Comparison of Feature Learning Methods for Metadata Extraction from PDF Scholarly Documents

Zeyd Boukhers[1,2,3]   Cong Yang[4]

Abstract—The availability of metadata for scientific documents is pivotal in propelling scientific knowledge forward and for adhering to the FAIR principles (i.e. Findability, Accessibility, Interoperability, and Reusability) of research findings. However, the lack of sufficient metadata in published documents, particularly those from smaller and mid-sized publishers, hinders their accessibility. This issue is widespread in some disciplines, such as the German Social Sciences, where publications often employ diverse templates. To address this challenge, our study evaluates various feature learning and prediction methods, including natural language processing (NLP), computer vision (CV), and multimodal approaches, for extracting metadata from documents with high template variance. We aim to improve the accessibility of scientific documents and facilitate their wider use. To support our comparison of these methods, we provide comprehensive experimental results, analyzing their accuracy and efficiency in extracting metadata. Additionally, we provide valuable insights into the strengths and weaknesses of various feature learning and prediction methods, which can guide future research in this field.

Index Terms—metadata extraction, document processing, neural networks, natural language processing, computer vision, multimodal approaches, scientific documents

I. INTRODUCTION

The widespread availability of scientific metadata has greatly contributed to the success and advancement of the scientific community by enabling the easy findability and accessibility of scientific documents. This is achieved by indexing and linking scientific papers in a large and consistent graph such as the OpenAIRE graph [1] or the Open Research Knowledge Graph [2]. As a result, the field of scientometrics has emerged to study and analyze scholarly literature. While it has become increasingly common for publishers and authors to collect and provide comprehensive metadata alongside the publication of scientific documents, to ensure data's accuracy, completeness, and integrity, this practice was not always popular. Historically, certain disciplines, such as Social Sciences, have seen a considerable portion of their publications become less discoverable due to inadequate metadata collection. This shortfall is particularly evident in works from smaller or mid-sized publishers, which may have lacked the resources or incentive to adequately document metadata, especially in the case of older publications [3], [4]. Consequently, numerous initiatives have been established to consolidate efforts towards enhancing the findability, accessibility, interoperability, and reusability of scholarly data. Prominent among these are The European Open Science Cloud (EOSC)[1], The German Research Data Infrastructure (NFDI)[2] and European Strategy Forum on Research Infrastructures (ESFRI)[3]. The primary focus of these initiatives is on the pivotal task of making metadata universally available.

Alternatively, the metadata can be directly extracted from scientific documents. However, manually extracting metadata from the vast number of published documents is a labour-intensive and time-consuming task, making automation essential. To automate the process, several approaches have been proposed, including classical natural language processing (NLP)-based approaches [5], [6], which aim to extract metadata from PDF documents efficiently and accurately.

With the recent advances in Deep Neural Networks (DNNs) on textual data, significant results have been achieved on this task [7]. This is due to the capability of these networks to capture latent features from the textual documents. However, the problem is still open and far from being solved because scientific documents come in different templates and layouts. This makes it difficult for any model to find common patterns in the order of the classes. To overcome this problem, some works [3], [8] propose to tackle the problem using image processing techniques and taking advantage of the remarkable advances in computer vision. To this end, these techniques view the scientific PDF documents as RGB images. Furthermore, to harness the strengths of both text and visual information, several studies [4], [9] have adopted multimodal approaches, demonstrating notable effectiveness.

This study explores a variety of feature learning and classification approaches to extract metadata from scientific PDF documents, emphasizing the use of methodologies best suited to the specific challenges of this task. We employ classical approaches such as Conditional Random Fields, advanced NLP techniques including BiLSTM with BERT representations, and innovative multimodal and Textmap methods. While generative LLMs like GPT-4 or LLAMA excel in natural language generation, they are not ideal for structured tasks such as metadata extraction from scientific PDFs. These mod-

[1]Fraunhofer Institute for Applied Information Technology FIT, Sankt Augustin, Germany.
[2]University Hospital of Cologne, Cologne, Germany.
[3]University of Koblenz, Koblenz, Germany (e-mail: zeyd.boukhers@fit.fraunhofer.de).
[4]Soochow University.

els, designed primarily for text generation from prompts, face difficulties with fixed formats, which can lead to inaccuracies from over-generalization and context sensitivity, and require substantial resources for task-specific tuning. By contrast, our chosen approaches leverage the strengths of BERT and other architectures to efficiently handle the unique layout variability and multimodal content of scientific documents, ensuring precise and reliable metadata extraction.

In addition to evaluating the technical aspects of these approaches, we also compare their performance and results on a large and unique dataset. One challenge in this area is that many techniques, such as those based on deep neural networks (DNNs), require an extensive ground truth dataset for training. However, creating such a dataset can be difficult, as the process of annotating the data is time-consuming and labor-intensive, and often requires quality checks. To address this issue, we created two challenging datasets, namely, *SSOAR-MVD* and *S-PMRD*. For *SSOAR-MVD*, we synthesized 50.000 samples using a predefined set of templates and available metadata. *S-PMRD* is an authentic subset of the Semantic Scholar Open Research Corpus. The main contributions of this paper are as follows:

- We present a variety of approaches for extracting metadata from scientific PDF documents.
- We created a large, labelled dataset for metadata extraction from scientific PDF documents.
- We conducted extensive experiments to compare the various approaches.
- To facilitate reproducibility and future development, we have made the implementations of all the approaches publicly available[4].

The remainder of this paper is organized as follows: In Section II, we review related works. In Section III, we introduce all the approaches covered in this paper. Section IV presents the dataset and experimental results, and finally, in Section VI, we provide concluding remarks and discuss potential future directions.

II. RELATED WORK

Metadata extraction, while a specialized subset of information extraction (IE), serves a distinct purpose and presents unique challenges. This section provides an overview of the most pertinent techniques for metadata extraction, categorizing them into three distinct groups for a clearer understanding of their applications and methodologies.

A. Natural Language Processing

Metadata extraction in Natural Language Processing (NLP) has primarily been approached through two distinct methodologies: rule-based and machine learning-based techniques [10]. Rule-based techniques rely on predefined rules developed through human expertise to guide metadata extraction [10]. These methods are generally more straightforward to implement but may lack the adaptability found in machine learning-based systems [11], [12]. On the other hand, machine

learning-based approaches, exemplified by platforms like Cite-SeerX [13], leverage supervised learning algorithms trained on labelled datasets to autonomously extract metadata from new documents. These algorithms range from Hidden Markov Models (HMM) [14], Conditional Random Fields (CRFs) [15], to Support Vector Machines (SVM) [16]. Although robust and effective, the drawback of these machine learning methods lies in the labour-intensive labelling of training data, especially when dealing with samples of high variability.

Recent advancements in Deep Neural Networks (DNN) have provided a new dimension to the field of metadata extraction. DNNs have been shown to considerably outperform traditional methods in effectiveness and efficiency [10]. [17] pioneered a Bidirectional LSTM-CRF model, combining Long Short-Term Memory (LSTM) with a Conditional Random Field (CRF) layer to encode word sequences and predict labels. Similarly, [18] employed a Bidirectional LSTM integrated with a Convolutional Neural Network (CNN) to generate character-level word representations. [19] introduced a DNN-based Segment Sequence Labeling for metadata extraction, setting new performance benchmarks. This approach outstripped existing works such as ParsCit [20], a CRF-based model, and BibPro [21], a neural network-based model, when evaluated on public datasets like UMass [22] and Cora [14].

B. Computer Vision

While Computer Vision (CV) approaches are not yet ubiquitously applied in the field of metadata extraction, emerging research indicates their promising capabilities, especially for Natural Language Processing (NLP) related tasks. One notable example is DeepPDF [23], which applies a unique perspective to PDF document segmentation. Instead of traditional text-based analysis, DeepPDF treats the document as an image and employs UNet-Zoo, a specialized architecture originally designed for biomedical image segmentation. This approach allows for accurate paragraph identification while ignoring other elements like headers, captions, figures, and references, thus substantiating the potential of CV-based techniques for textual document analysis.

Building upon the groundwork laid by [23], MexPub [24] introduced an innovative technique for extracting metadata from German PDF documents. The methodology utilizes a pixel-by-pixel analysis through the MASK-RCNN architecture [25], specifically engineered for object detection and classification. It incorporates the ResNeXt backbone [26] and Feature Pyramid Networks (FPN) for feature extraction from raw images. While MexPub has shown promising results, it encounters limitations in certain areas. For example, the model struggles with generalizing to scientific literature that diverges structurally from the training dataset. Additionally, MexPub faces challenges in precisely detecting smaller patterns or those placed in unconventional positions. These limitations suggest that the method's performance could be further enhanced by incorporating text processing elements into a unified architecture.

[4]Willbereleaseduponpublicaiton.

C. Multimodality

Multimodal deep learning has made significant inroads across various applications, including but not limited to audio-visual and image classification, showcasing impressive performance. Specifically within the realm of metadata extraction, there's growing evidence that multimodal approaches are superior to their unimodal counterparts, as highlighted in studies by [27], [28], and [4].

Balasubramanian et al. [27] employed a combined audio and video modality strategy to extract metadata from video lectures. Their technique harnessed the potential of a Naive Bayes classifier in tandem with a rule-based refiner. The essence of this methodology was capitalizing on the interplay between audio transcripts and the content of slides embedded within video streams. Astonishingly, this synergy yielded a marked 114.2% improvement in metrics such as F-score, precision, and recall when benchmarked against solely audio-based methodologies.

Liu et al. [28] pioneered a multimodal deep-learning strategy tailored for metadata extraction from scientific documents. Their model seamlessly ingests both image and textual data, negating the need for handcrafted classification features. On the textual front, Recurrent Neural Networks (RNNs) were employed, while image data was processed using Convolutional Neural Networks (CNNs). The amalgamated representation was then processed via a BiLSTM network, culminating in classification through a CRF classifier. The potency of this composite approach was evident when juxtaposed against unimodal strategies.

Further enriching the field,[4] presented an intriguing approach to address metadata extraction challenges specific to German scientific papers, which frequently exhibit a vast array of layouts due to the varied publishing standards of small to mid-sized publishers. The paper proposed a multimodal approach that perceives a PDF document simultaneously as an RGB image and a textual document, using BiLSTM and MexPub, respectively. The outputs from both sub-models are subsequently merged and processed by another BiLSTM model for token classification.

III. Approach

This section discusses various feature learning and classification methods for extracting metadata from scientific PDF documents. Like many studies in this area, we assume that metadata may only be present on the first page of a PDF document and that its availability may vary across documents. For example, all scientific PDF documents may not include the Digital Object Identifier (DOI).

Let $\mathcal{P}$ be the first page of a scientific PDF document, consisting of a set of observed words $\omega = \langle \omega_1, \omega_2, \cdots, \omega_n \rangle$, where $n = |\omega|$. Let S be a set of states in a finite state machine, each corresponding to a label $l \in L$ (e.g., *Title*, *Authors*, etc.). The task is to formalize $\gamma(\mathcal{P}) = \mathbf{s}$, where $\mathbf{s} = \langle s_1, s_2, \cdots, s_n \rangle$ is the sequence of states in S that correspond to the labels assigned to the words in the input sequence ω. Table I represents the used variables and their descriptions

Variable	Description
γ	The metadata extraction model
$\mathcal{P}$	The first page of the PDF document
$\mathbf{S}$	The outcome of the model, which is a set of strings associated with their labels
y	The metadata label
$\mathbf{s}_i$	The output metadata value of the yth label
K	Section
w	Classified token
ω	Unclassified token

TABLE I
OVERVIEW OF KEY VARIABLES USED IN THIS PAPER ACROSS THE DIFFERENT FEATURE LEARNING AND CLASSIFICATION METHODS FOR METADATA EXTRACTION.
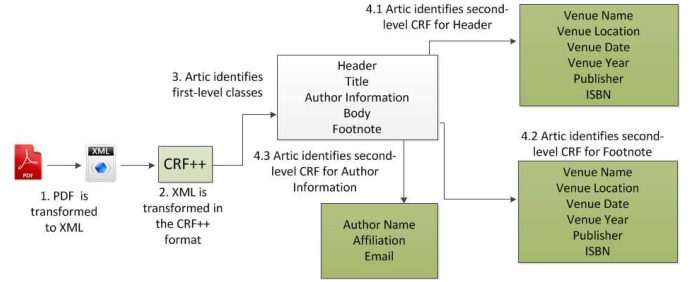

Fig. 1. Schematic representation of the two-layer Conditional Random Field (CRF) model for metadata extraction [29].

In this study, we compare several approaches for extracting metadata from scientific PDF documents, including foundational techniques like Conditional Random Fields (CRF) [29] and GROBID [30], which have established the groundwork for metadata extraction. We also implement and explore novel neural sequence labeling approaches using BiLSTM and BiLSTM-CRF architectures (Sections III-B and III-C). This work introduces three new methodologies that take different approaches to the problem: a computer vision approach using Fast R-CNN (Section III-E), a multimodal neural architecture (Section III-F), and our proposed TextMap framework (Section III-G). All approaches are evaluated following the aforementioned formalization of the metadata extraction task, enabling a comprehensive comparison of their effectiveness.

A. Conditional Random Fields (CRF)[29]

The approach proposed by Souza, Viviane, and Heuser [29] employs a two-layer Conditional Random Field (CRF) model for extracting metadata from scientific PDF documents. As illustrated in Figure 1, the extraction process is divided into two main steps: identifying main sections and extracting metadata from these sections.

Given the extracted lines from the first page $\mathcal{P}$, the first layer of the CRF model classifies each line into one of the five main sections that may contain metadata information: *Header*, *Title*, *Author Information*, *Body*, and *Footnote*. To achieve this, the model processes font features such as size, style, and alignment from each line and uses them as input. Once the main sections have been identified, the second layer of the CRF model is responsible for extracting metadata from these sections. Some content is automatically excluded from certain sections during this process. For instance, content that appears
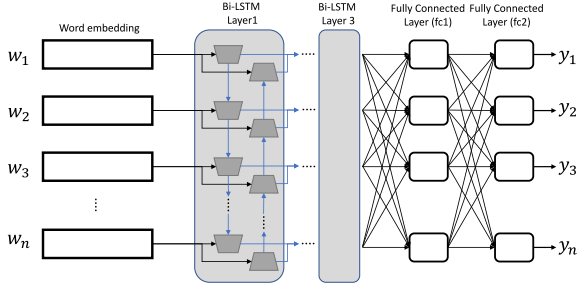
Fig. 2. Diagram of the Bi-Directional LSTM network architecture for metadata extraction

Fig. 3. Diagram of the Bi-Directional LSTM network architecture with CRF classifier for metadata extraction

in the Footnote section would not be included in the model's output.

For each identified section k, the model processes a sequence of observed words $\omega = \langle \omega_1, \omega_2, \cdots, \omega_n \rangle$. Each word ω_i is represented with a feature vector $\mathbf{x}_i$, which comprises m handcrafted features such as length, whether it follows a year format, presence of special characters, and capitalization, among others. The model calculates the probability of a section sequence given a handcrafted feature sequence using the following equation:

$$P(\mathbf{s} \mid \mathbf{x}) = \frac{\exp\left(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(y_{i-1}, y_i, \mathbf{x}, i)\right)}{Z(\mathbf{x})} \quad (1)$$

where f_j denotes the feature function for the jth feature, and λ_j is the corresponding weight parameter. $Z(\mathbf{x})$ is the normalization factor, ensuring that the sum of probabilities over all possible label sequences equals 1:

$$Z(\mathbf{x}) = \sum_{y'} \exp\left(\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f_j(y'_{i-1}, y'_i, \mathbf{x}, i)\right) \quad (2)$$

To find the optimal weights $\lambda_{j=1}^{m}$, a training process is conducted by maximizing the log-likelihood of the training data:

$$\mathcal{L}(\lambda) = \sum_{u=1}^{|D|} \log P\left(y^{(u)}, \mathbf{x}^{(u)}\right) - \frac{\sum_{j=1}^{m} \lambda_j^2}{2\sigma^2} \quad (3)$$

where $(x^{(u)}, y^{(u)})$ are the pair features and label of the uth training instance in the training dataset D, and σ^2 is a hyperparameter for L2 regularization that controls the model's complexity.

B. Bi-Directional LSTM

For this solution, we employed a Bidirectional Long Short-Term Memory (BiLSTM) model with three layers. The BiLSTM has 112 hidden dimensions and is followed by two fully connected layers. The final layer uses a softmax activation function to assign each word to a specific class. Given a sequence of observed words $\omega = \langle \omega_1, \omega_2, \cdots, \omega_n \rangle$, the embedding vector of each word ω_i is obtained the BERT model:

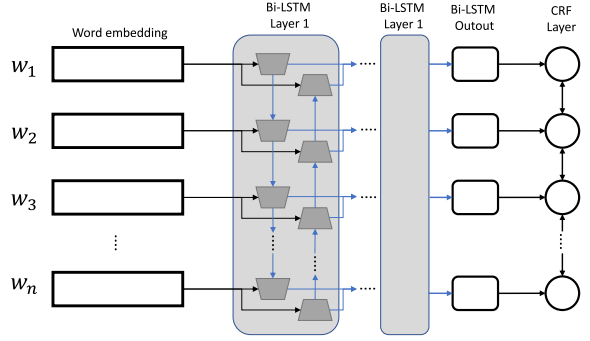$$\mathbf{x_i} = \text{BERT}(\omega_i), \quad i = 1, 2, \cdots, n \quad (4)$$

resulting in a sequence of embedding vectors $\mathbf{x} = \langle \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n \rangle$.

This sequence of embedding vectors is fed into the three bidirectional LSTM layers with hidden dimensions $= 112$. Let $\mathbf{h}_i^{(f)}$ and $\mathbf{h}_i^{(b)}$ denote the forward and backward hidden states at position i in the sequence. The hidden states are updated as follows:

$$\begin{aligned} \mathbf{h}_i^{(f)} &= \text{LSTM}^{(f)}(\mathbf{x}_i, \mathbf{h}_{i-1}^{(f)}), \\ \mathbf{h}_i^{(b)} &= \text{LSTM}^{(b)}(\mathbf{x}_i, \mathbf{h}_{i+1}^{(b)}) \end{aligned} \quad (5)$$

For each BiLSTM layer t, the outputs of the forward and backward LSTM units are concatenated to form the hidden state of the BiLSTM layer:

$$\mathbf{h}_i^{(t)} = \left[\mathbf{h}_i^{(f,t)}; \mathbf{h}_i^{(b,t)}\right] \quad (6)$$

The output of the last BiLSTM layer is passed through two fully connected layers with weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ and bias vectors $\mathbf{b}_1$ and $\mathbf{b}_2$:

$$\mathbf{o}_i = \text{ReLU}(\mathbf{W}_2 \text{ReLU}(\mathbf{W}_1 \mathbf{h}_i^{(\text{BiLSTM})} + \mathbf{b_1}) + \mathbf{b_2}) \quad (7)$$

A softmax activation function is applied to the output of the last fully connected layer to compute the probability distribution over the predefined set of labels for each word in the sequence:

$$\hat{\mathbf{y}}_i = \text{SoftMax}(\mathbf{o}_i) = \frac{\exp(\mathbf{o}_i)}{\sum_{l=1}^{L} \exp(\mathbf{o}_i, l)} \quad (8)$$

C. BiLSTM-CRF

As BiLSTM-CRF is used in many NLP tasks and specifically extracting information from textual data [31], [32], we assume that it would perform similarly on the task of extracting metadata from PDF documents. Figure 3 illustrates the developed model that takes as input the embeddings of the words extracted from $\mathcal{P}$. The embeddings are obtained using a pre-trained BERT model. The assumption is that most of the metadata classes are represented in structured phrases that BERT can capture. For the other classes (e.g. Author name),

The proposed model consists of a 4-layer BiLSTM network with 115 hidden dimensions followed by a CRF layer for
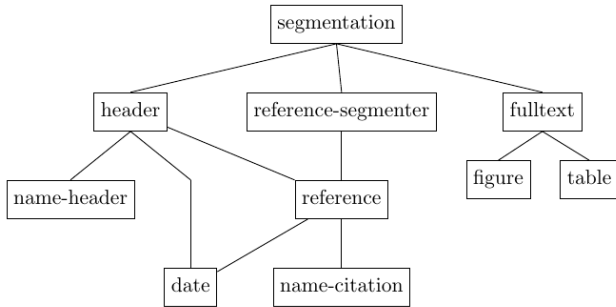
Fig. 4. Overview of the GROBID Framework for structured metadata extraction [30]

sequence labelling. The sequence of observed words goes through the same steps as mentioned in the Equations 4, 5 and 6. Then, the output of the last BiLSTM layer $H = \sum_i \mathbf{h}_i^{\text{BiLSTM}}$ is passed through a CRF layer to calculate the probability of a label sequence $P(\mathbf{s} \mid H)$, using Equation 1.

D. Grobid[30]

GROBID is a machine-learning library that is designed to extract, parse, or restructure raw documents into structured XML/TEI documents. It employs a cascade of sequence labelling models to parse each document, allowing it to adapt to the different hierarchical structures present in the documents. By utilizing a cascade approach, GROBID can handle a wide variety of document layouts and structures.

The main idea behind GROBID's approach is to break down the complex task of document parsing into a series of smaller, more manageable tasks. Each model in the cascade focuses on a specific aspect of the document structure, such as headers, titles, author information, or other metadata. The models have a small number of labels, which makes it easier to manage and train. When combined, the full cascade provides a detailed end-result structure.

In GROBID, the models are organized hierarchically to address the inherent hierarchical structure of the documents. The original GROBID model produces 55 different "leaf" labels, which are the final labels assigned to the text elements after the document has been processed by the entire cascade of models. Each "leaf" label corresponds to a specific element in the structured XML/TEI output.

Model Training: Train a cascade of sequence labeling models on the training dataset. Each model in the cascade is responsible for recognizing and classifying specific elements of the document structure. The models are organized hierarchically, with each model feeding its output to the subsequent model in the cascade.

Model Inference: Given a new document, apply the trained cascade of models to parse the document and extract metadata. The output of each model is fed into the next model in the cascade, refining the document structure at each step. Finally, the "leaf" labels are assigned to the text elements, resulting in a structured XML/TEI representation of the document.

Metadata Extraction: Once the document has been parsed and structured, the metadata can be easily extracted from the XML/TEI representation by querying the relevant elements and their associated "leaf" labels.

E. Fast-RCNN

In earlier work [3], we addressed this problem by viewing the PDF document as an image and leverage from the advanced progress in computer vision.

The model is an adaptation of Mask R-CNN, a cutting-edge object instance segmentation technique proposed by He et al [33]. It identifies objects within images at the pixel level by extending Faster RCNN with an additional branch for predicting object masks and utilizing Region of Interest (RoI)-Align instead of RoI-Pooling. The binary object mask highlights the position of each object in its bounding box on a pixel-by-pixel basis. In this implementation, Mask R-CNN is combined with a ResNeXt [34] backbone architecture and a Feature Pyramid Network (FPN), following the approach, outlined in [35].

As illustrated in Figure 5, the PDF page $\mathcal{P}$ is first transformed into a pixel image, which serves as input for the RCNN model. The model is composed of three main components: (i) a Feature Pyramid Network (FPN) with ResNeXt as a backbone network, (ii) a Region Proposal Network (RPN), and (iii) RoI (Region of Interest) Heads. As detailed in TableII, the ResNeXt backbone includes a stem block and four stages, each containing multiple bottleneck blocks.

The stem block down-samples the input image twice through a 7×7 convolution with a stride of 2, and max-pooling with a stride of 2, generating a feature map at a 1/4 scale. The subsequent four stages contain bottleneck blocks, each featuring three convolutional layers with kernel sizes of 1×1, 3×3, and 1×1. These stages consist of 3, 4, 23, and 3 bottleneck blocks respectively, and produce feature maps at scales of $1/4$, $1/8$, $1/16$, and $1/32$ [34]. A max-pooling layer with a kernel size of 1 and a stride of 2 is introduced to the final stage of ResNeXt, yielding a feature map at a $1/64$ scale [33].

The second component, the Region Proposal Network (RPN), suggests candidate object bounding boxes utilizing the outputs from the FPN's five stages. Subsequently, a fully convolutional mask prediction branch is integrated into the head [33]. The RoI head employs fully-connected layers to generate refined box locations and classification results from multiple fixed-size features, which are obtained by cropping and warping feature maps. The box head then filters out up to 100 boxes using non-maximum suppression (NMS) to eliminate redundant detections.

Transfer learning is a widely used technique in deep learning for computer vision tasks. It involves retraining pre-trained convolutional networks on smaller, task-specific datasets to fine-tune the weights and biases, leveraging the knowledge gained from one classification task to another [36]. In our study, we employ a source model based on the Detectron2 [37] implementation of Mask R-CNN ResNeXt-101 32x8d FPN. This model was initially fine-tuned on 191,832 images from the PubLayNet dataset [38], which includes annotated images of articles from PubMed Central™ Open Access (PMCOA)
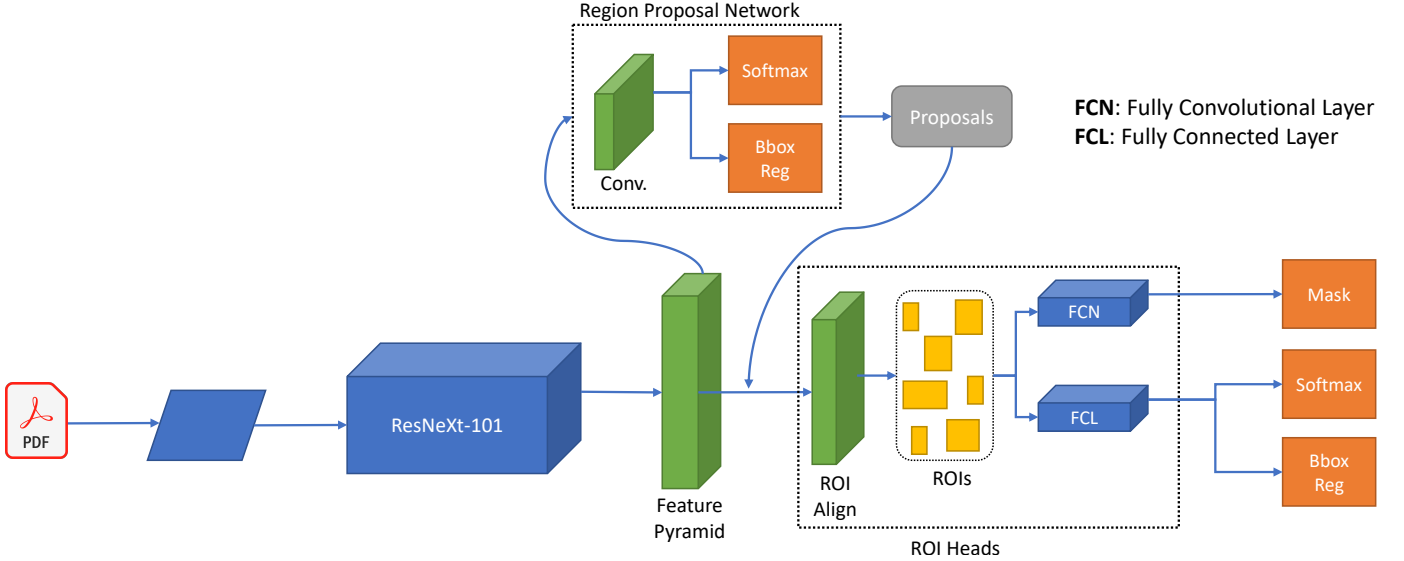
Fig. 5. Mask R-CNN architecture employed for metadata extraction from PDF pages.

Layer name	scale	kernel size		stride
stem	1/4	7×7		2
backbone 1	1/4	$\begin{matrix} 1 & \times & 1 \\ 3 & \times & 3 \\ 1 & \times & 1 \end{matrix}$	$\times 3$	1
backbone 2	1/8	$\begin{matrix} 1 & \times & 1 \\ 3 & \times & 3 \\ 1 & \times & 1 \end{matrix}$	$\times 4$	1
backbone 3	1/16	$\begin{matrix} 1 & \times & 1 \\ 3 & \times & 3 \\ 1 & \times & 1 \end{matrix}$	$\times 23$	1
backbone 4	1/32	$\begin{matrix} 1 & \times & 1 \\ 3 & \times & 3 \\ 1 & \times & 1 \end{matrix}$	$\times 3$	1
max pooling layer	1/64	1×1		2

TABLE II

OVERVIEW OF THE RESNEXT STRUCTURE IN THE RCNN MODEL, HIGHLIGHTING THE NUMBER OF BOTTLENECK BLOCKS AND THE SCALES OF FEATURE MAPS AT EACH STAGE.

featuring five classes: title, text, list, table, and figure. The model is well-suited for extracting metadata from scientific papers since it (i) has a backbone trained on the extensive COCO dataset, (ii) underwent fine-tuning on a large dataset of scientific document images, and (iii) is designed for a task closely related to ours.

To adapt this model for extracting metadata patterns from scientific documents, we first modified the final layer of the source model to output nine target classes (title, authors, journal, abstract, date, DOI, address, affiliation, and email addresses) instead of the original five. Empirical experiments on a subset of 103 random samples from our training dataset showed that the best-performing architecture has two frozen layers and 15k iterations. Based on these findings, we fine-tuned the model using the full training dataset, setting the learning rate to $2.5 \times 10^{(-3)}$.

F. Vision and Natural Language

In earlier work [4], we addressed this problem using a multimodal neural network model that employs NLP together with Computer Vision for metadata extraction.

Figure 6 illustrates the initial step of our process, wherein the text is extracted from $\mathcal{P}$ using CERMINE [39]. Known for its reliability in handling diverse layouts at the line level, CERMINE also provides geometric structural information such as text position and font style.

From each extracted token ω, a set of 16 handcrafted features, denoted as F_{hand}, is derived. A word embedding for the token, denoted as F_{embed}, is also generated, which encapsulates the context and meaning of the words. These two sets of features are then concatenated to form a single feature vector, such that $F_{total} = Concat(F_{hand}, F_{embed})$.

The consolidated vector F_{total} is used as the input for the Natural Language Processing (NLP) sub-model, described in section III-B. Simultaneously, the image of $\mathcal{P}$ is supplied as input to the Computer Vision (CV) model, described in section III-E.

1) Natural Language-based Model: To model the extracted text, we utilized a BiDirectional Long-Short-Term Memory (BiLSTM) due to its proven accuracy in handling textual data, as detailed in section III-B. This sub-model comprises two layers of LSTM models, each with 256 hidden dimensions; the first layer is a forward LSTM, and the second layer is a backward LSTM. Please refer to section III-B for more details about BiLSTM

The input to this model is a word representation vector with a length of 1041. As previously described, this vector is the concatenation of two vectors. The first vector, consisting of 16 units, encapsulates layout features such as the font size of the word, font style, the spacing between the word and the line above or below it, and flags denoting whether the text is italicized, bolded, or adheres to a specific common format like date or email, among others. The second vector contains the
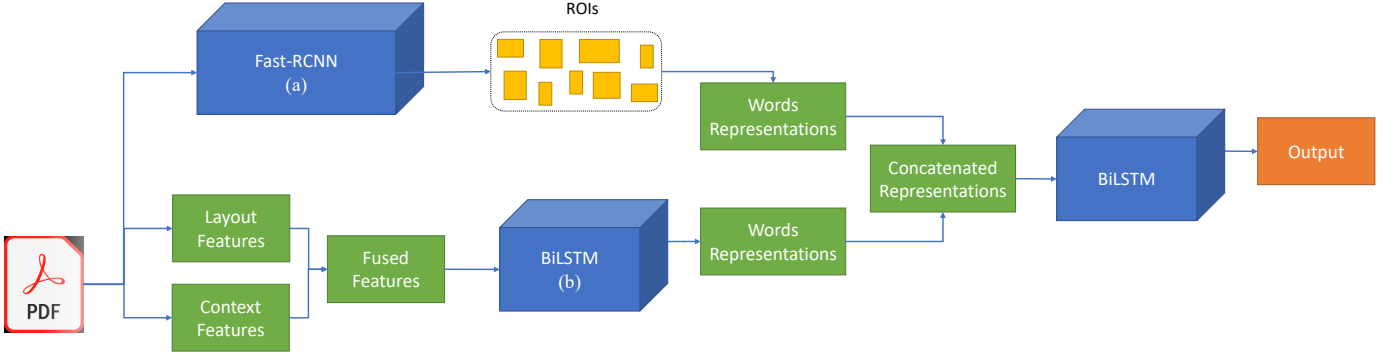
Fig. 6. Multimodal extraction approach, where *(a)* refers to the model described in section III-E and *(b)* refers to the model described in section III-B

ELMO [40] embedding results, derived from a model trained on German documents.

Following the two LSTM layers, a fully connected layer of 512 units is in place, ending with an output layer of 10 neurons, representing the metadata classes. The output layer employs a softmax activation function to generate probability scores for the word's affiliation with each of the classes.

2) Computer Vision-based Model: Building on the proven efficiency of MexPub [3], detailed in section III-E, we leverage it as the Computer Vision (CV) sub-model, feeding it with $\mathcal{P}$. This model yields output in the form of bounding boxes labelled with metadata classes. Subsequently, we extract the text enclosed within the bounding boxes as identified by the CV sub-model and compile the probabilities for all potential classes within that box prior to their submission to the classifier. It's important to note that the CV sub-model may also generate bounding boxes that are unclassified, meaning they do not associate with any of the predefined classes.

3) Classifier: In the final stage of our architecture pipeline, the output of the NLP and CV sub-models is fused using a SoftMax classifier. Specifically, all words from the document are extracted and sequentially traversed. Their vector representations, generated by the sub-models, are then concatenated. A BiLSTM, notable for its bidirectional operation and capability to preserve information from both past and future states, is employed in this context as well. This is especially advantageous for understanding context and discerning patterns within sentences or paragraphs (e.g., if the adjacent words are titles, the current word is highly likely to be a title as well). The model specifically takes in a vector of length 20, resulting from the concatenation of both sub-model outputs. The model's output is a probability distribution of length 10 corresponding to all classes. As depicted in Figure 2, the classifier comprises two stacked LSTM layers (forward and backward LSTMs) each with 256 hidden dimensions. A fully connected layer follows these two layers, encompassing 512 input nodes and 10 output nodes activated by a SoftMax function.

G. Text Map Approach

The text map approach presents a novel framework that jointly optimizes spatial and semantic information for metadata extraction. Given the first page $\mathcal{P}$ of a PDF document and its sequence of observed words $\omega = \langle \omega_1, \omega_2, \cdots, \omega_n \rangle$, our goal is to learn a mapping function γ that assigns metadata labels while preserving both spatial and semantic relationships.

1) Two Phase Processing: The approach processes documents through two complementary phases as depicted in Figure 7:

a) Phase 1: Spatial Representation: transforms $\mathcal{P}$ into a grayscale representation G that preserves structural information:

$$G = \phi(\mathcal{P}) \in \mathbb{R}^{H \times W} \quad (9)$$

where H and W are the height and width of the page, respectively. This transformation preserves the spatial distribution of text and structural elements across the document.

b) Phase 2: Semantic Mapping: The semantic mapping differs based on the chosen embedding function ψ. For Word2Vec, each token ω_i is embedded individually:

$$E_i = \psi_{Word2Vec}(\omega_i) \in \mathbb{R}^d \quad (10)$$

For BERT, entire text blocks $B_j = \omega_1, ..., \omega_k j$ are embedded together to capture contextual relationships:

$$E_j = \psi BERT(B_j) \in \mathbb{R}^{k \times d} \quad (11)$$

where d is the embedding dimension, and k is the number of tokens in the block.

2) Spatial-Semantic Integration: The key innovation in our approach is the integration of spatial and semantic information through a carefully designed interpolation process:

1. **Region Identification**: The regions of interest $R = R_1, ..., R_k$ are determined by the locations of text content in the document. Each region R_i corresponds to a bounding box containing embedded text:

$$R_i = \{(x, y, w, h) \mid$$
$$\text{text content exists at } (x, y) \text{ with width } w \text{ and height } h\} \quad (12)$$

The regions are naturally defined by the presence of text content that has been extracted and embedded. This ensures that our regions directly correspond to actual textual content in the document.
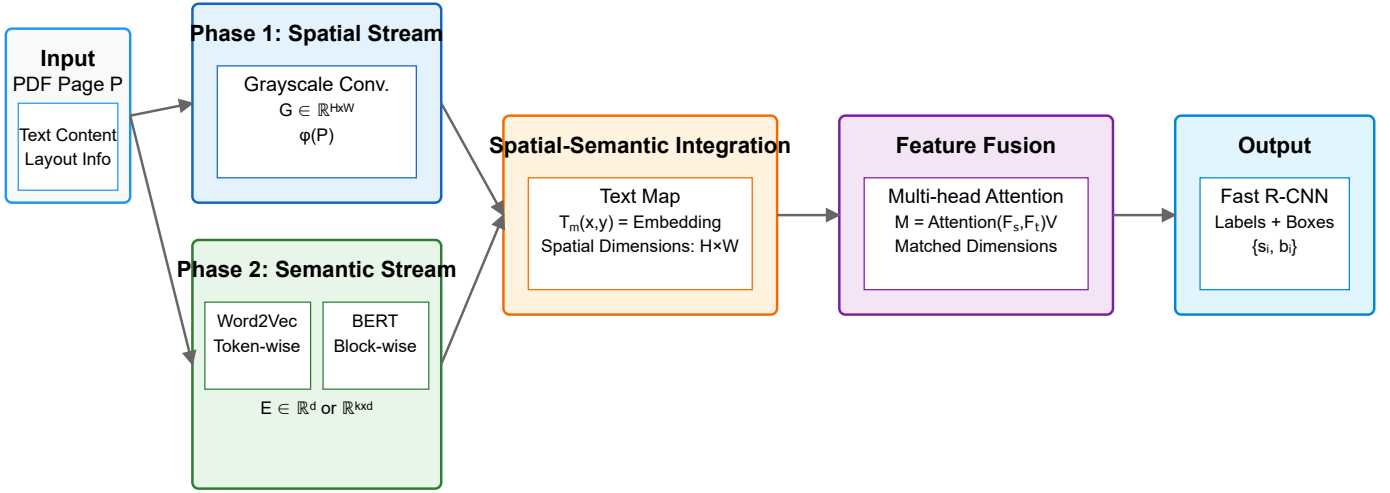
Fig. 7. Overview of the TextMap approach

a) Embedding Interpolation: For each region R_i, the embedding is directly mapped into the spatial coordinates of that region to create the text map T_m. We perform a straightforward mapping to ensure that each spatial location in the text map contains the embedding of the text (token or block) that appears at that location in the original document. For Word2Vec, where each token has its own embedding:

$$T_m(x,y) = E_j \quad \text{where} \quad (x,y) \in R_i \text{ contains token } \omega_j \tag{13}$$

For BERT, where entire blocks are embedded together:

$$T_m(x,y) = E_i \quad \text{where} \quad (x,y) \in R_i \text{ contains block } B_i \tag{14}$$

3. **Feature Fusion**: After interpolation, both the grayscale representation G and the text map T_m have matching spatial dimensions, as the embeddings have been mapped to their corresponding regions' coordinates in the document space. Specifically:

$G \in \mathbb{R}^{H \times W}$ from the spatial stream $T_m \in \mathbb{R}^{H \times W \times d}$ from the interpolated embeddings, where d is the embedding dimension

This dimensional alignment allows us to apply convolutional operations to both streams:

$$F_{spatial} = Conv2D(G) \in \mathbb{R}^{H' \times W' \times C} \tag{15}$$

$$F_{semantic} = Conv2D(T_m) \in \mathbb{R}^{H' \times W' \times C} \tag{16}$$

where H' and W' are the reduced spatial dimensions after convolution, and C is the number of output channels. These spatially-aligned feature maps are then fused using a multi-head attention mechanism:

$$M = Attention(F_{spatial}, F_{semantic})V \tag{17}$$

where V is a learnable value matrix. This fusion process effectively combines the structural information from the spatial stream with the semantic information from the text embeddings, while maintaining spatial correspondence between the two streams.

3) Segmentation and Classification: The fused features M are processed through a Fast R-CNN architecture for final segmentation and classification. For each identified region, we predict both the class label and bounding box refinements:

$$\{s_i, b_i\} = FastRCNN(R_{refined}) \tag{18}$$

where s_i is the metadata label and b_i are the refined coordinates.

4) Joint Optimization Framework: The model is trained through a joint optimization framework that combines three objectives:

1. **Semantic Objective** ($\mathcal{L}_{semantic}$):

$$\mathcal{L}_{semantic}(\theta) = -\sum_{i=1}^{n} \log P(s_i | \omega_i, E_i) \tag{19}$$

This term ensures accurate metadata label assignment based on textual content.

2. **Spatial Objective** ($\mathcal{L}_{spatial}$):

$$\mathcal{L}_{spatial}(\theta) = \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} \|f(G_i) - f(G_j)\|_2^2 \cdot \mathbb{K}[s_i = s_j] \tag{20}$$

where $\mathcal{N}(i)$ represents the spatial neighbors of token i, f is a feature extraction function, and $\mathbb{K}$ is the indicator function.

3. **Cross-modal Objective** ($\mathcal{L}_{cross}$):

$$\mathcal{L}_{cross}(\theta) = -\sum_{i=1}^{n} \log P(s_i | E_i, f(G_i)) \tag{21}$$

The complete optimization objective is:

$$\mathcal{J}(\theta) = \alpha \mathcal{L}_{semantic}(\theta) + \beta \mathcal{L}_{spatial}(\theta) + \gamma \mathcal{L}_{cross}(\theta) \tag{22}$$

where α, β, and γ are learnable parameters that balance the contribution of each term.

5) Training and Inference: During training, we optimize $\mathcal{J}(\theta)$ using mini-batch stochastic gradient descent:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{J}(\theta_t) \tag{23}$$

where η is the learning rate.

At inference time, for a new document page $\mathcal{P}$, we: 1. Generate spatial features $G = \phi(\mathcal{P})$ 2. Compute embeddings $E_i = \psi(\omega_i)$ for each token 3. Identify regions 4. Apply the trained model to obtain metadata labels:

$$s_i = \arg\max_{l \in L} P(l|E_i, f(G_i)) \tag{24}$$

This formulation ensures that:

- Tokens with similar semantics and spatial proximity are likely to share labels
- The model can handle variable document layouts
- Both local and global document structure are considered
- The extraction is robust to variations in formatting

IV. EXPERIMENTS

In this section, we compare the performance of the described methods in the previous section on two challenging datasets.

A. Dataset

This section presents a comparative analysis of the different methodologies aimed at extracting metadata from academic PDF documents. To this end, we prepared two challenging datasets, namely SSOAR-MVD and S-PMRD.

1) SSOAR Multidisciplinary Vision Dataset (SSOAR-MVD): To ensure a fair comparison of all the methods described in the section, we ensured they were all applied to the same dataset. As a result, we collected a challenging dataset of 50,000 documents from the SSOAR repositor[5]. The SSOAR stores publications from various publishers, including small and mid-sized ones, covering a range of disciplines known for their challenging layout formats, such as Social Sciences, Humanities, Law, and Administration. This guarantees that a wide variety of templates are included in the dataset. During the scraping process, each document was downloaded along with its textual metadata provided by the SSOAR repository. However, since most computer vision approaches require labelled images (i.e., bounding boxes), a preprocessing phase was conducted to ensure this. Each document underwent the following steps:

- The document is converted into an image.
- Using an open-source tool provided by TensorFlow, blocks of text were extracted from the document along with their respective bounding boxes.
- The similarity between each text block and metadata class was measured (using the collected textual metadata from the SSOAR).
- If a certain block had a near-perfect similarity with a specific class, the corresponding bounding box was assigned to that class. Otherwise, it was assigned a class "other".

[5]https://www.gesis.org/en/ssoar/home

2) S2ORC PDF Metadata Refinement Dataset (S-PMRD): To evaluate the efficacy of these methods on an authentic corpus, we meticulously curated a subset from the Semantic Scholar Open Research Corpus (S2ORC) [41]. While S2ORC offers a vast repository of millions of scholarly articles, our preliminary analyses revealed significant discrepancies between the raw textual data provided by S2ORC and the content within the corresponding PDF documents. Notably, certain text segments available in the S2ORC dataset were absent from the PDFs, and this does not explain the extraction methodologies employed by S2ORC. These inconsistencies pose substantial challenges for conducting detailed academic analyses that necessitate precise text alignment, such as citation context analysis, text-based data mining, and metadata extraction.

To address these challenges, we developed a specialized sub-corpus of S2ORC, specifically aimed at extracting metadata directly from PDF documents, thereby bypassing the potential inaccuracies inherent in pre-extracted text. The processing pipeline for each document included in this sub-corpus is as follows:

- Using the Digital Object Identifier (DOI) from S2ORC, additional metadata, including links to the actual PDF documents, is retrieved from CrossRef [42] using their API.
- PDFs are downloaded when available, acknowledging that some links may be inactive or access-restricted.
- Text is extracted from the first page of each PDF. This extracted text undergoes a normalization process to eliminate irregularities such as inconsistent spacing and line breaks, thereby ensuring uniformity across the dataset.
- We employ both exact and fuzzy matching techniques to extract critical metadata elements, including author names, titles, abstracts, and affiliations. This dual-method approach accommodates minor discrepancies due to text recognition errors or formatting variations.
- Each document is converted into an image representation.
- For each identified metadata element, bounding boxes are determined based on their locations within the text.

Ultimately, each instance in this dataset comprises:

- The original PDF file.
- The normalized text extracted from the PDF.
- An image of the first page of the PDF.
- Metadata attributes annotated with both their positions in the extracted text and their coordinates on the corresponding image.

V. SETTINGS

In this study, we employ a token-level evaluation to validate each model's effectiveness, where each predicted token is compared against the ground truth annotations in our dataset. This evaluation is conducted using standard metrics such as Precision, Recall, and F1-Score.

A. Results

In the first analysis, we evaluate the performance of the presented methods on the datasets SSOAR-MVD and S-PMRD, as depicted in Tables Tables III and IV, illustrating

the outcomes in terms of Precision, Recall, and F1-Score to provide a multi-dimensional comparison. Notably, the proposed TextMap-Word2Vec method achieves the highest F1-score of 0.913, suggesting that capturing both the semantics of the PDF content and its layout is particularly effective for this task. This is evident by the performance of the Fast-RCNN and Vision-Langauge methods which also leverage the layout and the appearance of the PDF document. In contrast, traditional models like CRF exhibit lower performance metrics, which may indicate difficulties in adapting to the multifaceted nature of PDF content, where layout and semantic context play crucial roles. An important observation is that the embedding approach in TextMap plays a significant role in the performance of the model.

To give a better overview of the performance of each method, we present below the result of each method for each attribute. Table V presents the detailed results of CRF on SSOAR-MVD. As demonstrated, CRF excels in attributes with structured and predictable formats such as Dates (F1-score: 0.731) and DOIs (F1-score: 0.749), where the patterned nature of the data plays to the strengths of the CRF's sequence modelling capabilities. However, CRF struggles with more complex and less structured text such as Titles (F1-score: 0.433) and Abstracts (F1-score: 0.392), where the variability in content and formatting challenges its ability to accurately predict boundaries and content. The moderate success in extracting Authors (F1-score: 0.482) and higher performance in Affiliation data (F1-score: 0.672) suggest that CRF can handle semi-structured text effectively when patterns are somewhat predictable.

Table VI presents the detailed results of Bi-LSTM on SSOAR-MVD. The BiLSTM method exhibits strong performance across several metadata categories such as 'Title', 'Author', etc. This reflects its robust capability to capture both the context and sequence of text within scholarly PDF documents. Specifically, the method maintains strong performance in handling Abstracts and Dates, with F1-scores slightly above 0.910. This indicates BiLSTM's adeptness at managing narrative content and specific formatted text. However, the slightly lower performance in extracting Affiliation data, with an F1-score of 0.833, suggests some challenges in dealing with categories of short strings.

Table VII presents the detailed results of LSTM-CRF on SSOAR-MVD. The LSTM-CRF method demonstrates moderate efficacy in extracting different metadata categories, notably outperforming traditional CRF models that rely on hand-crafted features. This enhancement suggests that the LSTM architecture provides a more robust feature representation for CRF to utilize effectively. However, despite its competencies across various categories, LSTM-CRF does not surpass the BiLSTM method, which exhibits superior handling of long-term sequential and contextual data dependencies.

Table VIII presents the detailed results of Grobid on SSOAR-MVD. Despite its simple design, GROBID exhibits robust performance across different categories. It particularly excels in extracting Authors and Abstracts, achieving impressive F1 scores of 0.958 and 0.935 respectively. These high scores can be attributed to GROBID's effective appli-

cation of its cascading sequence labelling models, which are adept at handling well-structured and clearly delineated data. However, GROBID encounters some variability in categories involving more complex or less standardized information, such as Address and Affiliation. This variability stems from these categories' inherent challenges, including inconsistent formatting and multifaceted data structures that can complicate the parsing process. While the cascading approach of GROBID generally enhances its capability to manage hierarchical document structures efficiently, it occasionally struggles with elements that lack a clear or uniform presentation.

Table IX presents the detailed results of Fast-RCNN on SSOAR-MVD. The model demonstrates high precision and recall across most categories, with solid performance in the Title, Abstract, and Journal categories, suggesting robustness in recognizing well-defined metadata fields. The Email and Authors categories also show commendable accuracy. However, the model indicates slightly weaker performance in the Address and Date categories, with F1-scores of 0.837 and 0.832, respectively. This could be due to variability in the formatting and presentation of these metadata elements across documents, which poses challenges in consistent extraction. The small variance between Macro Average and Micro Average metrics indicates a balanced performance across different categories, without significant bias toward any particular type of metadata.

Tables X, XI, XII present the performance metrics across three different configurations of the TextMap model, using BERT, Word2Vec, and Char2Vec embeddings. we can observe varied performances that highlight the strengths and weaknesses of each embedding strategy with the TextMap approach. TextMap using BERT Embeddings configuration (Table X) demonstrates strong performance across several categories, particularly in Authors, Abstract, and Journal, with F1-scores > 0.92. This suggests that BERT's deep contextual embeddings are particularly effective at extracting structured text like titles and authors' details, typically well-defined in the document. The lower performance in the Affiliation and Address categories, with relatively lower F1 scores, could indicate challenges in capturing less consistently formatted information.

TextMap using Word2Vec Embeddings configuration (Table XI) generally performs well, particularly in the Authors and Abstract categories. This suggests good generalization in capturing both the semantic and structural patterns in data, though slightly less effectively than BERT in terms of overall averages. However, as shown in Table XIII, it has a lower computational cost in both training and inference. Consequently, this configuration provides a good balance between performance and computational efficiency, especially suitable for environments where computational resources or training data are limited.

TextMap using Char2Vec Embeddings configuration (Table XII) demonstrates a notable decline in performance across most categories compared to the BERT and Word2Vec models. It performs best in the DOI category with an F1-score > 0.9 but struggles particularly with Abstract and Journal metadata, with F1-scores < 0.8. In conclusion, Char2Vec, while useful

TABLE III
OVERALL PERFORMANCE COMPARISON OF DIFFERENT METHODS ON SSOAR-MVD

Method	Precision Macro	Precision Micro	Recall Macro	Recall Micro	F1-score
CRF	0.609	0.544	0.524	0.471	0.57
BiLSTM	0.901	0.89	0.898	0.861	0.9
LSTM-CRF	0.778	0.713	0.745	0.697	0.761
GROBID	0.854	0.671	0.794	0.551	0.821
Fast-RCNN	0.9	0.915	0.896	0.904	0.898
Vision-Language	0.935	**0.94**	0.902	0.904	0.92
TextMap-Bert	0.908	0.887	0.902	0.897	0.905
TextMap-Word2Vec	**0.917**	0.92	**0.91**	**0.904**	**0.913**
TextMap-Char2Vec	0.845	0.8	0.849	0.797	0.847

TABLE IV
OVERALL PERFORMANCE COMPARISON OF DIFFERENT METHODS ON S-PMRD

Method	Precision Macro	Precision Micro	Recall Macro	Recall Micro	F1-score
CRF	0.573	0.521	0.501	0.45	0.511
BiLSTM	0.883	0.872	0.898	0.863	0.889
LSTM-CRF	0.740	0.707	0.736	0.692	0.724
GROBID	0.822	0.651	0.787	0.542	0.791
Fast-RCNN	0.874	0.906	0.886	0.912	0.893
Vision-Language	0.91	**0.923**	0.904	0.911	0.903
TextMap-Bert	0.882	0.860	0.916	0.887	0.894
TextMap-Word2Vec	**0.892**	0.902	**0.91**	**0.902**	**0.901**
TextMap-Char2Vec	0.815	0.786	0.841	0.792	0.821

TABLE V
PERFORMANCE METRICS FOR CRF METHOD ON SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.568	0.35	0.433
Abstract	0.457	0.344	0.392
Authors	0.57	0.418	0.482
Email	0.612	0.607	0.609
Address	0.522	0.481	0.5
Date	0.754	0.71	0.731
Journal	0.547	0.577	0.561
Affiliation	0.663	0.682	0.672
DOI	0.795	0.709	0.749
Macro Average	0.609	0.524	0.57
Micro Average	0.544	0.471	N/A

TABLE VII
PERFORMANCE METRICS FOR LSTM-CRF METHOD ON SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.741	0.699	0.719
Abstract	0.688	0.7	0.693
Authors	0.84	0.815	0.827
Email	0.801	0.782	0.791
Address	0.725	0.76	0.742
Date	0.89	0.822	0.854
Journal	0.739	0.727	0.732
Affiliation	0.774	0.68	0.723
DOI	0.81	0.724	0.764
Macro Average	0.778	0.745	0.761
Micro Average	0.713	0.697	N/A

TABLE VI
PERFORMANCE METRICS FOR BiLSTM METHOD ON SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.931	0.91	0.920
Abstract	0.908	0.914	0.911
Authors	0.944	0.93	0.937
Email	0.881	0.86	0.870
Address	0.9	0.882	0.891
Date	0.916	0.905	0.910
Journal	0.865	0.891	0.878
Affiliation	0.814	0.853	0.833
DOI	0.952	0.94	0.946
Macro Average	0.901	0.898	0.9
Micro Average	0.89	0.861	N/A

TABLE VIII
PERFORMANCE METRICS FOR GROBID METHOD ON SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.764	0.667	0.951
Abstract	0.84	0.79	0.935
Authors	0.934	0.855	0.958
Email	0.91	0.812	0.893
Address	0.722	0.78	0.872
Date	0.855	0.877	0.873
Journal	0.887	0.75	0.927
Affiliation	0.859	0.813	0.818
DOI	0.911	0.8	0.916
Macro Average	0.854	0.794	0.821
Micro Average	0.671	0.551	N/A

in certain niche applications (like OCR and typo-sensitive extractions), may not be suitable for tasks requiring deep semantic understanding such as understanding the semantics of a scholarly text.

In addition to comparing the models' performance in terms of precision, recall, and F1 score, we compare their computational complexity using the SSOAR-MVD dataset.

This comparison reveals a range of trade-offs between computational efficiency and performance accuracy. CRF offers the quickest inference time and requires the least training time, making them ideal for environments where speed is prioritized over cutting-edge accuracy. BiLSTM models, while requiring more extensive training, provide rapid inference capabilities, suitable for real-time applications once the model is deployed. LSTM-CRF models combine the deep learning prowess of LSTMs with the structured output of CRFs did

TABLE IX
PERFORMANCE METRICS FOR FAST-RCNN METHOD ON SSOAR-MVD

Category	Precision Macro	Recall Macro	F1-score
Title	0.966	0.95	0.958
Abstract	0.915	0.922	0.918
Authors	0.91	0.938	0.924
Email	0.933	0.917	0.925
Address	0.875	0.802	0.837
Date	0.825	0.84	0.832
Journal	0.94	0.925	0.932
Affiliation	0.839	0.876	0.857
DOI	0.901	0.893	0.897
Macro Average	0.9	0.896	0.898
Micro Average	0.915	0.904	N/A

TABLE X
PERFORMANCE METRICS FOR TEXTMAP USING BERT EMBEDDINGS.

Category	Precision Macro	Recall Macro	F1-score
Title	0.954	0.949	0.951
Abstract	0.921	0.95	0.935
Authors	0.967	0.949	0.958
Email	0.91	0.877	0.893
Address	0.889	0.856	0.872
Date	0.855	0.891	0.873
Journal	0.924	0.93	0.927
Affiliation	0.822	0.815	0.818
DOI	0.931	0.902	0.916
Macro Average	0.908	0.902	0.905
Micro Average	0.887	0.897	N/A

TABLE XI
PERFORMANCE METRICS FOR TEXTMAP USING WORD2VEC
EMBEDDINGS

Category	Precision Macro	Recall Macro	F1-score
Title	0.962	0.922	0.941
Abstract	0.933	0.952	0.942
Authors	0.978	0.949	0.963
Email	0.93	0.899	0.914
Address	0.904	0.86	0.881
Date	0.852	0.907	0.878
Journal	0.924	0.94	0.931
Affiliation	0.834	0.849	0.841
DOI	0.933	0.915	0.923
Macro Avrerage	0.917	0.91	0.913
Micro Average	0.92	0.904	N/A

TABLE XII
PERFORMANCE METRICS FOR TEXTMAP USING CHAR2VEC EMBEDDINGS

Category	Precision Macro	Recall Macro	F1-score
Title	0.851	0.874	0.862
Abstract	0.77	0.8	0.785
Authors	0.849	0.815	0.832
Email	0.902	0.89	0.896
Address	0.86	0.881	0.870
Date	0.875	0.842	0.858
Journal	0.782	0.809	0.795
Affiliation	0.804	0.83	0.817
DOI	0.917	0.905	0.911
Macro Average	0.845	0.849	0.847
Micro Average	0.8	0.797	N/A

not achieve higher accuracy compared to Bi-LSTM models and has longer training times and moderately slow inference speeds. GROBID, tailored specifically for document processing tasks, demands the most extended training period and

TABLE XIII
AVERAGE TRAINING AND INFERENCE TIMES FOR DIFFERENT MACHINE LEARNING MODELS USED IN METADATA EXTRACTION. THE TABLE LISTS THE ESTIMATED AVERAGE TRAINING AND INFERENCE TIMES FOR EACH MODEL AND STANDARD DEVIATIONS FOR THESE ESTIMATES.

Model	Training time	Inference time
CRF	36 ± 7.2 hours	0.5 ± 0.01 seconds
BiLSTM	84 ± 7.1 hours	0.1 seconds
LSTM-CRF	126 ± 3.7 hours	0.2 seconds
GROBID	156 ± 7.2 hours	1.2 ± 0.6 seconds
Fast-RCNN	60 ± 6.8 hours	1.3 ± 0.4 seconds
Vision-Language	192 ± 14.5 hours	3.5 ± 1.11 seconds
TextMap-Bert	172 ± 13.3 hours	1.3 ± 0.58
TextMap-Word2Vec	92 ± 6.2	0.4 seconds
TextMap-Char2Vec	90 ± 7.0	0.4 seconds

exhibits slower inference times, reflecting its comprehensive analytical depth. Fast R-CNN, effective in precise localization of content within documents, also shows a moderate training duration with slower inference, suited to applications where precision is more critical than speed. Vision-language models, though offering superior performance where an understanding of both visual cues and textual information is necessary, involve the longest training durations and the slowest inference rates, which could be a significant drawback in time-sensitive scenarios.

Lastly, TextMap models using BERT, Word2Vec, and Char2Vec embeddings demonstrate a spectrum of efficiencies, with BERT providing high accuracy but slower inference and longer training times, whereas Word2Vec offers a more balanced approach, making it preferable for scenarios that demand both efficiency and effectiveness.

B. Limitations

While the models examined in this study demonstrate considerable potential for metadata extraction from scholarly PDF documents, several limitations must be acknowledged to fully appreciate their applicability and scope of use.

- **Dependency on Training Data**: All models, particularly deep learning-based ones like BiLSTM, LSTM-CRF, and TextMap with BERT embeddings, exhibit a high dependency on the quantity and quality of the training data. Their performance is contingent upon the availability of large, annotated datasets. This reliance can limit their practical deployment in scenarios where such datasets are not readily available or are too domain-specific.
- **Adaptability to Rapid Changes**: The field of digital publishing is evolving, with new standards and formats emerging. The adaptability of these models to such rapid changes has not been thoroughly tested, raising concerns about their long-term viability without continuous updates and retraining.
  Error Propagation: In multi-stage models like GROBID or Fast-RCNN, errors in early processing stages can propagate, leading to compounded errors in metadata extraction outcomes. This cascade effect can significantly affect the overall quality of the extracted metadata.

VI. Conclusion

This study has conducted a comprehensive comparison of various machine learning models to evaluate their effectiveness in extracting metadata from two challenging datasets. The analysis revealed significant variations in performance and computational demands across the models, underscoring the importance of selecting an appropriate model and architecture tailored to specific use case requirements.

The CRF and BiLSTM models demonstrated rapid inference capabilities coupled with robust performance, making them ideal candidates for real-time applications. In contrast, the LSTM-CRF hybrid model, despite combining the strengths of LSTMs and the structured output capabilities of CRFs, did not achieve results on par with its component technologies.

Models that integrate vision and language modalities, while resource-intensive, deliver depth and precision in analysis that simpler models cannot achieve. This sophistication makes them particularly valuable in scenarios where the accuracy of extracted metadata critically impacts the outcomes of subsequent processes.

The TextMap models, which leverage various embeddings such as BERT, Word2Vec, and Char2Vec, offer a spectrum of choices balancing training and inference times with performance. Among these, BERT embeddings stand out for their exceptional accuracy, albeit at a higher computational cost, illustrating the fundamental trade-offs between resource investment and extraction efficacy.

Ultimately, the selection of a metadata extraction model should be driven not only by dataset characteristics but also by the practical constraints of the use case—available computational resources, required inference speed, and the trade-offs between precision and performance that stakeholders are prepared to accept. Future research should consider the potential of hybrid models and the development of more efficient training algorithms to further optimize the application of machine learning in metadata extraction tasks, enhancing both their efficiency and accessibility.

References

[1] P. Manghi, C. Atzori, A. Bardi, M. Baglioni, J. Schirrwagen, H. Dimitropoulos, S. La Bruzzo, I. Foufoulas, A. Mannocci, M. Horst *et al.*, "Openaire research graph dump," 2022.

[2] M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker, and S. Auer, "Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge," in *Proceedings of the 10th international conference on knowledge capture*, 2019, pp. 243–246.

[3] Z. Boukhers, N. Beili, T. Hartmann, P. Goswami, and M. A. Zafar, "Mexpub: Deep transfer learning for metadata extraction from german publications," in *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 2021, pp. 250–253.

[4] Z. Boukhers and A. Bouabdallah, "Vision and natural language for metadata extraction from scientific pdf documents: a multimodal approach," in *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, 2022, pp. 1–5.

[5] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn *et al.*, "Clinical information extraction applications: a literature review," *Journal of biomedical informatics*, vol. 77, pp. 34–49, 2018.

[6] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[7] P. R. Nayaka and R. Ranjan, "An efficient framework for metadata extraction over scholarly documents using ensemble cnn and bilstm technique," in *2023 2nd International Conference for Innovation in Technology (INOCON)*. IEEE, 2023, pp. 1–9.

[8] D. Ali, K. Milleville, S. Verstockt, N. Van de Weghe, S. Chambers, and J. M. Birkholz, "Computer vision and machine learning approaches for metadata enrichment to improve searchability of historical newspaper collections," *Journal of Documentation*, 2023.

[9] V. Balasubramanian, S. G. Doraisamy, and N. K. Kanakarajan, "A multimodal approach for extracting content descriptive metadata from lecture videos," *Journal of Intelligent Information Systems*, vol. 46, pp. 121–145, 2016.

[10] M.-Y. Day, R. T.-H. Tsai, C.-L. Sung, C.-C. Hsieh, C.-W. Lee, S.-H. Wu, K.-P. Wu, C.-S. Ong, and W.-L. Hsu, "Reference metadata extraction using a hierarchical knowledge representation framework," *Decision Support Systems*, vol. 43, no. 1, pp. 152–167, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923606001205

[11] A. Kawtrakul and C. Yingsaeree, "A unified framework for automatic metadata extraction from electronic document," in *Proceedings of The International Advanced Digital Library Conference. Nagoya, Japan*, 2005.

[12] H. Han, E. Manavoglu, H. Zha, K. Tsioutsiouliklis, C. L. Giles, and X. Zhang, "Rule-based word clustering for document metadata extraction," in *Proceedings of the 2005 ACM Symposium on Applied Computing*, ser. SAC '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 1049–1053. [Online]. Available: https://doi.org/10.1145/1066677.1066917

[13] H. Li, I. Councill, W.-C. Lee, and C. L. Giles, "Citeseerx: an architecture and web service design for an academic document search engine," in *Proceedings of the 15th international conference on World Wide Web*, 2006, pp. 883–884.

[14] K. Seymore, A. Mccallum, and R. Rosenfeld, "Learning hidden markov model structure for information extraction," in *In AAAI 99 Workshop on Machine Learning for Information Extraction*, 1999, pp. 37–42.

[15] F. Peng and A. McCallum, "Information extraction from research papers using conditional random fields," *Inf. Process. Manage.*, vol. 42, no. 4, p. 963–979, 2006. [Online]. Available: https://doi.org/10.1016/j.ipm.2005.09.002

[16] H. Han, C. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. Fox, "Automatic document metadata extraction using support vector machines," in *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, 2003, pp. 37–48.

[17] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *CoRR*, vol. abs/1508.01991, 2015. [Online]. Available: http://arxiv.org/abs/1508.01991

[18] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *CoRR*, vol. abs/1511.08308, 2015. [Online]. Available: http://arxiv.org/abs/1511.08308

[19] D. An, L. Gao, Z. Jiang, R. Liu, and Z. Tang, "Citation metadata extraction via deep neural network-based segment sequence labeling," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1967–1970. [Online]. Available: https://doi.org/10.1145/3132847.3133074

[20] I. G. Councill, C. L. Giles, and M.-Y. Kan, "Parscit: an open-source crf reference string parsing package," *LREC, Vol. 8.*, p. 661–667, 2008.

[21] C.-C. Chen, K.-H. Yang, C.-L. Chen, and J.-M. Ho, "Bibpro: A citation parser based on sequence alignment," *IEEE Transactions on Knowledge and Data Engineering 24, 2 (2012)*, p. 236–250, 2012.

[22] S. Anzaroot and A. Mccallum, "A new dataset for fine-grained citation field extraction," *ICML Workshop on Peer Reviewing and Publishing Models.*, 2013.

[23] C. G. Stahl, S. R. Young, D. Herrmannova, R. M. Patton, and J. C. Wells, "Deeppdf: A deep learning approach to extracting text from pdfs," 2018. [Online]. Available: https://www.osti.gov/biblio/1460210

[24] Z. Boukhers, N. Beili, T. Hartmann, P. Goswami, and M. A. Zafar, "Mexpub: Deep transfer learning for metadata extraction from german publications," in *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, 2021.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[26] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *arXiv preprint arXiv:1611.05431*, 2016.

[27] V. Balasubramanian, S. G. Doraisamy, and N. K. Kanakarajan, "A multimodal approach for extracting content descriptive metadata from

lecture videos," *J. Intell. Inf. Syst.*, vol. 46, no. 1, p. 121–145, 2016. [Online]. Available: https://doi.org/10.1007/s10844-015-0356-5

[28] R. Liu, L. Gao, D. An, Z. Jiang, and Z. Tang, "Automatic document metadata extraction based on deep networks," in *Natural Language Processing and Chinese Computing*, X. Huang, J. Jiang, D. Zhao, Y. Feng, and Y. Hong, Eds. Cham: Springer International Publishing, 2018, pp. 305–317.

[29] A. Souza, V. Moreira, and C. Heuser, "Arctic: metadata extraction from scientific papers in pdf using two-layer crf," in *Proceedings of the 2014 ACM symposium on document engineering*, 2014, pp. 121–130.

[30] "Grobid," 2008–2021.

[31] R. Alzaidy, C. Caragea, and C. L. Giles, "Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents," in *The world wide web conference*, 2019, pp. 2551–2557.

[32] S. Dai, Y. Ding, Z. Zhang, W. Zuo, X. Huang, and S. Zhu, "Grantextractor: Accurate grant support information extraction from biomedical fulltext based on bi-lstm-crf," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 18, no. 1, pp. 205–215, 2019.

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[34] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[36] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[37] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[38] X. Zhong, J. Tang, and A. J. Yepes, "Publaynet: largest dataset ever for document layout analysis," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1015–1022.

[39] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek, and L. Bolikowski, "Cermine: Automatic extraction of structured metadata from scientific literature," *Int. J. Doc. Anal. Recognit.*, vol. 18, no. 4, p. 317–335, 2015. [Online]. Available: https://doi.org/10.1007/s10032-015-0249-8

[40] W. Che, Y. Liu, Y. Wang, B. Zheng, and T. Liu, "Towards better ud parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation," *arXiv preprint arXiv:1807.03121*, 2018.

[41] K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. S. Weld, "S2orc: The semantic scholar open research corpus," *arXiv preprint arXiv:1911.02782*, 2019.

[42] G. Hendricks, D. Tkaczyk, J. Lin, and P. Feeney, "Crossref: The sustainable source of community-owned scholarly metadata," *Quantitative Science Studies*, vol. 1, no. 1, pp. 414–427, 2020.