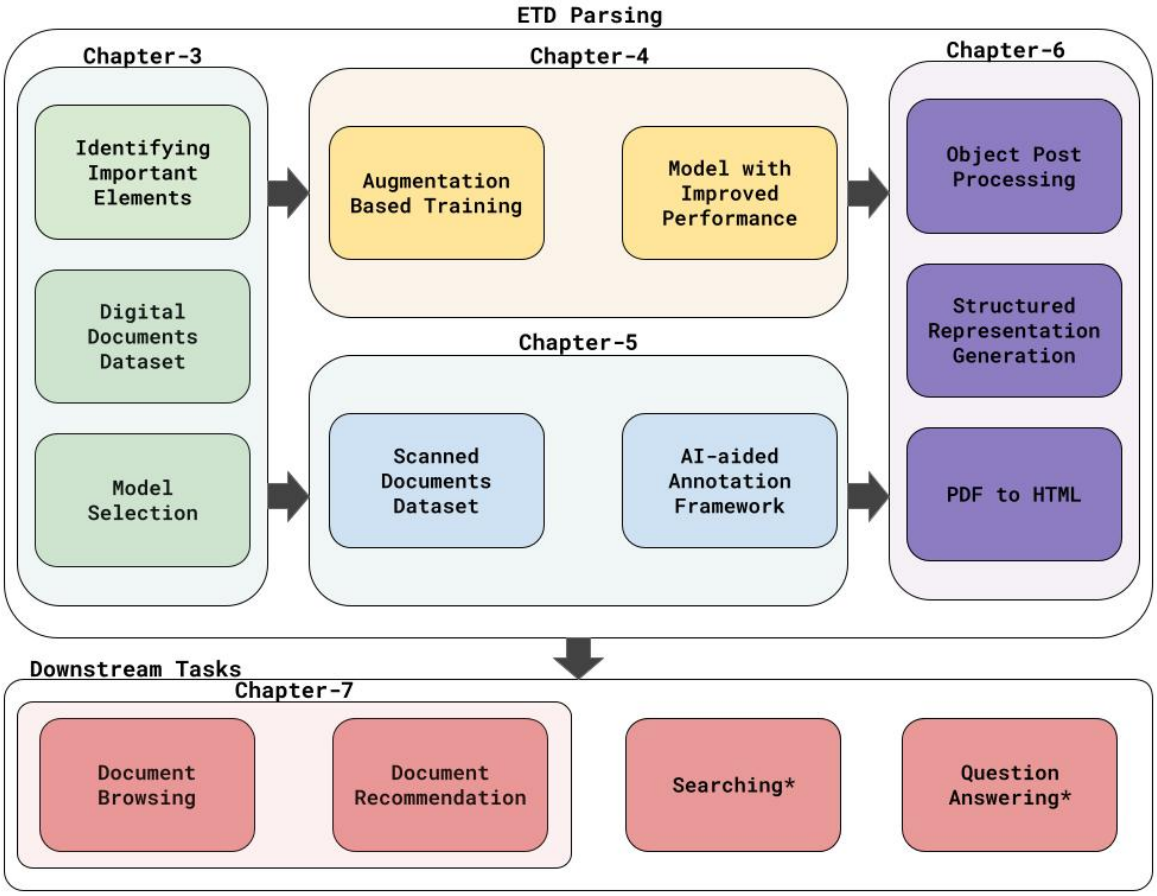


Figure 1.1: An overview of the different chapters, along with their key components, as proposed in this thesis. (* indicates the components are largely beyond the scope of this thesis.)

https://doi.org/10.1145/3543873.3587609.

3. **Aman Ahuja**, Alan Devera, and Edward A. Fox, "Parsing Electronic Theses and Dissertations Using Object Detection." *In Proceedings of the First Workshop on Information Extraction from Scientific Publications (WIESP 2022, held in conjunction with AACL-IJCNLP 2022)*, https://aclanthology.org/2022.wiesp-1.14.pdf.
4. **Aman Ahuja**, Chenyu Mao, William A. Ingram, and Edward A. Fox, "Analyzing and Navigating ETDs Using Topic Models." *In The Journal of Electronic Theses and Dissertations* (to appear).

5. Ming Zhu, **Aman Ahuja**, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. "Question Answering with Long Multiple-Span Answers." *In Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3840-3849. 2020.
6. **Aman Ahuja**, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan K. Reddy. "Language-Agnostic Representation Learning for Product Search on E-Commerce Platforms." *In Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 7-15. 2020.
7. Xuan Zhang, Zhilei Qiao, **Aman Ahuja**, Weiguo Fan, Edward A. Fox, and Chandan K. Reddy. "Discovering Product Defects and Solutions from Online User Generated Contents." *In The World Wide Web Conference*, pp. 3441-3447. 2019.
8. Ming Zhu, **Aman Ahuja**, Wei Wei, and Chandan K. Reddy. "A Hierarchical Attention Retrieval Model for Healthcare Question Answering." *In The World Wide Web Conference*, pp. 2472-2482. 2019.
9. **Aman Ahuja**, Ashish Baghudana, Wei Lu, Edward A. Fox, and Chandan K. Reddy. "Spatio-Temporal Event Detection from Multiple Data Sources." *In Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 293-305. Springer, Cham, 2019.
10. Vineeth Rakesh, Weicong Ding, **Aman Ahuja**, Nikhil Rao, Yifan Sun, and Chandan K. Reddy. "A Sparse Topic Model for Extracting Aspect-Specific Summaries from Online Reviews." *In Proceedings of the 2018 World Wide Web Conference*, pp. 1573-1582. 2018.
11. **Aman Ahuja**, Wei Wei, Wei Lu, Kathleen M. Carley, and Chandan K. Reddy. "A Probabilistic Geographical Aspect-Opinion Model for Geo-tagged Microblogs." *In 2017 IEEE International Conference on Data Mining (ICDM)*, pp. 721-726. IEEE, 2017.

Chapter 2

Review of Literature

2.1 Document Layout Analysis: Datasets

With the growing interest in using object detection based methods for document layout analysis, several datasets have been introduced. Many of these datasets focus on specific object types. For instance, TableBank [23], ScanBank [19], and MFD [3] consist of tables, figures, and equations, respectively. Several datasets that consist of a diverse set of objects have also been introduced. HJDataset [34] consists of historical Japanese documents. PRImA [4] consists of document images from magazines and research papers. PubLayNet [47] is based on PDF articles from PubMed Central. The number of different objects, however, is limited in these datasets. DocBank [24] is a large dataset that consists of a diverse set of objects from research papers. But given the differences between research papers and long documents such as ETDs, models trained on DocBank do not generalize well to ETDs.

2.2 Document Layout Analysis: Annotation Methods

Due to the intensive nature of dataset annotation in terms of time and cost, researchers have proposed several techniques to annotate training datasets for object detection models. For PDF documents with an accompanying MS-Word, XML, or LaTeX file, automatic extraction based on tags is possible [23, 24]. However, in the case of scanned documents, existing rule-

based approaches do not yield high-quality results. In such cases, techniques have been explored that can help annotators, or guide them in annotating samples about which the model is most uncertain [48].

2.3 Document Layout Analysis: Techniques

Early works in the domain of document layout understanding used rule-based approaches [14, 22]. Other approaches, e.g., GROBID [26] and CERMINE [38], designed for parsing scientific documents, primarily focused on short documents such as research papers, and use an ensemble of sequence labeling methods for document parsing. With the advent of deep-learning based object detection methods such as Fast-RCNN [12], Faster-RCNN [32], and YOLO [30, 40], document layout analysis based on object detection has been proposed. LayoutParser [35] uses object detection models that have been pre-trained on different object detection datasets to support layout understanding. However, since it primarily uses research-paper based datasets, it doesn't perform well on ETDs. Moreover, the number of object types it supports is very limited. More recently, layout-based language models [17, 45, 46] have been proposed. This line of work uses a multimodal architecture, i.e., a combination of visual and textual features, to pre-train the model on a large corpus of unlabeled data consisting of document images and their corresponding text. Although these models can then be fine-tuned on other downstream tasks such as object detection, they still require domain-specific annotated data for fine-tuning. Recently, to make the documents more accessible, services such as SciA11y [41] have been developed. However, their scope is limited to research papers, rather than long documents such as books and ETDs.

2.4 Analysis of ETDs

With the growing number of ETDs that are publicly available on the web, techniques aimed at analyzing ETDs have also gained interest in the research community. [39] proposes a framework for automatic crawling of ETDs from public repositories, as well as the resultant corpus of ETDs. An important line of work in the analysis of ETDs aims to extract elements, such as metadata [8, 9], URLs [33], etc. [29] proposes an XML schema for ETDs in a digital library.

2.5 Topic Modeling

Topic modeling has been widely studied in the domain of text mining to discover latent topics. One of the earliest methods to discover topics in text documents was probabilistic Latent Semantic Indexing (pLSI) [16]. However, since pLSI was based on the likelihood principle and did not have a generative process, it cannot assign probabilities to new documents. This was alleviated by Latent Dirichlet Allocation (LDA) [6], which models each document as a mixture over topics, and topics as a mixture over words.

With advances in the field of deep learning, neural topic models have gained increasing interest. Neural Variational Document Model (NVDM) [27] is a neural topic model that uses an unsupervised generative model based on Variational Autoencoders (VAE) [21]. Several other topic models that use a VAE-based architecture have been proposed [10, 28, 36]. More recently, pre-trained language models like BERT [20] and RoBERTa [25] have shown significant performance improvements in many NLP-related tasks due to their ability to learn contextualized representations of text. Consequently, several topic models that incorporate the representations from pre-trained language models have been proposed. BERTopic [13]

uses a clustering-based approach to first cluster documents based on their language model extracted representations, and then extracts the most representative words, i.e., topics, for each cluster using a TF-IDF based approach. In this process, however, the topics are not learnt, and are rather extracted using a post-processing mechanism. Contextualized Topic Model (CTM) [5] proposed an end-to-end learnable architecture that uses language model derived representations from Sentence-BERT [31] along with bag-of-words embeddings, in a VAE-based architecture similar to ProdLDA [36].

Chapter 3

Parsing Long PDF Documents Using Object Detection

3.1 Chapter Overview

In this chapter, we propose a set of elements in an ETD that are important for downstream tasks like searching, browsing, question-answering, etc. We also introduce ETD-OD, a new object detection dataset that contains over 25K page images originating from 200 ETDs, consisting of elements that can be important sources of information in an ETD. Finally, we investigate the performance of various state-of-the-art object detection models for document layout understanding on ETDs using the proposed dataset.

3.2 ETD Elements

Historically, ETDs do not conform to a universally accepted format, since different colleges and universities have their own specific standards and requirements for ETDs. In this section we discuss the elements that are typically found in ETDs and would be important to extract for further analysis and downstream tasks. This list was curated after extensive discussions with digital librarians and researchers. We broadly categorize the different elements of ETDs

into the following two-level taxonomy, i.e., set of broad and narrower classes.

3.2.1 Metadata

The metadata consists of elements that contain unique identifiable information about an ETD, including information found on the front page. Key metadata elements are:

- **Title:** The main title of the document.
- **Author:** Name of the document author.
- **Date:** Date (or month/year) when the document was published.
- **University:** University/institution of the author.
- **Committee:** Committee that approved the document, e.g., the student's graduate committee.
- **Degree:** Degree (e.g., Master of Science, Doctor of Philosophy) being earned.

3.2.2 Abstract

The abstract is an important element of an ETD, as it contains a summary of the work, typically about a page long. Its elements include:

- **Abstract Heading:** Since many ETDs contain multiple abstracts, such as a technical abstract and general audience abstract, or an abstract in English as well as the original language, extracting the abstract heading makes it easier to segment, and could be helpful in categorizing the abstract by audience type.
- **Abstract Text:** The actual text of the abstract.

3.2.3 List of Contents

The list of contents (also referred to as table of contents) of an ETD determines where different components are located based on their page numbers. This helps with accurately mapping the chapters and sections, as well as figures and tables, since they are generally included in the list of contents. This subcategory includes the following elements:

- **List of Contents Heading:** This helps identify the specific type of list (e.g., list of chapters/sections, list of figures, list of tables).
- **List of Contents Text:** This is the actual list of entries for this type of content.

3.2.4 Main Content

Chapters are one of the most important components of an ETD, as they contain detailed information about the research described in the document. This subcategory consists of elements that can typically be found in the chapters of an ETD.

- **Chapter Title:** The title of the chapter.
- **Section:** Quite often, chapters themselves can be long. It may be desirable to have further delimiters such as sectional headers. Hence, we include the section names (along with other identifiers such as numbers) which can be used for further splitting of the document.
- **Paragraph:** The main textual content of the ETD.
- **Figure:** This includes figures, charts, and other visual illustrations included in the document.
- **Figure Caption:** The text caption that describes the figure.
- **Table:** The table element category.
- **Table Caption:** The text caption that describes the table.

- **Equation:** Mathematical equation.
- **Equation Number:** Quite often, equations are numbered, which can be helpful in linking them to the list of equations that may be included in the document.
- **Algorithm:** Algorithms, such as pseudo-code.
- **Footnote:** We separate footnotes from regular paragraphs, as they typically provide auxiliary information which might be undesirable in many downstream tasks, such as summary generation.
- **Page Number:** Page numbers, which could be helpful in cross-referencing pages and the objects contained therein to the list of contents.

3.2.5 Bibliography

We also include bibliographic elements in the list of objects. They are described below:

- **Reference Heading:** The header that indicates start of the references list.
- **Reference Text:** The actual list of references cited in the document.

In our dataset, we regard appendices as chapters, since they contain many elements that are found in the main chapters. They can however, be easily differentiated from main chapters based on the title.

3.3 Dataset

In this section we introduce ETD-OD, an object detection dataset for layout analysis on scholarly long documents such as ETDs.

3.3.1 Dataset Source

The ETD-OD dataset consists of 25K page images from 200 theses and dissertations. These documents were downloaded from publicly accessible institutional repositories, and were uniformly sampled with regards to degree, domain, and institution. Since object detection requires images as the input data, the documents were split into page images using the `pdf2image`[1] Python library. These images were then used for annotation.

3.3.2 Annotation

We use Roboflow[2] for annotating the page images in our dataset. Each annotation was done by one of the 6 undergraduate students, each of whom was a computer science student from junior year or above. Each data sample was further validated for correctness by two graduate students.

3.3.3 Dataset Statistics

Table 3.1 shows the detailed statistics for different object categories in our dataset. The dataset consists of ∼25K page images and ∼100K bounding boxes spanning across different object categories. Owing to the variation in the frequency of occurrence of various object categories in documents, some categories have many more samples as compared to others. Elements such as paragraphs can be found on most pages, and hence, it is the dominant category in our dataset. 80% of the images and their corresponding objects were used for training, while the remaining 20% were used as the validation set.

[1] https://pypi.org/project/pdf2image/
[2] https://roboflow.com/

Category	**# Instances**
Title	439
Author	404
Date	338
University	309
Committee	282
Degree	279
Abstract Heading	169
Abstract Text	183
List of Contents Heading	512
List of Contents Text	1059
Chapter Title	2211
Section	9337
Paragraph	30359
Figure	6359
Figure Caption	5722
Table	2654
Table Caption	2213
Equation	5092
Equation Number	3051
Algorithm	96
Footnote	5722
Page Number	24543
Reference Heading	313
Reference Text	2088
Total Objects	99859
Total Images	25073

Table 3.1: Distribution of different object categories in our dataset. *Note: Some of the documents were accompanied with front matter (metadata) pages that are sometimes generated by the digital libraries. We include annotations for such documents as well, and hence, the number of metadata elements does not exactly match the number of documents.*

3.4 Proposed Framework

We now introduce the proposed framework for extracting important elements from an ETD by means of object detection. The architecture of our framework is illustrated in Figure 5.3. The different modules shown can broadly be divided into the following three categories.

3.4.1 Data and Preprocessing

Since our framework is primarily built for parsing long scholarly documents, it takes the PDF version of the document as input. The input file is converted to individual page images (.jpg format) using Python-based PDF libraries such as `pdf2image`. Next, the page images are individually fed to the Element Extraction module for further processing.

3.4.2 Element Extraction using Object Detection

This module forms the backbone of our system. It takes the individual page images as input, and uses an object detection model such as Faster-RCNN or YOLO for object detection. These models are first trained on the ETD-OD dataset. The specific details about training object detection models are included in later sections of this chapter. While using the object detection models as a part of this module, only inference is performed, and no updates are made to the model parameters. The output of object detection will be a list of elements, where each element contains information about the bounding boxes such as the coordinates, along with the category labels. This process is repeated for all of the pages in the document, and finally, a list of pages accompanied by their respective elements is populated.

In some instances, the object detected by the model is classified as one belonging to a different, yet similar category. In such cases, we use certain post-processing rules to correct

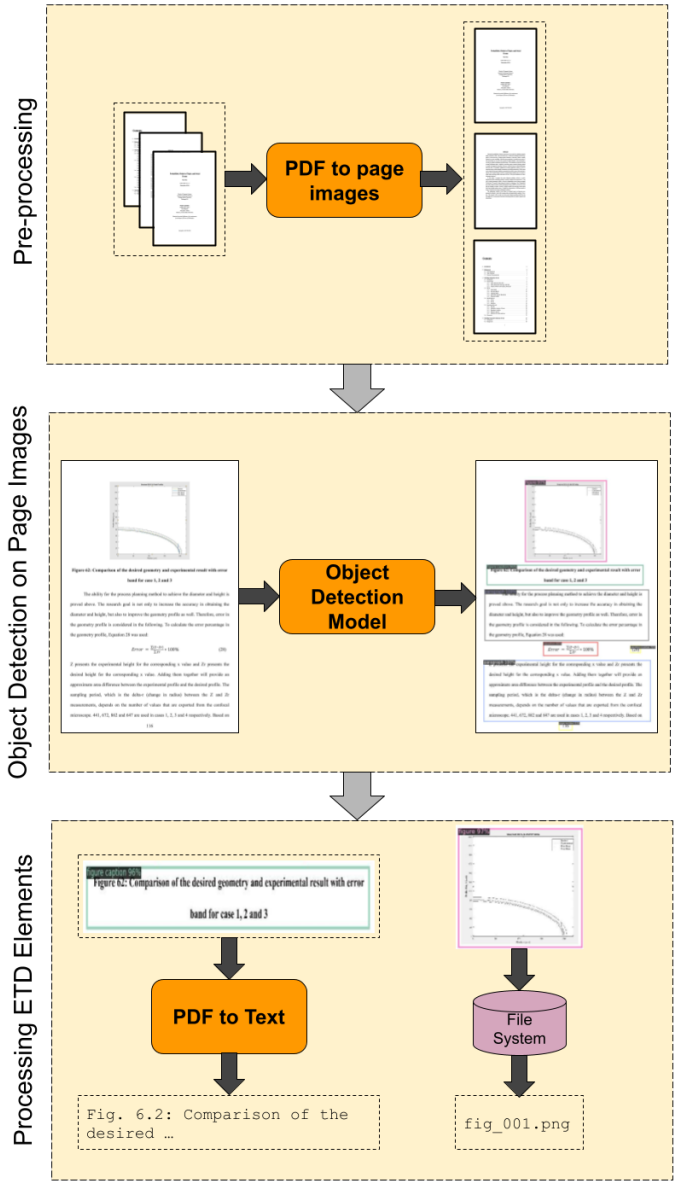


Figure 3.1: Architecture of the proposed object detection based parsing framework.

the predictions. For example, *abstract heading* being mis-classified as chapter heading is one of the common errors, since both of these elements are often found in bigger font size at the beginning of a page. This can, however, be corrected by enforcing a constraint such as: a chapter heading in the first 10 pages with matching keyword "abstract" will be the abstract heading. We use a set of such rules for different object types to correct mis-classifications. This component is discussed in detail in Chapter 6.

3.4.3 Post-Processing Extracted Objects

After extracting all of the elements for all of the pages in the document, we regard the objects as broadly belonging to two types. The first type includes **image-based** objects such as figures, tables, algorithms, and equations, that need to be stored on the file system as an image. We regard tables as image-based objects even though they might contain text, since further extraction of information in structured format from tables is beyond the scope of this work. The second type of object includes **text-based** elements such as paragraphs, titles, etc., which need further processing to be converted to plain text. We regard all object categories excluding the image-based ones as textual elements.

For converting text-based objects to plain text, we use off-the-shelf tools and libraries. Some PDF documents are born-digital, where the text can be easily extracted using Python libraries such as `pymupdf`[3] based on page ID and bounding box coordinates. For scanned documents we use optical character recognition (OCR) tools such as `pytesseract`[4].

For image-based elements, we record the path of the image that is cropped based on the coordinates. Figures and tables are mapped to their respective captions based on proximity. For any figure/table element, the caption object closest to them based on Euclidean distance

[3] https://pymupdf.readthedocs.io/en/latest/
[4] https://pypi.org/project/pytesseract/

w.r.t. bounding box coordinates is assumed to be the caption. A similar method is followed to map equations with their equation numbers, with an added constraint that the y-coordinate of the center of the equation number should fall between min and max y-coordinates of the equation object.

3.5 Object Detection Training

We use the ETD-OD dataset introduced in this chapter for training object detection models for our framework. The models currently supported are:

- **Faster-RCNN** [32]: Faster-RCNN is an object detection model that has two stages. A region proposal network generates regions of interest, which are fed to another network for final detection. We use the version of Faster-RCNN that uses ResNeXt-101 [44] as the backbone model.
- **Faster-RCNN pre-trained on DocBank** [24]: Faster-RCNN (with ResNeXt-101 backbone) is pre-trained on DocBank, and then fine-tuned on ETD-OD. Although DocBank does not include all of the elements found in ETDs, we hypothesize that the scholarly nature of documents used in pre-training should help improve the performance over the vanilla version of the model.
- **YOLOv5** [18]: YOLO is a family of single stage object detection models that perform the processes of localization and detection using a single end-to-end network. This improves the speed without any significant drop in performance. These models have shown impressive performance on various datasets [42].
- **YOLOv7** [40]: This is the most recent version of YOLO, which has been shown to outperform many object detection models.

Both of the Faster-RCNN models were trained on our dataset for 60K iterations with an

inference score threshold of 0.7. The models were based on the implementation included in the open-source detectron2 [43] framework. For the DocBank-pretrained version of the model, we used the original set of weights and configurations open-sourced by the authors. Both of the versions of YOLO were based on the open-source implementations, and were trained for 150 epochs.

3.6 Experimental Results

In this section, we discuss the results obtained in the experimental analysis of our work.

3.6.1 Evaluation Metrics

For the quantitative evaluation of object detection models, the commonly used metrics are average precision (AP) and mean average precision (mAP). AP is defined as the area under the precision-recall curve for a specific class. mAP is the average of AP values for all object classes. Both of these metrics have different versions based on the overlap threshold (also referred to as *Intersection over Union or IoU*) used for comparing the predicted object against ground truth. For example, in *mAP@0.5*, all of the objects with an intersection of 50% or more with the ground truth will be regarded as correct predictions. Another commonly used version of mAP is *mAP@0.5-0.95*, which is the average mAP over different thresholds, from 0.5 to 0.95 with step 0.05.

Model	mAP@0.5	mAP@0.5-0.95
Faster-RCNN	39.1	19.6
Faster-RCNN*	76.2	44.0
YOLOv5	83.4	52.1
YOLOv7	<u>85.3</u>	<u>52.7</u>

Table 3.2: mAP comparison for object detection models on ETD-OD. Faster-RCNN* represents the model pre-trained on DocBank and fine-tuned on ETD-OD. Underlined values indicate best performing models.

3.6.2 Analysis of Various Object Detection Models Trained on ETD-OD

Table 3.2 shows performance of different object detection models on the validation set of our dataset. The following observations can be made from the mAP values shown.

- **Pre-training on scholarly documents improves model performance:** The basic version of Faster-RCNN without any pre-training on scholarly documents has the lowest performance among all the models. The same model, after pre-training on DocBank, and then fine-tuned on the ETD dataset, gives much better performance. Since DocBank also consists of scholarly documents, albeit of different type, the pre-training process exposes the model to a diverse dataset, which eventually results in better generalization and predictive performance.
- **YOLO outperforms Faster-RCNN on ETD dataset:** YOLO models belong to the class of single stage detectors, which are designed with an emphasis on speed. YOLO typically performs worse than Faster-RCNN in scenarios where the objects are smaller or multiple objects are close to each other. However, in the case of documents, most objects are typically of large size and have minimal overlap with each other due to white spaces and line breaks around objects (such as between a header and paragraph). Hence, it outperforms Faster-RCNN on the ETD dataset.

3.6.3 Analysis of Detection Performance on Different Object Categories

Category	AP@0.5	Category	AP@0.5
Title	92.5	Paragraph	97.4
Author	89.5	Figure	98.4
Date	68.3	Fig. Caption	95.4
University	91.1	Table	94.7
Committee	96.5	Tab. Caption	89.8
Degree	68.3	Equation	72.6
Abs. Heading	94.2	Eqn. Number	55.0
Abs. Text	86.7	Algorithm	66.6
LOC Heading	75.5	Footnote	98.9
LOC Text	99.3	Page Number	51.3
Chapter Title	88.8	Ref. Heading	80.7
Section	90.9	Ref. Text	99.3

Table 3.3: AP@0.5 values for different object categories for YOLOv7 (Abs. = Abstract, LOC = List of Contents).

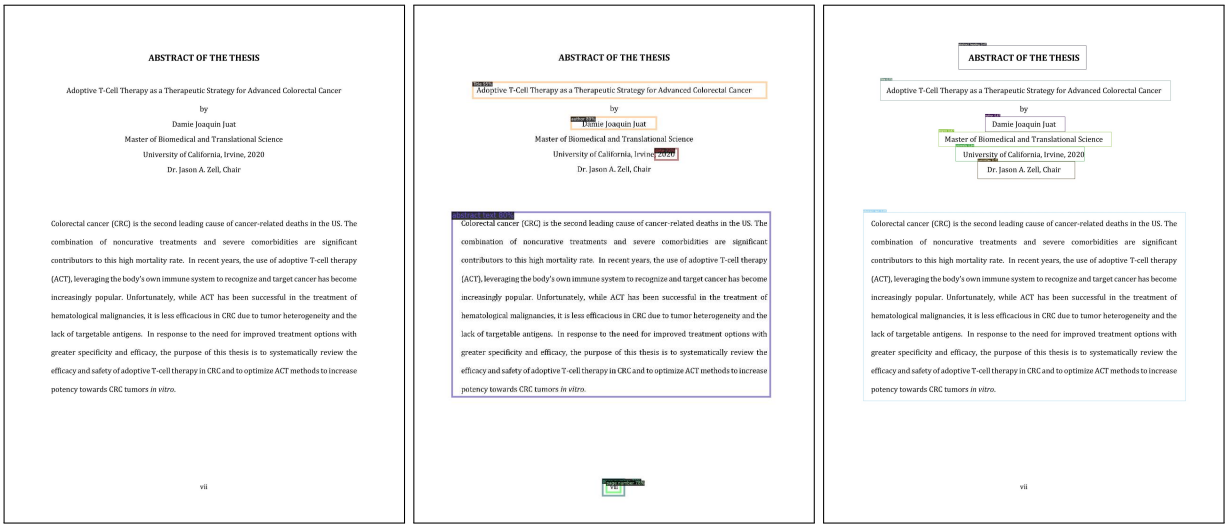
In Table 3.3, we show the performance of the best performing model (YOLOv7) on various object categories in our dataset. The lower performance of certain categories can generally be attributed to two reasons:

- **Limited Number of Training Samples:** Elements such as degree, date, and algorithm have very few instances in our dataset. As such, the performance on these classes is lower than others.
- **Smaller Object Sizes:** Elements such as page number and equation number tend to be of smaller size as compared to other elements. Since object detection models tend to struggle with localization of smaller objects, performance of such classes is impacted.

3.6.4 Comparison against Other Layout Detection Datasets

Categories	DocBank only	ETD-OD only	DocBank ETD-OD
Abstract	2.29	0.0	67.42
Author	5.8	19.27	73.27
Caption	42.72	55.04 / 18.27	97.46 / 89.03
Date	0.0	0.0	76.28
Equation	8.13	62.28	76.19
Figure	72.44	78.21	95.01
Footer	69.38	85.03	97.64
List	NA	NA	NA
Paragraph	5.01	80.64	94.34
Reference	2.94	75.43	97.92
Section	19.88	66.99	77.63
Table	33.25	49.04	89.7
Title	1.1	11.3	73.85

Table 3.4: AP@0.5 values for categories supported by DocBank using Faster-RCNN trained on different datasets and evaluated on the validation set of ETD-OD. For *Caption*, we list the Figure Caption / Table Caption values for models trained on ETD-OD.



(a) Original Image (b) Faster-RCNN* (DocBank, ETD-OD) (c) YOLOv7

Figure 3.2: Examples of outputs generated by the Faster-RCNN* and YOLOv7 models.

Chapter 4

Augmentation-Based Training for Layout Analysis Models

4.1 Chapter Overview

In Chapter 3, we introduced a dataset that can be used to train object detection models to extract scholarly elements from ETDs. While having high quality manually annotated datasets is an ideal method to train supervised machine learning methods, the high costs of manual annotation often restrict researchers from getting access to large datasets. Hence, there is a need to develop methods that can exploit the limited amount of manually annotated datasets to the highest capacity. One such method is data augmentation, which augments the existing training data curated for object detection training, by applying one or more augmentation steps to each training image, while utilizing the annotations of the original image. In this chapter, we explain an augmentation-based training approach for training object detection models. We used this approach to train layout analysis for ETDs, and experimental results show that augmentation-based training yields better performing models.

4.2 Image Augmentation

We start by introducing data augmentation for images. We are given a set of N images $\mathbf{I} = \{\mathbf{i_1}, ..., \mathbf{i_N}\}$ and annotations $\{\mathbf{b_1}, ..., \mathbf{b_N}\}$, where $\mathbf{b_k}$ denotes the set of bounding box coordinates and the corresponding labels associated with image $\mathbf{i_k}$. We also consider a set of image transformation functions $\mathcal{F} = \{f_1, .., f_M\}$ and number of augmentation steps $m < M$.

For each image $\mathbf{i_k}$, our image augmentation process first samples the m transformation functions from $\mathcal{F}$. Each of these transformations is iteratively applied on the image to generate an augmented version of the image $\widehat{\mathbf{i}}_\mathbf{k}$. While many different types of augmentations have been proposed for images, for our setting we limit it to techniques that do not modify the underlying size or orientation of the image, but rather modify the visual aspects of the image. The derived image can thus use the annotation $\mathbf{b_k}$ of the source image without any modifications. This process can be repeated multiple times, each time with a different value of m and the corresponding sample of augmentation steps, to generate multiple augmented versions of an image.

4.3 Types of Image Transformations

The different types of image transformations that we use to generate augmented dataset are discussed below. An example of a page along with each of its augmented versions is shown in Fig. 4.1. While the example here shows the versions generated by applying each of the image transformation individually, in practice, we apply a series of augmentation steps to generate harder samples. The augmented images thus generated are more likely to match real world distortions that can be found in scholarly documents.

4.3.1 Brightness and Contrast

This step supports modifying the brightness and contrast of the original image. Since scholarly documents often contain multiple figures and tables, each with a varied range of colors, and can often be scanned, we hypothesize that models trained on images of varying brightness and contrast can be helpful.

4.3.2 Erosion

Many academic documents, especially the scanned ones, often contain eroded text, i.e., text with broken boundaries. Due to erosion, the elements lose their clarity. This transformation can allow models to better adapt to such examples.

4.3.3 Dilation

Like erosion, often times scanned documents may contain dilated text resulting from the process of scanning. Dilation happens an element expands, resulting in some objects being merged. To perform well on such cases, training on dilated versions can be helpful.

4.3.4 Borders

Many documents, when scanned, can contain borders resulting from the edges of binding. To allow object detection models to be able to identify such noise, training on border-augmented images can be helpful.

4.3.5 Downscale

Downscaling reduces the number of pixels in an image, thus reducing the sharpness of each object in the image.

4.3.6 Blur

Documents have a wide range of variance in terms of resolution. Training on blurred images can allow models to become more robust to such variance.

4.3.7 Salt and Pepper Noise

Noisy patches such as those resembling small dots of white/black colors like salt/pepper sprinkles are common in the case of scanned documents. This augmentation can be helpful to deal with such samples.

4.3.8 Random Lines

Another type of noise that is common in scanned documents is jagged lines, which are a result of the scanning process. To allow layout analysis on such documents, we include this augmentation.

4.4 Results

In this section, we discuss the experimental results obtained in our evaluation. We focus our evaluation on two aspects, as discussed below. For each setting below, we use the

ETD-OD dataset introduced in Chapter 3 as the original dataset. For each of the images in the training set, we generate 2 augmented versions, by applying up to 3 augmentation functions per augmented image. The number and type of augmentation functions is sampled individually for each generated image.

4.4.1 Models

We use the following models for our experimental evaluation. Each setting uses YOLOv7 [40] as the object detection model, as this was the best performing object detection model on ETD-OD, as discussed in Chapter 3.

- **YOLOv7$_{base}$:** This is the version of YOLOv7 trained on the original object detection dataset. This model serves as the baseline model that has been trained without using any augmented data.
- **YOLOv7$_{aug}$:** This is the version of YOLOv7 trained on the original object detection dataset, along with the derived data consisting of 2 augmented versions per image. Due to the inclusion of the augmented dataset in training, the training dataset size becomes 3× the dataset used in YOLOv7$_{base}$. This is the model being evaluated for augmentation-based testing.

4.4.2 Layout Detection of Digital ETDs

In this experiment, we want to determine if co-training on augmented images derived from digital ETDs along with original images can improve the performance of layout analysis on digital ETDs. Hence, we use the test split of ETD-OD as the evaluation dataset. We evaluate the performance each of the two models, i.e., YOLOv7$_{base}$ and YOLOv7$_{aug}$. These results are shown in Table 4.1.

Model	mAP@0.5	mAP@0.5-0.95
YOLOv7$_{base}$	85.3	52.7
YOLOv7$_{aug}$	85.7	53.6

Table 4.1: mAP scores of two different versions of YOLOv7 on test set consisting of digital ETDs.

4.4.3 Layout Detection of Scanned ETDs

In this experiment, we evaluate if the augmentation-based training can be helpful in the layout analysis of scanned ETDs. Hence, we use the test split of the scanned images from ETD-ODv2 (introduced in Chapter 5) as the evaluation dataset. The result for each of the two models is shown in Table 4.2.

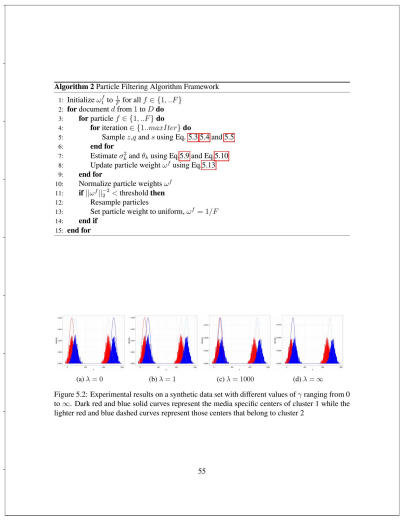
Model	mAP@0.5	mAP@0.5-0.95
YOLOv7$_{base}$	44.9	25.2
YOLOv7$_{aug}$	57.6	34.3

Table 4.2: mAP scores of two different versions of YOLOv7 on test set consisting of scanned ETDs.

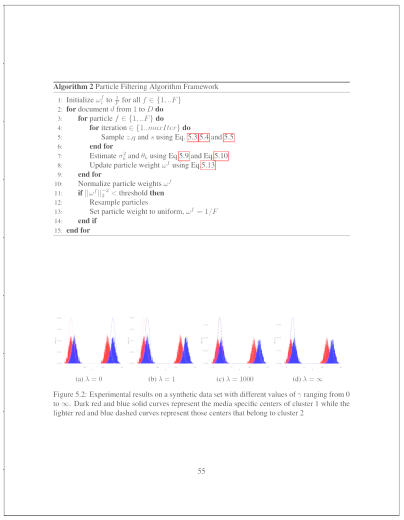
4.4.4 Analysis

Based on the results shown in Tables 4.1 and 4.2, it can be observed that YOLOv7$_{aug}$, i.e., the model trained on augmented dataset alongside original dataset, outperforms the baseline model in both of the settings. The performance improvement on digital ETDs is marginal, which can be attributed to the fact that the validation set only consists of clean page images with limited distortions. Thus, the improved prediction capability of the model does not get tested in this setting. However, there is a significant performance improvement when tested on page images from scanned documents. Since scanned documents are more likely to contain distortions, obtaining good predictive performance requires the model to be robust

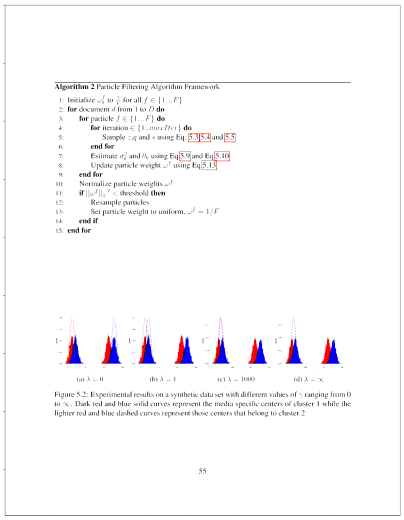
to such distortions. The model trained on augmented images is more likely to be robust to such distortions, which can be seen from the better performance of YOLOv7$_{aug}$ over the baseline model.



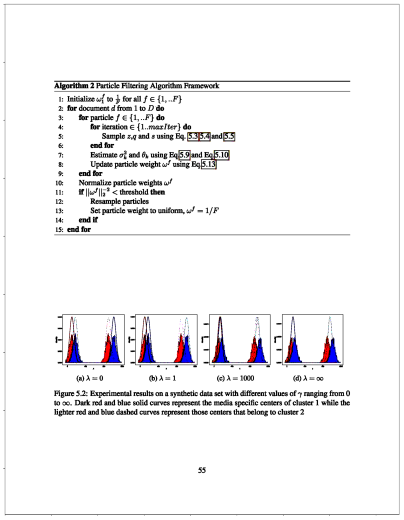
(a) Source Image.



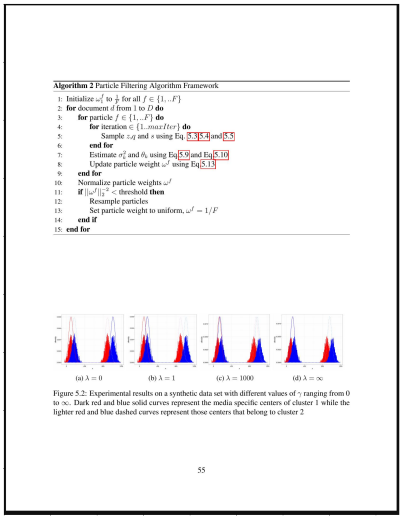
(b) Brightness/Contrast.



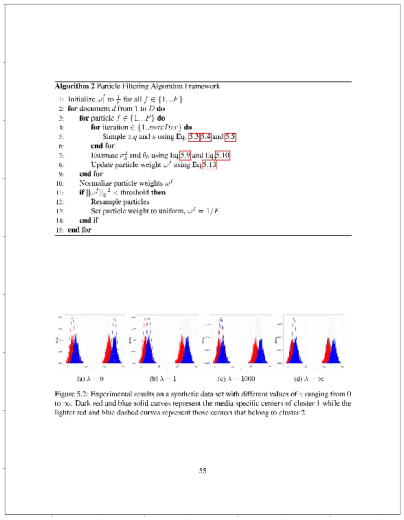
(c) Erosion.



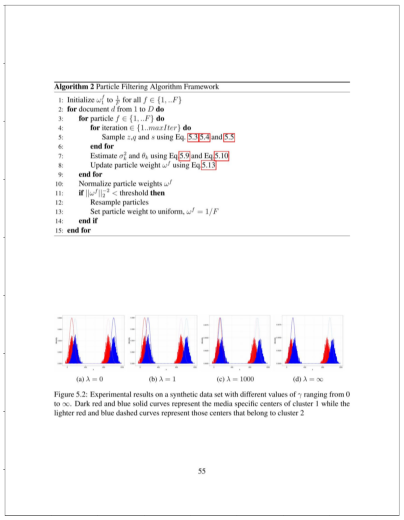
(d) Dilation.



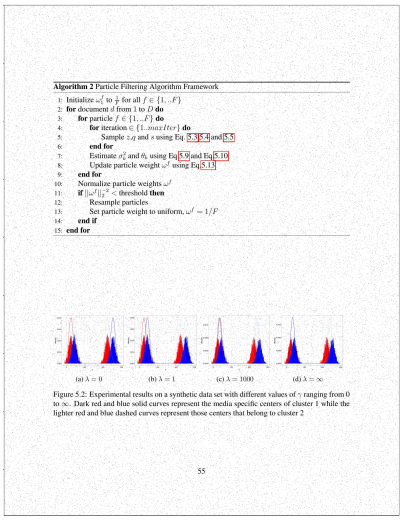
(e) Borders.



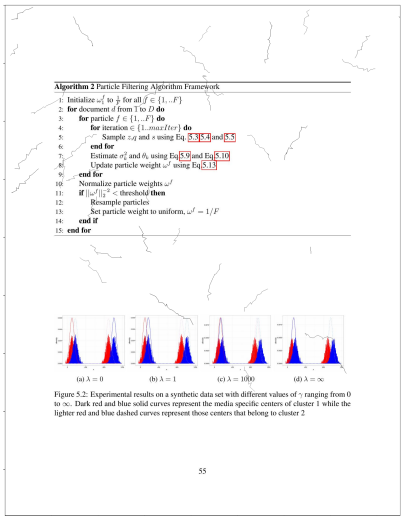
(f) Downscale.



(g) Blur.



(h) Salt and pepper.



(i) Lines.

Figure 4.1: An example of a page with its augmented versions.

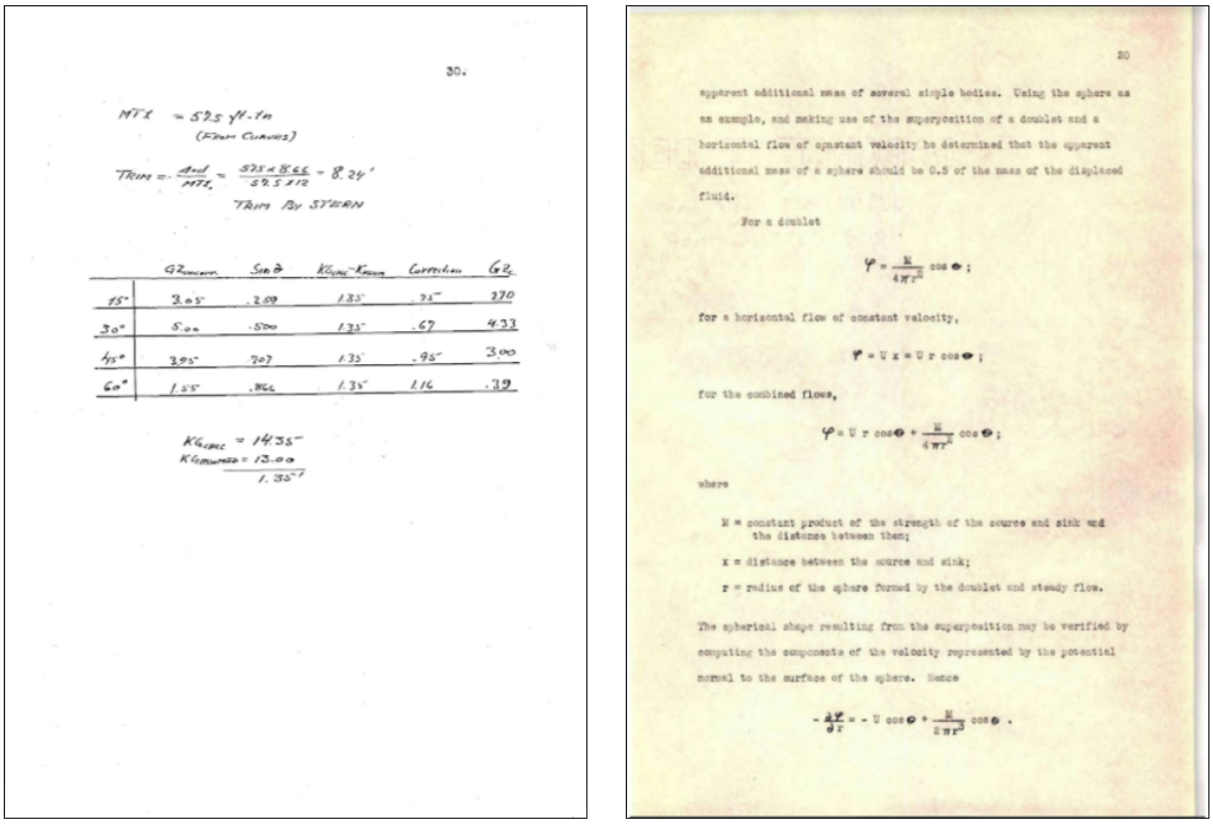
Chapter 5

AI-Aided Annotation for Developing Layout Analysis Datasets

5.1 Chapter Overview

An important aspect of object detection-based methods is that they often require a huge amount of labeled training data. For digital documents, especially those written in LaTeX, it is often possible to obtain annotations using rule-based automatic annotation methods [24]. However, in the case of scanned documents, as well as digital documents without accompanying LaTeX source code, annotating data is a cumbersome process that requires a great amount of manual effort. In the case of ETDs, many documents present in digital libraries, especially the older ones, tend to be scanned documents that were written using legacy text editing software or with a typewriter. These documents were then microfilmed and/or scanned and converted to PDF. Consequently, these documents contain a large amount of noise that was introduced during the PDF conversion process, as shown in Figure 5.1. Furthermore, given that these documents were prepared using legacy methods, they differ significantly from newer documents, such as digital ETDs, in terms of layout and structure. Additionally, some of the elements, such as metadata elements like ETD title and author name, can only be found on a few pages, while others, such as a paragraph, can be found on many pages in a document. As such, the distribution of different object categories

in the training data varies. This also affects the performance of object detection models in classes with a limited number of training instances.

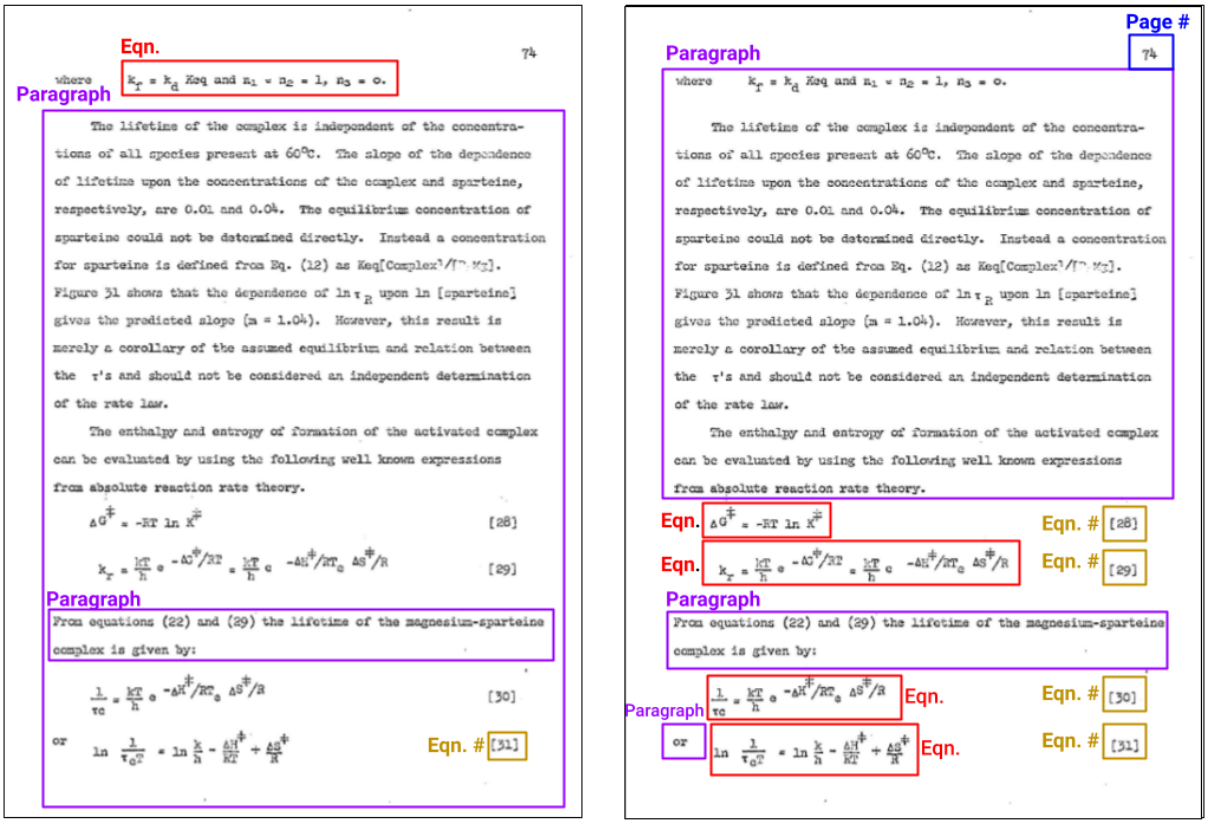


(a) Handwritten elements. (b) Noisy patches.

Figure 5.1: Examples of pages from scanned documents.

In this chapter, we propose an AI-aided annotation framework to minimize the amount of resources such as annotation time associated with developing training datasets for layout analysis. Our proposed framework utilizes the predictive capabilities of models trained on existing datasets to assist human annotators. As illustrated in Figure 5.2, although the annotations generated by the model might not all be correct, many of them are correct. Having humans only enter annotation corrections can reduce the number of instances that need to be manually labeled. This significantly speeds up the annotation process, without compromising the quality of the generated dataset. It also helps to address the problem of

class imbalance in object detection datasets, by guiding annotators to selectively label images, e.g., those that are more likely to contain elements from a predefined set. Experimental results show that our proposed annotation scheme significantly reduces the annotation time and class imbalance, thus resulting in models with improved performance across the set of object classes. We also introduce ETD-ODv2, a new dataset for object detection-based layout analysis of long documents such as theses and dissertations. ETD-ODv2 supplements the page images included in ETD-OD, adding 20K page images originating from scanned theses and dissertations. It also adds annotations for page images that are likely to con-



(a) Model generated annotations.

(b) Corrected annotations.

Figure 5.2: An illustration showing a page from a scanned document, the annotations generated by an object detection model trained on a small dataset, and the final annotations after correction by a human annotator.

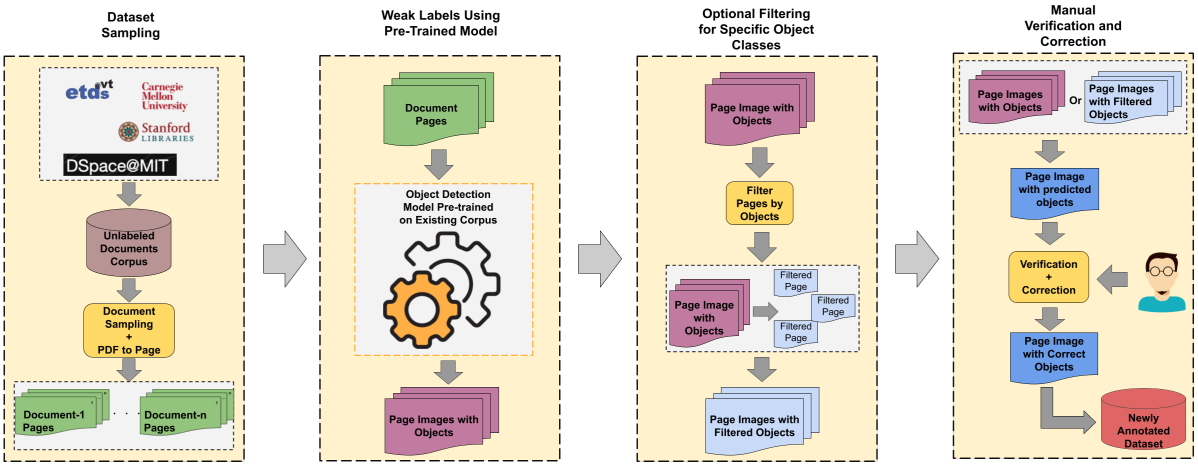


Figure 5.3: Architecture of the proposed AI-aided annotation framework.

tain low-frequency elements, such as *document title* and *algorithm*, since they can only be found on selected pages of a document, or in documents from specific domains (e.g., equations in a physics work). These pages were sourced from a large corpus consisting of both scanned and digital documents, making them helpful for mitigating the class imbalance in existing datasets as well. ETD-ODv2 thus addresses the limitations of existing datasets for ETD layout analysis, whose scope is limited to digital documents only, and suffers from a class imbalance problem. Our experimental results show that models trained on our newly annotated dataset perform much better than those trained on other datasets.

5.2 Proposed AI-aided Annotation Scheme

Due to the resource-intensive nature of the dataset annotation process, labeled data for training supervised machine learning models are always scarce. However, unlabeled data are generally available in abundance. This is also the case with document layout analysis, where getting high-quality annotations for documents and their respective pages is not easy. However, given the numerous documents that exist on the Internet and in digital libraries, many

unlabeled scholarly documents are publicly available. Although labeling document page images is a cumbersome task, we hypothesize that models trained on existing datasets can be used to assist human annotators in the labeling process, thus reducing the time required to annotate training datasets. These models can be used to generate weak labels for the huge corpus of unlabeled ETDs, which can then be filtered, validated, and corrected by human annotators. Based on this assumption, in this section, we propose an AI-aided annotation framework for developing datasets to train supervised object detection models. Figure 5.3 gives an overview of our proposed framework. The key components of this framework are discussed in detail below.

5.2.1 Dataset Sampling

We use a large corpus of unlabeled ETDs, sourced from multiple open access digital libraries. We first sample a set of documents from this unlabeled corpus that can be used for AI-aided annotation. Each of these documents is then split into page images, since object detection models require images as input.

5.2.2 Weak Labels Using Pre-Trained Model

Once we have a set of documents as well as their respective page images, they are sent to an object detection model such as YOLO [40] or Faster-RCNN [32] that has been pre-trained on an existing labeled dataset, such as ETD-OD [1]. The labels thus inferred for each image serve as weak annotations for further processing and manual verification/annotation.

5.2.3 Optional Filtering for Specific Object Classes

In some cases, such as in the case of academic documents like theses and dissertations, labeling the entire set of pages found in the sampled documents could result in a highly unbalanced dataset. In such cases, it might be desirable to use weak labels to identify images containing a pre-defined set of object categories. We refer to these object categories as objects of interest. These categories include minority classes, such as those containing very few instances in the labeled dataset, or those that have lower performance as compared to other categories. This could enable researchers to produce datasets with balanced class distributions.

5.2.4 Manual Verification and Correction

The filtered set of pages, along with their predicted bounding boxes and their respective labels, is then verified by human annotators for correctness. For page images with correctly predicted objects, no changes are made and the respective page is added to the verified dataset. For page images with incorrect predictions, whether in terms of missing or incorrect labels, the correct bounding boxes are drawn by human annotators before being added to the verified dataset.

The new dataset can then be used to fine-tune existing pre-trained models or in combination with existing datasets for model training.

5.3 ETD-ODv2 Dataset

In this section, we introduce ETD-ODv2, a new dataset for layout analysis of electronic theses and dissertations. Although existing datasets like ETD-OD [1] can be helpful in

Category Name	Description	#Digital Instances	#Scanned Instances	#AI-Aided Instances	#Total Instances
Title	Title of the document	439 (0.4%)	253 (0.4%)	2186 (1.6%)	2878 (1.0%)
Author	Name of the document author	404 (0.4%)	249 (0.4%)	2548 (1.9%)	3201 (1.1%)
Date	Date of publication, or of final research defense	324 (0.3%)	224 (0.4%)	2415 (1.8%)	2963 (1.0%)
University	University/institution of the author	340 (0.3%)	203 (0.3%)	1873 (1.4%)	2416 (0.8%)
Committee	Committee that approved the document	305 (0.3%)	83 (0.1%)	1472 (1.1%)	1860 (0.6%)
Degree	Degree (e.g., Master of Science) being earned.	281 (0.3%)	202 (0.3%)	1834 (1.3%)	2317 (0.8%)
Abstract Heading	A header that indicates the start of abstract text	169 (0.2%)	113 (0.2%)	807 (0.6%)	1089 (0.4%)
Abstract Text	The actual text of the abstract	183 (0.2%)	73 (0.1%)	952 (0.7%)	1208 (0.4%)
List of Contents Heading	A header that identifies the content of a list	512 (0.5%)	300 (0.5%)	3151 (2.3%)	3963 (1.3%)
List of Contents Text	The actual list of entries for the type of content	1059 (1.1%)	460 (0.7%)	3172 (2.3%)	4691 (1.6%)
Chapter Title	The title of the chapter	2199 (2.2%)	1926 (3.1%)	1263 (0.9%)	5388 (1.8%)
Section	The header of a section which splits a document	9337 (9.4%)	2946 (4.7%)	5196 (3.8%)	17479 (5.8%)
Paragraph	The main textual content of the document	30359 (30.4%)	17962 (28.5%)	34601 (25.2%)	82922 (27.6%)
Figure	A figure, chart, or other visual illustration	6359 (6.4%)	2977 (4.7%)	2148 (1.6%)	11484 (3.8%)
Figure Caption	The text caption that describes a figure	5722 (5.7%)	2370 (3.8%)	1564 (1.1%)	9656 (3.2%)
Table	The table element category	3145 (3.1%)	2192 (3.5%)	656 (0.5%)	5993 (2.0%)
Table Caption	The text caption that describes a table	2225 (2.2%)	1872 (3.0%)	399 (0.3%)	4496 (1.5%)
Equation	A mathematical equation/formula	5092 (5.1%)	5579 (8.8%)	27266 (19.8%)	37937 (12.6%)
Equation Number	Used to reference an equation with a number	1834 (1.8%)	3727 (5.9%)	20943 (15.2%)	26504 (8.8%)
Algorithm	An algorithm description, e.g., as pseudo-code	96 (0.1%)	224 (0.4%)	787 (0.6%)	1107 (0.4%)
Footnote	Auxiliary information at the end of content	2029 (2.0%)	2340 (3.7%)	1045 (0.8%)	5414 (1.8%)
Page Number	A number of a specific page in a document	24543 (24.6%)	15800 (25.0%)	17454 (12.7%)	57797 (19.2%)
Reference Heading	A header that indicates the start of a reference list	271 (0.3%)	189 (0.3%)	1830 (1.3%)	2290 (0.8%)
Reference Text	The actual list of reference cited in the document	2632 (2.6%)	864 (1.4%)	1839 (1.3%)	5335 (1.8%)
Total Objects		99859	63128	137401	300388
Total Images		25073	16766	20204	62043

Table 5.1: ETD-ODv2 dataset statistics.

layout extraction from digital documents, they suffer from a class imbalance problem and do not contain scanned documents.

5.3.1 Scanned Documents

There are several attributes related to scanned documents that are not found in digital documents. These include the following.

- **Noisy patches:** A common observation found in scanned documents is that a large number of pages contain noisy patches that result from the process of converting such documents into an electronically readable PDF file.
- **Low resolution:** Given that these documents are essentially images of hard-copy versions of the original document, they tend to have relatively low resolution.

- **Dilated or eroded text:** Another common observation regarding many scanned documents is that the text is eroded (i.e., has a thinner font than the original document) or dilated. This can also be attributed to the PDF conversion process.
- **Handwritten elements:** Some of the pages of scanned documents contain elements – such as tables, figures, and equations – that were written or drawn by hand and were not typed or created using software.

Due to the presence of such attributes, object detection models trained on the digital documents dataset generally do not perform well on scanned documents. Hence, our new dataset includes manually annotated page images from scanned documents, to support layout analysis on scanned documents.

5.3.2 Page Images with Minority Elements

While it is desirable to have images of pages from scanned documents, this does not prevent the dataset from being subject to a class imbalance problem. This is because some elements – such as *document title* and *author name* – typically only appear on a small set of pages in the document, such as the front page. Therefore, a dataset constructed by labeling all pages appearing in a document will always be prone to the class imbalance problem. Moreover, some element classes such as *algorithm* might only appear in documents in certain domains, such as computer science. Hence, a set of documents uniformly sampled from several different domains will have few pages with such instances. To alleviate this problem, we use the proposed AI-aided annotation method to identify/filter and annotate pages that are more likely to contain such minority elements. These page images were sourced from both digital and scanned documents. The elements that we consider to be minority elements are listed below.

- **Elements found on a limited number of pages:** Title, Author, Date, University, Committee, Degree, Abstract Text, List of Contents Heading.
- **Elements found in documents from select disciplines:** Equation, Equation Number, Algorithm, Reference Heading.

5.3.3 Dataset Source and Object Classes

To ensure compatibility with existing datasets, we use the object categories defined in ETD-OD for annotation. The documents in both subsets of our data set (i.e., the scanned and AI-aided) were sourced from a uniformly sampled set of theses and dissertations from open access institutional repositories of U.S. origin [39].

5.3.4 Dataset Statistics

Table 5.1 shows the detailed statistics of different object categories in our dataset.

Scanned Documents

The subset of scanned documents in our dataset consists of images and bounding box annotations of ∼16K pages, derived from 100 theses and dissertations. These documents were annotated by a group of five undergraduate students [49]. To ensure the correctness, each sample also went through another round of review by one of the authors of [2]. We use Roboflow[1] as the dataset annotation platform.

[1] https://roboflow.com/

Pages with low-frequency elements

Our dataset also consists of ∼20K page images from ∼1,200 documents that were annotated using our proposed AI-aided annotation framework. The pages were then filtered based on the labels listed above and reviewed and corrected as needed by a group of four annotators [11].

5.4 Experiments

In this section, we report the experimental results obtained during our evaluation. Our experiments focus on determining the improvements in terms of human resources, such as annotation time, obtained using the AI-aided annotation strategy. We also analyze whether the new dataset, consisting of scanned documents and pages with instances from lower-frequency categories, can be helpful in improving the performance of object detection models.

5.4.1 Annotation Time

Experimental Setup

To construct our proposed AI-aided annotation framework, we used the bounding box widget from the open source framework `pylabel`[2], which was integrated with a pre-trained object detection model. We trained a YOLOv7 model [40] on ETD-OD [1] and a small set of ∼2K scanned documents. We only used a small number of samples from the scanned documents dataset, as that was the only sample available at the time. The model obtained was then used in our AI-aided framework to generate the proposed labels. We will refer to this model as

[2] https://pylabel.readthedocs.io/en/latest/

YOLOv7_base in the remainder of the discussion. As noted in [1], YOLOv7 outperforms other models in the object detection task, so we use it as the detection model for empirical evaluation.

Evaluation Settings

To determine whether the proposed AI-aided annotation scheme reduces resource requirements, we compare the time required to label images under different settings.

- **No Model Assistance:** This is the classical labeling setting under which the annotators are shown neither bounding boxes nor the respective labels for page images.
- **AI-Aided-v1:** Under this setting, for each image, the annotators were shown the bounding boxes generated by the **YOLOv7_base** model.
- **AI-Aided-v2:** For this setting, we fine-tuned the **YOLOv7_base** model on a set of 10K page images labeled using our AI-aided annotation scheme. This was done to evaluate whether the assistance of a model trained on an additional new dataset affects the annotation time. We then used this model to generate bounding boxes for each image shown to the annotators.

In the two AI-aided settings, annotators were asked first to review the model-generated annotations. All correct annotations were left unchanged, and only missing, incorrect, or extra-bounding boxes were asked to be modified. For each of the three settings, each of the four annotators annotated ∼500 pages, and the time spent on annotation was recorded.

Results

In Figure 5.4, we report the average time spent per page by each of the annotators under different annotation settings. The following observations can be made:

- **Model assistance significantly reduces annotation time:** As we can observe from the graph, the average time required to annotate a page without the assistance of a model (i.e., without any proposed bounding boxes) is 2-3 times longer than for each of the AI-aided settings. This is likely because even though the models used for assisting annotators might have been trained on limited data and coverage (in terms of document types and object classes), they still possess predictive power to help with many of the elements found in pages, such as paragraphs and figures. Thus, we can conclude that the assistance of models trained on existing data significantly helps in annotating more data by reducing the time required for annotation.
- **Model assistance increases with better trained models:** Another observation that can be made from Figure 5.4 is that as we obtain models with better predictive power, the suggested labels of the model become more accurate, further reducing the time required to annotate a page. The model used for the `AI-Aided-v2` setting had been trained on 10K more samples than the one used in `AI-Aided-v1` setting. The samples used were also more balanced in terms of object classes. Therefore, it has better predictive power,

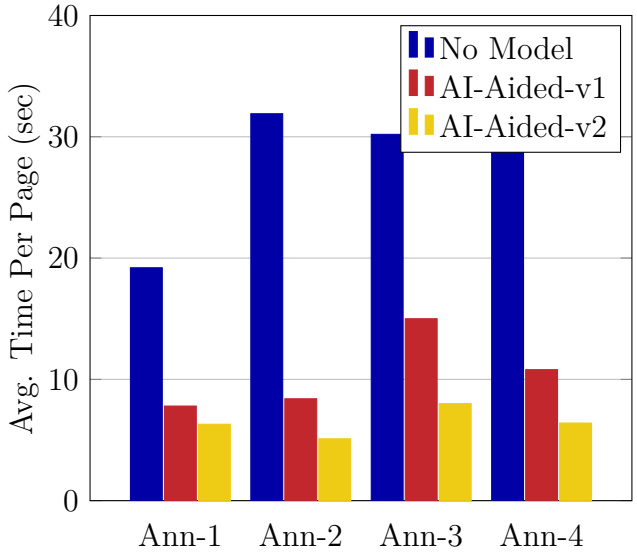


Figure 5.4: Annotation time for each annotator under different annotation settings.

enabling it to be more helpful to human annotators.

5.4.2 Object Detection Performance

In this analysis, we present our findings on how the AI-aided annotated dataset helps improve object detection performance. The specific details of this analysis are described below.

Object Detection Model

As stated above, we use YOLOv7 as the benchmark object detection model for this analysis. Since the purpose of this analysis is to determine how training on different datasets impacts model performance, the specific choice of object detection model is beyond the scope of this analysis. Moreover, previous studies have shown that YOLOv7 is the state-of-the-art model for object detection tasks [1, 40, 42].

Test Dataset

Since the AI-aided subset of our dataset was constructed with the objective of mitigating the class imbalance problem, it consists of page images from documents of several types, such as scanned and digital. Therefore, to analyze how training with the AI-aided dataset helps object detection models on various types of documents, we construct a test dataset consisting of page images sampled from ETD-OD [1], as well as the scanned and low-frequency element pages from ETD-ODv2. This is done to ensure that the test set is representative of diversity in terms of both document types and object types. The breakdown of images and objects in the test dataset is shown in Table 5.2.

Source	# Images	# Objects
Digital	3760	14319
Scanned	9353	9294
AI-Aided	3031	20718
Total Test	9353	44331

Table 5.2: Distribution of the test dataset.

Baselines

We use the versions of the dataset listed below to evaluate object detection performance. All versions used YOLOv7 as the object detection model. The number of images and objects in each version is listed in Table 5.3.

- **Digital:** This version of the model was trained only on the digital document images from ETD-OD. As such, the training dataset contained a small number of samples from the minority classes due to the class imbalance in the scanned subset.
- **Scanned:** This version of the model was trained only on the scanned subset of the ETD-ODv2 dataset. As in the previous setting, the training dataset used in this setting also has the class imbalance problem.
- **Digital + Scanned:** Under this setting, the YOLOv7 model was trained on the combined images of scanned and digital documents, that is, a merged set consisting of the two dataset splits described above.

Version	# Images	# Objects
Digital	21313	85540
Scanned	14204	53834
Digital + Scanned	35517	139374
Digital + Scanned + AI-Aided	52690	256057

Table 5.3: Statistics of different versions of the data set used for training.

- **Digital + Scanned + AI-Aided:** This setting uses the **Digital + Scanned** split described above, along with the AI-aided subset of ETD-ODv2. This setting represents a model that has been trained on diverse types of document (i.e., digital and scanned) and consists of a larger number of training instances from each object category.

	AP@0.5				AP@0.5:0.95			
Categories	**Digital**	**Scanned**	**Digital+ Scanned**	**Digital+ Scanned+ AI-Aided**	**Digital**	**Scanned**	**Digital+ Scanned**	**Digital+ Scanned+ AI-Aided**
Title	0.861	0.538	0.888	0.924	0.688	0.340	0.672	0.732
Author	0.814	0.471	0.833	0.927	0.556	0.221	0.523	0.624
Date	0.676	0.393	0.731	0.852	0.454	0.124	0.398	0.545
University	0.730	0.312	0.788	0.874	0.539	0.156	0.529	0.628
Committee	0.822	0.327	0.856	0.926	0.622	0.167	0.620	0.692
Degree	0.524	0.060	0.551	0.732	0.385	0.024	0.380	0.532
Abstract Heading	0.897	0.320	0.929	0.948	0.636	0.127	0.628	0.672
Abstract Text	0.812	0.703	0.837	0.872	0.786	0.629	0.811	0.845
List of Contents Heading	0.880	0.782	0.884	0.915	0.655	0.293	0.555	0.690
List of Contents Text	0.939	0.926	0.955	0.966	0.875	0.790	0.889	0.896
Chapter Title	0.503	0.460	0.761	0.786	0.273	0.211	0.406	0.425
Section	0.861	0.706	0.882	0.890	0.495	0.306	0.509	0.541
Paragraph	0.944	0.925	0.964	0.969	0.805	0.728	0.825	0.841
Figure	0.855	0.854	0.917	0.965	0.674	0.609	0.754	0.797
Figure Caption	0.809	0.716	0.881	0.897	0.518	0.359	0.564	0.576
Table	0.864	0.824	0.919	0.941	0.668	0.602	0.748	0.761
Table Caption	0.763	0.590	0.891	0.903	0.424	0.317	0.519	0.524
Equation	0.857	0.825	0.875	0.920	0.652	0.521	0.635	0.719
Equation Number	0.832	0.594	0.890	0.916	0.565	0.122	0.486	0.657
Algorithm	0.368	0.231	0.463	0.665	0.327	0.173	0.406	0.527
Footnote	0.697	0.854	0.881	0.950	0.488	0.574	0.638	0.687
Page Number	0.519	0.346	0.630	0.670	0.206	0.098	0.216	0.261
Reference Heading	0.836	0.612	0.808	0.871	0.631	0.238	0.561	0.655
Reference Text	0.911	0.927	0.964	0.974	0.838	0.819	0.894	0.904
Combined (mAP)	0.774	0.596	0.832	0.886	0.573	0.356	0.590	0.655

Table 5.4: Object detection performance results.

Evaluation Metrics

We use the two commonly used object detection metrics to evaluate the results of different models discussed above. Both metrics are based on the average precision (AP), which is

calculated based on the number of predicted objects that overlap with the ground-truth object over a certain threshold in terms of the area. The two metrics are described in detail below.

- **AP@0.50 / mAP@0.50:** For a given object category, AP@0.50 is the percentage of predicted bounding boxes that overlap with the true bounding boxes by more than 50% in terms of area. mAP@0.50 is the average of AP@0.50 for all object categories.
- **AP@0.50:0.95 / mAP@0.50:0.95:** This is calculated by first calculating the AP at different thresholds, from 0.50 to 0.95, with a step of 0.05. All these AP values are averaged to compute AP@0.50:0.95 for an object category. mAP@0.50:0.95 is the average of AP@0.50:0.95 for all object categories.

Results

Table 5.4 shows the results obtained on the test dataset described above in each of the training settings. Based on the results shown, the following observations can be made.

- **Performance w.r.t. document type:** The subset of images used to train the **Scanned** model had the highest amount of noise and lowest quality (e.g., blurred) as compared to the training dataset used for other models. This results in the lowest overall performance of the model.
- **Size of the training dataset:** The **Scanned** model was trained on the smallest training dataset. Consequently, it has the lowest performance among all four variants. The large size of the training dataset used in **Digital + Scanned + AI-Aided** helps achieve the best overall performance.
- **Performance on minority classes:** We also find that training on a dataset with a better distribution in terms of object classes significantly improves performance. As can

be seen from the results shown, the performance of certain categories, such as *Degree* and *Algorithm*, increased by ∼20%. This shows that model performance on certain low-performing categories can be improved by training on a larger number of samples from such categories.

- **Weak labels can be helpful signals for targeted annotation:** Another observation that can be made from the performance improvements achieved on low-frequency categories is that weak labels generated from an existing model can serve as a good indicator for more targeted annotation. Although using such labels cannot guarantee coverage, they can still address performance issues to a great extent.
- **Overall performance:** Finally, we can also observe that performance improvements are achieved in other categories that were not included in the filter set. This can be attributed to the fact that while the AI-Aided data consisted of pages filtered based on the occurrence of minority elements, these pages also contained other elements in addition to those from the filter set. This helped the model to be trained on more samples from other object categories as well, thus improving the performance across all object classes.

Chapter 6

Structured Representations of Long Scholarly Documents

6.1 Chapter Overview

In Chapters 3, 4, and 5, we discussed how object detection can be used to detect and extract important scholarly elements from long documents such as ETDs. However, the scope of these chapters was limited to extracting elements from the document pages. In reality, a long PDF document such as an ETD consists of many pages, each of which contributes to the overall organization of the document, which can be represented as a hierarchical structure. Converting the *"unordered set"* of objects extracted from layout parsing methods to a structured format which can represent the organization of information in an ETD can be very helpful to support downstream tasks such as document/figure search, chapter summarization, etc. The structured versions can also be used to support accessibility needs of those with disabilities, by means of accessibility tools such as on-screen readers. However, generating structured versions of ETDs is a non-trivial task, and involves several challenges, as discussed below:

- **Identifying delimiters:** Delimiters, such as chapter and section elements, are one of the most important components of the information structure of an ETD. The inherently long nature of ETDs makes correct identification of delimiters an important component

in ETD parsing. They are useful in segmenting the document into multiple smaller components, thus making it easier for the reader. They are also useful in downstream tasks that rely on segmented units of a long document, such as chapter summarization.

- **Linking objects:** Many object types have relationships between each other, and correct identification of such relationships can be useful in several downstream tasks. For example, linking figures and/or tables to the respective captions can be useful in figure/table search. As such, identifying such relationships is important during information extraction from scholarly documents.

In this chapter, we address the task of converting the extracted set of elements to a structured format, such as XML, so that the information in a document can be made useful for other downstream tasks. We also present a system that can allow for easy navigation of a long PDF document, using the information from the generated XML format.

6.2 XML Schema

Based on the structure of an ETD, in Schema 6.1, we present an XML schema that can be used to capture the organization of content in an ETD in a structured format. The schema is based on the following observations.

- The overall information in an ETD can broadly be encapsulated into three high-level categories. `front` consists of elements that can give key identifiable information, as well as an overall summary about the work. These include metadata elements, abstract, and lists(s) of contents, figures, and tables.
- `body` consists of elements that can give in-depth information about the content of a document. It contains a list of chapters, each of which further contains a list of sections. The sections encapsulate detailed informational elements contained therein.

- `back` consists of information that often is not critical for the understanding of a document. This includes a list of references and the appendices.

```
<etd>
  <front>
    <title>Document Title</title>
    <author>Author Name</author>
    <university>University</university>
    <degree>Degree Type</degree>
    <committee>Committee</committee>
    <date>Date or Month/Year</date>
    <abs_heading>Abstract</abs_heading>
    <abs_text>In this..</abs_text>
    <toc_heading>Table of..</toc_heading>
    <toc_text>1. Intro ...</toc_text>
  </front>
  <body>
    <chapter>
      <title>Chapter-1..</title>
      <page_no>1</page_no>
      <sections>
        <section>
          <name>1.1..</name>
          <paragraphs>
            <para>In this...</para>
            <para>Next, we...</para>
          </paragraphs>
          <figures>
            <figure>
              <path>fig_001.png</path>
              <caption>Fig.1...</caption>
            </figure>
          </figures>
          <tables>
            <table>
              <path>tab_001.png</path>
              <caption>Table.1.. </caption>
            </table>
          </tables>
```

```
        <equations>
          <equation>
            <path>eqn_001.png</path>
            <eq_no>1</eq_no>
          </equation>
        </equations>
        <algorithms>
          <algorithm>
            <path>alg_001.png</path>
          </algorithm>
        </algorithms>
        <footnotes>
          <footnote>...</footnote>
        </footnotes>
      </section>
    </sections>
  </chapter>
 </body>
 <back>
  <ref_heading>Ref..</ref_heading>
  <ref_text>..</ref_text>
 </back>
</etd>
```

Schema 6.1: XML Schema for Representing ETDs in Structured Format.

6.3 XML Generation

As discussed earlier, two challenges hinder the process of converting a PDF and its respective objects from each of the pages into the XML format shown above. We will address these challenges by observing the errors found in a uniformly sampled set of documents, and then formulating a set of rules derived based on domain expertise regarding document structure.

6.3.1 Identifying Delimiters

We discuss some of the commonly found errors in delimiters below. While the list is not exhaustive and might not cover all possible errors, it is based on a user study of a sample consisting of 25 ETDs by 2 undergraduate students from Virginia Tech's course CS 4624 (Multimedia, Hypertext, and Information Access) in Spring 2023.

- **Error:** Last line of a paragraph on a new page being detected as a chapter heading.

  Reason: Many chapter headings in ETDs appear as first line of a document, and are only a few words (less than a line) long. The last line of a paragraph resembles such chapter headings.

  Proposed Rules:

  - Chapter headings that do not start with a capital case letter are re-labeled as paragraph.
  - The last paragraph of the previous page should have its last character as an end punctuation.

- **Error:** Chapter headings in headers and footers being regarded as start of new chapters.

  Reason: Many documents contain contain headers and footers on every page, which contains the title of the current chapters. Due to similarities between such elements and the actual chapter title, such as the presence of "chapter" keyword, the model might regard them as chapter titles.

  Proposed Rules: When identifying a new instance of a `chapter` element in the parser, ensure that the title of the new chapter differs from the previous chapter.

Some other rules that are applied to chapter elements include:

- **Presence in Table of Contents:** For all the detected chapters, we check for their

existence in the table of contents. A list of all the entries from the table of contents is extracted, and then each of the elements is checked against this set of entries. The matching is done using fuzzy string matching, to make sure the chapter titles overlap with at least one table of contents entry, with similarity above a certain threshold. This threshold will be derived empirically. Additionally, since we also detect the *page number* as one of the objects, we can match the page number in the table of contents entry against the chapter title and its detected page number as a further validation step.

- **Location in the page:** Chapter titles often appear on the top of the page. Based on this observation, we can filter out all the chapter titles that occur in the first half of the page based on the y-coordinate of the tentatively detected chapter titles.

While such rules may be helpful in fixing incorrect predictions, their scope is limited to false positive predictions only. This means that the rules cannot help us identify the objects that were not detected by the object detection method. Those objects can only be identified by a better object detection model, and we leave that as future work.

6.3.2 Linking Figures and Tables with their Captions

For each of the elements types (e.g., figures and tables) that need to be linked with their captions, we first identify the order of element and their caption. Some documents may contain a caption below the figures, while others might contain captions above. The same also applies to tables. Hence, for each document, we iterate through all the detected figures and count the number of figures that have a caption above them, and the number of figures that have a caption below. Based on the maximum of the two numbers, we determine the order of figures and their captions. The same process is followed for tables to determine the

table-caption order.

Next, for each figure and table, based on the determined order, we find the nearest corresponding caption element. A special case is figures that have captions on different pages. A methodology to link such figures with their captions would be a direction for future work in this domain.

6.3.3 Linking Equations and Equation Numbers

Equation elements are linked to the nearest equation number elements based on the y-coordinate.

6.4 PDF to HTML Browser for Improved Accessibility

In addition to generating structured representations of the entire PDF using objects detected from individual page images, wedevelop a working system that allows users to view ETDs in an accessible format. The system allows users to upload the document of their preference and then view it in web-based UI. This system is built as a Flask application, which first generates the structured version of a document based on the XML format shown earlier, and then displays the document in the browser. This system offers multiple use-cases, as listed below.

6.4.1 User-friendly View of Long Documents

One of the well-known problems of ETDs is that they are inherently long documents, and navigating them is hard. Some existing studies [41] have shown that allowing users to be

able to read long PDF documents in a web-based application is helpful and can improve the readability of such documents. By allowing users to view a long ETD in a web-based application, we expect increased usage and adoption of such documents by researchers.

6.4.2 Improved Accessibility for Those with Disabilities

A common limitation of PDF documents is their limited compatibility with accessibility tools such as on-screen readers. This is crucial for users with special needs, such as those with blindness, as such users often rely on accessibility tools for access to knowledge. In recent years, tools such as PREP[1] have been developed, to allow with tagging PDFs to make them compatible with on-screen readers. However, based on our analysis, it was found that automatic tagging feature of PREP does not work well in the case of ETDs, thus limiting the usability of such documents by users with accessibility needs. On the other hand, HTML based applications can be very well integrated with on-screen readers.

6.5 System Design

Figure 6.1 shows an overview of our PDF to HTML parsing system for ETDs. The system allows users to view long PDF versions of ETDs in a user-friendly and accessible format in a web-based interface. While currently the system requires users to upload a PDF file they want to parse and view, in the future this will be merged with the integrated system for ETDs, expected to be developed by the CS5604 class in Fall 2023. This will allow offline processing, so that users can view one of the many ETDs in an institutional repository in an accessible format. The different components of the UI are explained in detail below.

[1] https://prep.continualengine.com/

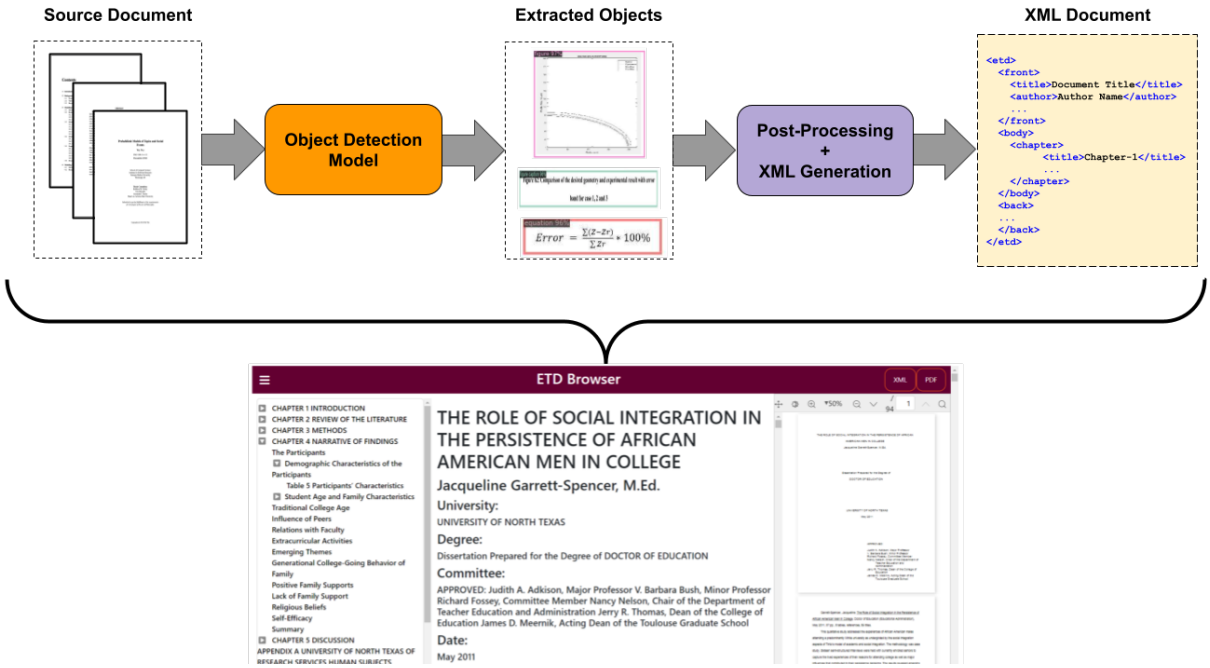


Figure 6.1: An overview of the PDF to XML to HTML system.

6.5.1 Side Bar for Navigation

ETDs are inherently long documents consisting of multiple components such as chapters. Each chapter often consists of multiple sub-components such as sections, wherein key information such as text, figures, and tables exist. To allow users to navigate through a long document, it is often desirable to have a high-level view of the document. While digital documents, such as those written in LaTeXcan support this via applications such as Acrobat Viewer, many ETDs do not support this either because they were written using a legacy tool, or are scanned documents. Hence, to help with such documents, our system allows navigation using a collapsible side bar. The side bar shows a list of chapters that were extracted from the document. Each chapter is a nested list that consists of the sections in the corresponding chapter. Some sections also contain elements such as figures and tables, which often contain important findings of a document. Hence, a third level of nesting shows

a list of tables and figures (based on the captions) for the corresponding section. Each of the entries in the sidebar have hyperlinks to the corresponding element in the main document.

6.5.2 PDF View

To allow users to keep track of the original documents, as well to support cross referencing, the original PDF document is shown in the right side bar. This sidebar can be extended in width for those who might want to have a detailed look at the PDF document. It also serves as a testing tool for the document parser, so that researchers can evaluate the quality of extracted components by directly cross-referencing them with the original document.

6.5.3 Document View

This is the main component of our document viewer, that shows the content of the document. The top part of this space shows the document metadata such as title, author name, and university. It is followed by the main content of the document. Figures, tables, equations, and algorithms are displayed as images. Each of the contents shown in this section can be cross-referenced in the original PDF being displayed in the right side bar using a click. This functionality allows users to cross-reference elements such as mathematical text, which are likely to become erroneous or confusing in the PDF-to-text extraction process.

Chapter 7

Topic Modeling based System for Analyzing and Browsing ETDs

7.1 Chapter Overview

As discussed earlier, many downstream tasks rely on NLP algorithms, which require specific elements of a long document, such as title, abstract, chapter text, etc. One such line of work that is of value in the analysis of ETDs is topic modeling, which aims to extract thematic collections of words that could represent topics, from a large corpus of text documents. The representations learned from topic models can be used for downstream tasks that rely on document representations, such as finding similar documents (document recommendation), finding similar topics, analyzing the variation of topics over time, etc.

In this work, we propose ETD-Topics, a topic modeling based framework for analyzing and discovering information contained in ETDs using several state-of-the-art topic models. Our framework allows users to extract topics present in an ETD collection using any one of the several topic models provided. Users can then select a topic of interest, and do further analysis of the topic using multiple end-user services supported in our framework. Supported services include searching documents associated with a particular topic, calculating the distribution of the documents w.r.t. topics, document recommendation, topic recommendation, and topic trend analysis based on time range and/or university. Moreover, since topic mod-

els are unsupervised in nature, our framework does not require any handcrafted labels such as categories, thereby making it easily deployable and scalable for new document collections.

7.2 System Architecture

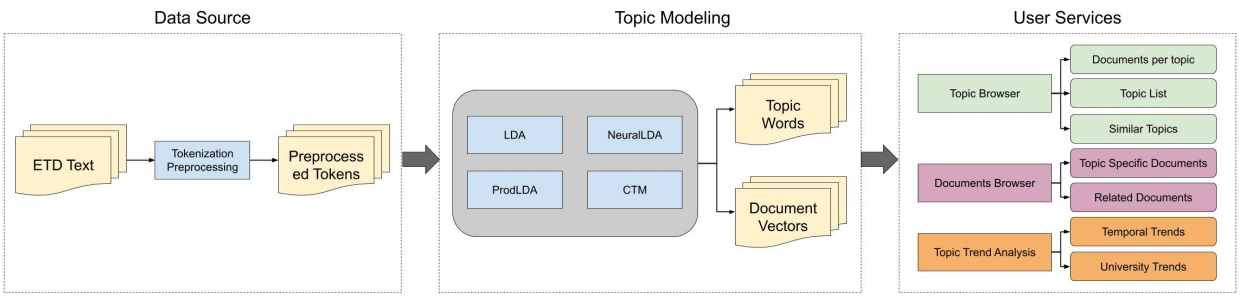


Figure 7.1: An overview of ETD-Topics.

Fig. 7.1 shows the architecture of our framework, as described below.

7.2.1 Data Source

Since our framework aims to assist in analysis of massive amounts of ETD data, we require a large collection of text ETDs. For each ETD, we use its title and abstract as the corresponding text. This text is then tokenized and goes through a series of preprocessing steps, such as stop word and punctuation removal, removing terms with low document frequency (infrequent words), and lemmatization. We also drop documents whose token count is less that a certain threshold number (20 in this case), as these are likely to be documents with limited or missing text. Finally, we obtain a list of tokens for each document that can be sent to the topic modeling module.

7.2.2 Topic Modeling

This module forms the main backbone of our system. It takes the preprocessed data as input and uses topic modeling algorithms to extract the topics from the document corpus. The topic modeling algorithms currently supported are:

- **LDA** [6]: LDA is one of the earliest topic models, that uses Bayesian priors as the initial document-topic and topic-word assignments, and then updates these distributions based on the probability with which a document or a word is associated with a certain topic.
- **NeuralLDA** [36]: This is the neural network based version of LDA, that utilizes a variational inference method for learning document-topic representations.
- **ProdLDA** [36]: This is an improved version of NeuralLDA, that is designed to give more coherent and interpretable topics.
- **CTM** [5]: In contrast to other topic models that use bag-of-words representations for text and hence ignore the order of words, this model combines representations from language models like BERT [20] in the topic modeling process, thus incorporating word context.

Since topic models require several iterations over the dataset for training, we train all the models offline, using different numbers of topics for each model. We set the number of topics (denoted as K) to $\{10, 25, 50, 100\}$ while training the models, thus resulting in 16 pre-trained models (4 models, each with a different value of K, for each of the 4 algorithms listed above).

Topic models typically give two types of outputs. The first is a $K \times V$ topic-word distribution matrix, where V is the vocabulary size. Each matrix row represents the importance of each of the words in the vocabulary for the respective topic. The second is an $M \times K$ document-

topic distribution matrix, where M is the number of documents in the corpus; each row represents the proportion of each of the topics in the respective document.

7.2.3 User Services

The front end user interface (UI) encapsulates multiple downstream tasks and services for users of a digital library. Below are descriptions of services illustrated in Fig. 7.2.

Topic Browser

Our framework allows users to select a topic modeling technique and number of topics. Users can then use the following services:

- **Documents per Topic Distribution:** This module helps users find the most popular topics in the document collection. Given a threshold value (on a scale of [0, 1], default = 0.3) and a topic, this component calculates the number of documents in the entire database for which the given topic constituted more than the threshold. The overall results are displayed as a histogram, where each bar shows the number of documents for that respective topic.
- **Topic List:** For every topic, this module shows the top 10 words that are representative of that topic; the set thus serves as a type of label. Because of the unsupervised nature of topic models, it is not possible to get a short semantically/disciplinary appropriate label for each topic. Hence, we display the top representative words for each topic.
- **Similar Topics:** Some users work in interdisciplinary fields. Often, a selected topic might not be directly related to users' preferences, but might still be correlated with the users' requirements, e.g., for researchers working in inter-disciplinary fields. In

Figure 7.2: A snapshot of different user services. (a) Documents per Topic Distribution and Topic List, (b) Similar Topics and Topic Specific Documents for one topic, (c) Document page showing Related Topics and Similar Documents for one document, (d) Trend Analysis.

such instances, it is often desirable to show a list of related topics to the user. To facilitate this process, this module shows related topics for a given topic. This is done based on similarities between different rows of the topic-word matrix.

Document Browser

The document browser allows users to get specific documents based on their interests. It mainly consists of two modules:

- **Topic Specific Documents:** This module allows users to get relevant documents for one of the many topics shown in the *Topic Browser*. It selects the documents based on the presence of the selected topic in the document using the corresponding values of the document-topic vectors. It then displays the title and abstract of the selected document. It also allows users to get more details of a specific document by clicking on it.
- **Related Documents:** This module assists users in finding documents that are similar to a selected document. This is especially useful in the case of scholarly documents, since users are typically interested in finding multiple related works.

Topic Trend Analysis

- **Temporal Analysis:** Many users of a digital library, such as university administrators and faculty members, are interested in analyzing how different research areas trend over time. This module allows users to filter documents associated with a topic in a given time range (in years).
- **University-Specific Analysis:** In some instances, users are interested in analyzing research trends in their institution, or in peer institutions. This module shows users

such research trends, by university. Additionally, users can combine this feature with temporal analysis to visualize institution-specific research trends over time.

7.3 System Setup and Analysis

The discussion in this section corresponds to what is reported in [15]. Since the study reported therein, our collection (both size and scope) and work with ETD-Topics has broadened.

7.3.1 Dataset and System Details

Our dataset has ∼320K ETDs from over 42 universities. They come from 1845 – 2020, with most published after 1945. Our topic models are from open source implementations included in OCTIS [37]. The UI was developed using Flask[1] with a Python backend.

7.3.2 Evaluation Metrics

We evaluate the different topic modeling algorithms on two commonly used metrics from the topic modeling literature. These are explained below:

- ***Diversity*** is a measure of how distinct the top words of a topic are w.r.t. top words in other topics. A score of 0 indicates redundancy, while 1 indicates very diverse topics.
- ***Coherence*** measures the degree of semantic similarity between top words from the same topic. Models with high coherence tend to give more interpretable topics.

[1] https://flask.palletsprojects.com/

	Diversity				***Coherence***			
Topics	*LDA*	*NeuralLDA*	*ProdLDA*	*CTM*	*LDA*	*NeuralLDA*	*ProdLDA*	*CTM*
10	0.75	<u>1</u>	0.96	<u>1</u>	0.044	-0.057	0.037	<u>0.104</u>
25	0.752	<u>1</u>	0.94	0.94	0.080	-0.038	0.077	<u>0.114</u>
50	0.792	<u>0.988</u>	0.92	0.948	0.076	-0.037	0.116	<u>0.136</u>
100	0.831	<u>0.937</u>	0.858	0.879	0.076	-0.039	0.117	<u>0.130</u>

Table 7.1: Quantitative comparison of different models, with underlined values indicating best performing models.

Model	***Words***
LDA	network communication user channel mobile security node wireless protocol
NeuralLDA	thesis network perform introduce efficient end describe integrate linear
ProdLDA	network problem challenge approach base provide system design framework
CTM	network protocol ad mobile node attack internet secure request

Table 7.2: Corresponding words for a topic from different models.

7.3.3 Comparison of Different Topic Models

Table 7.1 shows the performance of the four different topic models, for each of the four numbers of topics, on our collection, for the two metrics discussed above.

We observe that NeuralLDA produces more diverse topics than other models, indicated by its high diversity score, with CTM being the second best performing model in terms of diversity. However, the coherence scores for CTM are much better than other models, indicating more interpretable topics. A good topic model should ideally have high coherence and diversity scores, since high diversity and low coherence could also mean that the topics are composed of unique, yet unrelated words which do not indicate any themes. In Table 7.2 we also show the corresponding words for one topic obtained from all the models. The topic produced by NeuralLDA is less coherent, indicated by words like *thesis* and *introduce*, in line with its low coherence scores. In contrast, the topics produced by LDA and ProdLDA are cleaner and have fewer words that are semantically different than the rest of the words, though they do have some open-ended words like *user* and *provide*. CTM produces the most coherent topic,

which is also reflected by its high coherence scores. It appears that CTM is the best overall performing model on our ETD corpus.

7.4 Integrating ETD-Topics with Other End-User Services

In addition to supporting browsing and navigation by means of end-user services illustrated in Fig. 7.1, our framework can be integrated with many other APIs and end-user services that require document representations for user satisfaction. An example of such a service is a search / information retrieval system, which allows users to search for documents related to user queries.

7.4.1 Overview of Information Retrieval Systems

Many modern information retrieval systems use search engine frameworks like Apache Lucene[2] and Apache Solr[3], which can be used to search for documents that match a user query in a large document collection. Users can then obtain detailed information that best satisfies their query by clicking on one or more documents returned by the search engine. However, often the document(s) returned by the search engine do not fully satisfy users' requirements. This is especially the case in scholarly document search, where many users are interested in a wide range of documents, e.g., while doing literature surveys.

[2] https://lucene.apache.org/
[3] https://solr.apache.org/

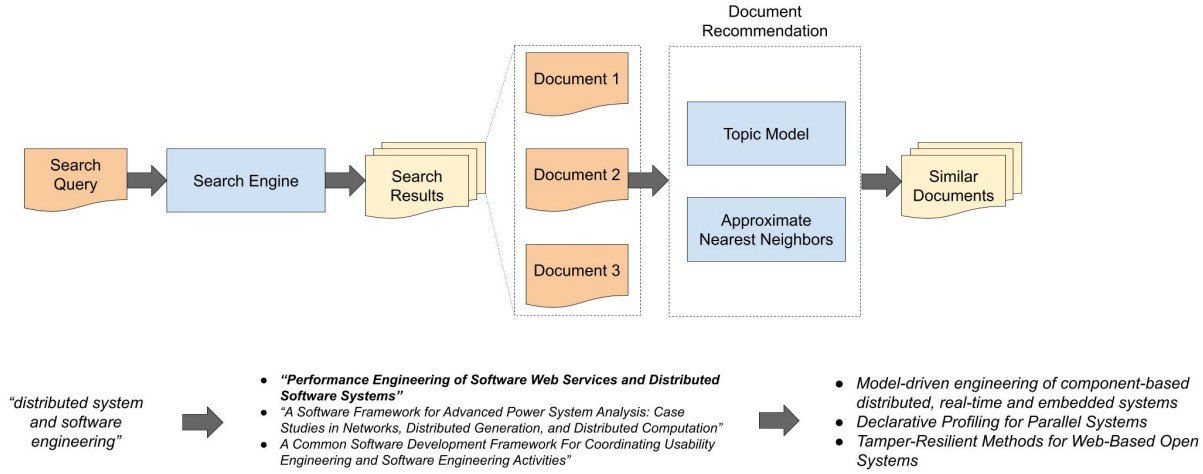


Figure 7.3: Integrating search engine module with topic models from ETD-Topics framework for document recommendation. An example of a search query, its search results returned by a BM25 based search engine, and recommended documents for one highlighted document are shown.

7.4.2 Integrating Document Recommendation with Document Retrieval

To improve user experience, a search and retrieval service is often integrated with a document recommendation service to allow users access to a wider range of possibly relevant documents. Traditional document recommendation systems primarily rely on historical user click logs. Such logs can be difficult to obtain for scholarly documents such as ETDs, since many ETD-related services are offered by university libraries which have a smaller user base as compared to commercially available services. Document recommendation services in such scenarios hence need to be supported using auxiliary information that does not rely on user logs.

Our framework supports document recommendation using the document representations learned from topic models, which can be used to find semantically similar documents. Since

the topic modeling based representations can be learned in an unsupervised way, it does not require large amounts of user logs to support such services. An overview of this integration is illustrated in Fig. 7.3. After a user sends a search query to the search engine, the search engine returns a set of documents presumed relevant to the query. A user can then select a document of interest to obtain more information about that document. This is further integrated with a document recommendation module that utilizes the topic models to first obtain the document representation, and then uses an approximate nearest neighbors technique to compute a list of similar documents. This list is then displayed to the user as recommended documents.

7.4.3 Extending Topic Modeling from Documents to Chapters

The CS 5604 class in Fall 2022 at Virginia Tech worked on an integrated system for ETDs that will support APIs for several services such as search, question-answering, chapter segmentation, chapter summarization, etc. [7]. In future, using the segmented chapters and their corresponding text for the ETD corpus, chapter-level topics could be extracted using the pre-trained topic models, by means of the framework proposed in this work. The end-user services proposed in this chapter, such as document recommendation and searching documents by topics, can then be supported at the level of chapters to support chapter recommendation and searching chapters by topics. This remains a work for the future.

7.5 Further Evaluation

Further evaluation of some of the components proposed in this chapter, such as evaluating the quality of recommended documents, requires user studies. The system developed by the

CS 5604 class is aimed at supporting user studies, and in the future can be extended and used to evaluate such components. These user studies, however, will be conducted by other graduate students and their results will be included as part of their research.

Chapter 8

Conclusion

8.1 Conclusion

This dissertation aims to address the needs of digital library users by developing datasets, techniques, and systems for analyzing and navigating long documents, such as ETDs. Since end-user services in a digital library often rely on NLP models that require data in a machine-friendly format, a significant part of this research aims to address the problem of document parsing, by means of object detection based layout analysis methods. As a use-case for the extracted data and to address the problem of limited training data for supporting end-user services such as document browsing and recommendation, we also present a topic-modeling based framework for ETDs.

In summary, the contributions of this research are as follows.

- We develop datasets for training object detection based layout analysis methods for long scholarly documents. These datasets cover a range of document types, such as born-digital and scanned documents. They could also be useful for layout analysis on other types of documents, such as books and patents, due to an overlapping set of object types such as figures, tables, and paragraphs. Hence, we expect these datasets to be a valuable resource for the document understanding community.
- We develop methodologies for document parsing and information extraction from long

scholarly documents. We hope that they will be helpful in making long documents more accessible and reader-friendly, by supporting other end-user services such as document search and retrieval, question-answering, and long document summarization. They will also allow easy testing of document understanding methods, and we expect them to be a valuable resource for the wider research community.

- To support the needs of digital library users by means of end-user services, we also propose a topic modeling framework for document browsing and recommendation. The unsupervised nature of topic modeling addresses the problem of a unified classification ontology as well as lack of labeled data by research topics.

8.2 Summary of Hypotheses

In this section, we give a brief summary of the hypotheses listed in Chapter 1, and the results obtained in their evaluation. For hypotheses that remain to be evaluated, an evaluation plan is summarized.

- **H1:** Object detection based document layout analysis methods for long scholarly documents, trained on high quality domain-specific labeled data, perform better than those trained on a larger dataset originating from other related domains, such as research papers.

  Status: True.

  Explanation: As shown in Table 3.4, models trained on a smaller dataset of objects originating from ETDs perform better than those on trained on a larger dataset of objects from research papers, like DocBank.
- **H2:** Pre-training on other scholarly datasets, albeit from a different domain such as research papers, improves the performance of document layout analysis methods on

long scholarly documents such as ETDs.

Status: True.

Explanation: As shown in Table 3.2, models like Faster-RCNN* that are pre-trained on other scholarly datasets and then fine-tuned on ETD dataset perform better than those that were not pre-trained.

- **H3:** Training on derived datasets, such as augmented versions of the original training data, can significantly improve the performance of layout analysis models.

  Status: True.

  Explanation: As shown in Table 4.1, training on a dataset obtained by augmenting images in the original dataset improves the object detection performance.

- **H4:** To perform well on other document types, such as scanned documents, object detection models trained on a specific type of documents, such as born-digital ones, require additional training using techniques, like augmentation, that help bridge the distribution gap.

  Status: True

  Explanation: The results shown in Table 4.2 show that augmentation-based training results in significant performance improvement for layout analysis of scanned documents.

- **H5:** AI-aided annotation methods, such as using models trained on existing smaller datasets to extract weak labels for unlabeled data, reduce the resources required for annotating additional data.

  Status: True.

  Explanation: The comparison of annotation time for manual annotation vs. AI-aided annotation shown in Figure 5.4 shows that model assistanvce significantly reduces annotation time.

- **H6:** Models trained on datasets with skewed distribution in terms of class labels

achieve better performance on minority classes when trained on additional data from those classes, such as from AI-aided annotation methods.

Status: True.

Explanation: The mAP values of models fine-tuned on the dataset resulting out of AI-aided annotation are higher than those of the initial models (i.e., the model without fine-tuning on the new dataset), as shown in Table 5.4.

- **H7:** Combining the predictive power of AI models with rules formulated based on domain expertise possessed by humans reduces errors in predictive tasks such as document structure parsing.

  Status: True.

  Explanation: A case study was done on a small sample of ETDs by a team from CS4624 class of Spring 2023. Some of the common errors, as well as rules to remediate them are discussed in Section 6.3.1. Based on the finding of aforementioned study, it was determined that post-processing rules are essential for document parsing.

- **H8:** Neural topic models can outperform other traditional topic models, such as LDA, while doing topic modeling on scholarly documents such as ETDs and their chapters.

  Status: True.

  Explanation: The results shown in Tables 7.1 and 7.2 show that neural topic models like CTM perform better than LDA.

Bibliography

[1] Aman Ahuja, Alan Devera, and Edward Alan Fox. Parsing electronic theses and dissertations using object detection. In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, pages 121–130. Association for Computational Linguistics, November 2022. URL https://aclanthology.org/2022.wiesp-1.14.

[2] Aman Ahuja, Kevin Dinh, Brian Dinh, William A Ingram, and Edward Fox. A new annotation method and dataset for layout analysis of long documents. In *Companion Proceedings of the ACM Web Conference 2023*, pages 834–842, 2023.

[3] Dan Anitei, Joan Andreu Sánchez, José Manuel Fuentes, Roberto Paredes, and José Miguel Benedí. ICDAR 2021 Competition on Mathematical Formula Detection. In *International Conference on Document Analysis and Recognition*, pages 783–795. Springer, 2021.

[4] Apostolos Antonacopoulos, David Bridson, Christos Papadopoulos, and Stefan Pletschacher. A Realistic Dataset for Performance Evaluation of Document Layout Analysis. In *2009 10th International Conference on Document Analysis and Recognition*, pages 296–300. IEEE, 2009.

[5] Federico Bianchi, Silvia Terragni, and Dirk Hovy. Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 759–766, 2021.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of machine learning research*, 3(Jan):993–1022, 2003.

[7] Satvik Chekuri, Prashant Chandrasekar, Bipasha Banerjee, Sung Hee Park, Nila Masrourisaadat, Aman Ahuja, William A. Ingram, and Edward Alan Fox. Parsing electronic theses and dissertations using object detection. In *Proceedings of the 23rd ACM/IEEE-CS Joint Conference on Digital Libraries*. ACM, June 2023.

[8] Muntabir Hasan Choudhury, Jian Wu, William A Ingram, and Edward A Fox. A Heuristic Baseline Method for Metadata Extraction from Scanned Electronic Theses and Dissertations. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 515–516, 2020.

[9] Muntabir Hasan Choudhury, Himarsha R Jayanetti, Jian Wu, William A Ingram, and Edward A Fox. Automatic Metadata Extraction Incorporating Visual Features from Scanned Electronic Theses and Dissertations. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 230–233. IEEE, 2021.

[10] Ran Ding, Ramesh Nallapati, and Bing Xiang. Coherence-Aware Neural Topic Modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 830–836, 2018.

[11] Kevin Dinh, Brian Dinh, Andrew Leavitt, and Annie Tran. Object Detection, 2022. URL http://hdl.handle.net/10919/114082. Virginia Tech CS4624 team term project.

[12] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[13] Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[14] Jaekyu Ha, R.M. Haralick, and I.T. Phillips. Recursive X-Y cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 952–955 vol.2, 1995. doi: 10.1109/ICDAR.1995.602059.

[15] Chongyu He, Jianchi Wei, and Chenyu Mao. Textmining. 2022. URL http://hdl.handle.net/10919/109986. Virginia Tech CS4624 team term project.

[16] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*, pages 50–57. ACM, 1999.

[17] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. *arXiv preprint arXiv:2204.08387*, 2022.

[18] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Kalen Michael, Jiacong Fang, Imyhxy, , Lorna, Colin Wong, (Zeng Yifu), Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Tkianai, YxNONG, Piotr Skalski, Adam Hogan, Max Strobel, Mrinal Jain, Lorenzo Mammana, and Xylieong. ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations, 2022. URL https://zenodo.org/record/7002879.

[19] Sampanna Yashwant Kahu, William A. Ingram, Edward A. Fox, and Jian Wu. ScanBank: A Benchmark Dataset for Figure Extraction from Scanned Electronic Theses and Dissertations. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 180–191. IEEE Computer Society, 2021.

[20] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[21] Diederik P Kingma and Max Welling. Auto-encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, 20164.

[22] Frank Lebourgeois, Zbigniew Bublinski, and Hubert Emptoz. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. In *11th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems*, volume 1, pages 272–273. IEEE Computer Society, 1992.

[23] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. TableBank: A Benchmark Dataset for Table Detection and Recognition. In *Proceedings of The 12th language resources and evaluation conference*, pages 1918–1925, 2020.

[24] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. DocBank: A Benchmark Dataset for Document Layout Analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 949–960, 2020.

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

[26] Patrice Lopez and et al. GROBID. https://github.com/kermitt2/grobid, 2008–2022.

[27] Yishu Miao, Lei Yu, and Phil Blunsom. Neural Variational Inference for Text Processing. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd*

International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, pages 1727–1736, New York, New York, USA, 20–22 Jun 2016. PMLR.

[28] Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering Discrete Latent Topics with Neural Variational Inference. In *International Conference on Machine Learning*, pages 2410–2419. PMLR, 2017.

[29] Aniket Prabhune and Edward A Fox. XML for ETDs. Technical report, Department of Computer Science, Virginia Polytechnic Institute & State University, 2002.

[30] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[31] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in neural information processing systems*, 28, 2015.

[33] Lamia Salsabil, Jian Wu, Muntabir Hasan Choudhury, William A Ingram, Edward A Fox, Sarah M Rajtmajer, and C Lee Giles. A Study of Computational Reproducibility using URLs Linking to Open Access Datasets and Software. In *Companion Proceedings of the Web Conference 2022*, pages 784–788, 2022.

[34] Zejiang Shen, Kaixuan Zhang, and Melissa Dell. A large dataset of historical Japanese documents with complex layouts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 548–549, 2020.

[35] Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. LayoutParser: A unified toolkit for deep learning based document image analysis. In *International Conference on Document Analysis and Recognition*, pages 131–146. Springer, 2021.

[36] Akash Srivastava and Charles Sutton. Autoencoding Variational Inference for Topic Models. In *5th International Conference on Learning Representations*, 2017.

[37] Silvia Terragni, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano, and Antonio Candelieri. OCTIS: Comparing and Optimizing Topic models is Simple! In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 263–270, 2021.

[38] Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. CERMINE: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 18 (4):317–335, 2015.

[39] Sami Uddin, Bipasha Banerjee, Jian Wu, William A Ingram, and Edward A Fox. Building A Large Collection of Multi-domain Electronic Theses and Dissertations. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 6043–6045. IEEE, 2021.

[40] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[41] Lucy Lu Wang, Isabel Cachola, Jonathan Bragg, Evie Yu-Yen Cheng, Chelsea Haupt, Matt Latzke, Bailey Kuehl, Madeleine N van Zuylen, Linda Wagner, and Daniel Weld. SciA11y: Converting Scientific Papers to Accessible HTML. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–4, 2021.

[42] Papers with Code. Real-Time Object Detection on COCO. https://paperswithcode.com/sota/real-time-object-detection-on-coco, 2022.

[43] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[44] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[45] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, 2021.

[46] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

[47] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. PubLayNet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, 2019.

[48] Yichao Zhou, James B. Wendt, Navneet Potti, Jing Xie, and Sandeep Tata. Radically lower data-labeling costs for visually rich document extraction models. *CoRR*, abs/2210.16391, 2022. doi: 10.48550/arXiv.2210.16391.

[49] Kecheng Zhu, Zachary Gager, Shelby Neal, Jiangyue Li, and You Peng. Object Detection, 2022. URL http://hdl.handle.net/10919/109979. Virginia Tech CS4624 team term project.