Deep integrative information extraction from scientific literature

by

Huichen Yang

B.S., Kansas State University, 2015

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

This dissertation presents deep integrative methods from both visual and textual perspectives to address the challenges of extracting information from documents, particularly scientific literature. The number of publications in the academic literature has soared. Published literature includes large amounts of valuable information that can help scientists and researchers develop new directions in their fields of interest. Moreover, this information can be used in many applications, among them scholar search engines, relevant paper recommendations, and citation analysis. However, the increased production of scientific literature makes the process of literature review laborious and time-consuming, especially when large amounts of data are stored in heterogeneous unstructured formats, both numerical and image-based text, both of which are challenging to read and analyze. Thus, the ability to automatically extract information from the scientific literature is necessary.

In this dissertation, we present integrative information extraction from scientific literature using deep learning approaches. We first investigated a vision-based approach for understanding layout and extracting metadata from scanned scientific literature images. We tried convolutional neural network and transformer-based approaches to document layout. Furthermore, for vision-based metadata information extraction, we proposed a trainable recurrent convolutional neural network that integrated scientific document layout detection and character recognition to extract metadata information from the scientific literature. In doing so, we addressed the problem of existing methods that cannot combine the techniques of layout extraction and text recognition efficiently because different publishers use different formats to present information. This framework requires no additional text features added into the network during the training process and will generate text content and appropriate labels of major sections of scientific documents.

We then extracted key-information from unstructured texts in the scientific literature using technologies based on Natural Language Processing (NLP). Key-information could include the named entity and the relationship between pairs of entities in the scientific literature. This information can help provide researchers with key insights into the scientific literature. We proposed the attention-based deep learning method to extract key-information with limited annotated data sets. This method enhances contextualized word representations using pre-trained language models like a Bidirectional Encoder Representations from Transformers (BERT) that, unlike conventional machine learning approaches, does not require hand-crafted features or training with massive data. The dissertation concludes by identifying additional challenges and future work in extracting information from the scientific literature.

Deep integrative information extraction from scientific literature

by

Huichen Yang

B.S., Kansas State University, 2015

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Major Professor
William H. Hsu

Copyright

Abstract

This dissertation presents deep integrative methods from both visual and textual perspectives to address the challenges of extracting information from documents, particularly scientific literature. The number of publications in the academic literature has soared. Published literature includes large amounts of valuable information that can help scientists and researchers develop new directions in their fields of interest. Moreover, this information can be used in many applications, among them scholar search engines, relevant paper recommendations, and citation analysis. However, the increased production of scientific literature makes the process of literature review laborious and time-consuming, especially when large amounts of data are stored in heterogeneous unstructured formats, both numerical and image-based text, both of which are challenging to read and analyze. Thus, the ability to automatically extract information from the scientific literature is necessary.

In this dissertation, we present integrative information extraction from scientific literature using deep learning approaches. We first investigated a vision-based approach for understanding layout and extracting metadata from scanned scientific literature images. We tried convolutional neural network and transformer-based approaches to document layout. Furthermore, for vision-based metadata information extraction, we proposed a trainable recurrent convolutional neural network that integrated scientific document layout detection and character recognition to extract metadata information from the scientific literature. In doing so, we addressed the problem of existing methods that cannot combine the techniques of layout extraction and text recognition efficiently because different publishers use different formats to present information. This framework requires no additional text features added into the network during the training process and will generate text content and appropriate labels of major sections of scientific documents.

We then extracted key-information from unstructured texts in the scientific literature using technologies based on Natural Language Processing (NLP). Key-information could include the named entity and the relationship between pairs of entities in the scientific literature. This information can help provide researchers with key insights into the scientific literature. We proposed the attention-based deep learning method to extract key-information with limited annotated data sets. This method enhances contextualized word representations using pre-trained language models like a Bidirectional Encoder Representations from Transformers (BERT) that, unlike conventional machine learning approaches, does not require hand-crafted features or training with massive data. The dissertation concludes by identifying additional challenges and future work in extracting information from the scientific literature.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank all of the people who have helped, supported, and encouraged me during my doctoral program at Kansas State University.

First, I thank my advisor Dr. William Hsu. He is a knowledgeable, kind, enthusiastic, and patient supervisor, mentor, and role model. He provided many insightful ideas that have enlightened me. He gave me clear guidance and sufficient freedom to develop research in my areas of interest. I appreciate his support and the research platform (The Laboratory for Knowledge Discovery in Databases (KDD)) that Dr. Hsu provided. Because of his generosity, I have developed my academic skills as well as other primary soft skills like team leadership, communication, and cooperation. In addition, I want to thank all my peers in the KDD group. It has been my honor to be a part of the group.

I would like to thank Dr. Daniel Andresen who patiently helped me as I worked on my research on the High-Performance Computing (HPC) project.

I also would like to thank Dr. Mitchell Neilsen, Dr. Pascal Hitzler, and Dr. Sanjoy Das who served as my committee members and provided valuable suggestions on my dissertation.

I also want to give my special thanks to Nora Ransom, who helped me revise my manuscripts.

Finally, I could not have reached my goals without the unconditional love, support, and encouragement from my parents and my wife, Tian Liu.

Dedication

To my lovely family for their support and love.

Chapter 1

Introduction

Scientific literature is a pathway for researchers to exchange ideas and communicate the results of scientific research.The research community has expanded dramatically, and the growth rate of published scientific reports across scientific fields has simultaneously increased. The scientific literature has become overloaded[2]. According to a report from US National Science Foundation, peer-reviewed science and engineering journal articles and conference papers increased about 4% annually over the last 10 years, from 1.8 million to 2.6 million publications from 2008 to 2018[3]. The actual number would be much larger if it included the non-peer-reviewed open-access archive services like arXiv and medRxiv. The numbers of published scientific articles is huge and the content rich. However, researchers must spend too much time not just finding the relevant research repositories, but reading published articles to gain insight from previous research and developing new ideas or methodologies for their own research. The whole process is immensely laborious, and searching such an enormous repository of information is far beyond any human ability. Just as an example, researchers in materials science must begin designing a new experiment by first extracting the results of past research. Just manually tracking all the latest scientific literature in countless journals or conferences is a time-consuming process. Even though some scientific publishers, such as IEEE, Elsevier, and Springer, provide abstracted information from published scientific articles for researchers to use, too much literature has not been included. Therefore, an

intelligent method of automatically identifying and extracting information from the scientific literature is necessary.

Information from scientific literature can be considered metadata. Metadata refers to data about data. It describes basic information about data and categorizes the data, which then can be easily retrieved. According to NISO[4], scientific documents have two types of metadata: descriptive information and structural information. Descriptive information refers to textual information that presents the content of scientific articles. It describes the purposes of the scientific article, the problem addressed in the article, the methodology, and the results. The structural information describes the layout of the article like the title, affiliation information, and keywords. The normal extracted structural information of scientific documents is in semi-structured format. The format helps readers determine how the paper is structured; it is the primary information source for downstream tasks like relevant article recommendation and citation analysis. Extracting information from scientific literature must consider both types of metadata. Traditional information extraction (IE) is a fundamental task of Natural Language Processing (NLP); IE automatically extracts machine-readable structured information from unstructured or semi-structured text data sources. IE techniques have been actively developed since 1980 and applied to many shared tasks through a series of conferences like the Message Understanding Conferences (MUC)[5], Conference on Computational Natural Language Learning (CoNLL), Automatic Content Extraction (ACE), and Text Analysis Conference (TAC). A set of sub-tasks can be completed with IE techniques, such as named entity extraction, co-reference resolution, relation extraction, and event extraction, but IE tasks focus primarily on descriptive (i.e., textual) information, not on structural information of scientific documents. Extracting structural information requires additional tasks.

1.1 Motivation

Based on the challenges discussed in the previous section, we considered how we could automatically extract potential information from such huge scientific document resources.

Extracted information is not just text but also the structural information of scientific documents. Therefore, we wanted to extract information from two type of data sources: text and document image. Solving these two challenges was motivated as explained in the following.

Unlike the simplest text information, structural information provides the layout of a scientific document in two parts: visual information and semantic information[6]. Visual information describes the document structure and identifies the boundaries of similar regions. Semantic information labels the detected regions of a document: the title, figures, and tables, for instance. This structural information can be broadly used to automatically build up large corpora for training machine learning models and extracting key insights from scientific literature. The existing tools, such as Optical Character Recognition (OCR)[7;8], extract text from scanned document images, but they cannot capture the structural information. Moreover, a lack of semantic information will lead to messy results. For example, all text will be generated together with no boundary information showing distinct zones. Such a mass of information requires extra work for cleaning. Meanwhile, the vast body of scientific literature is released by various publishers who have diverse preferences in formatting and layout for their articles. Therefore, finding a robust method to extract structural information from scientific documents remains elusive.

The other motivation comes from extracting information from unstructured scientific text, also known as key-information extraction. This kind of extraction aims to extract valuable information from scientific articles like chemical entities, methodologies for experiments, and materials properties. This valuable information can help researchers gain insight from large literature pools, speed up the discovery of new material, and construct a domain-specific knowledge graph. However, different fields of science include specific terminologies and heterogeneous datasets like experiment descriptions and materials from across scientific fields[9], heterogeneous concepts from clinical electronic medical records (EMRs) data[10;11], and other such information, which significantly differs from the general information for lay audiences. Meanwhile, even though machine learning methods, especially deep learning, can be efficiently applied to large data sources, the models still require expensive labeled data to train. The existing labeled datasets from the scientific domain are usually insufficient.

Hence, finding more effective methods of extracting key-information from scientific text is necessary.

1.2 Problem Statement

This dissertation addresses research problems of a deep integrative method that responds to issues with extracting information from scientific literature. We, therefore, considered two data sources from which information should be extracted to make searching the literature easier for scientists: scientific document images and document text. The strategies introduced are as follows:

- Developing a machine learning-based pipeline of recipe information extraction from the scientific literature;
- Using computer vision from deep learning to automatically detect document structures;
- Considering metadata information extraction as an object detection task and developing an end-to-end trainable framework for extracting visual and semantic metadata information from scientific documents;
- Using transfer learning by fine-tuning a pre-trained language model for entity extraction in the scientific domain.

1.3 Contributions

Most of the research for this dissertation has been published in conference proceedings, with the rest submitted and under review. The research contributions can be summarised as follows:

- We developed a pipeline for extracting procedural information from scientific documents with machine learning and data science technology. The pipeline was implemented as an open-source tool derived from document acquisition and filtering, payload

extraction, recipe step extraction, recipe assembly, and presentation in an information retrieval interface with question answering (QA) functionality.

- Further, we developed a novel approach for adapting convolutional neural networks for object recognition and classification for detecting the layout of scientific articles.
- We created a transformer-based framework, the first to introduce a fully transformer-based detector for uncovering document layout.
- We developed a novel, vision-based, deep learning approach for extracting metadata as both a central component of and an ancillary aid to extracting structured information from scientific literature with a variety of formats.
- We applied the attention-based deep learning approach to the task of Named Entity Recognition (NER) from synthesis procedural text of scientific literature in the materials science domain.

1.4 Dissertation Outline

The dissertation has the following sections:

- **Chapter** 2 - Background

  This chapter introduces background information with a literature review of articles covering information extraction from scientific literature, both metadata extraction and key-information extraction.

- **Chapter** 3 - Pipelines for Procedural Information Extraction from Scientific Literature

  This chapter describes a machine learning and data science pipeline for extracting structured information from documents, implemented as a suite of open-source tools and extensions to existing tools. It centers around a methodology for extracting procedural information in the form of recipes: stepwise procedures for creating an artifact (in this case synthesizing a nanomaterial) from published scientific literature.

This chapter is based on the published work[12].

- **Chapter** 4 - Document Layout Understanding

  This chapter presents two vision-based approaches for the task of detecting document layout: Recurrent Convolutional Neural Network (RCNN) and transformer-based approach.

  This chapter 4.1 is based on the published work[13] and chapter 4.2 is based on the manuscript under review by ICIP 2022.

- **Chapter** 5 - Automatic Metadata Information Extraction from Scientific Literature using Deep Neural Networks

  This chapter introduces an end-to-end trainable neural network for segmenting and labeling the main regions of scientific documents while simultaneously recognizing text from the detected regions. The proposed framework combines object detection techniques based on RCNN for scientific document layout detection with Convolutional Recurrent Neural Network (CRNN) for text recognition.

  This chapter is based on the published work[14].

- **Chapter** 6 - Named Entity Recognition from Synthesis Procedural Text in Materials Science Using an Attention-Based Approach

  This chapter proposes the attention-based deep learning approach to the task of Named Entity Recognition (NER) from synthesis procedural text of scientific literature in materials science.

  This chapter is based on published work[15].

- **Chapter** 7 - Conclusions and Future Research

  This chapter summarizes achievements and offers suggestions for future research.

Chapter 2

Background

This chapter provides background and reviews relevant research on information extraction from scientific literature with three primary aspects: document layout analysis 2.1, metadata information extraction from scientific literature 2.2, and information extraction from scientific text 2.3.

2.1 Document Layout Analysis

Document layout analysis is the task of automatically understanding, recognizing, and analyzing the regional information (e.g., text, figures, tables) and positional relationships between different layout components in the document. This is also called document layout understanding and includes analysis of the physical layouts (e.g., columns, paragraphs, text zones, tables, figures), analysis of logical layouts (e.g., titles, authors, abstracts, sections), or both. This is the key step for document understanding; to completely understand a document requires both reading the text and seeing the relationships of the parts in the layout as a human does. The document layout analysis task can be used to extract pre-defined semantic units from a document. In other words, given a document $\mathcal{D}$ comprising various regions (r_0, r_1, ..., r_n), define the semantic categories $\mathcal{C} = \{c_0, c_1, ..., c_m\}$. The goal of a document layout detector is to find a function $\mathcal{F}$: $(\mathcal{C}, \mathcal{D}) \rightarrow \mathcal{S}$, where $\mathcal{S}$ is the prediction set:

$\mathcal{S} = \{(r_0, c_0), ..., (r_n, c_m)\}$. Figure 2.1 provides an example of the scientific document layout analysis process. The given document is segmented by pre-defined layout categories through the document layout detector.

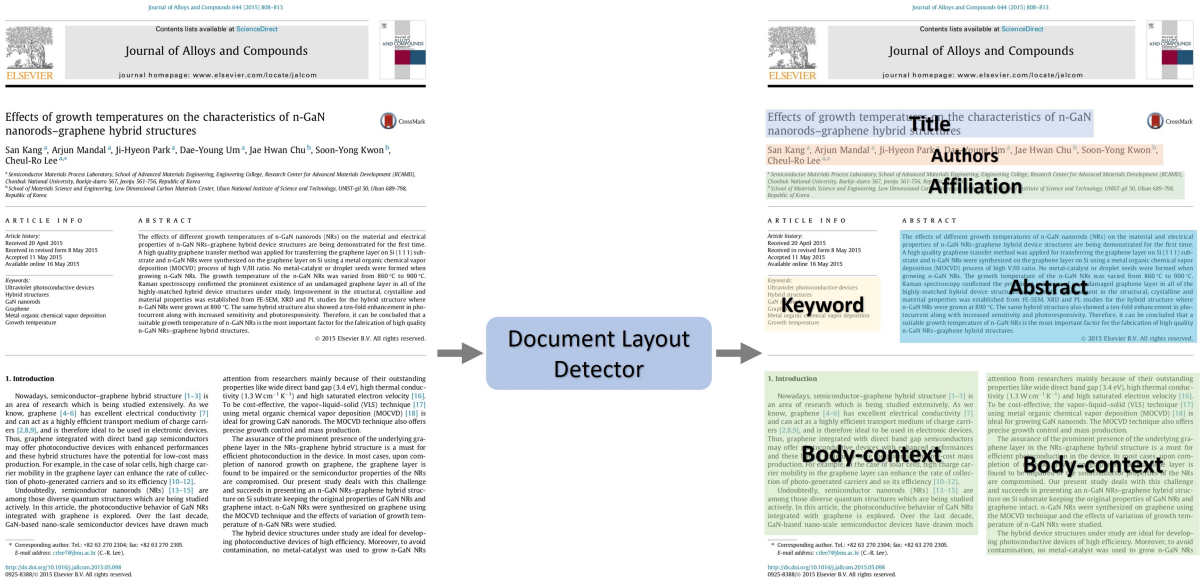


Figure 2.1: Scientific document layout analysis process.

For different types of document layout analyses, two common approaches have been proposed and researched by researchers: rule-based and machine learning-based.

Rule-based methods

The rule-based methods of document layout analysis fall into three main categories: top-down, bottom-up, and hybrid[16]. The *top-down* (model-driven) strategy is based on a document layout whose structure is known in advance. It begins with large components in document pages. Each page of the document is recursively processed and segmented from larger components into smaller sub-components. For instance, a document page with two columns is split into two blocks of text, each block is then split into several sectional blocks, each sectional block is split into text lines, and so on. The analysis process stops when there are no more regions to split, or certain conditions are met. The *Bottom-up* (data-driven) strategy, on the other hand, is based on a document layout whose structure

is unknown in advance. It begins with pixels or small connected-components of document elements. For example, characters can be be grouped into words, lines, or text zones. Then, these components are grouped and merged to create a larger homogeneous region to form document structures. The analysis process repeats until pre-defined conditions are met. The hybrid strategy integrates the top-down and bottom-up strategies. In general, the top-down approach is faster and more efficient for specific document layouts, while the bottom-up approach requires more time to analyze although it performs better for complex document layouts. The hybrid approach addresses the deficiencies of top-down or bottom-up algorithms to generate better outputs. The following section provides the common analytic algorithms for both top-down and bottom-up strategies.

The three main categories of top-down strategies are texture-based analysis, run length smearing algorithm (RLSA), and projection profile analysis.

- *Texture-based analysis* is widely used in document layout analysis with both top-down and bottom-up strategies depending on the method of implementation to segment document images based on extracted texture features into regions of interest and, using statistical methods, to classify those regions from a given set of pre-defined classes. Jain et al.[17] introduced a texture feature-based algorithm using two-dimensional Gabor filters for segmenting text and non-text regions of document images without knowing the document structure. The proposed method also works for skewed images and handwritten text. Subsequently, Jain and Zhong[18] presented a language-free page segmentation algorithm based on texture-based analysis that can recognize and categorize a grey-scale document image into three main texture regions: halftone, background, and text and line-drawing. Lee and Ryu[19] proposed a texture-based, parameter-free method for segmenting a document image beginning with maximal homogeneous regions in a document image and identifying them as different zones (texts, images, tables, and rule lines). This method used a pyramidal quadtree structure for multi-scale analysis with a top-down approach to reduce the computation of time complexity.

- *Run length smearing algorithm (RLSA)* segments blocks and discriminates text regions

from binary document images. In the binary sequence image, the pixel value 0 represents white, and 1 represents black. The 0 in binary sequence changes to 1 if the number of adjacent 0_s is less than or equal to a pre-defined threshold T. This analysis process starts from horizontal (row) and vertical (column) directions to generate two bitmaps. Then, these two bitmaps are combined with AND operation in the final step. The RLSA was first introduced by Wahl et al.[20] to segment and classify text and image regions from digitized and printed documents. Later, Wong et al.[21] used RLSA in a document analysis system. Shi and Govindaraju[22] presented an adapted RLSA for text line detection of handwritten documents. However, RLSA is very sensitive to irregular text like handwritten and skewed text[23].

- *Projection profile analysis*, or X-Y cut, is broadly used in document layout analysis as a top-down method. The basic concept of the X-Y cut algorithm is to decompose recursively a document image into rectangular blocks. The process starts from the whole document, both horizontally and vertically, to find a potential zone that is close to the content and to segment it as a block. This analysis process is repeated until no more zones can be found. Ha et al.[24] implemented the recursive X-Y cut algorithm with bounding boxes of connected components of black pixels for document page segmentation. This implementation speeds up the recursive process after the connected components are obtained. Furthermore, many modified X-Y cut algorithms have been introduced for document layout analysis[25–27]. Usually, projection profile analysis is suitable for structured document layout, such as Manhattan layout, but it cannot segment complex document layouts that include overlapped rectangular blocks.

The three main categories of bottom-up strategy are connected component analysis, Voronoi-based analysis, and Delaunay triangulation analysis.

- *Connected component analysis* starts at the pixel level and scans the document image row by row. The analysis process merges black pixels if neighboring pixels are also black; otherwise, the model assigns a new label for the white pixel. The process

repeats for the entire image, classifying regions into different layouts. O'Gorman[28] proposed Docstrum, a bottom-up based method that uses k-nearest-neighbor clustering of connected components for page layout analysis. Bukhari et al.[29] used the multi-layer perceptron (MLP) classifier to classify each connected component area as either text or non-text. In other research, Rabaev et al.[30] presented a method to detect text lines from gray scale images by analyzing the connected components generated by a sliding threshold.

- *Voronoi-based analysis* uses a Voronoi diagram to detect irregular layouts (e.g., a polygon) of documents by defining boundary points around the irregular regions. Kise et al.[31] used an approximated area Voronoi diagram to obtain the candidate boundary of the area to solve the problem of text skew. In this method, the Voronoi diagram is generated by connected components; the method is effective for areas with any angle of inclination. Lu et al.[32] implemented an algorithm that can quickly generate Voronoi diagram areas of connected components to shorten the time-consuming process of constructing a Voronoi diagram. The Voronoi-based analysis methods must calculate the space between characters and lines during document segmentation. The process is inefficient for manuscripts with wide-character spacing.

- *Delaunay triangulation analysis* uses Delaunay triangulation to segment document layouts. In mathematics, Delaunay triangulation is a dual Voronoi diagram. Xiao and Yan[33] applied Delaunay triangulation to extract triangular features from 2D space of document images, thereby distinguishing the text area. Further, Xiao and Yan[34] used a Delaunay triangulation-based method to detect the title and author regions from document images.

The hybrid method integrates the top-down and bottom-up strategies. For example, Kruatrachue et al.[35] described a hybrid approach for document segmentation that used an edge following algorithm using a small window of 16 by 32 pixels to scan a page of document before using an X-Y cut algorithm to reduce errors if the space was smaller than the window.

Lin et al.[36] presented a hybrid method using K-Means to cluster the GLCM (Grey Level Co-occurrence Matrix) features for document image segmentation.

Rule-based methods were important to document layout analysis before machine learning algorithms became popular. However, rule-based methods require knowledge of document structure and good feature selection, and they are sensitive to noise from the input document format. Such traditional knowledge-based systems are limited in generality of purpose and robustness across different cases if the rules are too domain-specific or quantitatively brittle and arbitrary.

Machine learning-based method

Machine learning methodologies have also been used for document layout analysis, and they gradually became the principal methods for document understanding. The methodologies are generally divided into non-deep learning and deep learning[6].

Non-deep learning is usually built on a conventional machine learning model that is itself either pixel-based or feature-based. For example, Marinai et al. used a pixel-based Artificial Neural Network (ANN) for document layout analysis[37], and Wei et al. introduced feature-based Support Vector machines (SVM)[38]. Usually, feature-based methods are better than pixel-based methods because they may raise missing contextual information issues[39]. Feature-based methods require feature extraction to empower training and build robust models. Features can be either handcrafted or generated automatically, as in texture features extraction[40] methods and geometric features extraction[29] methods for text-line extraction tasks.

Deep learning The main purpose of visual analysis is to detect the structure of the document and determine the boundaries of similar regions.

Deep learning methods like convolutional neural networks (convnets) have become the preeminent architecture for many pattern recognition and computer vision tasks since convnets were first used successfully to recognize handwritten characters. They were eventually adapted for object detection and recognition (particularly the PASCAL VOC and ISLVRC ImageNet challenges). Three key strengths of convnets include differentiable representation, scalable GPU computing, and large data set availability (a resource that is notably lack-

ing at present in scientific literature layout detection). Models trained using deep learning methods can address more complex document layout for both physical layout and logical layout analysis[41–43]. For example, Grüning et al.[44] presented ARU-Net deep neural network as an extension of the U-net[45] to fix the pooling issue of previous deep learning methods for text-line detection in historical documents, and Yang et al.[46] addressed an end-to-end multimodal method to extract semantic structures from document images with text features using a fully convolutional network. Deep neural networks need large data sets to learn crucial parameters for segmentation or classification tasks. Those parameters can be initialized with random weights or using transfer learning with pre-trained networks. Moreover, deep learning methods require long training, but they are more robust than traditional strategies for complex document layouts.

2.2 Metadata Extraction from Scientific Literature

Metadata is data that describes data. It eases data management and data tracking processes by capturing basic data information. Metadata is generated when data is created or updated. For instance, digital video includes metadata like authors, location at which the video was recorded, and video format. The metadata of a book, for another example, may include author information, publication date, and ISBN number. Such metadata information is increasingly critical in digital information to improve data visibility in search engines as well as for cataloging and organizing items in digital libraries.

In scientific literature, metadata is a crucial feature because it describes basic information such as titles, keywords, authors, affiliation, journal name, body text, and references that define attribute information of articles. Extracting metadata from scientific literature is very important because it enables machines to process the data and provide necessary information for downstream information extraction tasks: scientific article recommendation and citation analysis. Figure 2.2 shows an example of extracting metadata from scientific literature. Specifically, given a document $\mathcal{D}$ comprising various regions (r_0, r_1, ..., r_n), sequences (w_0, w_1, ..., w_t), and semantic categories $\mathcal{C} = \{c_0, c_1, ..., c_m\}$, the

metadata detector searches for a function $\mathcal{F}$: $(\mathcal{C}, \mathcal{D}) \rightarrow \mathcal{S}$, where $\mathcal{S}$ is the prediction set: $\mathcal{S} = \{((w_0^0, ..., w_t^0), r_0, c_0), ..., (w_0^n, ..., w_t^n), r_n, c_m)\}$.

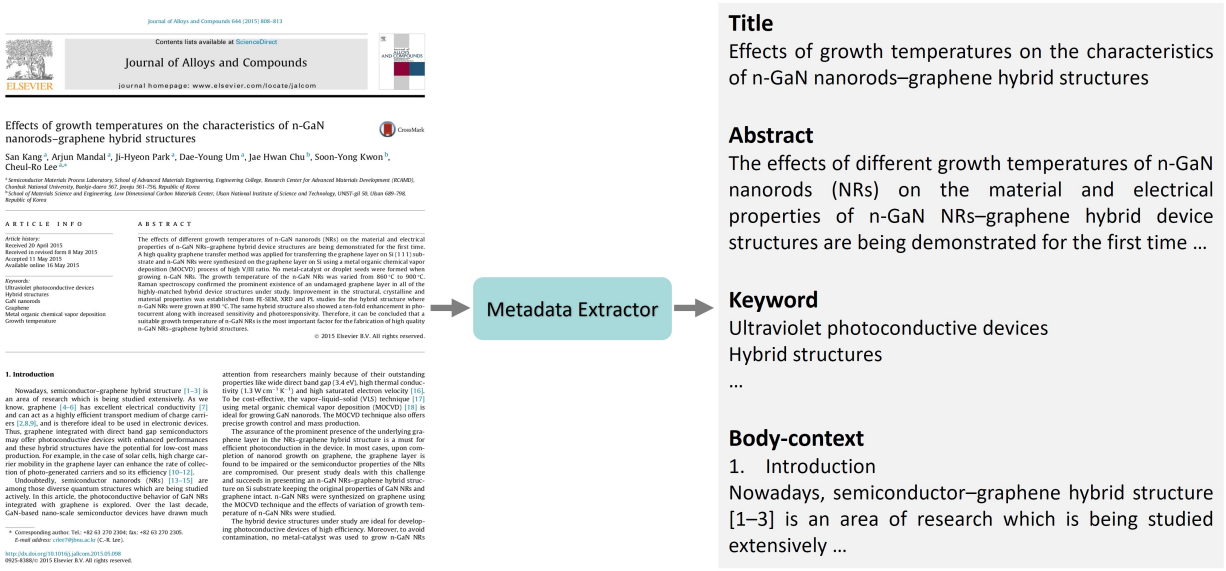


Figure 2.2: Example of metadata information extraction from scientific literature.

Several existing, but different, approaches can automatically extract metadata information from published scientific literature. These methods usually encompass both extracting metadata information from original source data, such as Microsoft Word documents or Latex source codes and either rules-based or machine learning-based extraction.

Extracting metadata information from original source data avoids the greater challenges created by the highly variable formats of publications. The original source data of scientific literature includes homogeneous document regions and the content of each region. Extraction simply focuses on target inquiries without undertaking extra document pre-processing tasks like layout analysis or content recognition. For instance, Scharpf et al.[47,48] introduced an annotation recommender system to identify mathematical formulas in documents in either Wikitext or LaTeX format in science, technology, engineering, and mathematics (also known as STEM). Swain et al.[49] developed a toolkit for extracting automated chemical information like chemical entities and measurements from multiple source data formats, usually HTML or XML in the chemical domain. These applications allow the corpus of scientific literature

to be annotated automatically, which can benefit downstream tasks in Natural Language Processing (NLP). However, these approaches require access to the source data, which might not be feasible for all published literature. Moreover, these approaches are limited to certain types of data sources.

The ***Rule-based method*** relies on hand-crafted templates and strategies specifying how to extract desired information from a document. It is usually based on text structure and document layout. For example, Flynn et al.[50] presented a template-based system to extract metadata from a large and diverse document collection. The system used a predefined template to process the converted XML file of a PDF input document to produce validated metadata through the system pipeline. Constantin et al.[51] developed PDFX, a rule-based system that extracts logical structure from scholarly literature published in PDF. This system can generate an XML file that presents an article's layout structure with such layout items as title, author, abstract, and references. To extract metadata from scientific literature, Giuffrida et al.[52] proposed a knowledge-based method. Huynh et al.[53] introduced a GATA framework, based on a predefined rule for automated metadata extraction from scientific papers. These rule-based methods are usually used for document layout analysis and require third-party tools like optical character recognition (OCR) to handle the post-processing task for text recognition. The rule-based metadata extractor system shows good performance due to a good, manually designed template, but it is sensitive to text format and diverse document layout.

The ***Machine learning-based method*** has demonstrated its usefulness for extracting metadata from scientific literature. This approach can be divided into *supervised machine learning* approaches that generally use a feature-based classification model with a labeled data set. Huy Hoang Nhat Do et al.[54] offered the Enlil system that integrated the conditional random field (CRF) to identify authors and affiliations and the support vector machine (SVM) to detect relationships between authors and their corresponding institutions. The Enlil system then extracts affiliation information from scholarly documents. Lopez[55] presented GROBID (GeneRation Of BIbliographic Data), which uses a machine learning technique to extract metadata information from raw scientific documents (like PDF) for an-

alyzing scientific text. *Unsupervised machine learning* approaches generally use clustering algorithms with a non-annotation data set. Tsai et al.[56] used an unsupervised bootstrapping algorithm to identify, categorize, and cluster scientific concepts from the literature. Moreover, deep learning approaches have been heavily researched because they can extract metadata from scientific literature from the growing numbers of publications in diverse domains. Tkaczyk et al.[57] introduced the CERMINE open-source system that combines supervised and unsupervised machine learning technologies to extract structured metadata from digital scientific articles. Deep learning relies on a massive data set to train or fine-tune parameters using pre-trained models to learn good feature representations for extracting metadata. Yang et al. considered scientific literature layout detection as an object detection problem; they used a pre-trained object detection network to fine tune detection[13]. Prasad et al. proposed Neural ParsCit[58], a way of extracting layout and bibliographic metadata from a research document with a long short-term memory (LSTM) network. Saha et al. presented graphical object detection from document images with Mask R-CNN[59].

2.3 Extracting Key-Information from Scientific Literature Text

Extracting key information finds structured and meaningful information in semi-structured or unstructured data. This is an essential step for tasks like document understanding or constructing knowledge graphs. In contrast to extracting metadata, information extraction emphasizes textual information in context and using logical reasoning to make inferences based on this extracted information. The purpose of information extraction from the scientific literature is to automatically turn the unstructured text of published scientific articles into structured information. Normally, the process can be categorized into entity extraction and relation extraction.

2.3.1 Entity Extraction

A representative task in entity extraction is named entity recognition (NER), a fundamental information extraction task that seeks to locate and classify pre-defined entities from unstructured text. Entities are typically noun phrases comprising one or a few tokens from unstructured text. The types of entities vary depending on the domain. For instance, named entities such as person, location, or organization are common to all general fields, but specific chemical terminologies and biological protein names are typically included only in science.

Given a sentence $S = \{w_1, w_2, ..., w_N\}$ where w represents the words in the sentence, NER can formalize the words to a list of tuples $\{E_s, E_e, t\}$, each of which is a named entity mentioned in sentence S. In addition, $E_s \in [1, N]$ and $E_e \in [1, N]$ are the start and end indices of a named entity, and t is the entity type of a pre-defined category set. NER approaches are usually classified as rule-based, unsupervised learning, feature-based supervised learning, or deep learning[60].

Rule-based methods

Rule-based NER systems use hand-crafted rules, lexicons[61], orthographic, and feature-engineering. They rely on lexicon resources and domain specific knowledge, as Quimbaya et al. proposed in a dictionary-based approach for NER for electronic health records[62]. The rule-based approaches work well when the lexicon is exhaustive but fail when definitions are not in the lexicon[63]. The approach, unfortunately, is not robust and cannot be used in other domains because of incomplete dictionaries and domain-specific rules.

Supervised learning

Supervised learning is also called feature-based supervised learning. The algorithm is designed for a model to learn to recognize similar patterns in unseen data using training examples from features of an annotated data set. Supervised learning for NER tasks can be divided into multi-class classification and sequence labeling. The common feature-based supervised machine learning algorithms applied to NER tasks are hidden Markov models (HMM)[64], maximum entropy models[65], support vector machines (SVM)[66], decision trees[67], and conditional random field (CRF)[68]. For example, Bikel et al. used HMM in an NER

system to identify and classify names and dates[69]. Li et al. implemented an SVM-based learning system for NER task[70]. Liu et al. proposed a CRF-based system in NER for drug name recognition[71].

Unsupervised learning

The key to unsupervised learning is hidden patterns from an unlabelled data set discovered through clustering. The clustering-based NER systems recognize named entities from the clustered groups based on context similarity. This approach leverages the recognition algorithms that can learn lexical patterns on large corpora that has no human annotation. Collins et al. used named entity classification with large unlabeled examples and offered two unsupervised algorithms to prove their ideas[72]. Zhang and Elhadad proposed an unsupervised approach with shallow syntactic knowledge and inverse document frequency (IDF) to extract named entities from biomedical text[73].

Deep learning

A deep learning-based NER model has several benefits compared to conventional machine learning methodologies. First, learning complex and intricate features from input data through non-linear activation functions is meaningful. Second, this model can automatically extract features from input data instead of using feature engineering, which requires domain expertise. Third, a deep learning network can be trained as an end-to-end paradigm that avoids the error cascade of a pipeline model[74].

The architecture of a deep learning-based NER model has three layers. The first is an embedding layer that accepts the input sequence. The embedding layer may be based on either word or character level[75,76] or on incorporating additional features like part-of-speech (POS) and gazetteer[77]. The second layer is the context encoder layer, which usually uses neural networks like CNN[78], RNN[79], or language model[80] to capture context dependencies from the output of the first layer. The final layer is a tag decoder that takes context-dependent representations as input and produces a sequence of tags for the input sequence. This layer usually uses Softmax[81], CRF[68], or RNN[82] as the decoder.

2.3.2 Relation Extraction

Relation extraction automatically identifies the semantic relationship among entities from text. It can be broadly used for NLP applications like information retrieval and question answering. Most relation extraction systems focus on extracting binary relations, such as is-a, part-of, and employment, possibly in the form of subject, relation, and object or what is referred to as relational triples. The subject and object represent two entities, and relation expresses the relationship between these entities.

The earliest algorithm for relation extraction was ***handwritten patterns***, based on the lexico-syntactic pattern developed by Hearst[83]. The hand-built patterns can be highly precise but low-recall, involving much work to create patterns. Another method uses ***supervised machine learning*** for relation extraction, finding pairs of named entities from sentences, then applying the supervised classifiers (e.g., random forest, logistic regression, SVM, RNN, or Transformer) to classify the relationship for each pair[84 ,85 ,86]. ***Distant supervision learning*** combines the advantages of bootstrapping with supervised learning for extracting relationships. This method acquires many seed examples from large unlabeled data sets, then applies a supervised classifier to these examples[87]. ***Joint information extraction*** focuses on extracting named entities and relations at the same time. This method uses a single model system that can effectively integrate entity and relationship information, extracting them simultaneously. For example, Li et al. presented a single framework based on shared parameters for entities and relation extraction[88], with a single model based on global features for entity, relation, and event extraction.

Chapter 3

Pipelines for Procedural Information Extraction from Scientific Literature

3.1 Introduction

In this chapter, we present a machine learning-driven document analysis pipeline for scientific literature that is designed to address challenges to automation of payload extraction and identification of recipe steps using natural language processing (NLP). This paper focuses on payload filtering and extraction in Portable Document Format (PDF) files, the most common format for sharing and dissemination of scientific knowledge. The overall goal of this work is to extract *recipes*, which are defined as procedural specifications in the form of sequences of steps centered around participating tagged entities and ultimately roles and operations, from scientific publications. In this paper we focus on the extraction task itself and consider each purpose and application as a use-case of document analysis.

Our pipeline is designed on principles of holistic document analysis - specifically, to use machine learning in multiple stages with shared objectives for document analysis. Each stage passes successively refined natural language and metadata features on to the next. Our study focuses on materials process engineering, with an emphasis on detecting and categorizing techniques for the synthesis of nanomaterials, an emerging research and development area.

As with our prior work on this application domain, our overall goal is to extract recipes from the scientific literature, using information extraction techniques that are based on machine learning, applied to both labeled and marked-up corpora. However, there is no existing system to date that solves holistic information extraction tasks of the desired form, such as automatic compilation of full recipes, single recipe steps, or even chemical unit operations, from published scientific papers. There is a wide technical gap between the intake of published literature from source collections and the output of actionable information such as the recipe-containing sentences of an article known to be relevant. Some existing tools, such as PDFBox[89], can convert PDF documents to text files, but cannot extract useful information from those text files. Conversely, other tools can help extract domain-related information from text files, but cannot filter the documents for relevance to a query, or segment and order the appropriate text payload within a PDF file. Our system narrows this gap by interfacing such tools using format and metadata standards that are shared from stage to stage of a pipeline, and providing a unified supporting framework for the algorithms and representations of all stages that is driven throughout by machine learning. The system diagram for this pipeline and framework is depicted in Figure 3.1. The central tasks of this paper are the extraction, classification (and automatic annotation), and federated web-based delivery of: plain text payloads, associated figures, recipe-related sentences, and finally recipe steps.

We begin by reviewing related work in Section 3.2 and in Section 3.3, introduces the entire extractor pipeline and the methodological details for each stage. We then fully describe an experiment design and present experimental results in Section 3.4, and finally derive conclusions and priorities for future work in Section 3.5.

3.2 Related Work

Recent work on empirical methods for NLP and supervised machine learning using extracted information has been applied to the domain of nanomaterials IE. For example, Kim et al. introduced synthesis parameters of oxide materials extraction which is based on machine

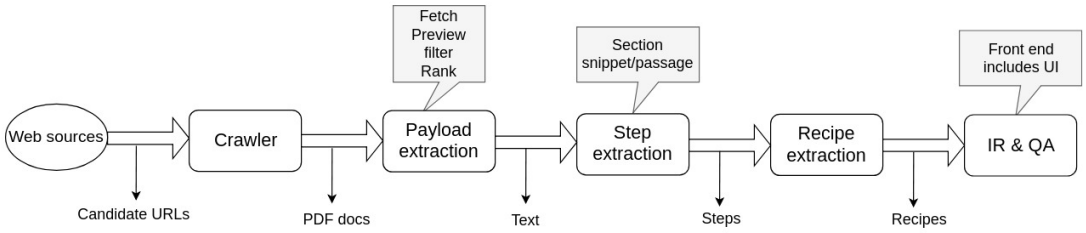


Figure 3.1: Extractor pipeline.

learning and NLP from relevant journal articles[90]. They used an existing application programming interface (API) of CrossRef[91] to retrieve relative articles and converted these to plain text from downloaded files with HTML and PDF formats. The 'bag of words' method has been used for relevant section classification with binary logistic regression in paragraphs. These paragraphs are collected based on manual annotation from 100 different journal articles. The final text is extracted from candidate sections in a scientific article, using a combination of pre-trained Word2Vec[92] and neural network[93] models to proceed text extraction. This Word2Vec is used to learn accurate vector representations for specifying domain-related words of oxide materials. This is a first step towards using machine learning for information extraction from large and comprehensive corpora in the nanomaterials domain, and yields some methods that are potentially transferable to other domains of scientific literature. Another significant research project[94] also addresses text extraction with machine learning techniques from scientific literature in the materials synthesis domain.

3.3 Information Extraction System

In this section, we describe the details of each stage of the extractor pipeline (Figure 3.1). Our system covers from scientific literature source collection to recipe steps displayed on web page, including the following steps:

- Synthesis literature materials are crawled from online resources. The type of this

literature is free texts in basic publication formats (PDF or HTML);

- Documents (PDF) conversion to plain text and extracting relevant figures and other images from published literatures, then filter the domain-related literature;
- Step extraction/sentence classification from experiment section of literatures;
- Recipe assembly from relevant sentences;
- Front end as user interface to display the contents of whole literature which includes title, authors, abstract, body of literature, sections, references and recipe results.

Crawler/Ranker/Filter

We use a crawler which automates the process by utilizing a set of seeds (e.g. URLs) to find sufficient links to look through in order to construct the corpus of our domain.

To complete the task of finding and downloading PDFs from the web, we created a Java based web crawler. To start, the crawler takes a newline delimited text file containing seeds (URLs) as the base index to start the crawl from. It then builds a B-Tree from the URLs given and the URLs crawled with a depth determined by the given depth in the configuration file. From the nodes on the B-Tree, the crawler determines which nodes are downloadable PDFs and the proceeds to download them into the output directory. This process runs until each node on the B-Tree has been covered. We also developed a backward citation component to help focus the crawler. After the first initial crawl and after the annotations of the gathered PDFs, the PDFs that were matched positive for relevancy are sent back to the crawler. The crawler then handles the metadata from the PDFs and uses the citations in each paper to query for new, more relevant, seeds. We used relevant seeds which are provided by expert and crawled 30K scientific literature from public resources as our corpus.

Payload Extraction

The payload extraction system used in this work is based on similar ones developed for speeding up the annotation process to identify relevant papers from a corpus of scientific documents using classification[95]. This tool shows the first few pages of any paper, lists the keywords which are domain-related, and highlights them on the paper simultaneously. That

is so users can quickly go through the paper indicated by the highlighted keywords to decide if the paper is related to target domain.

The text extraction and section classification steps are crucial for recipe extraction in the pipeline. Classifying the different sections of a synthesis literature allows for the region of interest in recipe-step search to be narrowed down. However, with the large crawled document corpora comes a disparity of document formatting and challenges for section classification. To address these format disparity challenges, *MATESC*, a tool for metadata-aware extraction developed by De La Torre et al.[96] is adapted and improved. This tool uses metadata features and heuristics, such as font size, font type and character spatial location to group words, lines and paragraphs to classify them with their corresponding section header.

Figure 3.2 shows the work flow of *MATESC (Metadata-Analytic Text Extractor and Section Classifier for Scientific Publications)*, an open-source tool developed by De La Torre et al. *MATESC* takes PDF documents as input and uses PyMuPDF[97] to extract text and the metadata of each character. The extracted text is filtered by removing irrelevant text usually found in the margins of each document page, using their spatial location. Then, words are merged into their corresponding line, while considering font and spatial location to differentiate between section titles and section content. Afterwards, those lines are then grouped into paragraphs and those paragraphs are sequentially ordered based on a calculated bounding box of a paragraph. Table 3.1 shows the accuracy of *MATESC* evaluated on random articles versus articles relevant to the nanomaterials synthesis domain. For detailed results regarding section classification accuracy which we use as a baseline, we refer the interested reader to De La Torre et al.[96].

Name	Accuracy	Precision	Recall	F1-score
Random Domain	0.85	0.63	0.63	0.57
Relevant Domain	0.88	0.78	0.74	0.72

Table 3.1: Accuracy for the MATESC extractor on random vs synthesis of nanomaterial relevant papers.

To improve *MATESC* 's text grouping and section classification, we used the *scikit-*

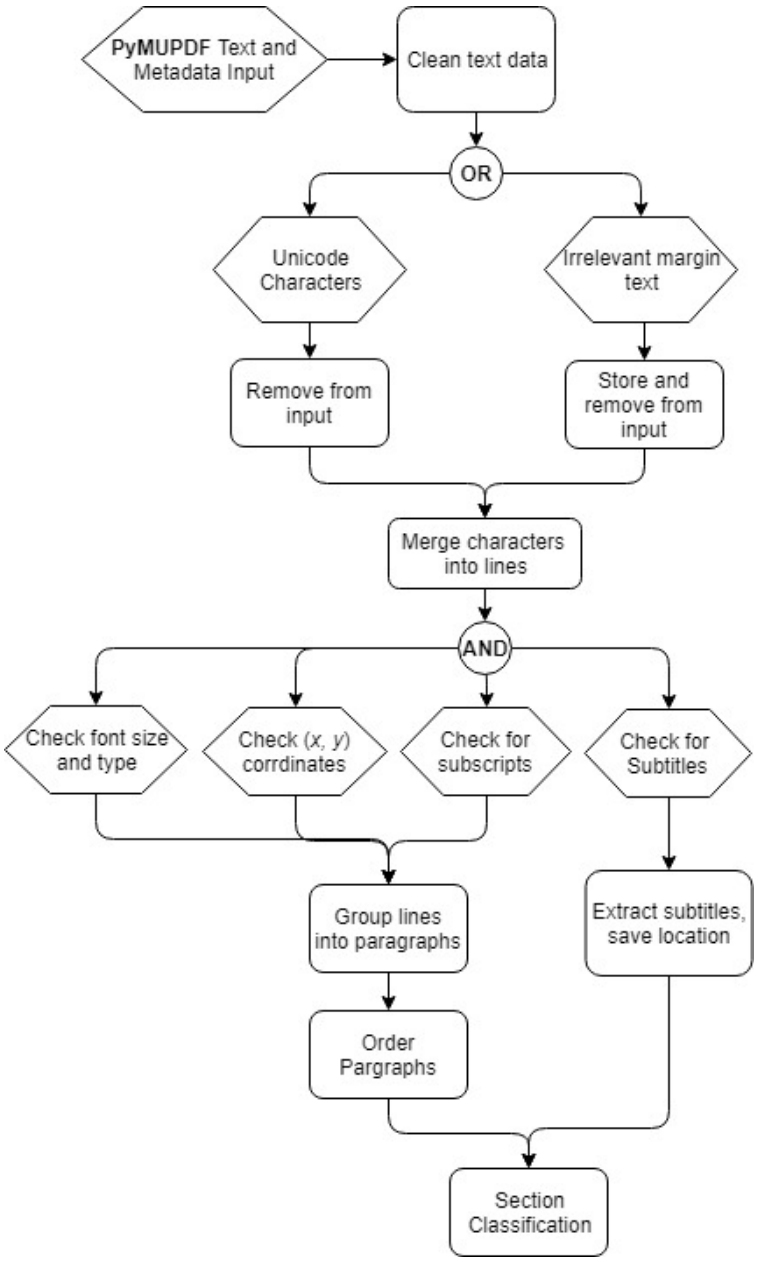


Figure 3.2: MATESC's input processing and section classification pipeline.

learn implementation of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to classify spans of text into their corresponding groups based on the euclidean distance between each span's metadata features. The features included x,y coordinates, font type, and font size of each text span. The current limitations of the algorithm include excessive splitting and (less frequently) merging due to the unordered

clustering nature of the algorithm. Some text groups-spans and lines-are over-split (under-merged) based on the difference distance threshold. Future work involves further research in DBSCAN's parameter estimation, including fine-tuning of radius and minimum point thresholds.

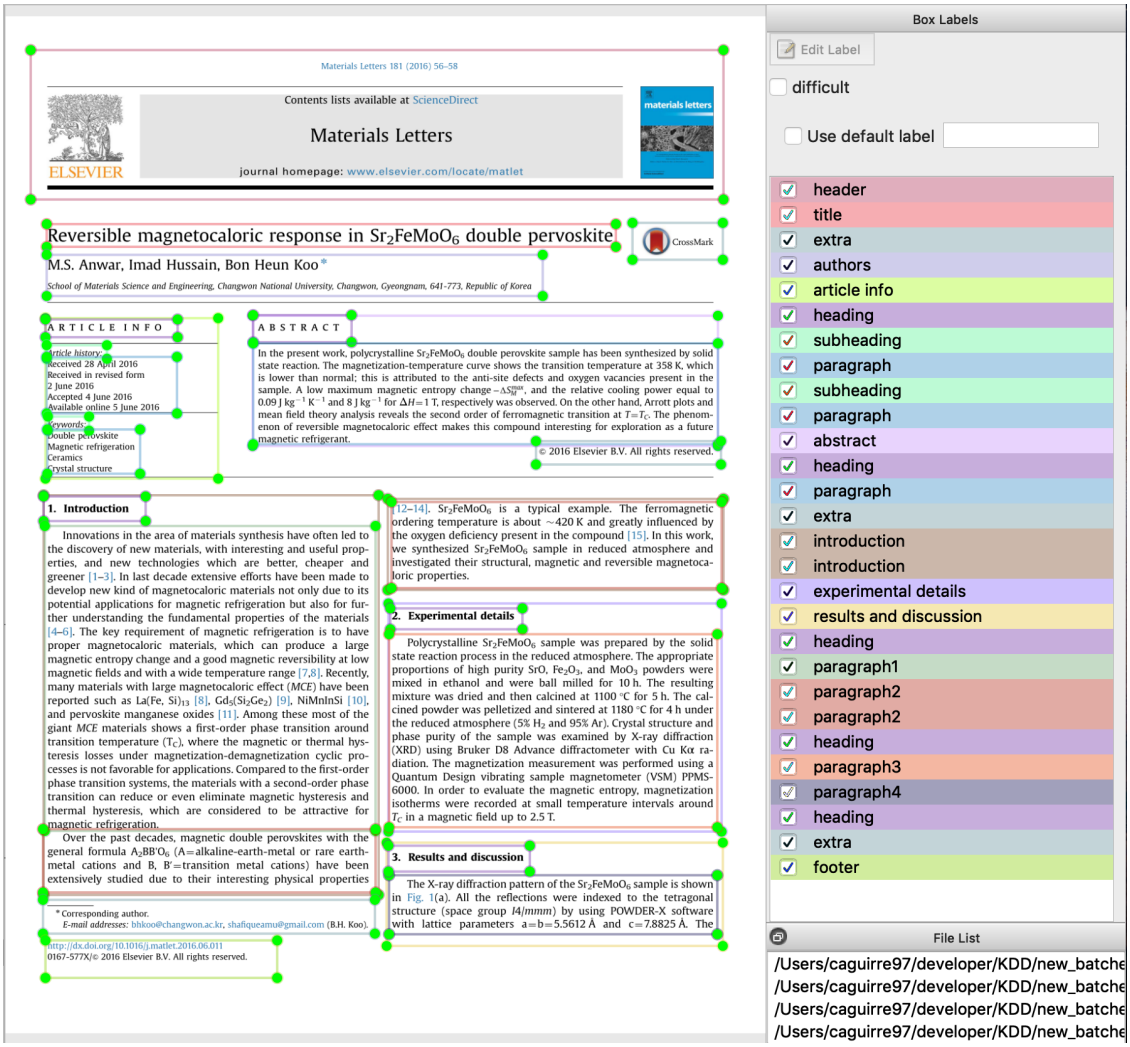


Figure 3.3: Payload extraction ground truth (markup annotation).

As our ground truth for testing *MATESC* and other clustering algorithm output, we developed an open-source dataset using *labelImg*[98] to obtain the regions of interest for headers, heading, and paragraphs. The data set allows us to use *PyMuPDF* to extract the text spans within each region of interest and create the ground truth paragraphs and sections. Figure 3.3 shows an example of a annotated page used as Payload Extraction ground truth.

Currently, this data set allows us to test grouping, clustering and sequential reading order of these groups.

Paragraph and Section Clustering

Due to PDF document format disparity among different journals and publications, the procedure to form paragraphs and sections from characters is laborious. Heuristics that consider the metadata features provided by it PyMuPDF and others calculated from it, such as spacing between lines and column recognition, has been integrated into *MATESC*. Nevertheless, learning to cluster paragraphs and sections is an interesting problem that the group has started to work on. Grouping lines into paragraphs and paragraphs into sections is key to finding the synthesis within the literature. In the future, we plan to test clustering algorithms, such as the DBSCAN algorithm to group lines into their corresponding paragraphs based on the distance measurement of each line's metadata features. Some of these features include the bounding box around the text, the font type and size and the page number the line belongs to. Furthermore, we would like to use `pyfaster-rcnn`[99] to obtain region proposals for our sections, we hope to see an increase in the accuracy of section classification and extraction.

Step Extraction

We use a binary Naïve Bayes (NB) classifier to perform sentence classification on the experiment section, which was an output of the payload extraction stage. We hand-labeled 2600+ relevant or irrelevant sentences from 98 relative literature as our training data set, $(\mathrm{s_1}, \mathrm{c_x}) \ldots (\mathrm{s_m}, \mathrm{c_x})$ where s represents a sentence, and c represents its label, where the value of c is 1 (relevant) or 2 (irrelevant). We then train the NB classifier to obtain a learned function γ: s $\rightarrow$c that predicts the class attribute of input sentence s through the classifier.

$$c_{\mathrm{NB}} =_{c \in \mathcal{C}} P(c_{\mathrm{i}}) \prod_{s \in S} P(s|c) \tag{3.1}$$

Here $S \equiv \{s_1, s_2, \ldots, s_{\mathrm{n}}\}$ represents the sentences from experimental section, and $C \equiv$

$\{c_1, c_2\}$ represents the binary classes. We also tried to train our model with different feature categories of transforming training data set to different vectors:

1. Count vectors as training features, converting a collection of sentences to a matrix of term frequency;

2. TF-IDF vectors as training features, converting a collection of sentences to a matrix of TF-IDF feature scores;

3. N-gram as training features, converting a collection of sentences to a matrix of TD-IDF scores of N-grams;

The model has been implemented in *scikit-learn*[100], and results are shown in table 3.2. The class 0 represents irrelevant sentences, and class 1 represents relevant sentences.

Name	Accuracy	Class	Precision	Recall	F1-score
Count	0.79	0	0.83	0.78	0.80
		1	0.76	0.81	0.78
TF-IDF	0.78	0	0.81	0.82	0.82
		1	0.77	0.76	0.76
N-gram	0.76	0	0.78	0.83	0.80
		1	0.76	0.69	0.72

Table 3.2: Accuracy of NB classifier with different categories.

Recipe Extraction

A *recipe* in our research is defined as a set of specific actions that are applied to a set of recognized base materials in experiments within the application domain of nanomaterials synthesis. After conducting informal elicitation sessions with subject matter experts concerning the format and content of recipes, we developed a written rubric; we then analyzed 27 relevant papers manually to extract recipes as our ground truth. *ChemicalTagger*[101], an open-source tool for semantic text-mining in the chemistry domain that is based on *OSCAR*[102] (a system for automated annotation of scientific articles in the field of chemistry or a subarea), is used in our research to extract recipes from relevant sentences generated

from the stage of step extraction in our pipeline. *ChemicalTagger* generates XML files from raw text. The XML files include different tags and some of which contain the verbs labeled as action phrases (e.g. dry, wait) by *ChemicalTagger*. We then parse the XML and extract recipes from the sentences containing action phrases.

It is worth noting that we did not use *ChemicalTagger* directly but rather as a source of functional features on which our new system is based. The reason is that *ChemicalTagger* defines some verbs as action phrases which are not applicable in the nanomaterials domain. For instance, "prepare" is recognized as a verb and extracted by *ChemicalTagger*; however, it is in the stage of preparation rather than the real outcomes of recipe steps in our domain. Because of that, we have compared the results extracted by *ChemicalTagger* with our ground truth and modified the action phrases in *ChemicalTagger* to be consistent with our domain knowledge (e.g. adding injection as action phrase).

Front End Intelligent UI for IR & QA

The front end interface has been developed to demonstrate our system's information retrieval capabilities. This interface presents the user with the option of querying a set of papers to view by selecting their material and morphology or the option of searching all papers by user provided search terms. Our front end shows the system's functionality after the payload extraction step has occurred. The resulting papers were initially the output of the crawler stage which were then modified for retrieval in the payload extraction stage. After a search is made, the user can view relevant paper titles with their corresponding images. If a paper is selected for viewing, a user is shown a paper's extracted contents and its DOI. A paper's extracted abstract, experimental section, and references are displayed. The capability of searching by user provided terms is implemented using the Apache Solr[103] search engine.

3.4 Recipe Evaluation

As described above, each desired recipe step output is a demarcated passage or set of passages within a sentence which includes an action followed by some specific materials and/or

metrics in our domain. A complete recipe thus consists of multiple recipe stems that are combined by sequential grouping (`begin/end`), with the eventual goal of developing a formal specification language that includes parallel execution (`cobegin/coend`) and iteration. This would facilitate the development of a materials synthesis planning language beginning with a data definition language (DDL) and leading to a formal ontology for specific recipe-based tasks such as question answering (QA) about ingredients, unit operations, and embedded numerical quantities, such as concentrations and temperatures. For the recipe evaluation we need to measure the difference between the recipe output from our system and recipe ground truth, which is annotated manually. Identifying whether the recipes are the same as the recipe ground truth is a complex task that is challenging to specify formally and difficult to automate, because the recipe output also depends on each stage of the extractor pipeline (Figure 3.1). For example, at the Payload stage, we might lose some information, such as Unicode characters: °, ≥, or there is some extra space generated between the material names in our domain, when converting PDF document to text file. Additionally, at the Step Extraction stage, we trained machine learning model for filtering the irrelevant sentences, but the accuracy of the model would also affect the final recipe outputs. For the Recipe Extraction stage, *ChemicalTagger*, an open-source tool, was used to extract sentences that include action phrases. Whether these action phrases can fit in our domain or not would determine the accuracy of the result of recipe extraction.

By examining the holistic output at the sentence level, we calculate precision, recall. and F1 scores to evaluate the recipe output generated from our system. In our system, precision ($T_r/T_r + E_r$) indicates the rate of recipes extracted by our system that correspond to the ground truth recipes, from a known reference set determined by annotation; recall ($T_r/T_r + M_r$) measures the rate of known reference recipes that are successfully captured by our system; F1 score is the harmonic mean of precision and recall.

- T_r **(true positives)** represents output recipes r from a system that are the same as ground truth recipes;
- E_r **(false positives)** represents **extra** recipes r that are captured by our system but

are not relevant recipes, compared with ground truth;

- M_r **(false negatives)** represents r that are part of the ground truth recipe but **missed** by our system;

The measurement of above parameters is based on cosine similarity, which will eventually calculate the precision, recall, and F1 score. Moreover, the recipes captured by the system are from the same document where the ground truth is located. Therefore, there is no meaning ambiguity between the two recipes in comparison and semantic analysis is not necessary.

Specifically, cosine similarity is used to evaluate how similar two documents are from each other. These two documents in the form of vectors represent the recipe output from our system and the recipe ground truth, respectively.

$$Score_{\text{similarity}} = \frac{\sum_{i=1}^{n} O_{\text{i}} \times G_{\text{i}}}{\sqrt{\sum_{i=1}^{n} (O_{\text{i}})^2} \times \sqrt{\sum_{i=1}^{n} (G_{\text{i}})^2}} \tag{3.2}$$

In Equation 3.2, O and G represent the two documents in terms of vectors, where O denotes recipe output and G denotes ground truth. We evaluate recipe accuracy for two perspectives: (a) measure the similarity of the two documents; (b) measure the accuracy of recipes by looping each sentence in the recipe output to compare with the recipe sentences in ground truth. Results generated from a and b will be compared and sometimes lead us to explore further. For example, a high score in terms of similarity between the two documents but a low score for accuracy of recipes is a red flag to us and further investigation is warranted. Regarding the accuracy in (b), we consider two situations in which a recipe is accurately extracted: if the similarity is equal or greater than 70%, the parameter T_r would be set up to 1, meaning the truth recipe has been outputted by our system; if the similarity is greater than 50% but less than 70%, we assign the value of 0.5 to parameter T_r. This is because some of the recipes captured by our system actually are partial ground truth recipe. A very strict restriction will filter out them.

Table 3.3 shows the average similarity between two documents, percentage of precision,

$Doc_{Similarity}$	Precision	Recall	F1
87.95%	74.76%	71.33%	70.27%

Table 3.3: System evaluation results.

recall and F1 score compared the system recipe output with the recipe ground truth of 27 papers which are annotated manually. The evaluation of these 27 papers has been done on a separate test set other than on the training dataset.

3.5 Conclusions and Continuing Work

The procedural extraction tasks and pipeline described in this paper also represent a test bed for the document analysis tasks of learning to rank and filter in a focused crawler, text extraction from typeset documents, snippet and passage extraction, and especially unstructured to structured information extraction. Each stage of the holistic system affects the final recipe output, demonstrating a challenging problem of credit assignment that may be amenable to representations for sequential decision making. In preliminary experiments on the entire pipeline, we noticed that raising the accuracy of payload extraction, training our models with more data sets, and extracting action phrases that better fit in our domain propagate gains downstream to improve recipe extraction accuracy.

Promising findings using corpora crawled using seeds described in[95] suggests that next steps in current and future work ought to involve expansion of the test bed to other nanomaterials domains and examine transfer learning between domains (one type of material to another), and between tasks (e.g., recipe knowledge base population to recipe QA or textual entailment). The existing system incorporates information extraction in the form of full-document payloads and sentential or sub-sentential units of recipe-bearing text, and also filters collections for relevance to a specified material of interest and incorporates a presentation module for the display of extracted figures and other embedded content.

Chapter 4

Document Layout Understanding

This chapter includes two sub-chapters regarding the task of document layout understanding.

4.1 Vision-Based layout Detection from Scientific Literature using Recurrent Convolutional Neural Networks

4.1.1 Introduction

The number of academic literature publications has been growing rapidly. These published literature documents include a great amount of free text containing potentially valuable information which can help scientists and researchers to develop new ideas in their fields of interest, to extract peers' key insights[12], or explore new research areas from various fields each having their own serials (journals) and conference proceedings[94]. Unfortunately, the increasing rate of production of scientific literature in many domains has outpaced the rate at which individual researchers and smaller laboratories can process new documents and assimilate knowledge. Millions of academic products (conference papers, journal papers, book chapters, etc.) are published each year and disseminated as digital documents consisting of generally unstructured text. Scientific Literature Layout Detection (SLLD) can increase

the effectiveness of automated information extraction tools by inferring the layout of scientific literature accurately based on geometric and logistic layout analysis, this facilitating the automatic extraction of metadata such as section delimiters, formatting information for equations and formulas, procedural data, special environments, captions for tables and figures, bibliographies, etc. It presents a possible solution for automatic construction of large corpus, and also provides assistance for some scientific literature-related downstream tasks of natural language processing (NLP).

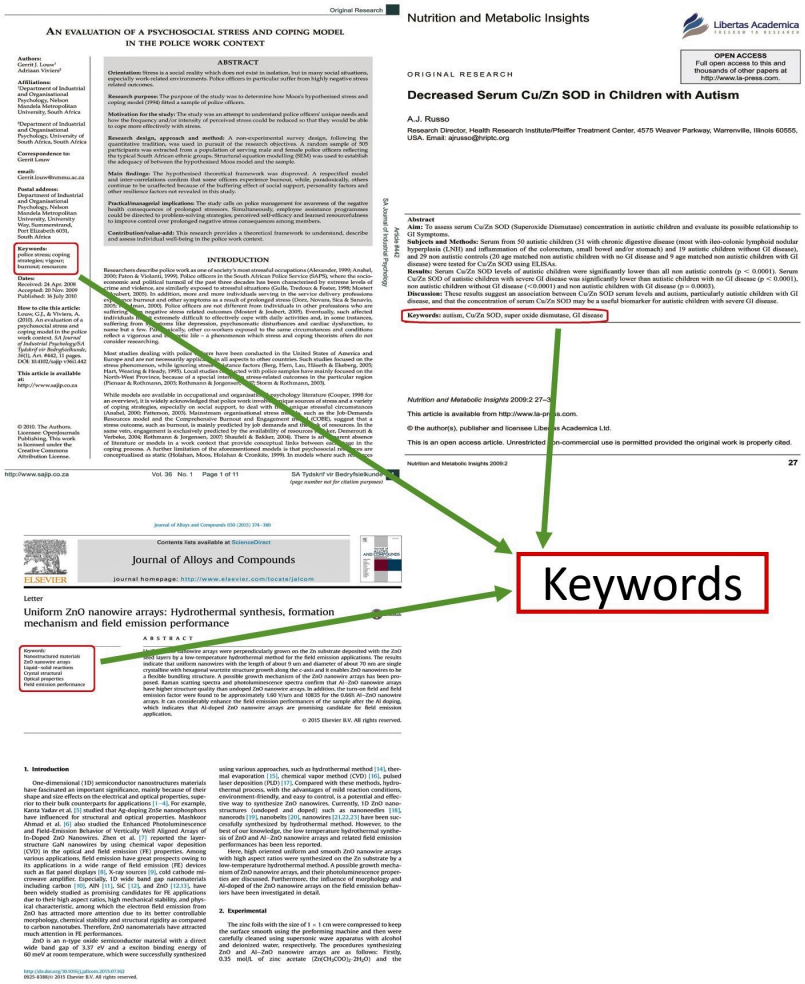


Figure 4.1: Examples of inconsistent layouts in scientific literature. The layouts of keyword vary by different articles.

Portable Document Format (PDF) is a form of digital document which has been most commonly used in scientific publications. Extracting good quality metadata from PDF

publications remains difficult and challenging, since publishers have very diverse preferences of formatting and layouts in their articles. Thus, a single type of information can be organized in various different formatting styles and fonts for different scientific publications (Figure 4.1). Existing tools such as optical character recognition (OCR) perform the extraction of raw text from scanned PDF documents automatically, but they are not used for metadata information extraction, and the lack of layout analysis will lead to messy results, such as the extraction of headers and footers together. Furthermore, raw text information extraction from whole documents tends to necessitate secondary data cleaning, which is extremely time-consuming. For instance, procedural information extraction in the nanomaterial synthesis domain may only require textual information from the experimental methods section rather than an entire scientific article[12]. In addition, the bottleneck of some existing works of document layout analysis is related to a lack of comprehensive extraction for different blocks, such as the title, authors, figures, etc., from scientific literature, including figure and table detection from digital documents[59], full-text extraction from scientific publications[104], or domain-specific figure analysis in scientific articles[105]. Therefore, a robust method for comprehensive and efficient information extraction is still needed for common section types that are typical for many scientific disciplines.

In this paper, we present an end-to-end learning framework that is based on Faster R-CNN[99], adapting the two-stage object detection framework from computer vision for scientific literature documents layout detection. This novel approach detects the main regions of scientific articles, and outputs the blocks and their corresponding labels, including title, authors, abstract, body of text, etc.(Figure 4.2). We also create a synthesis data set by merging and rendering key pages from two scientific document corpora. This allows training and evaluation of models that are challenging to evaluate given the insufficiency of existing data set for SLLD tasks. We begin by reviewing related work in Section II and introducing the methodology in detail in Section III. We then fully describe an experiment design in Section IV, present evaluation part in Section V, and finally draw conclusions and priorities for future work in Section VI.

Figure 4.2: Ground truth: major regions have been annotated.

4.1.2 Methodology

Object detection using deep learning networks has grown popular and for the past decade convnets have held first position and attained state-of-the-art results on standard data sets for object detection (e.g. MS COCO[106]). The task of SLLD is similar to that of object detection, both of which can locate objects with bounding boxes, and classify these objects in images with labels. However, digital images in scientific publications are not like the normal images that include distinct objects like a car, a bird, or a flower. Most of the scientific document pages only have a few images or tables, and the rest of the parts are the body text of sections, titles, author lists and affiliations, etc., and the types of font tend to be different from each other. To address this problem, we try to use different neural networks for feature extraction from input images, and to combine them with appropriate anchor ratios for SLLD tasks. Section IV shows significantly improved results generated by our approach. Moreover, our end-to-end learning framework is more robust compared with others (e.g., complex document layout analysis[57]), and it can be applied to any language

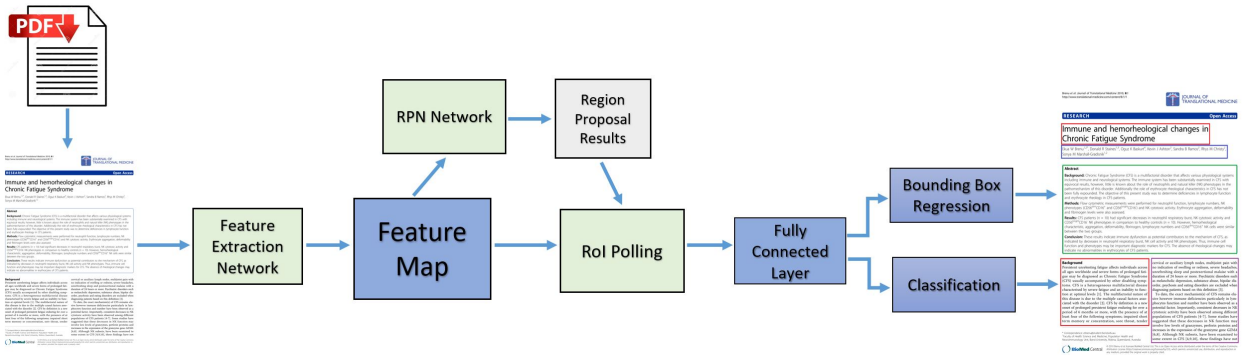


Figure 4.3: Scientific literature layout detection framework.

without additional features (e.g. text embedding[46]) but document page image for input.

Our approach uses Faster R-CNN[107] (Figure 4.3) as our baseline, a classic two stage object detection framework that is built on Fast R-CNN[108]. The main improvement of Faster R-CNN is to replace a selective search algorithm with a region proposal network, which is based on anchor boxes, to generate proposals for a detection network. Therefore, there are two neural networks that are involved in the training process: one neural network is used for proposal region generation which might include target object, and the other is used for selected regions classification and object detection. Faster R-CNN as our baseline framework uses ResNet-50[109] with Feature Pyramid Networks (FPN)[110] as backbone neural networks for feature extraction and potential region selection. These selected regions are then fed to the second neural network for object classification and bounding box regression.

The two strategies are applied based on baseline of Faster R-CNN: backbone network replacement, and appropriate anchor ratio selection for the SLLD task. The experiments demonstrate that the results are indeed improved using our strategies.

Backbone replacement with VoVNet-v2[111] At present, deep learning-based object detection models rely on Convolutional Neural Network (CNN) as feature extractors, such as ResNet for Faster R-CNN and DarkNet for YOLOV3[112]. Although deep neural networks have performed well on feature extraction, it also incurs the problems of high computational cost and slow training speed. VoVNet[13] network is proposed to resolve the efficiency problem, and it has been known to have better performance than other deep neural networks based on

many experimental results. VoVNet is built on One-Shot Aggregation (OSA) modules which aggregate concatenation feature only once in the last feature map rather than aggregate previous feature at every subsequent layers, for instance, by DenseNet[113]. VoVNet has several architectures based on different numbers of CNN and OSA on different layers, such as VoVNe-39 and VoVNet-57. VoVNet-v2 further improves performance and efficiency of VoVnet by adding (1) residual connection which enables us to train deeper networks, such as VoVNetV2-99; (2) Squeeze-and-Excitation (eSE) attention module on the last feature layer to improve the performance. We use VoVNetV2-39 as a backbone for feature extraction within Faster R-CNN framework.

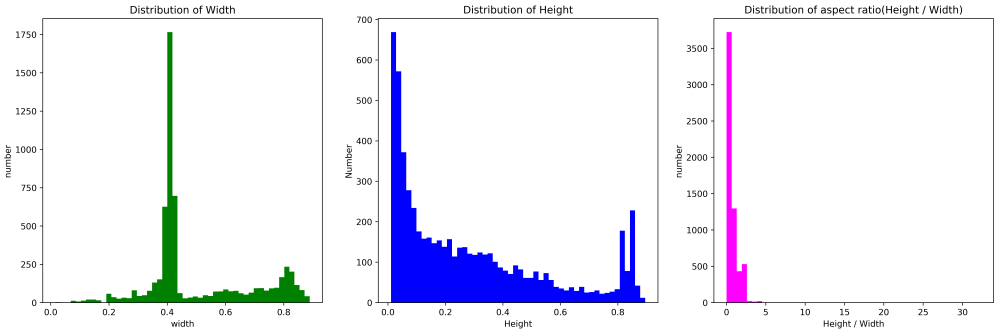


Figure 4.4: Aspect ratio analysis for anchor box.

Anchors Aspect Ratio Selection Anchor boxes use a similar mechanism as sliding-window to capture the most likely regions which contain objects with different scales and aspect ratios on RPN stage. Different objects have different aspect ratios (width/height) to be detected in images, such as the aspect ratio of a car, which is around 1:2, or that a utility pole, which might be 1:10 or lower. To accurately identify the proper schemes of different anchor sizes of pre-detected objects will improve the prediction results. In SLLD tasks, for instances, the aspect ratio box of text body is different from the aspect ratio box of author. We analyze the distribution of bounding box sizes based on the ground truth object bounding box coordinator of our synthesis data set with K-means cluster anchor box selection[114], and got the aspect ratios for different blocks ranging from 0.1 to 4.0 (Figure 4.4) by choosing 50 clusters. The details of Anchor parameters configuration are presented in Implemented Details.

4.1.3 Experiment

Synthesis Data Set

There are few available data set for SLLD task. Soto et al. introduced a good but small data set that is annotated manually for SLLD[115]. This data set incldues 822 images from 100 PDF scientific literature, and 9 labeled region classes to cover major region layout of scientific literature. However, it has an instance imbalance issue such that there are 1275 instances of body, which are way more than the 100 instances of title. This is because a paper normally has only one title, a few authors, but the number of body of text are at least as many as the number of pages, and sometime even twice or three times more than the number of pages depending on the layout of the document. Moreover, this data set missed the label of Keywords which usually indicates the most refined and significant information from the scientific literature. In order to solve the above issue, we create a synthesis data set, which is not only an extension of the data set from Soto et al.[115] but also integrates two other data sets as follows:

- ICDAR-2013[116]: This data set includes 150 tables from 67 PDF documents, 40 PDFs among which are collected from US Goverment and the rest are from EU. The purpose of constructing such a data set is to increase the diversity of tables in use. All of the PDF documents are converted to images.
- GROTOAP[117]: This data set has 113 annotated PDF documents from scientific literature. It achieves 100% accuracy since it is annotated manually. We only chose the first page of each PDF scientific document, and converted them into images for our synthesis data set in order to increase the minority instances and solve the instance imbalance issue.

After combining these three data sets, we extracted 1550 image pages from 363 PDF documents. All of images have been converted to a fixed size of 612×729 at 200 dpi. We use 10 labels to classify the major regions from the scientific literature with the synthesis data set:

- ***Title***: the title and subtitles.
- ***Authors***: the author names.
- ***Address***: the affiliation information of authors, including authors' address, email, etc.
- ***Abstract***: an abstract section.
- ***Keyword***: the selected keywords.
- ***Body***: the main block of articles.
- ***Figure***: all figures but excluding logos or icons from publishers.
- ***Table***: the tabular contents.
- ***Caption***: the captions for both figures and tables
- ***Reference***: the bibliography information, excluding post-references notes.

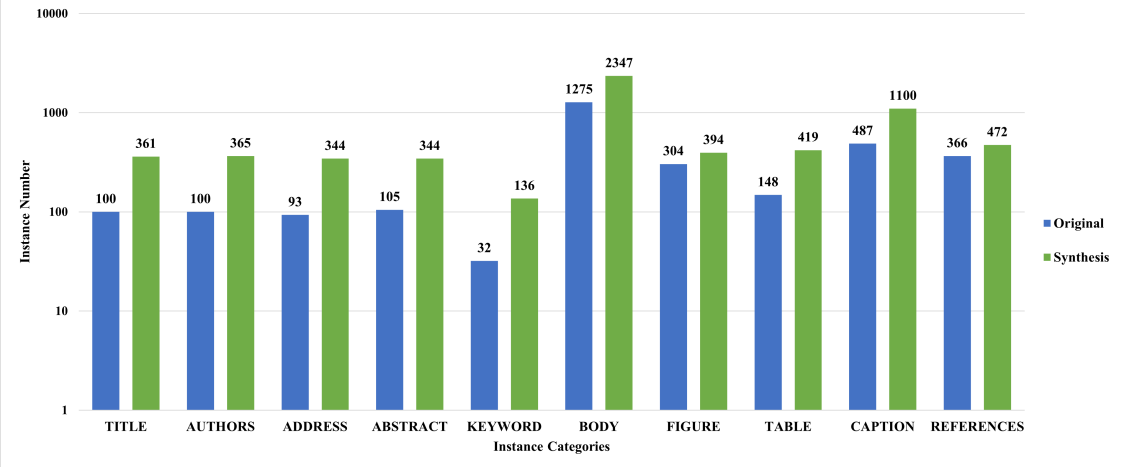


Figure 4.5: Instances comparison between two data sets by labels.

We added one more label *keyword*, merged *table_caption* and *figure_caption* as ***caption*** from[115] data set, and keep the rest as the same. Figure 4.5 shows instances comparison between Soto et al. [28] data set and our Synthesis data set.

Implementation Details

The models are trained with pre-trained weights on MS COCO data set (∼37 epochs)[106]. Two stage object detection frameworks Faster R-CNN and Mask R-CNN are implemented using Detectron2[118]. All models are trained and tested on a single NVIDIA Tesla P100 GPU with a batch size of 8 for 150 epochs. SGD (Stochastic Gradient Descent) is used as the optimization algorithm. The initial learning rate is 0.002, and decays by 0.1 after 100 epochs. We used different neural networks as a backbone for feature extraction from input images: ResNet-50+FPN, VoVNetV2-39. There are 5 anchor scales in powers of 2 from 32 to 512, 8 anchor aspect ratios from 0.2 to 2.8 based on K-means selection results for VoVNet-v2 backbone model, and the reset of models are trained by standard aspect ratios [0.5, 1.0, 2.0]. We also trained a single-stage object detection framework YOLOv3[112] with our synthesis data set for comparison. This model uses DarkNet53 base network that were pre-trained on MS COCO data set.

4.1.4 Evaluation

We constructed object detection frameworks with different configurations, and train them with two different data set to compare them with our design SLLD model. All of these models start by being trained using pre-trained models on MS COCO[106] data set. For obtaining a thorough performance evaluation, we use MS COCO[106] evaluation metrics to measure SLLD results rather than other simple evaluation methods, such as precision and recall, because it can help to evaluate various sizes of objects from detection results. For SLLD tasks, it can help to evaluate different region sizes of scientific literature based on the detection results.

MS COCO[106] evaluation metrics with different IoU thresholds

- mean Average Precision (**mAP**): mean average precision at IoU = 0.5:0.05:0.95
- Average Precision 50 (**AP50**): $AP^{IoU=0.50}$
- Average Precision 75 (**AP75**): $AP^{IoU=0.75}$
- Average Precision for small object (**APs**): area (pixel-wise) $< 32^2$

- Average Precision for medium object (**APm**): 32^2 < area (pixel-wise) < 32^2
- Average Precision for large object (**APl**): area (pixel-wise) < 96^2
- Average Recall (**AR**)

Data Set

- ***data set1*** (D1): original data set[115] - 600 image for training, 222 images for testing.
- ***data set2*** (D2): synthesis data set - 1225 images for training, 325 images for testing.

Table 4.1: Overall comparison among SLLD results (%) with different methodologies and data set at IoU = 0.5:0.05:0.95. D1 represents data set1 and D2 represents data set2.

Detector	Backbone	Data Set	mAP	AP50	AP75	APs	APm	APl	AR
Soto et al.(30 epochs) [28]	ResNet101	D1	-	70.30	-	-	-	-	-
Faster R-CNN **(baseline)**	ResNet50_FPN	D1	69.76	87.46	76.49	-	51.65	77.41	62.70
Faster R-CNN	ResNet50_FPN	D2	77.48	92.39	84.42	35.00	63.32	77.65	69.50
Mask R-CNN	ResNet50_FPN	D1	70.68	87.60	82.90	-	52.05	75.05	65.50
Mask R-CNN	ResNet50_FPN	D2	77.66	91.79	85.80	40.00	64.378	75.604	69.50
YOLOV3 (49 epochs) [28]	-	D1	-	68.90	-	-	-	-	-
YOLOV3	DarkNet53	D2	45.90	66.50	57.10	-	-	-	46.33
Faster R-CNN	VoVNetV2-39	D1	67.12	89.01	72.84	-	47.66	73.56	60.50
Faster R-CNN **(ours)**	VoVNetV2-39	D2	76.39	**95.02**	**86.46**	**75.00**	62.25	74.22	68.80

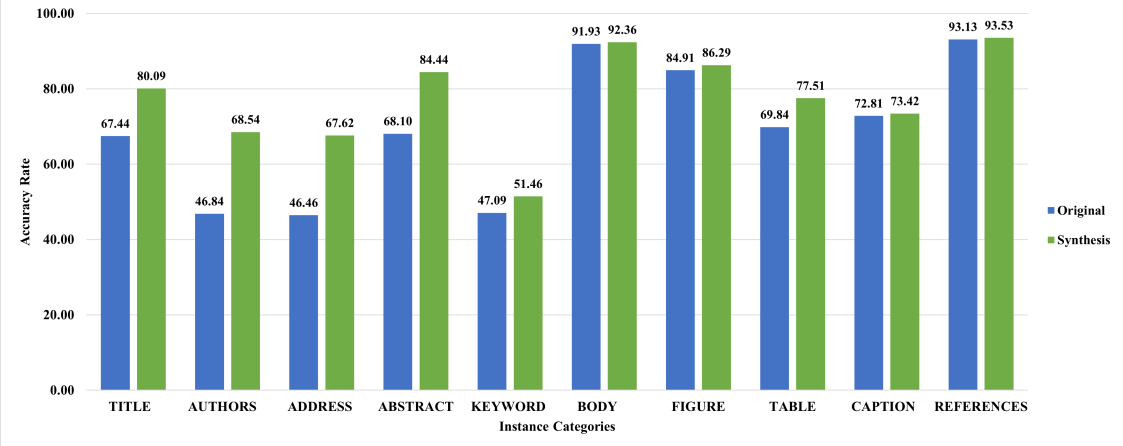


Figure 4.6: Detection results comparison between two data sets by labels at 0.5 IoU.

Our algorithm of SLLD tasks has significantly improved results compared with others (Table 4.1), especially for the object detection in small area, such as the area displaying

keyword and the area presenting authors in scientific literature. We believe that this result is benefited by our appropriate anchor analysis. We redesign aspect ratio of anchor box rather than using simple numbers [0.5, 1.0, 2.0] for the most commonly used detection tasks to cover more labeled regions in scientific literature documents. Likewise, the result indicates that our framework also obtains improvements in each labeled region of SLLD tasks. (Figure 4.6).

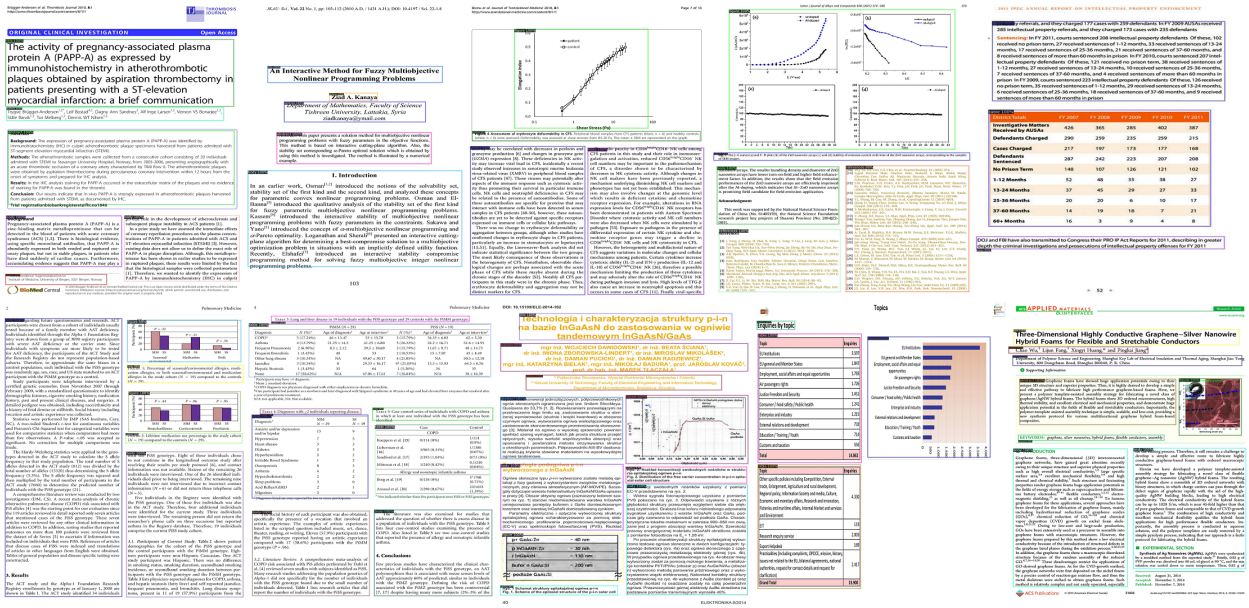


Figure 4.7: Detection results with corresponding labels.

Our framework performs well on major region detection from scientific literature (Figure 4.7). However, it still fails to detect non-rectangle region in the images of scientific documents (left figure in Figure 4.8). We attribute this to the restriction of annotation methods. For data set annotation, we used rectangles instead of bounding polygons, which could be a viable and more flexible representation for non-rectangular regions. Moreover, duplicated bounding boxes are found to detect a single region area in the images of scientific documents (right figure in Figure 4.8). We think this issue will be improved by training by deeper network, such as VoVNetV2-99., and with more epochs.

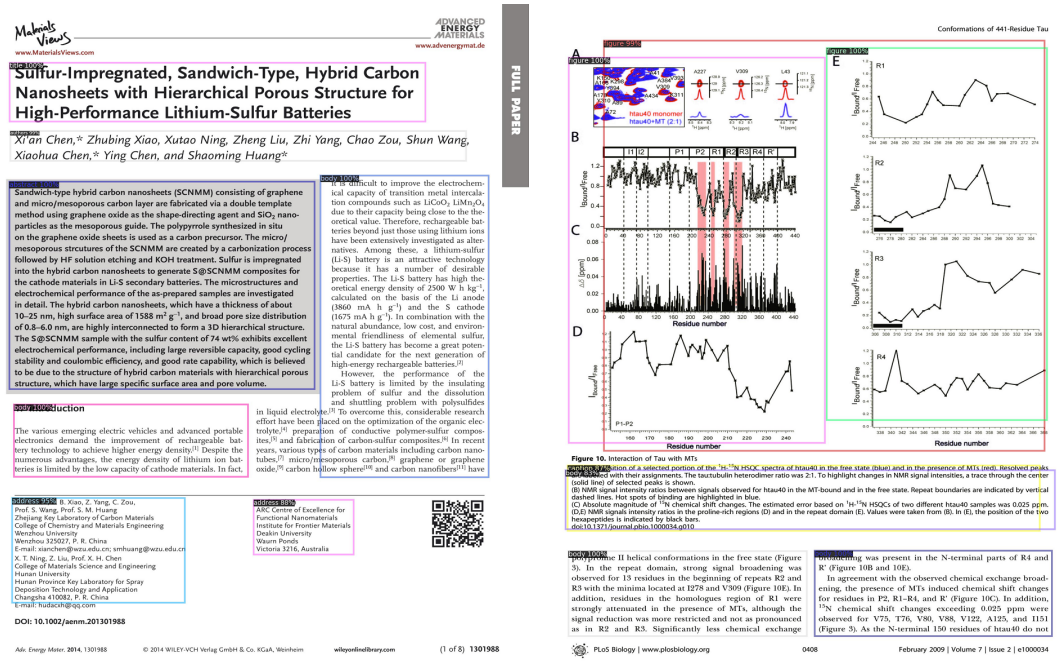


Figure 4.8: Examples of failure: left figure represents failure of locating polygon regions, and right figure represents overlap of bounding boxes for single region.

4.1.5 Conclusion

For SLLD tasks, we introduce a novel end-to-end learning and vision-based framework. The major regions in scientific literature documents will be detected through the model which is trained by our framework. This model not only detects text regions but also figures and tables. Our approach is easy to adapt and implement for a broad range of scientific literature formats and domains, since it does not require extraction of additional features (e.g. text information of document). Fine-tuning pre-trained model which is generated from irrelevant tasks is feasible through our experiment. Specifically, we used a pre-trained model with MS COCO[106] which does not have any classes in the data set that are related to our SLLD work. Compared with other approaches of document layout analysis that are only working on some parts of the documents, our approach is built on the entire document rather than some regional information, and therefore it offers a possible way to construct large corpus for downstream NLP.

Regarding the continuing work, more instances of minority classes should be added into

training data set, such as title, authors, keyword, to improve minority object detection results. Although we apply transfer learning to our models, the training process is still not as efficient as what we expected. We are going to replace complex network with lite neural network, such as MobileNetV2.

4.2 Transformer-based Approach for Document Layout Understanding

4.2.1 Introduction

Rapidly growing digital documents have become a key part of information transformation. However, due to the various layouts and the complex structures of documents, automatically structured analysis of documents is crucial to speed up the transformation process (Figure 4.9). Document Layout Understanding (DLU) is a central step in automatic analysis, recognition of document structure, and information extraction out of document images. It leads to an important research direction for both Computer Vision (CV) and Natural Language Processing (NLP), and is a fundamental task of Document AI, which aims to automatically read, understand, and analyze documents.[119].

DLU plays an essential role in object detection tasks for document images to detect and recognize the fundamental components such as title, text body, figures, and tables in the document as objects. Some well-known deep learning-based object detection methods have been applied to DLU tasks, such as using a CNN-based neural network at pixel-level for document segmentation[46;120] and Faster R-CNN based architecture for document layout detection[13]. Meanwhile, recent work introduces and integrates text, visual features, spatial features as the multi-modal model for DLU tasks[121;122]. These additional information could help models obtain SOTA performance on relevant datasets. In this paper, we only use visual features for DLU tasks.

The attention-based transformer architecture has been widely employed in Natural Lan-

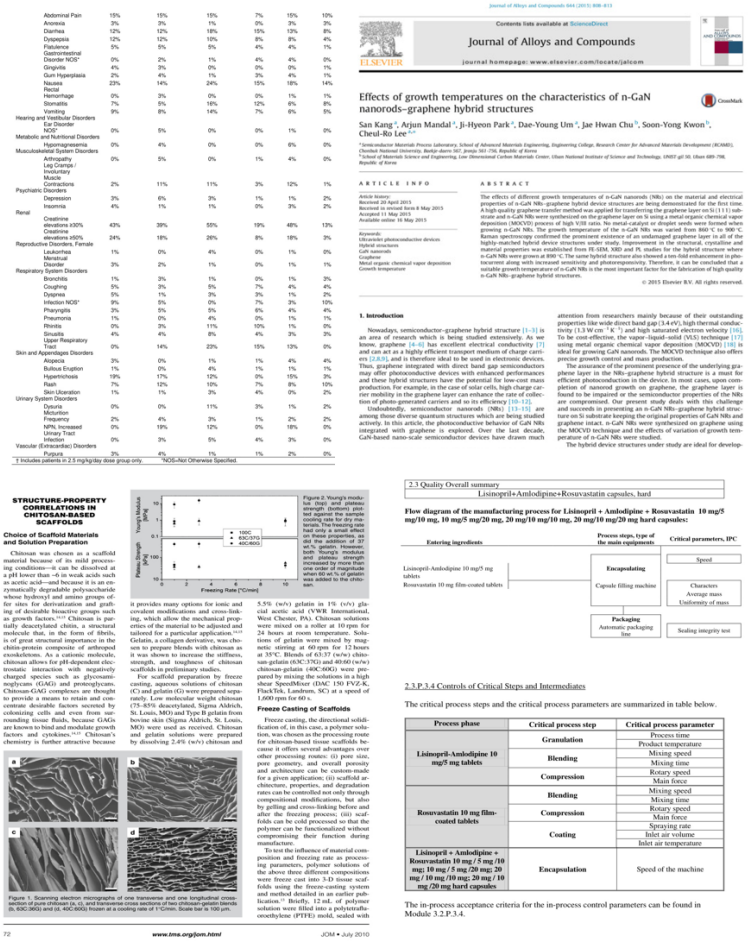


Figure 4.9: Examples of complexity layouts of document image.

guage Processing domain, and has been approved for its performance. It is also becoming increasingly attractive in recent object detection fields. Carion et al.[123] develop DETR which is the first transformer-based end-to-end object detection framework. In contrast to conventional one-stage and two-stage detection networks, it utilizes prediction methods which directly conduct bounding box predictions with Hungarian bipartite matching, instead of using the anchor and non-maximum suppression (NMS) mechanism. However, the slow convergence and inaccuracy of small object performance of DETR make it inefficient. Zhu et al.[124] propose Deformable DETR based on DETR architecture which solves DETR's issue of slow convergence and high complexity, and they have introduced the idea of deformable convolution[125] to the attention module.

Given the impressive performance of the transformer in the CV field, it will be interesting to see if we can also take advantage of it in the DLU area. Therefore, we propose a fully transformer-based framework for document layout understanding, namely TRDLU. The TRDLU is an end-to-end DLU detector with vision transformer - Swin Transformer[126] as the backbone for feature extraction from the input image, and connect with transformer encoder-decoder for document layout detection and recognition. This study integrates the most recent work in the transformer of the object detection area and outperforms the previous transformer-based as well as CNN-based object detection frameworks[123;124] for DLU task.

The main contributions of our paper are presented as follows:

- This study is the first one to introduce a fully transformer-based detector pipeline for the task of DLU method, namely TRDLU.
- the proposed detector pipeline outperforms the previous transformer-based detector on DLU tasks, and is even better than the multi-modal feature-based detectors;
- the experiment results show that TRDLU outperforms the previous state of the art in DLU benchmark datasets

4.2.2 Methodology

The overall TRDLU contains three main components: a transformer backbone, transformer encoder-decoder, and set prediction.The transformer backbone is used for visual feature extraction from the input images. The transformer encoder takes the feature in, and outputs the potential object features. The transformer decoder uses encoder outputs and object queries to generate final predictions for feed forward network (FFN). The final output will be generated by the set prediction process. The details of the detector pipeline are shown in Figure 4.10.

Transformer Backbone We use Swin Transformer[126] which is one of the state of the art architecture in the vision in transformer family as backbone for visual feature extraction

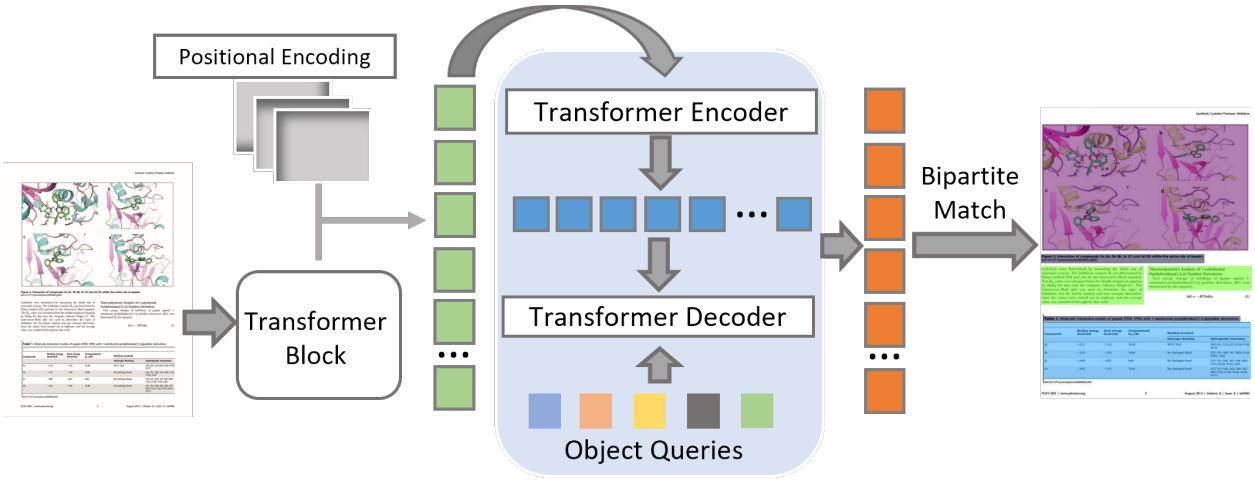


Figure 4.10: The architecture of TRDLU.

from input images. Considering the input image is $H \times W \times 3$, Swin Transformer first splits the image into 4 non-overlapping patches as tokens with the patch splitting module. Then it sets the patch feature as a concatenation of the pixel values, and feeds it into the first stage of the two-stage module through a linear embedding layer, followed by two Swin Transformer blocks. Starting from the second stage, the patch features will be concatenated into 4C-dimensional by the first patch merging layer and converted into 2C-dimensional features with a linear layer. Finally, the feature transformation will be achieved by applying Swin Transformer blocks. The steps in stage 2 will be repeated in the rest of the stages. We use Swin tiny version (Swin-T) which has 4 stages as backbone, and the layer number of each stage is 2, 2, 6, and 2, respectively. The final feature output is $f \in \mathcal{R}^{\frac{H}{32} \times \frac{W}{32} \times 8C}$, where C represents the channel dimension. In addition, the position information is added into the feature map, flattened to spatial feature map $f \in \mathcal{R}^{N \times D}$, and sent to the multi-layer transformer encoder, similar to the Deformable DETR[125].

Transformer Detector The novelty of this study is to combine merits of the most recent transformer-based works in CV, including the top-k object query[127], bounding box refinement and two-stage strategy[125], and auxiliary losses in encoder layer[128] to improve the performance in terms of accuracy and efficiency. This combination is integrated into the implementation of the transformer-based encoder-decoder detector which follows the

structure in Deformable DETR.

Transformer encoder-decoder We construct the basic architecture of the transformer encoder-decoder following the structure in Deformable DETR[125].

Transformer encoder employs a multi-scale deformable attention module. The output of the previous layer is considered as the input of the current layer, which will be combined with the positional embedding as object queries. The deformable-attention reduces computational complexity by considering only the relevant keys for each query instead of every pair of queries and keys. Because of the decrease in computational complexity, we add auxiliary detection heads into the encoder layer, which will not increase the cost pressure, but improve the model performance.

Transformer decoder employs self-attention and multi-scale deformable attention modules which contain object queries as query elements. The reference point is predicted for each query and used for the multi-scale deformation attention model to extract image features. To optimize the model result, the detection head is applied to bounding box prediction to predict the deviations from the box center where the reference point was placed initially. Hence, this process facilitates the speed of model convergence.

Top-k object query The Top-k object query mechanisms is introduced by Efficient DETR[127], where the encoder outputs can be used as decoder inputs and each of them is associated with an auxiliary detection head which computes a class score as a measurement of each output's objectness. The top-k encode outputs are then selected as the decoder queries based on the class score. We employ the top-k decoder query selection because it is identified to generate better results compared with the methods used in DETR[123] and Deformable DETR[125].

Bounding box refinement The implementation of bounding box refinement (BBR) follows the structure that is used in Deformable DETR[125]. The key idea of BBR is to refine the predicted bounding boxes by the current decoder layer based on the previous layer predictions. The predicted bounding box is represented by $b^d_{p\{,y,w,h\}} \in \mathcal{R}$, where d is the decoder layer and p is the coordinator of prediction bounding box. The BBR process is repeatable from the first decoder layer to the last decoder layer. The final refinement result

is returned by the last decoder layer. This iterative bounding box refinement mechanism can effectively improve detection performance.

Two-stage We apply the two-stage method which is introduced from two-stage Deformable DETR to our transformer detector. The object queries of the decoder layer in one-stage method are generated by predefined embeddings directly. Unlikely, the two-stage method first selects the top-k proposal boxes in the first stage based on their class scores, and feeds the selected boxes into the decoder and set positional embeddings of object queries as positional embeddings of region proposal coordinates during the bounding box refinement process. These object queries are more relevant to the current image. Following the two-stage Deformable DETR, we use multi-scale feature maps to generate anchors for each position and set the base anchor scale to be equal to 0.05. Then C (C is number of classes) category scores and 4 offsets per anchor are predicted by the detection head.

Loss Function For the bounding box loss function, we use Distance Intersection over Union[129] with l_1 loss:

$$\mathcal{L}_{\text{box}}(b_{\sigma(\text{i})}, \hat{b}_{\text{i}}) = \lambda_{\text{diou}}\mathcal{L}_{\text{diou}}(b_{\sigma(\text{i})}, \hat{b}_{\text{i}}) + \lambda_{\text{L1}}||(b_{\sigma(\text{i})} - \hat{b}_{\text{i}})||_1 \tag{4.1}$$

where λ_{diou}, λ_{L1} are hyper-parameters, $\mathcal{L}_{\text{diou}}$ is the distance IoU loss. The Hungarian loss function is used to calculate the classification loss and bounding box regression loss between prediction and ground truth:

$$\mathcal{L}_{\text{Hungarian}}(\bar{y}, \hat{y}) = \sum_{i=1}^{N} \left[\mathcal{L}_{class}^{i,\hat{\sigma}(i)} + 1_{\{\bar{y_i} \neq \emptyset\}} \mathcal{L}_{box}^{i,\hat{\sigma}(i)} \right] \tag{4.2}$$

4.2.3 Experiment

We evaluate TRDLU on three different benchmark datasets. Two of them are document layout related datasets, and one is a table detection dataset. For fair comparisons, we use MS-COCO evaluation metric which is the same evaluation metric used by each benchmark.

Benchmark Datasets

Scientific Literature Regions (SLR) is a synthesis dataset of DLU. It contains 1660 document images which are captured from three existing datasets: Article Regions[115], ICDAR-2013[116], and GROTOAP[117]. This dataset includes 11 classes corresponding to the main regions of documents, including Title, Author, Address, Abstract, Keyword, Body, Figure, Table, Caption, Reference, and Text.

PubLayNet[130] is a large dataset for document layout analysis. The document layout is labeled by bounding boxes and polygonal segmentations. This dataset contains 360K document images and 5-region annotation classes: Title, List, Text, Figure, and Table. The ground truth of the test set is not released because the authors want to keep it for the competition. Therefore, we evaluate our model on the validation dataset.

TNCR[131] is a table detection dataset. It contains 9428 labels with 6612 document table images. This dataset includes 5 different classes to present the various table formats of scanned document images: No lines, Partial Lined, Merged Cells, Partial Lined Merged Cells, and Full lined.

Table 4.2: Detection results comparison on Scientific Literature Regions (SLR) and TNRC datasets.

Detector	Dataset	mAP	AP50	AP75	APs	APm	APl	AR
Faster R-CNN[13]	SLR	76.24	93.52	85.77	62.78	63.67	76.33	81.31
Cascade Mask R-CNN[132]	SLR	79.92	94.36	88.30	70.75	69.84	81.26	88.60
Deformable_detr	SLR	80.61	95.50	88.50	58.70	66.90	83.30	87.70
TRDLU (ours)	SLR	**82.70**	**96.40**	**90.70**	**75.40**	**73.30**	**83.60**	**89.20**
Deformable_detr[131]	TNRC	86.70	93.80	87.40	-	-	-	89.60
TRDLU(ours)	TNRC	**90.60**	**93.90**	**92.50**	-	-	-	**98.10**

Implementation Details We use pre-trained Swin-Tiny Transformer[126] backbone network. The transformer includes 6 encoder and 6 decoder layers associated with the auxiliary detection head for each layer. The models are trained on Nvidia A40 GPU machine. We set batch size to 2 to train the models for 50 epochs on Scientific Literature Regions and for 30 epochs on TNCR datasets, respectively. The initial learning rate is set to 0.0002 and decays by 1/10 after the 40th and 25th epochs. For PubLayNet dataset, the model is trained by

Table 4.3: Detection result comparison on PubLayNet dataset.

Method	Text	Title	List	Table	Figure	mAP
VSR[121]	96.70	93.10	94.70	97.40	96.40	95.70
DocSgeTr[133]	89.90	73.60	89.50	97.50	96.60	89.40
TRDLU (ours)	95.82	92.13	**97.55**	**97.62**	**96.62**	**95.95**

batch size 4 for 10 epochs with an initial learning rate of 0.0002 and decays by 1/10 after the 8th epoch. The rest hyperparameters are the same as those in Deformable DETR.

Performance comparison We compare TRDLU on three DLU task-related benchmark datasets with the same tasks using state-of-the-art detection approaches. For SLR and TNCR datasets (Table 4.2), TRDLU outperforms all other methods and improves the mAP (mean average precision) up to 2.9 percent on SLR and 3.0 percent on TNCR. It also increases the AR (average recall) by 1.5 percent and 8.5 percent on SLR and TNCR, respectively. Table 4.3 shows the comparison results on PubLayNet. The TRDLU outperforms most other methods, and it is even better than the results using the VSR[121], a multi-modal framework.

Attention result analysis Figure 4.11 shows the attention map visualization results. The encoder could recognize the potential objects. It participates in the instance separation process, and gives the approximate object location. The decoder cloud gives the precise bounding boxes for different objects after model training. The attention visualization results can help us gain intuitions regarding how attention mechanisms work.

4.2.4 Conclusion

In this chapter, we present an end-to-end transformer-based framework for document layout understanding, namely TRDLU. It integrates the merits of the most recent research works in this field. It is the first study of a fully transformer-based framework, and outperforms the experiential results generated by other research on both CNN-based and transformer-based frameworks.

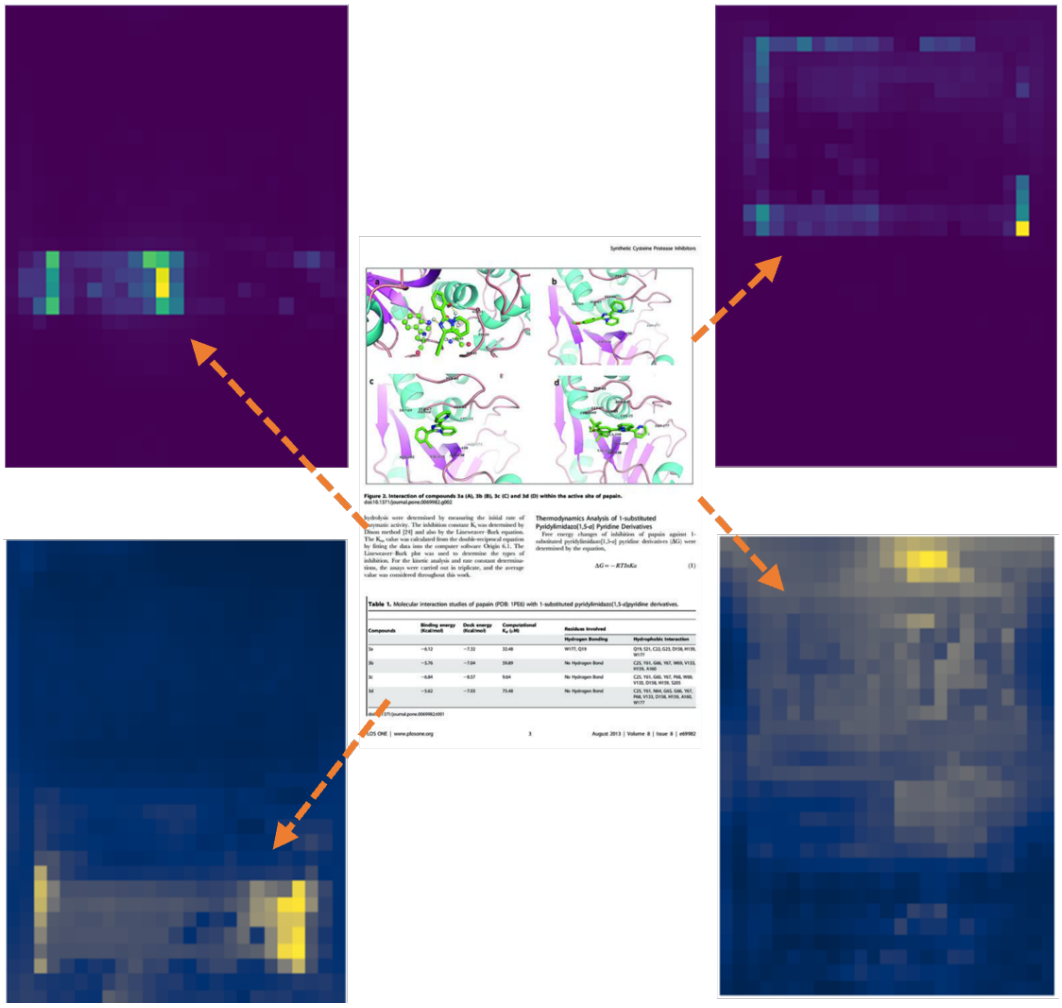


Figure 4.11: Attention map visualization of TRDLU. The middle image is the input image. The two upper figures represent the decoder attention map, the lower two figures represent the encoder attention map.

Chapter 5

Automatic Metadata Information Extraction from Scientific Literature using Deep Neural Networks

5.1 Introduction

This chapter addresses the task of simultaneous layout and free text recognition using a hybrid deep learning architecture that combines mask and cascade variants of Recurrent Convolutional Neural Networks (RCNN) and Convolutional Recurrent Neural Network (CRNN). We are motivated by the fact that the research community has expanded dramatically and the number of published scientific reports across scientific fields grows enormously each year. Research shows that the volume of daily publication doubles every 15 days[134]. Scientific literature thus includes much valuable information for researchers to help them extract key insights[94] and potential methods[12] in their respective research area. The time required to select and systematically read this increasing body of scientific literature presents a major challenge to researchers. Natural Language Processing (NLP) technology provides an efficient way for scientists and researchers to gain key insights from published articles, particularly by helping to find relevant papers related to their research area[135]. These efficient

methods must, however, be trained using large corpora. Reconstructing this large corpus automatically is an ideal task for scientific literature-related NLP.

The digital documents have been most used in scientific publications, such as Portable Document Format (PDF) documents which are not machine-readable. The information of text, image, or table from these digital documents must be extracted for further processing. Some existing tools can help extract text information from digital scientific literature. For instance, PyMuPDF[97] or Tesseract Optical Character Recognition (OCR) engine[136] can help extract plain text from PDF documents. These tools, however, do not provide a comprehensive solution that combines layout segmentation and text recognition to produce structured metadata for the scientific literature such as a paper's title, authors, abstract or bibliographic references. CERMINE[57], an open-source system for extracting structured metadata information from scientific articles, can generate an XML file that includes labels for each region of an article as well as content text, but it is not flexible enough to adapt to various formats of scientific literature, e.g., it cannot process three-column PDF scientific documents. Thus, a robust comprehensive framework is urgently needed.

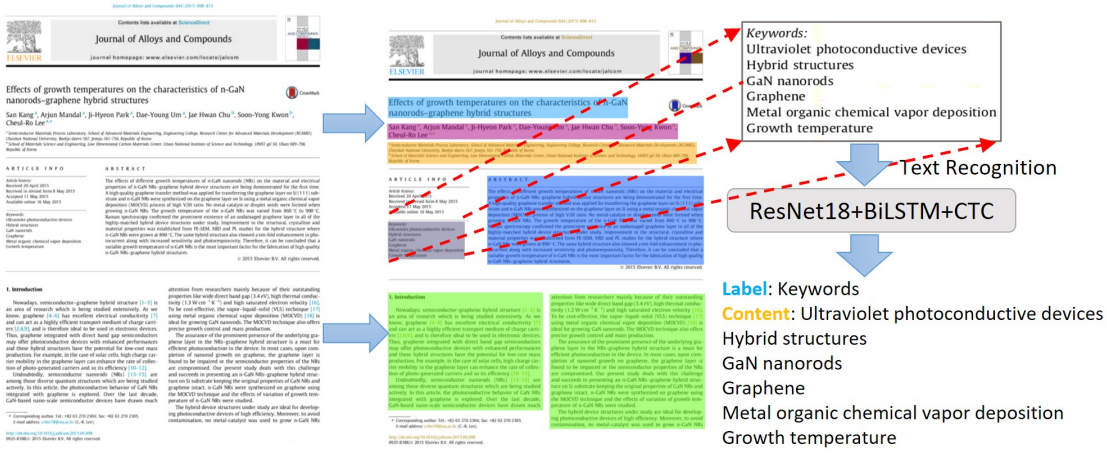


Figure 5.1: Example of metadata information extraction from a scanned scientific literature with end-to-end framework.

To address the limitations mentioned above, we present an end-to-end learning framework for neural networks, as designed for the task of comprehensive metadata information extraction from scientific literature. In this work, our novel contribution integrates an ob-

ject detection model based on Cascade R-CNN[132] and Mask R-CNN[137] that can be used to detect main region layout in scientific literature and text recognition model based on CRNN[138]. The output gives the labels of corresponding regions, such as title, author, and abstract, and the content text of each main region (Figure 5.1). We also extract images and tables without processing the content into text. We train the framework separately, and the experiment results demonstrate that the applied transfer learning with fine-tuning in a pre-trained model works well even for tasks that lack large, annotated corpus. Meanwhile, we create a new annotated data set for scientific literature layout detection tasks. The data set will be available at: https://github.com/huichentt/scientific_literature_regions.

5.2 Methodology

We treat metadata information extraction from scientific literature as detection and recognition tasks, which our system performs in two independent stages whose results are then integrated within our end-to-end framework. In the first stage, the detection model identifies the potential key regions of the scientific literature in the rectangular boxes as well as the corresponding text blocks. In the second stage, the recognition model recognizes and transcribes the words from the regions that have been detected. This two-stage process has two main advantages: the flexibility of training process and the independent recognition of different languages. We describe our approach in depth in the following section.

Scientific Literature Layout and Text Detection

In scientific literature, metadata elements or attributes such as title, author, abstract, etc., are considered major regions which can be delimited using different bounding boxes within a scanned document image. We consider the problem of scientific literature layout detection as one of object detection due to the similar characteristics between the underlying formal tasks. Besides the layout of major regions in scientific literature, lines of text also need to be detected in the text recognition stage. Unlike the scene text detection task, text in scientific literature does not involve complex backgrounds or irregular fonts. Therefore, we perform the text line detection using the same object detection model. Our experiment

shows the feasibility of this approach.

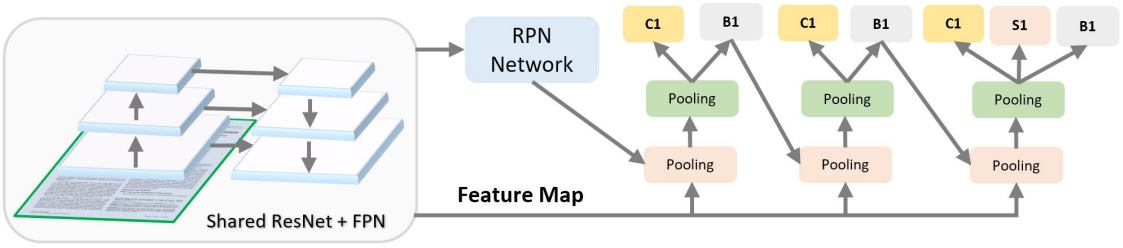


Figure 5.2: Architecture of Cascade Mask R-CNN. The segmentation branch is added to each cascade stage."C" is classification,"S" is segmentation branch, and "B" is bounding box.

Our novel neural architecture combines Cascade R-CNN and Mask R-CNN within our scientific literature layout and text detection framework. Cascade R-CNN is an extension of the Faster R-CNN[107] which is a classic two-stage object detection network. The first stage of Faster R-CNN uses region proposal network that takes the feature map as input and generates the hypotheses of object proposals. These hypotheses would be processed by a region-of-interesting network to generate the detection head in the second stage. Each hypothesis is generated based on a final classification score and a bounding box. Unlike Faster R-CNN with single R-CNN network, Cascade R-CNN extends the Faster R-CNN to multi-stage with cascaded regressor. The different stages have different heads $\{H_1, H_2, ..., H_N\}$ (where N represents the number of heads) and each of the heads is assigned to a specific IoU (Intersection over Union) threshold which defines the quality of a detector to produce the bounding boxes that are loosely aligned with (small threshold) or tightly aligned with (large threshold) their ground truth. The multiple specialized regressors $\{f_T, f_{T-1}, ..., f_1\}$ (where T represents the total number of cascade stages) are optimized for the distributions that are resampled for the different heads. This resampling procedure of cascaded regression provides good positive samples for the next stage. Those features of Cascade R-CNN show better performance than Faster R-CNN. On the other hand, Mask R-CNN is another state-of-the-art object detection network and is modified based on Faster R-CNN. Compared with Faster R-CNN, the Mask R-CNN introduces ROIAlign to replace the ROI Pooling layer of Faster R-CNN to resolve the pixel-to-pixel misalignment issue between network inputs and

outputs. Moreover, Mask R-CNN adds a segmentation branch for instance segmentation. The final loss function is composed of three parts: classification, bounding box regression, and mask loss.

To integrate Cascade R-CNN and Mask R-CNN, we use ResNet-50[109] with a Feature Pyramid Network (FPN)[139] as the backbone network for feature extraction from input images. The RPN network generates the proposal regions which are then fed into ROI pooling layer. The network head takes ROI pooling as input and generates three predictions: classification (C), mask (S), and bounding box regression (B). The output of one stage is used as input for the next stage (Figure 5.2). Therefore, the current bounding box distribution is generated by the previous regressor to optimize the current regressor. The final loss function combines classification and location (bounding box regression) loss functions at different stages (1).

$$\mathcal{L} = \sum_{i=0}^{2} (\mathcal{L}_{\text{cls\_S}_i} + \mathcal{L}_{\text{loc\_S}_i}) + \mathcal{L}_{\text{cls\_FPN}} + \mathcal{L}_{\text{loc\_FPN}} + \mathcal{L}_{\text{mask}} \tag{5.1}$$

In the equation above, cross-entropy loss is used as the classification loss function, and smooth L1 is used as the bounding box regression loss function. The total loss function is calculated at three cascade stages (denoted by i ranging from 0 to 2), FPN network, and mask branch. We choose GIoU (Generalized Intersection over Union) (2)[140] instead of IoU as increasing thresholds over stages. The GIoU handles the case of non-overlapping bounding boxes that is inapplicable for IoU.

$$GIoU = IoU - \frac{|C_{ab} - U|}{|C_{ab}|} \tag{5.2}$$

In the equation above, C represents the smallest enclosing convex object of a and b where a and b are two arbitrary convex shapes. U represents the area of C that does not belong to either a or b. The bounding box regressor trained for a certain GIoU threshold tends to

produce bounding boxes of higher GIoU threshold. The segmentation branch has been added to the last cascade stage. Furthermore, the anchor box aspect ratio is another important thing to be considered in object detection tasks since different objects have different aspect ratios. For instance, the aspect ratio box of reference is different from the aspect ratio box of text block in scientific literature layout detection tasks. Our experiment results demonstrate that selecting appropriate aspect ratio brings a higher accuracy level.

Text Recognition

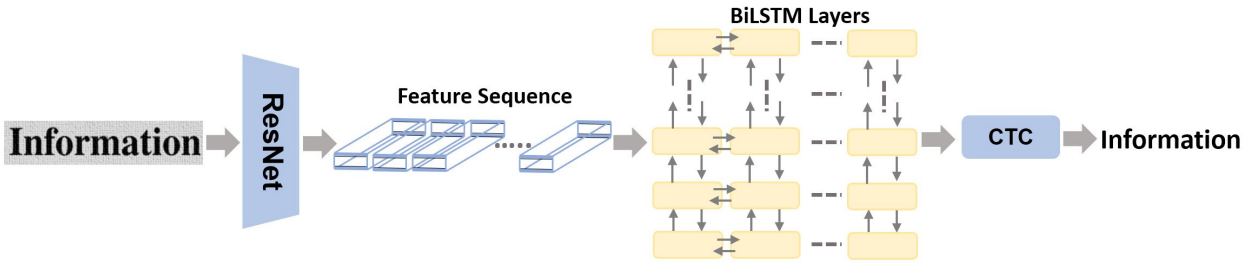


Figure 5.3: Architecture of text recognition model.

We propose a text recognition model based on an CRNN architecture which outputs a sequence of characters (Figure 5.3). The model uses Convolutional Neural Network (CNN) to extract visual features from input images. We use ResNet18 as backbone for visual feature extraction. Then the feature vector will be used to predict the character label distribution of each frame through the sequence modeling stage - a multi-layer Bidirectional Long-Short Term Memory (BiLSTM) network[141] which is used for sequence feature extraction. The BiLSTM layer will generate a number of layer-specific hidden state sequences, h^1, h^2, ..., h^N, where N is the number of layers in the BiLSTM. The encoder layer is followed by a softmax to produce a posterior probability matrix $y = (y^1, y^2, ..., y^T)$ where T is the length of the sequence. The last transcription layer is implemented by Connectionist Temporal Classification (CTC) loss[142]. CTC allows us to find the optimal label with the highest conditional probability by using dynamic programming to compute all potential alignment paths for predicting the probability distribution of all characters of the alphabet at each position in the image.

$$\mathcal{L} = \log \sum_{c \in A(s)} \prod_{t=1}^{T} p(c_t | h_t^N) \tag{5.3}$$

In the above loss function, h_t^N represents the probability of having N_{th} hidden layer at time stamp t. The $c_t \in C$ is a label set, consisting of all the possible output labels and a "blank" symbol. A(s) includes all possible paths of sequence $\mathbf{s}$, where $\mathbf{s} = \text{s}_1, \text{s}_2, ..., \text{s}_\text{L}$. The alignments between those predictions could include duplicate or blank characters. For instance, the word "STUDY" could be processed into "S-T-UU–DD-Y" such that there are duplicate and blank characters. Assuming that the optimal labeling is from the most probable path, we can find the most probable path based on the most likely character at each position of the sequence. The duplicate characters will be removed then if there is no blank character between them. The loss function is used to jointly learn all the model parameters.

5.3 Experiment

In this section we describe the data set that we use to train the models, together with implementation details for training process.

Data Set

We use two data sets to train the framework independently.

Scientific Literature Layout and Text detection

Due to the limited availability of training corpora, we developed a new corpus by annotating the composite corpus (synthesis data set) from Yang et al.[13]. This experimental test bed combines three existing data sets: region-labeled articles[115], ICDAR-2013[116], and GROTOAP[117]; however, it does not include text block-related annotation. Therefore, we labeled text blocks using 110 images from 20 scientific literature in PDF format. The final data set has 1660 images from 383 scanned scientific articles in PDF format, and it includes 11 labels corresponding to main regions.

- ***Title***: the title of scientific literature.
- ***Authors***: the authors' names.
- ***Address***: the affiliation information of authors, including authors' addresses, email, etc.
- ***Abstract***: an abstract section.
- ***Keyword***: the selected keywords.
- ***Body***: the main block of articles.
- ***Figure***: all figures but excluding logos or icons from publishers.
- ***Table***: the tabular contents.
- ***Caption***: the captions for both figures and tables.
- ***Reference***: the bibliography information, excluding post-references notes.
- ***Text***: text block.

Text Recognition

We use a combined corpus of data sets to train the network for text recognition. Three different data sets are integrated into the composite data set. The first one is English2k which is the sub-data set of SCUT_FORU_DB[143]. It has 1715 natural images for text detection and recognition from the Flickr website. We crop 7136 images of characters from this data set since we only focus on text recognition tasks. The second and third data sets are generated by Text Recognition Data Generator, which is an open-source tool to generate synthetic data for text recognition. We use three categories to produce the composite data randomly for generating:

- 2000 images of letters and numbers, containing 3, 4, 5, 6, 7 characters, respectively.
- 2000 images of letters only, containing 3 to 7 characters, respectively.

- 5000 images of letters, numbers and symbols, containing 5 characters.

The final combined corpus for text recognition consists of 30136 images. We split them into 24000 images for training and the rest are used for testing.

Implementation Details

All models are trained and tested on a single NVIDIA 2080 Ti GPU. The framework of scientific literature layout detection and text detection is implemented on Pytorch using Detectron2[118] and fine-tuned with pre-trained weight on MS COCO data set[106] (37 epochs). It has been trained with a batch size of 4 for 50 epochs. The SGD (Stochastic Gradient Descent) is used as an optimization algorithm. We use 0.002 as the initial learning rate which decays by 0.1 after every 20 epochs. The Cascade Mask R-CNN model is trained with GIoU threshold values of 0.5, 0.6, and 0.7, with 5 anchor scales from 32 to 512, and with 4 anchor aspect ratios: [0.1, 0.5, 1.0, 2.0]. The rest of models are trained by standard aspect ratios with [0.5, 1.0, 2.0]. We also use YOLOv3[112] which is trained based on DarkNet53 network that was pre-trained on MS COCO data set.

Compared to the detection model which processes the images with sizes close to 612 $\times$ 729 pixels, the text recognition model processes much smaller input images at 32 x 280 pixels. Thus, we use ResNet18 as CNN layer to extract the feature from images. Our experiment indicates that using ResNet18 increases the accuracy by 7% compared with VGG16[95]. To consider the robustness of the recognition model, we add some scientific literature-related symbols, such as $\pm$, °C, $\neq$, to the dictionary to train the model. The text recognition model has been trained from scratch with 50 epochs.

5.4 Evaluation

We evaluate our end-to-end framework from two different perspectives. First, we use the same data set to train and test other different configuration models to compare with our Cascade Mask R-CNN model for scientific literature layout and text detection. The detection results may affect the performance of the final framework, because the text recognition model

relies on the layout and text detection results. Second, we manually determine ground truth from 10 scientific articles for metadata extraction evaluation. It includes the main regions of scientific literature that are consistent with our training data set. Given the various layouts from different publishers, this ground truth is collected from different publishers for robustness testing.

Scientific Literature Layout and Text Detection Evaluation

We use MS COCO evaluation metrics for layout and text detection testing. COCO provides 12 indicators to evaluate the performance of object detector. We use 7 of these for our detector evaluation. The Faster R-CNN with ResNet50_FPN is baseline, and we compare it with different other object detection models using different backbones. Our Cascade Mask R-CNN model achieves better performance than others for scientific literature layout and text detection tasks (Table 5.1).

Table 5.1: Overall comparison of bounding box results for scientific literature layout and text detection.

Model	**Backbone**	**mAP**	**AP50**	**AP75**	**APs**	**APm**	**APl**	**AR**
Faster R-CNN (**baseline**)	ResNet50_FPN	75.79	92.51	84.24	59.18	63.82	76.71	78.95
Faster R-CNN$_{_GIoU}$	ResNet50_FPN	76.24	93.52	85.77	62.78	63.67	76.33	81.31
Faster R-CNN	VoVNetV2-39	76.52	93.23	86.50	72.32	61.63	74.66	77.28
YOLOV3	DarkNet53	55.36	73.65	64.10	-	-	-	53.26
Mask R-CNN	ResNet50_FPN	77.25	92.22	85.60	64.38	65.23	77.65	83.96
Cascade Mask R-CNN (**ours**)	ResNet50_FPN	**79.92**	**94.36**	**88.30**	70.75	**69.84**	**81.26**	**88.60**

Likewise, for each label, our Cascade Mask R-CNN model also outperforms the baseline Faster R-CNN (Figure 5.4). More examples of layout and text detection are shown in Figure 5.5.

Metadata information extraction evaluation

We use precision, recall and F1 to evaluate the performance of metadata information extraction.

$$Precision = \frac{N_{\mathrm{C}}}{N_{\mathrm{T}}}, \quad Recall = \frac{N_{\mathrm{C}}}{N_{\mathrm{G}}} \tag{5.4}$$

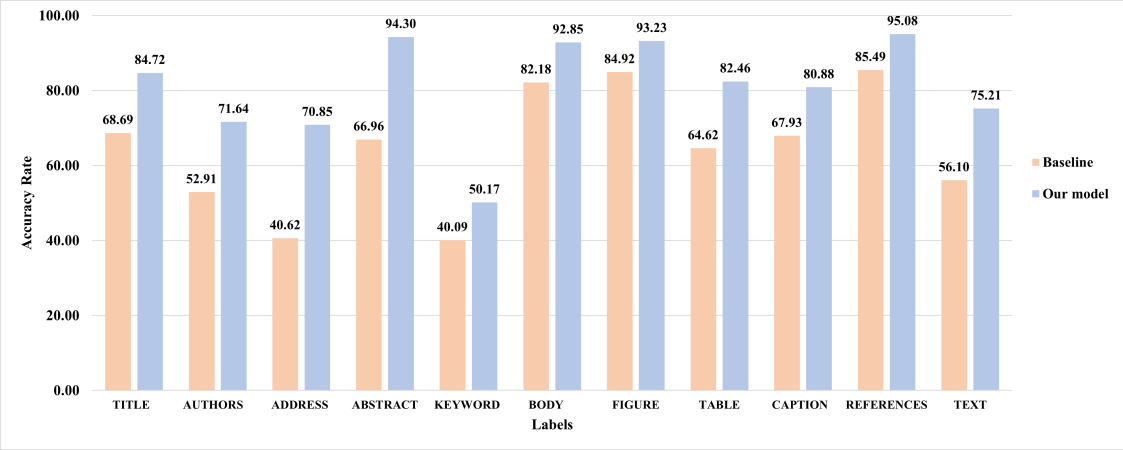


Figure 5.4: Detection results comparison with baseline by labels at 0.5 IoU.

where N_C is the number of words extracted correctly by our framework, N_T is the total number of words extracted, and N_G is the number of words in ground truth. To be clear, the extracted word is considered correct if all characters match the word of ground truth. In addition, we consider the subscript or superscript character as a regular part of the word and ignore the different font styles. Moreover, the metadata information does not include any text from images or tables. We conduct these three tests for each label and compare them with CERMINE (Table 5.2).

Table 5.2: The evaluation results comparison between our approach and CERMINE.

Label Name	**Precision(%)**	**Recall (%)**	**F1 (%)**	**Precision(%)**	**Recall (%)**	**F1 (%)**
	Our approach			CERMINE		
Title	90.92	88.73	89.81	91.67	89.27	90.45
Authors	**82.57**	**80.36**	**81.45**	74.75	67.37	70.87
Address	**87.22**	**79.69**	**83.29**	87.16	79.29	83.04
Abstract	**81.26**	**78.59**	**79.90**	75.18	75.17	75.17
Keyword	**92.93**	**91.56**	**92.24**	77.84	64.84	70.75
Body	75.23	77.49	76.34	83.56	79.79	81.63
Caption	**79.16**	**77.86**	**78.50**	22.50	22.50	22.50
Reference	**62.73**	58.52	60.55	59.77	62.55	61.13

The metadata information extraction results from text recognition model heavily depend on the training data set. Our ground truth in this evaluation is collected from scientific papers, particularly related to biology and materials science. Some domain-specific characters such as unit symbols in chemistry actions need to be considered to add for further improve-

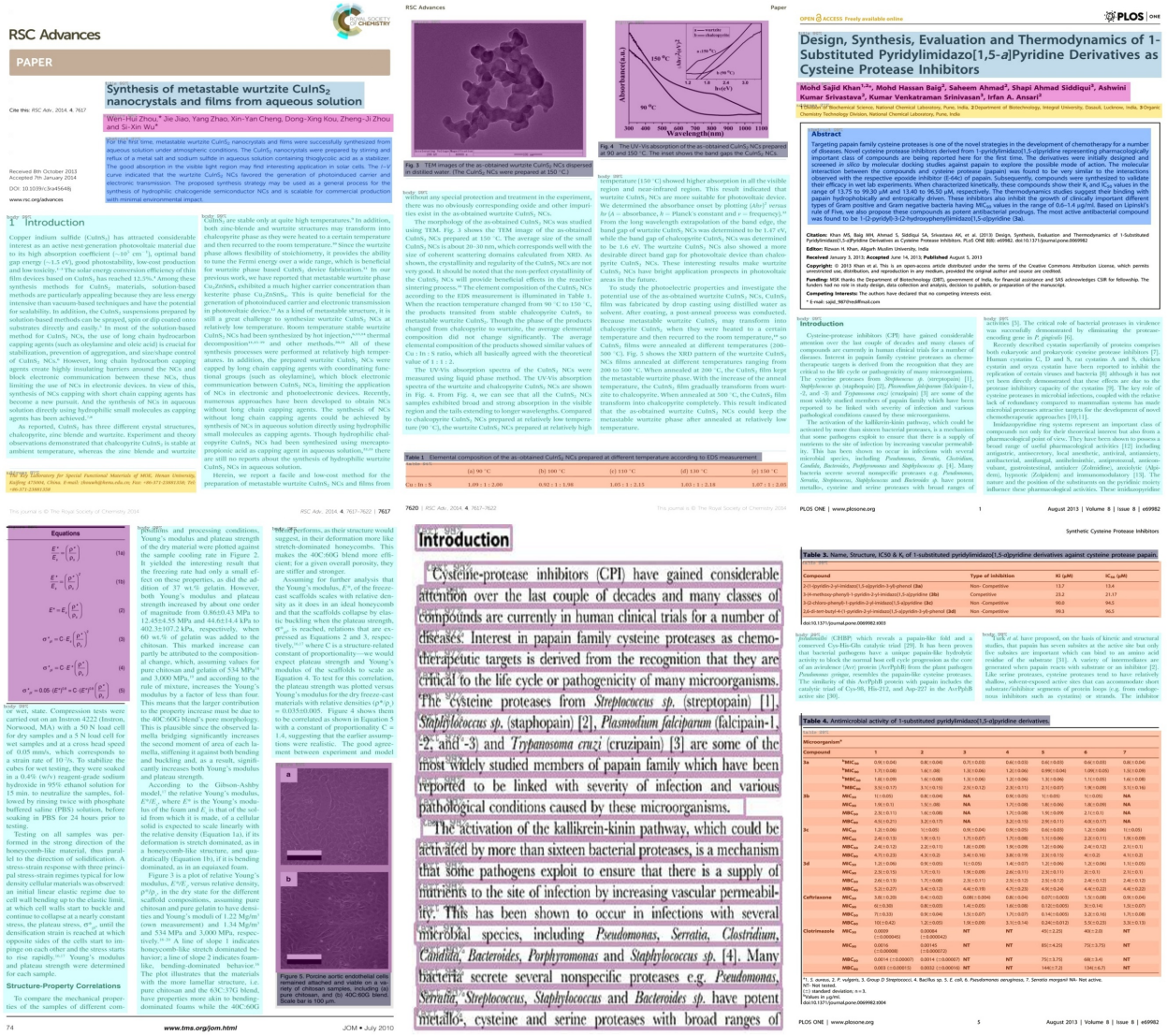


Figure 5.5: Layout and text detection results with corresponding labels.

ment. Moreover, evaluation results are also affected by resolution of input images. Cropping clear images from layout and text detection stage is a necessary preprocessing step.

5.5 Conclusion

In this chapter, we have introduced an end-to-end framework for metadata information extraction from scientific literature. The framework integrates two models: scientific literature layout and text detection based on Cascade Mask R-CNN, and text recognition based on

CRNN. Such integration enables us to both input scanned scientific article images and output the text with corresponding labels at once. This framework has flexibility for training to adopt diverse layouts of scientific literature from different publishers. Meanwhile, it can recognize different languages independently. In addition, we create a novel data set for scientific literature layout and text detection tasks. Our proposed Cascade Mask R-CNN model provides a benchmark with this data set.

In future work, we will examine the application of this end-to-end framework to the development of a large corpus for a range of domain-specific NLP tasks that currently lack such corpora, such as information extraction for knowledge base population, question answering, and dynamic search and indexing in the materials science domain. As we have seen from this work, there is a potential trade-off between the model performance and its accuracy that requires further attention, such as balancing between reducing the number of parameters of the model and maintaining/improving the accuracy.

Chapter 6

Named Entity Recognition from Synthesis Procedural Text in Materials ScienceDomain with Attention-Based Approach

6.1 Introduction

The number of published materials science articles has grown rapidly over the past few decades. Much potentially useful information in these published articles could help the materials design group explore and study new material synthesis. Conventionally, new materials are discovered mainly through published experiments in literature, which, however, are usually stored as unstructured text format. This requires great effort to sort and organize. Furthermore, researchers and scientists in materials science cannot access much more than a fraction of such information because their research time is limited. The inevitable result is, therefore, the need to enhance their ability to identify new technologies and find the appropriate literature[144].

Natural Language Processing (NLP) with machine learning technology can accelerate

the rate of materials science discoveries. Many materials science areas, thermoelectrics, photovoltaics, batteries, and pharmaceuticals, could use these techniques[145]. The fundamental task, then, of NER in NLP is to recognize named entities in the text of published experimental research and group them into pre-defined categories through classification[146]. In this paper, we focus on NER in the synthesis of procedural text in materials science. The synthesis procedures are defined as the order of the steps based on "participating tagged entities and ultimately roles and operations" that should be in methods sections of materials science research literature[12]. Those tagged entities could be material names, operations, and devices, among others. They are essential to extracting procedural information from materials science literature. Figure 6.1 shows an example of named entities from synthesis procedure text in a materials science article.

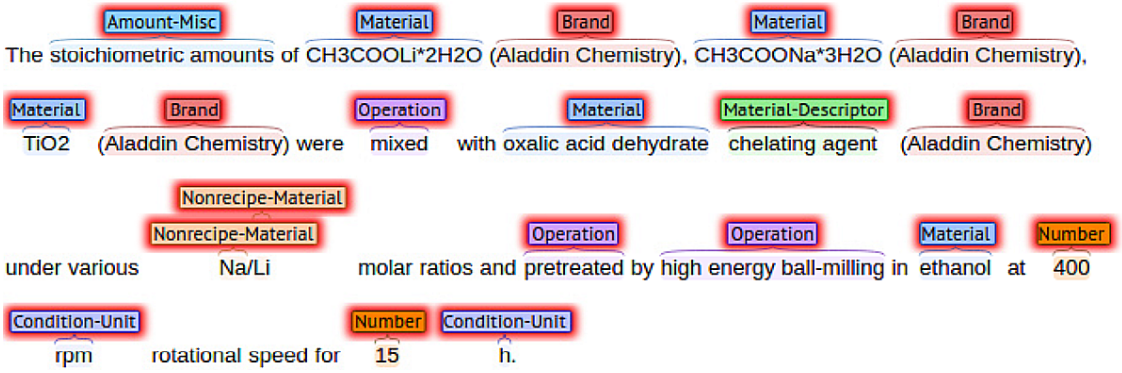


Figure 6.1: Example of named entities from synthesizing procedural text in materials science literature[1]. The highlighted words and phrases indicate entities involved in synthesis procedures.

In materials science, the particular challenge is insufficient annotated corpora; domain experts find labeling very expensive and time-consuming. To address the challenge in materials science, we used word embedding[92] with a BiLSTM (bidirectional LSTM) and a CRF (Conditional Random Fields) layer[147] as our base line model. We used BERT[81] pre-trained language model to compare contextual embedding to word embedding and then fit the output form BERT into a BiLSTM CRF model to learn the appropriate context information that would predict named entities. Our experiment results were based on three corpora of materials science and show the BERT-BiLSTM-CRF model improves significantly on other

models.

6.2 Related Work

Named entity extraction from published experimental research is an emerging field, attracting attention from many researchers. The most recently used approaches can be summarized into two types:

The first approach is entity extraction from materials science literature. This approach uses NER for extracting summary-level information from materials science documents. These named entities are broadly pre-defined in materials science as material name, sample descriptors, and material properties, among others[144]. The common extraction method collects relevant literature, uses unsupervised learning methods like K-means and Word2Vec[92], extracts word representation features from large unlabeled corpora, and then fits these word representation vectors along with small annotated corpora to machine learning models like CRF, decision tree with a linear classifier, and hierarchical neural networks for named entity extraction[90;94;148]. The extraction results can be stored in a database as structured data for queries.

The second approach is named entity extraction from synthesis procedural text of materials science literature. This approach uses NER to synthesize procedural text (or experimental methods) in the methodology sections of materials science publications. Compared with summary-level NER in materials science, this approach centered on details of entities involved in the experiment itself, including material names and operations in the experiment steps. Some previous research focused on this approach. Mysore et al.[149] extracted procedural information with action graphs, and Huo et al.[135] used semi-supervised learning methods with latent Dirichlet allocation (LDA), and random forests to classify inorganic materials from methodology information. We chose to use NER to synthesize procedural text as our main methodology.

6.3 Methodology

We treated NER as a sequence labeling problem. BERT-BiLSTM-CRF, the attention-based, deep learning, end-to-end model, was used to solve this problem. Figure 6.2 shows the structure of BERT-BiLSTM-CRF model. The pre-trained BERT model[81], as the embedding layer, received the raw input sentences. Then the BERT model output the contextual embedding vectors for each word as input to the BiLSTM layer for syntactic and semantic feature representation learning. The final CRF layer output possible tag sequences based on their conditional probability.

BERT Pre-trained language model

BERT[81] is a pre-trained language model based on a deep transformer encoder[150]. It introduced a masked language model (MLM) and next sentence prediction (NSP) to optimize the training process. These mechanisms allowed BERT to use an attention-based, multilayer, bidirectional transformer mechanism and a normal nonlinear layer to learn contextual information from large unlabeled corpora. Moreover, the pre-trained BERT language model can be easily fine-tuned for a particular downstream task. It is precise because BERT can use contextual information learnability and transferability instead of context-independent word embedding like Word2Vec[92]. This meant we could use BERT as the embedding layer. After fine-tuning, BERT performed well, even though it was pre-trained with corpora irrelevant to materials science; our NER task demonstrated this ability in our experimental results.

Bidirectional LSTM layer

The bidirectional LSTM is an extension of LSTM that applies a forward and backward LSTM network to sequence processing and links the network to the output layer[147]. The BiLSTM structure enables the output layer to gather contextual information simultaneously from past (backward) and future (forward). In addition, the BiLSTM has LSTM characteristics that avoid gradient vanishing and exploding that occur in RNN. Both forward and backward LSTM networks use the same equations in LSTM.

The BiLSTM takes the embedding result from BERT as an input vector for extracting sentence features. The output of the hidden state of BiLSTM will concatenate the forward

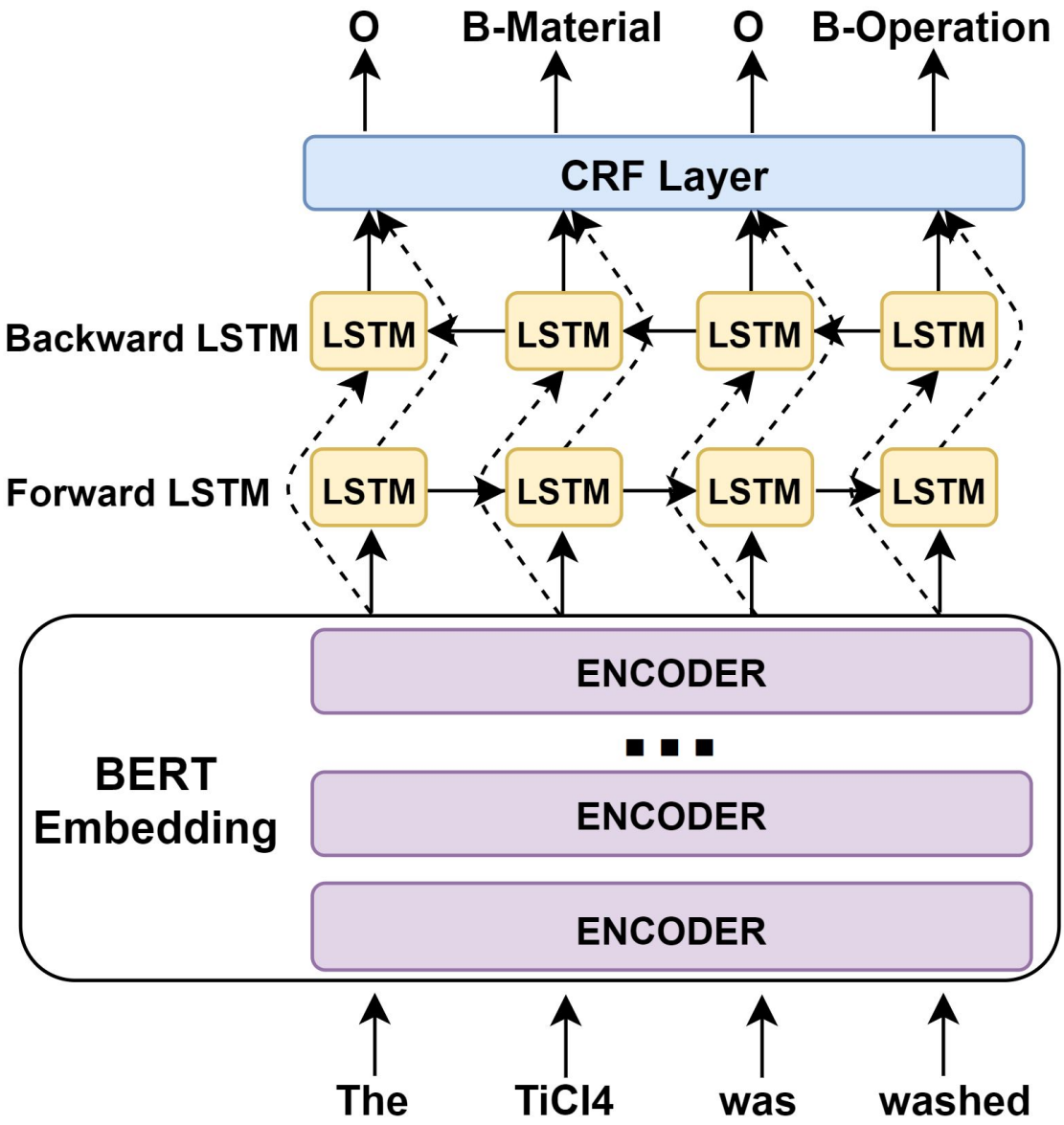


Figure 6.2: The architecture of the BERT-BiLSTM-CRF model.

LSTM H_f and backward LSTM H_b networks as final output $[H_l, H_r]$.

CRF layer

CRF is discriminative probabilistic method subject to a certain correlation constraint among tags. Using CRF as the last layer can help models learn the joint relationship between tags, as well as learn the constraints that ensure the sequences are valid. For instance, in BOI tagging format, the label of the first word in a sentence should start with the tag of "B" or "O", but not "I". These constraints are learned automatically using the training dataset

created by the CRF layer during the training process.

Label prediction of the CRF layer combines the output P from the BiLSTM layer, which represents the score of the i_{th} word in the sentence where y_i is the tag of the i_{th} word, and the transition matrix T represents the transition probability from tag y_i to tag y_{i+1}. We used the following equation to calculate the score of the labels sequence:

$$Score(X, Y) = \sum_{i=1}^{n} P_{i,\, y_i} + \sum_{i=0}^{n} T_{y_i,\, y_{i+1}} \tag{6.1}$$

Our goal is to minimize the loss function by maximizing the total score of the probability of sequence $s^{(X,\, y_i)}$. The log loss function is given as follows:

$$\mathcal{L} = Score(X, Y) - log \sum_{y_i \in Y} e^{s^{(X,\, y_i)}} \tag{6.2}$$

6.4 Experiment and Results

In this section, we describe the experiment and the results from three corpora.

Corpora

We used three corpora grouped into two categories to evaluate the model.

- Corpus 1 is a materials synthesis procedural (**MSP**) annotated corpus that was published in 2019[1]. This corpus is annotated by domain-experts from 230 experiment paragraphs describing synthesis procedures in materials science domain.
  - ***MSP***: Contains the operations and their arguments in synthesis experiments, such as material name, operation descriptor, synthesis apparatus, which have 21 different named entities.
- Corpus 2 is an annotated corpus in solid oxide fuel cells (**SOFC**) that is a sub-area of materials science published in 2020[151]. This corpus is annotated using four annotation

schemes based on 45 open-access scholarly articles by domain-experts. We use two of the four corpora, both related to the NRE task; the other two corpora are not related to the NRE task:

- ***SOFC***: Major entity mention types in experiment-describing sentences that include three different named entities.
- ***SOFC_Slot***: Experiment slot types in experiment-describing sentences that include 16 different named entities.

All of the corpora are annotated using the BOI format, where B is the word beginning entity, I is words inside the entity, and O is outside of the entity. The BOI labels should be predicted by the NER model; they were then transformed to pre-defined named entities.

Implementation details

We chose two different embedding layers for comparison. The Word2Vec was used as the word embedding layer for the BiLSTM-CRF model. For the BERT-CRF and BERT-BiLSTM-CRF models, we considered BERT as the embedding layer. We used a BERT-based-cased language model, which was pre-trained on cased English text. We chose SciBERT, a BERT model trained on scientific text[152], for comparison. Both pre-trained models have 12 attention heads, 12 layers and 768 hidden dimensions. We set maximum sequence length at 512, batch size at 16, initial learning rate at 0.05, warm up proportion rate at 0.1, and the dropout rate at 0.2. We used 10 epochs in the BERT-related fine-tuning models: BERT, SciBERT, BERT-CRF, and SciBERT-CRF. We used 100 epochs for training in the BiLSTM related models. In addition, the BERT language models were tuned as BERT embedding during the training process for BiLSTM-related models.

Evaluation methods

We used micro precision, recall, and F1 to evaluate the models because the corpora have a potential class imbalance issue. For example, the sample tagged as Material, Operation, Number, and Amount-Unit dominate the ***MSP*** corpus and reflect most synthesis procedures, but some named entities are not as important. However, macro precision, recall, and F1

treat all classes equally, which could have affected the accuracy of extraction results. The corresponding equations are presented below:

$$micro - Precision = \sum_{i=1}^{N} \frac{set_{\text{pre}} \cap set_{\text{true}}}{set_{\text{pre}} \cap set_{\text{true}} + set_{\text{pre}} \setminus set_{\text{true}}} \tag{6.3}$$

$$micro - Recall = \sum_{i=1}^{N} \frac{set_{\text{pre}} \cap set_{\text{true}}}{set_{\text{pre}} \cap set_{\text{true}} + set_{\text{true}} \setminus set_{\text{pre}}} \tag{6.4}$$

$$microF1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.5}$$

In these equations, the set_{pre} represents the prediction set, and the set_{true} represents the true labels set.

Results and analysis

We ran three different corpora using the same models. We used word embedding with BiLSTM-CRF as the baseline model and connected BERT embedding layer with CRF or BiLSTM-CRF. The results showed that the BERT-BiLSTM-CRF model achieved the best performance in most cases. Table 6.1 shows the results of our evaluation.

From the results, the pre-trained BERT language model used as embedding layer instead of Word2Vec showed significant improvement over the baseline model. That means the contextual feature of sentence was very helpful in the NER task in synthesis procedural text of materials science literature. In addition, the pre-trained BERT model worked better in the scientific text than in general English text. The results also showed that a fine-tuned, pre-trained language model with small corpora in a domain specific NER task got decent results in general. In addition, the corpus of ***SOFC*** had the best performance because it had only three different named entities with more balanced numbers.

Corpora	Model	Precision	Recall	F1
MSP	BiLSTM-CRF (Word2Vec)	78.51	74.84	76.63
	BERT	78.94	80.76	79.84
	BERT-CRF	79.75	80.60	80.60
	BERT-BiLSTM-CRF	85.25	83.53	84.38
	SciBERT	79.25	82.84	81.01
	SciBERT-CRF	80.48	82.96	81.70
	SciBERT-BiLSTM-CRF	**86.38**	**85.15**	**85.62**
SOFC	BiLSTM-CRF (Word2Vec)	75.33	74.35	74.84
	BERT	93.01	88.46	90.67
	BERT-CRF	93.32	88.54	91.10
	BERT-BiLSTM-CRF	93.38	90.09	91.43
	SciBERT	93.98	88.77	91.30
	SciBERT-CRF	**94.11**	89.28	**91.62**
	SciBERT-BiLSTM-CRF	93.14	**91.17**	91.57
SOFC_Slot	BiLSTM-CRF (Word2Vec)	63.24	56.29	59.56
	BERT	78.41	71.85	74.99
	BERT-CRF	80.00	72.49	76.06
	BERT-BiLSTM-CRF	89.31	82.08	86.16
	SciBERT	77.35	71.80	74.47
	SciBERT-CRF	78.45	70.46	74.24
	SciBERT-BiLSTM-CRF	**90.31**	**84.25**	**87.17**

Table 6.1: Evaluation results for three different corpora.

To the best of our knowledge, the ***MSP***[1] corpus has not been evaluated in any other publication. We compared our results with the evaluations in Friedrich et al.[151] based on ***SOFC*** and ***SOFC_Slot*** corpora. Table 6.2 provides a comparison of evaluation results.

Corpora	Model	macro F1
SOFC	SciBERT[151]	81.50
	SciBERT-BiLSTM-CRF (**ours**)	**85.61**
SOFC_Slot	BiLSTM SciBERT[151]	62.60
	SciBERT-BiLSTM-CRF (**ours**)	**64.59**

Table 6.2: Comparison of evaluation results with SOFC corpus.

Table 6.2 shows that our SciBERT-BiLSTM-CRF model outperforms both ***SOFC*** and ***SOFC_Slot*** corpora. Please note we chose the macro F1 in our evaluations to remain consistent with Friedrich et al.'s[151] evaluation methodology.

6.5 Conclusion

In this chapter, we introduce a promising attention-based deep learning approach, BERT-BiLSTM-CRF, for the NER task for synthesis procedural text of materials science. We evaluated our approach using three synthesis procedural text relevant corpora. The results showed that our BERT-BiLSTM-CRF model improved significantly over the baseline model. We have presented several models that got better results with the pre-trained language model BERT as the embedding layer compared than with word embedding models like Word2Vec. We also compared our model (using the SOFC corpora) to Friedrich et al.'s[151] model (using the SOFCSlot corpora). Our model was the better one based on the comparison results. Our work contributes to the community of materials science by demonstrating success in applying an attention-based, deep learning approach to NER of synthesis procedural text. Moreover, our work provides a competitive benchmark with these three corpora.

A few challenges in using NER in materials science will be further investigated in future work. For example, material name acronyms or abbreviations are a source of ambiguity; named entity detection of mention boundaries is also worth attention. The other concern is the entity label imbalance. For instance, there are 4843 named entities of materials in the ***MSP*** corpus, but only 122 named entities of Condition-Type. Future work should improve the application of our model in materials science domain.

Chapter 7

Conclusions and Future Directions

In this chapter, we summarize the main contributions of this dissertation, the application that we have developed based on our research, and the direction of future research.

7.1 Summary of Contributions

This dissertation presented an investigation of the integrative method for extracting information from scientific literature using deep learning techniques. The integrative method contains a series of strategies for extracting information from two primary data sources (document image and text). The principal purpose is to effectively extract information from expanding and diverse scientific literature and to provide feasible methods and comprehensive frameworks to do so. In this dissertation, we focused on the three fundamental research problems below:

- Explore vision-based deep learning methods of detecting and segmenting the physical structure of scientific documents;
- Automatically extract metadata from scientific documents based on visual features;
- Extract key structured information (e.g., entity information) from scientific text using a deep neural network.

We summarize our contributions to solving each of these problems as follows.

Chapter 3 presents a pipeline for procedural information extraction from published scientific literature (domain: synthesis of nanomaterials). This system meets computational information and knowledge management (CIKM) requirements of metadata-driven payload extraction, named entity extraction, and relationship extraction from text. Functional contributions described in this paper include semi-supervised machine learning methods for PDF filtering and payload extraction tasks, followed by structured extraction and data transformation tasks beginning with section extraction, recipe steps as information tuples, and finally, assembled recipes. Measurable objective criteria for extraction quality include precision and recall of recipe steps, ordering constraints, and QA accuracy, precision, and recall. Results, key novel contributions, and significant open problems derived from this work center around the attribution of these holistic quality measures to specific machine learning and inference stages of the pipeline, each with performance measures. The desired recipes contain identified preconditions, material inputs, and operations, constituting the overall output generated by our computational information and knowledge management (CIKM) system. Within the overall pipeline, we have applied machine learning approaches to step classification and, in continuing research, are applying these approaches to the subtasks of feature extraction, document filtering and classification, text payload extraction, recipe step identification, and multi-step assembly.

Chapter 4 introduces two deep learning-based methods for understanding scientific document layout.

One is a novel approach to developing an end-to-end learning framework to segment and classify major regions of a scientific document. We consider scientific document layout analysis as an object detection task over digital images, without any additional text features that must be added into the network during the training process. Our technical objective is to implement transfer learning via fine-tuning pre-trained networks and thereby demonstrate that this deep learning architecture is suitable for tasks that lack very large document corpora for training ab initio. As part of the experimental test bed for empirical evaluation of this approach, we created a merged multi-corpus data set for scientific publication layout

detection tasks. Our results show good improvement with fine-tuning of a pre-trained base network using this merged data set compared to the baseline CNN architecture.

The second method is an end-to-end transformer-based framework named TRDLU for the task of document layout understanding (DLU). We consider DLU a detection task and introduce TRDLU, which integrates a transformer-based vision backbone and transformer encoder-decoder as a detection pipeline. This is the first study using a fully transformer-based framework for DLU tasks. TRDLU may only be a visual, feature-based framework, but its performance is even better than multi-modal, feature-based models. We evaluated TRDLU on three different DLU benchmark datasets, each with strong baselines. TRDLU outperformed the current state-of-the-art methods on all of them.

Chapter 5 proposed a trainable end-to-end neural framework to extract metadata from scientific documents. The framework integrated RCNN for understanding document layout and CRNN for content recognition. The evaluation showed that the proposed model outperformed state-of-the-field baselines.

Chapter 6 focused on extracting named entities from synthesis procedural text from the materials science domain with the attention-based pre-trained language model. Unlike conventional machine learning approaches that need hand-crafted features or training with massive data, our attention-based, deep learning method enhanced contextualized word representations using a language model pre-trained with BERT and then associating with a BiLSTM and CRF layer, called BERTBiLSTM-CRF. Our method showed the feasibility of using limited annotated corpuses with a pre-trained language model for entity extraction from synthesis procedures in the materials science domain.

7.2 Application

Machine Learning-based procedural information extraction and knowledge management system (PIEKM) is a prototype based on nanomaterial scientific literature. The goal of PIEKM is to extract procedural information and demonstrate information retrieval capabilities. We developed this machine learning-based system to extract procedural information (in this

case, synthesizing a nanomaterial), figures, and tables from materials science articles; this system can not only retrieve information but has statistics visualization functionality. The application is based on our research as discussed in Chapter 3 and Chapter 6. Three crucial contributions of PIEKM are summarized as follows:

- This system helps researchers obtain procedural information related to material science efficiently and effectively from a multitude of publications.
- The system uses transfer learning approaches, such as chemical entity extraction, which can solve issues due to the small training dataset.
- The flexible application in this system can be deployed easily in other materials science domains.

The proposed system consists of three modules: (A) information processing, (B) user interface, and (C) query processing and information storage. Figure 7.1 shows the architecture of the PIEKM system. The details of each module follow.

(A) Information Processing Module: This module processes information from digital scientific literature. We focused on PDFs of digital scientific literature in PIEKM system. The input corpus of digital scientific literature was segmented into both text and non-text (figures, tables) parts. Then the procedural information and name entities are extracted. The extracted figures and tables are stored in corresponding folders. The rest of the extracted text information is stored as semi-structured format into a database for quick query response.

(B) User Interface Module: This module is in charge of responding to user queries and showing the results for each query, providing a preview of figures and tables from articles available in the system and presenting details from each article, including procedural information and chemical entities.

(C) Query Processing and Information Storage Module: This module is responsible for processing queries and information storage. Users submit queries, and the answers are acquired from an information storage database and returned to the user interface module for display. The module supports different material compositions and morphology searches.

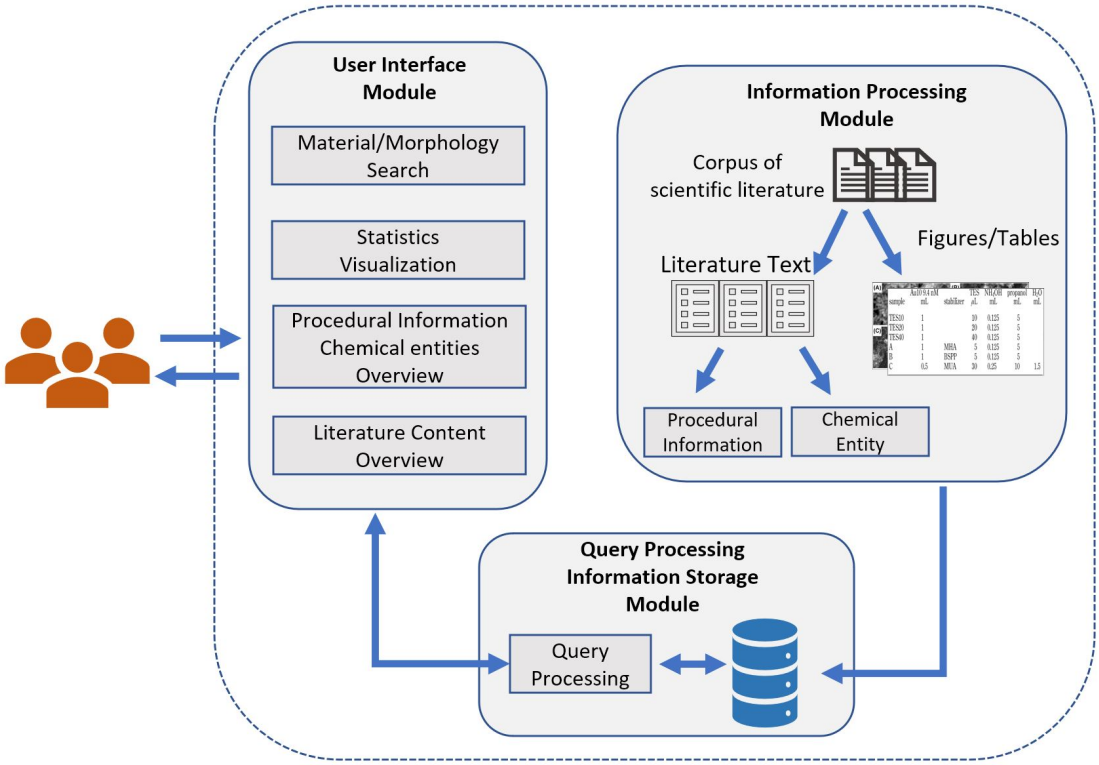


Figure 7.1: Architecture of the PIEKM system.

The PIEKM system was written in Python and deployed by Flask[1] framework. We used MongoDB[2] to store the queries for information data and the responses. Plotly[3] and Dash[4] were used for interactive visualizations.

Figure 7.2 shows the home page of PIEKM system. It provides a visual overview of the association between the number of articles within the database covering nanomaterial composition and the corresponding morphology. The user can click the material name to see the numbers of relevant articles across different morphologies. The relevant literature is searchable by material or morphology name, and the search result page will show a preview figure browser that provides all options for material or morphology names for selection; all figures in relevant articles are shown (Figure 7.3a). In addition, the chemical entities, recipe, and the full content of extracted literature can be displayed after clicking the title at the top

[1]https://flask.palletsprojects.com/
[2]https://www.mongodb.com/
[3]https://plotly.com/javascript/
[4]https://plotly.com/dash/

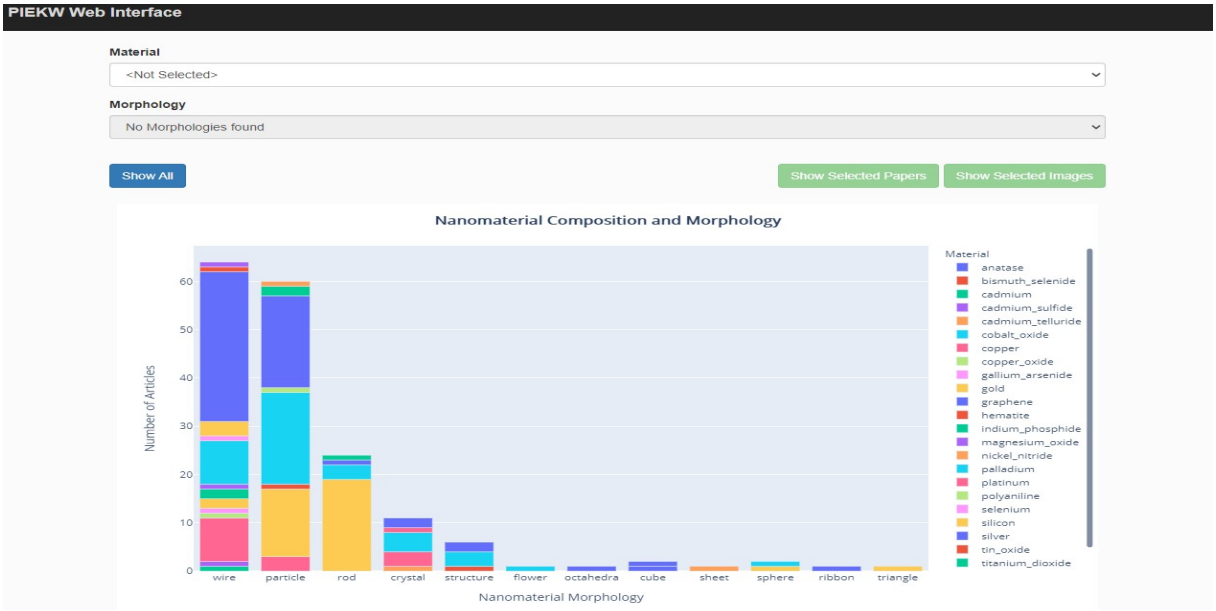
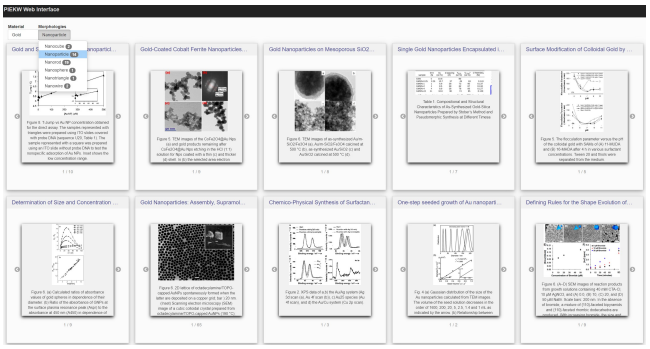
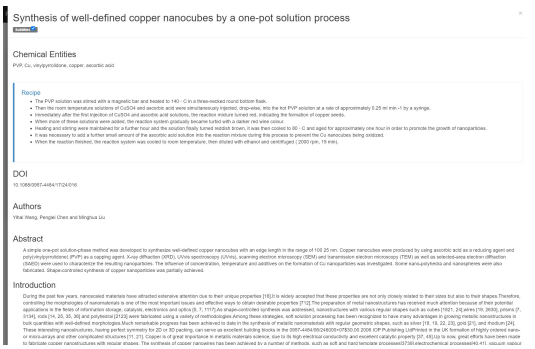


Figure 7.2: Home page of the PIEKM system.



(a) Search result page showing extracted papers and a preview figure browser.



(b) Chemical entities, recipe, and full content of extracted literature.

Figure 7.3: Search results visualization.

of each figure on the browser ((Figure 7.3b).

Overall, This PIEKM system provides an efficient way for researchers to gain insight from large number of articles from well-established publications and offers a feasible way to manage knowledge and publications in not only the nanomaterials domain but also other areas in material science.

7.3 Future Directions

In this section, we discuss future research directions.

Complex and Diverse Document Layout Detection

Scientific documents usually have Manhattan-based layout with either single-column or multi-column styles. Using a vision-based, deep learning approach is promising for detecting the layout of this kind of document, as illustrated in Chapter 4. However, other documents, handwriting manuscripts (both old and recent), magazines, newspapers, and business documents, may have non-Manhattan layouts (e.g., arbitrary complex layout or overlapping horizontal layout[153]). These kinds of layouts usually require detecting arbitrary boundaries then segmenting them into non-rectangular shapes. Finding an efficient way to detect these complex layouts requires further investigation.

Information Extraction in the Sciences

Research into information extraction using deep learning has been studied for a decade; plenty of research shows that deep learning outperforms traditional extraction methods, especially with huge corpora. However, applying this approach to scientific literature area is still in its infancy. We showed that extracting domain-specific entities from procedural texts with a pre-trained language model is possible in Chapter 6, but many challenges remain.

- *Data Acquisition.* Unlike news or social media, data is heterogeneous and diverse in scientific fields (e.g., molecular structure or reaction equations) coming from independent scientists and laboratory sources[9]. In other words, meaningful information comes not only from the text of scientific articles, but also the figures and tables from the scientific literature that cover experimental parameters as well as results. Successfully acquiring such data sources requires deeper exploration.

- *Low-resource.* The deep learning model achieves better performance using large corpora for training. Compared to ImageNet[154], which has millions of annotated instances, domain-specific, especially science-specific, corpora may include as few as several thousand[155] labeled tokens. Given such challenges, the research must consider and inte-

grate other strategies into the study, for instance, data augmentation, to generate more training data and use weak supervision learning to automatically label more data from unlabeled text.

- *Domain Knowledge Injection.* The science domain differs from the general domain, which focuses on categories like identifying the name of a person, location, or organization. Scientific articles include domain-related terminologies: drug names in bio-medicine, chemical entities in materials science, properties of the material, and acronyms or abbreviations of terminologies. Thus, we must consider adding relevant external knowledge into the model to make better predictions. For example, linking predicted chemical entities with an existing knowledge database like PubChem[156] would provide more accurate chemical names.

- *Document-level Information Extraction.* The sequential model presented in Chapter 6 is based on sentence-level extraction. However, the information on the experiment recipe can be found throughout the document and even in supplementary material, so a sentence-level model may miss contextual information. Usually, adding constraints during the inference process as post-processing generates more accurate results, but such additions are not automatic and require more human intervention. Thus, we need a model that can capture long-range dependencies and automatically extract information from scientific documents.

Multi-modal Information Extraction

The methods found in this dissertation use document layout analysis, metadata extraction, and key-information extraction based on scientific document images and text, but each method is independent of the others. Automatically processing and understanding the scientific document requires attention to all content simultaneously, including body text, figures, and tables, as a human reader does, to better understand the document. Current methods can be extended to multi-modal approaches that combine visual and text feature representation within a single model and generate a better model for downstream tasks like questions answering task and domain knowledge graph construction.

Integration Intelligence System

The pipeline test bed that we suggested in Chapter 3 integrates different stages as a system for extracting recipes from scientific publications. However, some stages of the pipeline need pre-processing or post-processing steps. We developed such steps based on the nanomaterial domain, which means it has not been evaluated in other scientific domains. The robust approaches mentioned in chapters 4 to 6 could be integrated into the pipeline and applied to other domains for interdisciplinary intelligence systems. For example, the current scholarly literature search engines, such as Google Scholar, and Semantic Scholar, provide a good service for relevant scientific literature recommendations, but the connections among different disciplines are insufficient. We still have a lot of room to improve interdisciplinary searches.

Bibliography

[1] Sheshera Mysore, Zach Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. *arXiv preprint arXiv:1905.06939*, 2019.

[2] Esther Landhuis. Scientific literature: Information overload. *Nature*, 535(7612):457–458, 2016.

[3] Karen White. Publications output: Us trends and international comparisons. *National Science Foundation*, 2019.

[4] NISO NISO. Understanding metadata. *National Information Standards*, 2004.

[5] Ralph Grishman and Beth M Sundheim. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[6] Galal M Binmakhashen and Sabri A Mahmoud. Document layout analysis: a comprehensive survey. *ACM Computing Surveys (CSUR)*, 52(6):1–36, 2019.

[7] Chirag Patel, Atul Patel, and Dharmendra Patel. Optical character recognition by open source ocr tool tesseract: A case study. *International Journal of Computer Applications*, 55(10):50–56, 2012.

[8] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[9] Elsa A Olivetti, Jacqueline M Cole, Edward Kim, Olga Kononova, Gerbrand Ceder, Thomas Yong-Jin Han, and Anna M Hiszpanski. Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews*, 7(4):041317, 2020.

[10] Hua Xu, Shane P Stenner, Son Doan, Kevin B Johnson, Lemuel R Waitman, and Joshua C Denny. Medex: a medication information extraction system for clinical narratives. *Journal of the American Medical Informatics Association*, 17(1):19–24, 2010.

[11] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*, 77:34–49, 2018.

[12] Huichen Yang, Carlos A Aguirre, F Maria, Derek Christensen, Luis Bobadilla, Emily Davich, Jordan Roth, Lei Luo, Yihong Theis, Alice Lam, et al. Pipelines for procedural information extraction from scientific literature: towards recipes using machine learning and data science. In *2019 International conference on document analysis and recognition workshops (ICDARW)*, volume 2, pages 41–46. IEEE, 2019.

[13] Huichen Yang and William H Hsu. Vision-based layout detection from scientific literature using recurrent convolutional neural networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6455–6462. IEEE, 2021.

[14] Huichen Yang and William Hsu. Automatic metadata information extraction from scientific literature using deep neural networks. In Wolfgang Osten, Dmitry Nikolaev, and Jianhong Zhou, editors, *Fourteenth International Conference on Machine Vision (ICMV 2021)*, volume 12084, pages 315 – 322. International Society for Optics and Photonics, SPIE, 2022. doi: 10.1117/12.2623554. URL https://doi.org/10.1117/12.2623554.

[15] Huichen Yang and William Hsu. Named entity recognition from synthesis procedural text in materials science domain with attention-based approach. In *SDU@ AAAI*, 2021.

[16] Song Mao, Azriel Rosenfeld, and Tapas Kanungo. Document structure analysis algorithms: a literature survey. In *Document Recognition and Retrieval X*, volume 5010, pages 197–207. International Society for Optics and Photonics, 2003.

[17] Anil K Jain and Sushil Bhattacharjee. Text segmentation using gabor filters for automatic document processing. *Machine vision and applications*, 5(3):169–184, 1992.

[18] Anil K Jain and Yu Zhong. Page segmentation using texture analysis. *Pattern recognition*, 29(5):743–770, 1996.

[19] Seong-Whan Lee and Dae-Seok Ryu. Parameter-free geometric document layout analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1240–1256, 2001.

[20] Friedrich M Wahl, Kwan Y Wong, and Richard G Casey. Block segmentation and text extraction in mixed text/image documents. *Computer graphics and image processing*, 20(4):375–390, 1982.

[21] Kwan Y. Wong, Richard G. Casey, and Friedrich M. Wahl. Document analysis system. *IBM journal of research and development*, 26(6):647–656, 1982.

[22] Zhixin Shi and Venu Govindaraju. Line separation for complex document images using fuzzy runlength. In *First International Workshop on Document Image Analysis for Libraries, 2004. Proceedings.*, pages 306–312. IEEE, 2004.

[23] Nikos Nikolaou, Michael Makridis, Basilis Gatos, Nikolaos Stamatopoulos, and Nikos Papamarkos. Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. *Image and Vision Computing*, 28(4):590–604, 2010.

[24] Jaekyu Ha, Robert M Haralick, and Ihsin T Phillips. Recursive xy cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 952–955. IEEE, 1995.

[25] George Nagy, Sharad Seth, and Mahesh Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, 1992.

[26] Francesca Cesarini, Marco Gori, Simone Marinai, and Giovanni Soda. Structured document segmentation and representation by the modified xy tree. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, pages 563–566. IEEE, 1999.

[27] Itay Bar-Yosef, Nate Hagbi, Klara Kedem, and Itshak Dinstein. Line segmentation for degraded handwritten historical documents. In *2009 10th International Conference on Document Analysis and Recognition*, pages 1161–1165. IEEE, 2009.

[28] Lawrence O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 15(11):1162–1173, 1993.

[29] Syed Saqib Bukhari, Mayce Ibrahim Ali Al Azawi, Faisal Shafait, and Thomas M Breuel. Document image segmentation using discriminative learning over connected components. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 183–190, 2010.

[30] Irina Rabaev, Ofer Biller, Jihad El-Sana, Klara Kedem, and Itshak Dinstein. Text line detection in corrupted and damaged historical manuscripts. In *2013 12th International Conference on Document Analysis and Recognition*, pages 812–816. IEEE, 2013.

[31] Koichi Kise, Akinori Sato, and Motoi Iwata. Segmentation of page images using the area voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.

[32] Yue Lu and Chew Lim Tan. Constructing area voronoi diagram in document images. In

Eighth International Conference on Document Analysis and Recognition (ICDAR'05), pages 342–346. IEEE, 2005.

[33] Yi Xiao and Hong Yan. Text region extraction in a document image based on the delaunay tessellation. *Pattern Recognition*, 36(3):799–809, 2003.

[34] Yi Xiao and Hong Yan. Location of title and author regions in document images based on the delaunay triangulation. *Image and Vision Computing*, 22(4):319–329, 2004.

[35] Boontee Kruatrachue, Narongchai Moongfangklang, and Kritawan Siriboon. Fast document segmentation using contour and xy cut technique. In *WEC (5)*, pages 27–29. Citeseer, 2005.

[36] Ming-Wei Lin, Jules-Raymond Tapamo, and Baird Ndovie. A texture-based method for document segmentation and classification. *South African Computer Journal*, 2006 (36):49–56, 2006.

[37] Simone Marinai, Marco Gori, and Giovanni Soda. Artificial neural networks for document analysis and recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 27(1):23–35, 2005.

[38] Hao Wei, Micheal Baechler, Fouad Slimane, and Rolf Ingold. Evaluation of svm, mlp and gmm classifiers for layout analysis of historical documents. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1220–1224. IEEE, 2013.

[39] Andreas Fischer, Micheal Baechler, Angelika Garz, Marcus Liwicki, and Rolf Ingold. A combined system for text line extraction and handwriting recognition in historical documents. In *2014 11th IAPR International Workshop on Document Analysis Systems*, pages 71–75. IEEE, 2014.

[40] Kai Chen, Mathias Seuret, Marcus Liwicki, Jean Hennebert, and Rolf Ingold. Page segmentation of historical document images with convolutional autoencoders. In *2015*

13th International Conference on Document Analysis and Recognition (ICDAR), pages 1011–1015. IEEE, 2015.

[41] Xiao-Hui Li, Fei Yin, and Cheng-Lin Liu. Page object detection from pdf document images by deep structured prediction and supervised clustering. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3627–3632. IEEE, 2018.

[42] Dafang He, Scott Cohen, Brian Price, Daniel Kifer, and C Lee Giles. Multi-scale multi-task fcn for semantic page segmentation and table detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 254–261. IEEE, 2017.

[43] Dario Augusto Borges Oliveira and Matheus Palhares Viana. Fast cnn-based document layout analysis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1173–1180, 2017.

[44] Tobias Grüning, Gundram Leifert, Tobias Strauß, Johannes Michael, and Roger Labahn. A two-stage method for text line detection in historical documents. *International Journal on Document Analysis and Recognition (IJDAR)*, 22(3):285–302, 2019.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[46] Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5315–5324, 2017.

[47] Moritz Schubotz, André Greiner-Petter, Philipp Scharpf, Norman Meuschke, Howard S Cohl, and Bela Gipp. Improving the representation and conversion of mathematical

formulae by considering their textual context. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, pages 233–242, 2018.

[48] Philipp Scharpf, Ian Mackerracher, Moritz Schubotz, Joeran Beel, Corinna Breitinger, and Bela Gipp. Annomathtex-a formula identifier annotation recommender system for stem documents. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 532–533, 2019.

[49] Matthew C Swain and Jacqueline M Cole. Chemdataextractor: a toolkit for automated extraction of chemical information from the scientific literature. *Journal of chemical information and modeling*, 56(10):1894–1904, 2016.

[50] Paul Flynn, Li Zhou, Kurt Maly, Steven Zeil, and Mohammad Zubair. Automated template-based metadata extraction architecture. In *International Conference on Asian Digital Libraries*, pages 327–336. Springer, 2007.

[51] Alexandru Constantin, Steve Pettifer, and Andrei Voronkov. Pdfx: fully-automated pdf-to-xml conversion of scientific literature. In *Proceedings of the 2013 ACM symposium on Document engineering*, pages 177–180, 2013.

[52] Giovanni Giuffrida, Eddie C Shek, and Jihoon Yang. Knowledge-based metadata extraction from postscript files. In *Proceedings of the fifth ACM conference on digital libraries*, pages 77–84, 2000.

[53] Tin Huynh and Kiem Hoang. Gate framework based metadata extraction from scientific papers. In *2010 International Conference on Education and Management Technology*, pages 188–191. IEEE, 2010.

[54] Huy Hoang Nhat Do, Muthu Kumar Chandrasekaran, Philip S Cho, and Min Yen Kan. Extracting and matching authors and affiliations in scholarly documents. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 219–228, 2013.

[55] Patrice Lopez. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*, pages 473–474. Springer, 2009.

[56] Chen-Tse Tsai, Gourab Kundu, and Dan Roth. Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM international conference on information & knowledge management*, pages 1733–1738, 2013.

[57] 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.

[58] Animesh Prasad, Manpreet Kaur, and Min-Yen Kan. Neural parscit: a deep learning-based reference string parser. *International journal on digital libraries*, 19(4):323–337, 2018.

[59] Ranajit Saha, Ajoy Mondal, and CV Jawahar. Graphical object detection in document images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 51–58. IEEE, 2019.

[60] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.

[61] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[62] Alexandra Pomares Quimbaya, Alejandro Sierra Múnera, Rafael Andrés González Rivera, Julián Camilo Daza Rodríguez, Oscar Mauricio Muñoz Velandia, Angel Alberto Garcia Peña, and Cyril Labbé. Named entity recognition over electronic health

records through a combined dictionary-based approach. *Procedia Computer Science*, 100:55–61, 2016.

[63] Isabel Segura Bedmar, Paloma Martínez, and María Herrero Zazo. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics, 2013.

[64] Sean R. Eddy. Profile hidden markov models. *Bioinformatics (Oxford, England)*, 14 (9):755–763, 1998.

[65] Jagat Narain Kapur. *Maximum-entropy models in science and engineering*. John Wiley & Sons, 1989.

[66] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[67] JR Quinlan. Induction of decision trees. mach. learn. 1986.

[68] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[69] Daniel M Bikel, Richard Schwartz, and Ralph M Weischedel. An algorithm that learns what's in a name. *Machine learning*, 34(1):211–231, 1999.

[70] Yaoyong Li, Kalina Bontcheva, and Hamish Cunningham. Svm based learning system for information extraction. In *International Workshop on Deterministic and Statistical Methods in Machine Learning*, pages 319–339. Springer, 2004.

[71] Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries. *Information*, 6(4):848–865, 2015.

[72] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[73] Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6):1088–1098, 2013.

[74] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[75] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*, 2017.

[76] Onur Kuru, Ozan Arkan Can, and Deniz Yuret. Charner: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 911–921, 2016.

[77] Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, 2019.

[78] Yonghui Wu, Min Jiang, Jianbo Lei, and Hua Xu. Named entity recognition in chinese clinical text using deep neural network. *Studies in health technology and informatics*, 216:624, 2015.

[79] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

[80] Marek Rei. Semi-supervised multitask learning for sequence labeling. *arXiv preprint arXiv:1704.07156*, 2017.

[81] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[82] Ashish Vaswani, Yonatan Bisk, Kenji Sagae, and Ryan Musa. Supertagging with lstms. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 232–237, 2016.

[83] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Coling 1992 volume 2: The 15th international conference on computational linguistics*, 1992.

[84] Gumwon Hong. Relation extraction using support vector machine. In *International Conference on Natural Language Processing*, pages 366–377. Springer, 2005.

[85] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.

[86] Nicolas Heist and Heiko Paulheim. Language-agnostic relation extraction from wikipedia abstracts. In *International Semantic Web Conference*, pages 383–399. Springer, 2017.

[87] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[88] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, 2014.

[89] Andreas Lehmkühler. Pdfbox - a java pdf library. https://pdfbox.apache.org/, May, 24, 2018.

[90] Edward Kim, Kevin Huang, Alex Tomala, Sara Matthews, Emma Strubell, Adam Saunders, Andrew McCallum, and Elsa Olivetti. Machine-learned and codified synthesis parameters of oxide materials. *Scientific data*, 4(1):1–9, 2017.

[91] Rachael Lammey. Crossref text and data mining services. *Insights*, 28(2), 2015.

[92] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[93] Marwin Segler, Mike Preuß, and Mark P Waller. Towards" alphachem": Chemical synthesis planning with tree search and deep neural network policies. *arXiv preprint arXiv:1702.00020*, 2017.

[94] Edward Kim, Kevin Huang, Adam Saunders, Andrew McCallum, Gerbrand Ceder, and Elsa Olivetti. Materials synthesis insights from scientific literature via text extraction and machine learning. *Chemistry of Materials*, 29(21):9436–9444, 2017.

[95] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[96] F Maria, Carlos A Aguirre, BreAnn M Anshutz, and William H Hsu. Matesc: Metadata-analytic text extractor and section classifier for scientific publications. In *KDIR*, pages 259–265, 2018.

[97] R Liu and JX McKie. Pymupdf. http://pymupdf.readthedocs.io/en/latest/, 2018.

[98] D Tzutalin. Labelimg. *GitHub Repository*, 6, 2015.

[99] 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.

[100] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[101] Lezan Hawizy, David M Jessop, Nico Adams, and Peter Murray-Rust. Chemicaltagger: A tool for semantic text-mining in chemistry. *Journal of cheminformatics*, 3(1):1–13, 2011.

[102] Peter Corbett and Peter Murray-Rust. High-throughput identification of chemistry in life science texts. In *International Symposium on Computational Life Science*, pages 107–118. Springer, 2006.

[103] Lucene.apache.org. (2019). Apache solr -. [online] available. http://lucene.apache.org/solr/, Feb, 11, 2019.

[104] Cartic Ramakrishnan, Abhishek Patnia, Eduard Hovy, and Gully APC Burns. Layout-aware text extraction from full-text pdf of scientific articles. *Source code for biology and medicine*, 7(1):1–10, 2012.

[105] Xiangyang Shi, Yue Wu, Huaigu Cao, Gully Burns, and Prem Natarajan. Layout-aware subfigure decomposition for complex figures in the biomedical literature. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1343–1347. IEEE, 2019.

[106] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[107] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

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

[109] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[110] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[111] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

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

[113] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[114] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[115] Carlos Soto and Shinjae Yoo. Visual detection with context for document layout analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3464–3470, 2019.

[116] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1449–1453. IEEE, 2013.

[117] Dominika Tkaczyk, Artur Czeczko, Krzysztof Rusek, Lukasz Bolikowski, and Roman Bogacewicz. Grotoap: ground truth for open access publications. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, pages 381–382, 2012.

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

[119] Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*, 2021.

[120] Sofia Ares Oliveira, Benoit Seguin, and Frederic Kaplan. dhsegment: A generic deep-learning approach for document segmentation. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 7–12. IEEE, 2018.

[121] Peng Zhang, Can Li, Liang Qiao, Zhanzhan Cheng, Shiliang Pu, Yi Niu, and Fei Wu. Vsr: A unified framework for document layout analysis combining vision, semantics and relations. In *International Conference on Document Analysis and Recognition*, pages 115–130. Springer, 2021.

[122] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003, 2021.

[123] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[124] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[125] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[126] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[127] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021.

[128] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse detr: Efficient end-to-end object detection with learnable sparsity. *arXiv preprint arXiv:2111.14330*, 2021.

[129] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12993–13000, 2020.

[130] 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.

[131] Abdelrahman Abdallah, Alexander Berendeyev, Islam Nuradin, and Daniyar Nurseitov. Tncr: Table net detection and classification dataset. *Neurocomputing*, 473: 79–97, 2022.

[132] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[133] Sanket Biswas, Ayan Banerjee, Josep Lladós, and Umapada Pal. Docsegtr: An instance-level end-to-end document image segmentation transformer. *arXiv preprint arXiv:2201.11438*, 2022.

[134] Daniel Torres-Salinas. Daily growth rate of scientific production on covid-19. analysis in databases and open access repositories. *arXiv preprint arXiv:2004.06721*, 2020.

[135] Haoyan Huo, Ziqin Rong, Olga Kononova, Wenhao Sun, Tiago Botari, Tanjin He, Vahe Tshitoyan, and Gerbrand Ceder. Semi-supervised machine-learning classification of materials synthesis procedures. *Npj Computational Materials*, 5(1):1–7, 2019.

[136] Anthony Kay. Tesseract: an open-source optical character recognition engine. *Linux Journal*, 2007(159):2, 2007.

[137] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[138] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[139] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[140] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.

[141] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[142] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[143] Zhuoyao Zhong, Lianwen Jin, and Shuangping Huang. Deeptext: A new approach for text proposal generation and text detection in natural images. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1208–1212. IEEE, 2017.

[144] Leigh Weston, Vahe Tshitoyan, John Dagdelen, Olga Kononova, Amalie Trewartha, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of chemical information and modeling*, 59(9):3692–3702, 2019.

[145] Surya R Kalidindi and Marc De Graef. Materials data science: current status and future outlook. *Annual Review of Materials Research*, 45:171–193, 2015.

[146] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[147] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[148] Tsendsuren Munkhdalai, Meijing Li, Khuyagbaatar Batsuren, Hyeon Ah Park, Nak Hyeon Choi, and Keun Ho Ryu. Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. *Journal of cheminformatics*, 7(1):1–8, 2015.

[149] Sheshera Mysore, Edward Kim, Emma Strubell, Ao Liu, Haw-Shiuan Chang, Srikrishna Kompella, Kevin Huang, Andrew McCallum, and Elsa Olivetti. Automatically

extracting action graphs from materials science synthesis procedures. *arXiv preprint arXiv:1711.06872*, 2017.

[150] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[151] Annemarie Friedrich, Heike Adel, Federico Tomazic, Johannes Hingerl, Renou Benteau, Anika Maruscyk, and Lukas Lange. The sofc-exp corpus and neural approaches to information extraction in the materials science domain. *arXiv preprint arXiv:2006.03039*, 2020.

[152] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[153] Koichi Kise. Page segmentation techniques in document analysis., 2014.

[154] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[155] Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv preprint arXiv:2010.12309*, 2020.

[156] Sunghwan Kim, Paul A Thiessen, Evan E Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1):D1202–D1213, 2016.