IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Metanest: A Survey Of AI-Powered Tools For Scientific Literature Metadata Extraction

[1]Paras Natekar, [2]Harsh Salunke, [3]Harsh Pardeshi, [4]Vighnesh Nair, [5] Dr. Rohini Palve (Professor)
Dept. of Computer Engineering
[1]Terna Engineering College
Nerul, Navi Mumbai, India

***Abstract:*** The rapid proliferation of scientific literature presents a significant challenge for researchers in efficiently discovering relevant information and extracting key metadata. While traditional keyword-based search engines exist, they often fail to capture the semantic nuances of complex research topics or handle the diverse layouts of scholarly documents. This has led to the development of intelligent metadata extraction systems that leverage layout analysis, Natural Language Processing (NLP), and machine learning techniques. This paper provides a survey of current approaches for automatic metadata extraction from scientific documents, particularly PDFs. We review and compare four distinct methodologies: a layout-aware BERT-based model (LAME), an automated framework combining layout analysis and specialized NLP models (AutoIE), a classification-based approach using Support Vector Machines (SVM), and a templatebased system focused on quality mining (PDFDataExtractor). We analyze these systems based on their core architecture, underlying techniques (e.g., PDFMiner, BERT, SVM, rule-based grammars), target metadata fields, and handling of document structure. This analysis is framed within the context of our own system, MetaScan, which integrates NLP enrichment and database indexing via a user-friendly interface. Our survey highlights the trade-offs between deep learning models, templatebased precision, and classical machine learning approaches in the complex domain of scientific document analysis.

***Index Terms***—Survey, Literature Review, Metadata Extraction, NLP, Layout Analysis, PDF Processing, Machine Learning, BERT, SVM, Information Retrieval, Scientific Documents

## I. INTRODUCTION

The volume of scientific research published daily continues to grow exponentially [4], creating an information overload that makes it increasingly difficult for researchers to stay current and efficiently locate relevant prior work. Traditional search methods, often limited to basic keyword matching over unstructured text, struggle with the diverse and often complex layouts found in standardized formats like the Portable Document Format (PDF) [1], [11]. Extracting key metadata— such as titles, authors, affiliations, abstracts, and keywords—is crucial for indexing, discovery, and interoperability within digital libraries and research platforms [9], [13], but remains a significant challenge due to inconsistent formatting across publishers and journals [1]. To address these limitations, numerous automated approaches leveraging Artificial Intelligence (AI), Natural Language Processing (NLP), and machine learning have been developed [4], [9]. These systems aim to parse the structure and content of scientific documents, identify specific metadata fields, and extract the relevant information in a structured format. Techniques range from rule-based systems and classical machine learning models like Support Vector Machines (SVMs) [9] to deep learning architectures, including transformer-based models like BERT [1], [4], and templatebased approaches tailored to specific publisher layouts [11]. This paper presents a survey of five distinct methodologies for automatic metadata extraction from scientific literature, primarily focusing on PDF documents. We analyze the LAyoutaware Metadata Extraction (LAME) framework [1], the Automated framework for Information Extraction (AutoIE) [4], an SVM-based classification approach [9], and the PDFDataExtractor tool [11], LayoutLMv3 [15], [16]. We compare their underlying technologies, approaches to layout analysis and text processing, target metadata, and overall

effectiveness. To ground this survey, we introduce our own system, MetaScan, which combines NLP enrichment with database indexing via a Streamlit interface, representing a user-centric approach to managing a curated document collection. The remainder of this paper is organized as follows. Section II details the architecture and goals of MetaScan. Section III outlines our survey methodology. Section IV provides a detailed review of the selected metadata extraction systems. Section V presents a comparative analysis, and Section VI discusses key findings and future research directions. Section VII concludes the paper.

## II. MOTIVATION: THE METASCAN SYSTEM

To ground our survey in a practical context, we first introduce MetaScan, an AI-powered research metadata indexing system we are developing. MetaScan is designed to provide researchers with an interactive dashboard for ingesting, analyzing, and searching scientific papers within a selfmanaged collection.
The system's architecture is built on a modular Python stack:

- **Streamlit Frontend:** An interactive web dashboard (dashboard.py) for document upload (JSON initially, PDF planned), search filtering, and analytics visualization.
- **MongoDB Backend:** A NoSQL database (db.py) storing flexible document metadata, suitable for potentially incomplete or varied information from research papers.
- **NLP Enrichment Pipeline:** A core module (enrich.py) using spaCy for text cleaning (lemmatization) and Named Entity Recognition (NER), and scikit-learn (TFIDF) for automatic keyword extraction relative to the corpus. It also includes rule-based categorization.
- **Data Ingestion & Extraction:** Supports ingestion from .json files (ingest.py) and includes a module (pdf _extractor.py) using PyMuPDF (fitz) and heuristic rules for extracting metadata (Title, Author, Abstract, Keywords) directly from PDF files.

The design of MetaScan involves several key challenges addressed by the systems in this survey: How to robustly handle diverse PDF layouts? Which NLP techniques offer the best trade-off between performance and complexity for metadata extraction? How to efficiently structure and index extracted data for effective search? These questions motivated our survey of the current state-of-the-art.

## III. SURVEY METHODOLOGY

To select systems for this survey, we reviewed the provided research papers focusing on automated metadata extraction from scientific documents [1], [4], [9], [11], [14]. Our selection criteria were:

1) The system or method must explicitly aim to extract bibliographic metadata (e.g., title, author, abstract) from research papers, particularly PDFs [1], [4], [9], [11].
2) The approach must involve AI, machine learning, NLP, or sophisticated rule-based/template techniques (going beyond simple regex) [1], [4], [9], [11].
3) The paper must provide sufficient detail on the methodology and technology used [1], [4], [9], [11].

Based on this, we selected the LAME framework [1], the AutoIE framework [4], the SVM-based method described by Han et al. [9], and the PDFDataExtractor tool [11] as representative of different significant approaches in the field.

## IV. REVIEW OF EXISTING SYSTEMS

This section details the four selected systems/methods for metadata extraction.

### *A. LAME: Layout-Aware BERT*

The LAyout-aware Metadata Extraction (LAME) framework proposed by Choi et al. [1] addresses the challenge of diverse PDF layouts by explicitly incorporating layout information into the extraction process [1]. The framework consists of three main stages [1]:

1) **Automatic Layout Analysis:** Uses the PDFMiner library [2] to extract text coordinates and font information from the first page of a PDF [1]. It then applies a series of reconstruction, refinement (using font info), and ordering steps to define logical text boxes corresponding to potential metadata fields [1].
2) **Training Data Construction:** Automatically generates a large dataset by matching the identified layout boxes with known metadata (obtained via DOI lookup if necessary) [1] using textual similarity measures (Levenshtein distance, BLEU score) [1]. This process created training data from 65,007 PDFs across 70 journals [1].
3) **Metadata Extractor:** Implements a novel pre-trained language model called **Layout-MetaBERT**, based on the BERT architecture [1], [3]. Crucially, during pretraining, layout structure is considered by treating each identified text box (layout) as a sequence separated by [SEP] tokens [1]. The model uses both Masked

Language Model (MLM) and Next Sentence Prediction (NSP) losses, where NSP predicts if two layout boxes are consecutive [1]. After pre-training, the model is finetuned for the downstream task of classifying each layout box into one of the target metadata categories (e.g., title ko, title en, author name ko, abstract_en) [1].
LAME demonstrated robust performance (Macro-F1 93.27%) on unseen journals [1], outperforming baseline models and showing that layout-awareness significantly aids extraction [1]. It specifically targets title, author, affiliation, abstract, and keywords in Korean and English [1].

### *B. AutoIE: Framework for Domain-Specific Extraction*

The AutoIE framework by Liu and Li [4] focuses on automating information extraction from scientific literature, with a specific application in the molecular sieve synthesis domain [4]. It aims to quickly locate and extract valuable information, acknowledging the length and complexity of scientific papers [4]. AutoIE integrates several components within three main units [4]:

1) **Layout and Location Unit:** This unit processes the input PDF to identify the document structure and locate relevant sections [4]. It uses [4]:

- *MFFAPD (Multi-Semantic Feature Fusion-based Approach for PDF Document Layout Analysis):* Leverages VTLayout [5] which fuses visual and text features to recognize coarse-grained blocks like titles, paragraphs, tables, etc. [4].
- *AFBRSC (Advanced Functional Block Recognition in Scientific Texts):* Uses HARGSD [6] to quickly locate specific sections relevant to the domain (e.g., method and experiment parts for molecular sieves) [4].

2) **Information Extraction Unit:** This unit takes the identified text sections and performs fine-grained information extraction [4]. It proposes a new model [4]:

- *SBERT (Span-BERT):* A joint entity and relation extraction model based on BERT [3] and span classification [4], [7]. It uses multi-dimensional features: span embeddings (from fine-tuned BERT), width embeddings (to filter long spans), CLS token (for sentence semantics), and Part-of-Speech (POS) embeddings (using NLTK [8]) [4]. It classifies spans into entity types and then classifies relationships between entity pairs [4]. Transfer learning is also employed [4].

3) **Display and Human Feedback Unit:** Extracted information (in JSON format [4]) is presented to domain experts for verification via a web interface [4]. This feedback is used to refine the dataset and model via an Online Learning Paradigm (OLPTM) [4].
AutoIE demonstrated high F1 scores on general datasets (CoNLL04, ADE) [4] and 78% average accuracy in the specific molecular sieve domain [4], significantly speeding up the extraction process compared to manual methods [4]. It targets both general metadata (title, author) and highly domainspecific fields (Alkali Source, Crystallization Conditions) [4].

### *C. SVM Method: Classification and Chunking*

Han et al. [9] proposed using Support Vector Machines (SVMs), a classification algorithm known for handling highdimensional data [9], for automatic metadata extraction from research paper headers [9]. They frame the problem in two main steps [9], leveraging the observation that most header lines belong to a single metadata class [9]:

1) **Line Classification:** Each line in the header is classified into one or more of 15 predefined metadata categories (Title, Author, Affiliation, Address, Note, Email, Date, Abstract, Introduction, Phone, Keyword, Web, Degree, Pubnum, Page) [9].

- *Feature Engineering:* A rich set of features is used, including both word-specific (e.g., capitalization, dictionary membership, presence in domain-specific lists like names, cities, affiliations) [9] and linespecific features (e.g., line length, line position, percentage of dictionary words, percentage of numbers, percentage of class-specific words) [9]. Word features are generated using rule-based clustering based on domain databases and orthographic properties [9]. Feature normalization (L-infinity norm) was found crucial for performance [9].
- *SVM Classifiers:* 15 binary SVM classifiers (using Gaussian kernels [9]) are trained in a ”one vs. all” approach [9] using SVM light [10].
- *Iterative Contextual Improvement:* An iterative procedure refines the classification by incorporating the predicted labels of neighboring lines (N=5) as additional binary features in subsequent rounds [9]. This converges within a few iterations and significantly improves performance for many classes [9].

2) **Chunk Identification:** For lines predicted as multi-class (a small percentage [9]), a subsequent step identifies the boundaries between different metadata chunks within the line [9].

- *Two-Class Chunking:* For the common two-class lines [9], the algorithm searches for the optimal boundary (punctuation or space) that maximizes the classification score difference between the resulting two chunks, using the line classifiers [9].
- *Author Name Recognition:* A specific SVM-based method is used to identify individual author names within multi-author lines (both punctuation- and space-separated) [9], by classifying potential name sequences generated based on valid name patterns [9].

This SVM-based approach achieved higher overall accuracy (92.9%) compared to a baseline HMM method (90.1%) on the same dataset [9], demonstrating the effectiveness of treating metadata extraction as a classification problem combined with feature engineering and contextual refinement [9].

### *D. PDFDataExtractor: Template-Based Quality Mining*

Zhu and Cole [11] developed PDFDataExtractor as a tool specifically designed to read PDF scientific articles and interpret their metadata [11], acting as a potential plug-in for the chemistry-aware NLP tool ChemDataExtractor [12]. It addresses the lack of semantic tags in PDFs [11], which hinders tools like ChemDataExtractor that perform better on structured formats like HTML/XML [11]. The core approach is templatebased, focusing on high precision (quality mining) for specific publisher layouts rather than generalized recall across all formats [11].

The workflow involves several stages [11]:

1) **Preprocessing:** Uses PDFMiner [2] to convert the PDF into text blocks with layout information (coordinates, font, etc.) [11]. PDFDataExtractor assigns additional features, including a 'universal sequence number' to track blocks across pages [11].

2) **Template Assignment:** Automatically selects a predefined extraction template based on the detected layout or publisher characteristics [11]. Templates contain rules and grammars specific to a layout [11].

3) **Metadata Extraction:** Applies template-defined rules to text blocks (primarily on the first page [11]) to extract key metadata like title [11], authors [11], abstract [11], keywords [11], DOI [11], journal information [11], etc. Specific heuristics are used for different fields (e.g., largest area for abstract if the keyword 'abstract' is missing [11], centered top block for title [11]).

4) **Section Detection:** Identifies section headings based on font size and formatting rules defined in the template [11]. It involves constructing lists of potential titles, their locations, and font sizes, cleaning these lists based on maximum font size [11], and then segmenting the document body accordingly [11]. It also identifies and filters noise like headers and page numbers [11].

5) **Reference Extraction:** Locates the reference section and attempts to parse individual reference entries, primarily by detecting sequence numbers (e.g., "[1]", "[2]") as anchors to segment the reference text block [11].

6) **Output:** Produces JSON and plain text outputs containing the extracted metadata and segmented body text [11], suitable for input into downstream tools like ChemDataExtractor [11].

PDFDataExtractor achieved high precision for core metadata fields like DOI (often 90%) across various publisher datasets (Elsevier, ACS, RSC, Wiley) [11], demonstrating the effectiveness of the template-based approach for targeted, highquality extraction, though performance varies more significantly for elements like journal name or references depending on layout consistency [11].

### E. LayoutLMv3: Unified Vision–Language Model for Document Understanding

LayoutLMv3 is a state-of-the-art, transformer-based document intelligence model designed to integrate text, image, and layout information for improved understanding of visually rich documents. Unlike traditional machine learning approaches such as Support Vector Machines (SVM), which rely heavily on handcrafted textual features for metadata extraction [9], LayoutLMv3 leverages deep multimodal pretraining to jointly learn from textual embeddings, spatial layout coordinates, and raw visual pixel information [15].

To enhance document comprehension, LayoutLMv3 incorporates both Masked Language Modeling (MLM) and Masked Image Modeling (MIM**)** during pretraining, enabling unified representation learning across modalities. This dualmasking strategy allows the model to effectively capture structural cues such as multi-column layouts, hierarchical headings, font variations, tables, figures, and mathematical elements within scholarly PDFs [15].

Compared to earlier versions like LayoutLM and LayoutLMv2—which combine layout and text features but rely on less sophisticated image modeling—LayoutLMv3 demonstrates stronger performance in document classification, key information extraction, and logical region segmentation tasks [16].

**Applications Relevant to MetaNest**

LayoutLMv3 is particularly effective for metadata extraction tasks such as:

- Title and author block identification
- Abstract boundary detection
- Logical segmentation of multi-column PDFs
- Figure and table caption recognition
- Extraction of structured elements across varied journal formats

**Relevance to MetaNest**

Integrating LayoutLMv3 into the MetaNest system would significantly enhance metadata extraction accuracy by leveraging the full layout and visual structure of research PDFs. Its inclusion in the survey also strengthens the methodological depth and demonstrates awareness of the latest advancements in document AI.

## V. COMPARATIVE ANALYSIS

Our comparative analysis, summarized in Table I, highlights the diverse strategies employed for automatic metadata extraction from scientific documents. These systems differ significantly in their core methodologies, handling of document layout, and the specific information they target.

LAME [1] and AutoIE [4] represent state-of-the-art deep learning approaches, leveraging BERT-based architectures. LAME's novelty lies in explicitly incorporating layout information (derived from PDFMiner [2]) into the pre-training of its Layout-MetaBERT model, demonstrating that structural awareness enhances robustness against varying formats [1]. AutoIE, while also using BERT [3] (in its SBERT variant [4]), focuses more on integrating layout analysis (VTLayout [5], HARGSD [6]) as a preliminary step to locate relevant sections for domain-specific entity and relation extraction, coupled with a human-in-the-loop verification process [4].

In contrast, the SVM Method [9] by Han et al. showcases a classical machine learning approach. It relies heavily on extensive feature engineering, capturing orthographic, lexical, and positional information [9], and employs SVMs to classify lines and subsequently identify chunks [9]. Its iterative contextual classification step acknowledges the sequential nature of document headers [9].

PDFDataExtractor [11] offers a pragmatic, templatebased solution. It prioritizes high precision by defining specific rules and grammars tailored to individual publisher layouts [11]. While less generalizable than machine learning models, this approach can achieve very high accuracy when a matching template exists [11], making it suitable for targeted database population efforts.

Our MetaScan system aims for a balance between generality and ease of implementation. It uses readily available NLP libraries (spaCy, scikit-learn) for enrichment and a heuristicbased approach (PyMuPDF with rules) for initial PDF metadata extraction. Compared to the reviewed systems, MetaScan's current PDF extraction is simpler than the deep learning models or the complex feature engineering of the SVM method, and less precise but more general than the template-based PDFDataExtractor. Its main focus is integrating these components with a database and user interface for managing a personal research collection.

The comparison highlights a fundamental trade-off: Deep learning models (LAME, AutoIE) offer powerful sequence modeling and potentially higher generalization but require large datasets and significant computational resources for training. Classical ML (SVM Method) relies on careful feature design and domain knowledge. Template-based systems

(PDFDataExtractor) offer precision but require ongoing effort to create and maintain templates for different layouts. MetaScan's approach prioritizes accessibility and integration for the enduser, using off-the-shelf NLP tools where possible.

## VI. TEXT SUMMARIZATION APPROACHES FOR SCIENTIFIC DOCUMENTS

Scientific papers are long and dense, and summarization helps researchers quickly understand important ideas. Modern NLP provides multiple approaches for summarizing research articles.Below we compare three models suitable for integration into MetaNest.

### 1. BERT Extractive Summarization

BERT-based extractive summarization utilizes the pretrained

Bidirectional Encoder Representations from Transformers (BERT) model to identify and select the most important sentences within a document [3]. Unlike generative summarization methods, BERT summarizers do not create new text; instead, they extract key sentences that best represent the original content.

Key Features:

1.Extractive summarization approach (selects original sentences rather than generating new ones).

2.Highly accurate for structured scientific text. 3.Computationally efficient compared to generative models.

Advantages

- Preserves the original meaning of the text.
- Well-suited for summarizing research article abstracts, introductions, and short sections.

Limitations

- Cannot paraphrase or generate new sentences.
- Limited flexibility in restructuring the summary content.
-

**2. T5 (Text-to-Text Transfer Transformer) Summarization**

The T5 model formulates all natural language processing tasks into a unified text-to-text framework, allowing summarization to be treated as a pure text generation problem. T5 generates abstractive summaries, providing concise rephrasings of long passages using natural-language constructs.

Key Features:

1. Performs well on research paper abstracts and complex scientific narratives.
2. Capable of rewriting long academic text into concise, coherent summaries.

Advantages

- Produces fluent and human-like summaries.
- Handles long contextual inputs effectively.

Limitations

- Higher computational requirements compared to extractive methods.
- Slower inference time on CPU-based systems.

**3. PEGASUS Summarization**

PEGASUS is among the most advanced models for scientific document summarization and excels in abstractive summarization tasks. Its pretraining strategy, known as *Gap Sentence Generation*, is designed to mimic real summarization objectives by masking whole sentences instead of single tokens.

Key Features:

1.Specialized pretraining for summary generation using the Gap Sentence Generation objective.
2.Highly effective for long research articles, review papers, and scientific domains.

Advantages

- Achieves state-of-the-art performance on multiple scientific summarization benchmarks.
- Produces high-quality, abstract-level summaries.

Limitations

- Large model size increases memory requirements.
- Requires GPU acceleration for efficient inference.

**Comparative Table**

| **Model** | **Type** | **Strength** | **Limitation** |
|---|---|---|---|
| **BERT** | Extractive | High accuracy, fast | No new sentences |
| **T5** | Abstractive | Natural rewritten summaries | Slow on CPU |
| **PEGASUS** | Abstractive | Best for scientific summarization | Heavy model |

Table I: Comparative Analysis of Text Summarization Models

## VI. DISCUSSION AND FUTURE WORK

Our survey reveals several key trends and challenges in automated metadata extraction. Firstly, layout analysis is critical [1], [4], [9], [11]. Systems like LAME [1], AutoIE [4], and the SVM method [9] (through positional features) explicitly or implicitly recognize that the visual arrangement of text with less data [9]. Template-based approaches like PDFDataExtractor prioritize precision over recall [11], suitable for controlled environments but brittle when encountering new layouts [11].

Thirdly, domain specificity is often necessary. AutoIE explicitly targets molecular sieve synthesis [4], while the SVM method uses domain-specific word lists [9].

| System | Primary Goal | Core Technology / Method | Key Techniques | Layout Handling |
|---|---|---|---|---|
| **MetaScan (Ours)** | Private indexing and enrichment of a user-curated corpus via dashboard. | Python,Streamlit,MongoDB, PyMuPDF | spaCy (NER, Lemma), TFIDF (Keywords), Rulebased Categorization, Heuristic PDF parsing. | Rule-based extraction from PDF text coordinates/content. |
| **LAME [1]** | High-performance metadata extraction robust to diverse layouts [1]. | Custom Pre-trained BERT (Layout-MetaBERT) [1]. | PDFMiner [1], [2], TextualSimilarity (BLEU,Levenshtein) [1], BERT (MLM, NSP on layouts)[1], [3]. | Explicit layout analysis (TextBox reconstruction); Layout info used in BERT pre-training [1]. |
| **AutoIE [4]** | Automated,domainspecific information Extraction (entities & relations) with human feedback loop [4]. | Framework integrating multiple models; Custom BERT-based model (SBERT) [4]. | VTLayout [4], [5], HARGSD [4], [6], SBERT(Span classification, Relation classification) [4], [7], NLTK (POS) [4], [8], Online Learning [4]. | Coarse-grainedblock recognition (VTLayout) and targeted section location (HARGSD) [4]. |
| **SVM Method [9]** | Metadata extraction from headers using classification [9]. | Support Vector Machines (SVM) [9]. | Feature Engineering (Word/Line specific, Orthography, Domain Lists)[9], Normalization [9], Iterative Contextual Classification [9], Chunking [9], Author Name Rec. [9]. | Implicit via line position features; Relies heavily on text/orthographic features within lines [9]. |
| **PDFDataExtractor [11]** | High-precision metadata and structure extraction using publisher-specific templates [11]. | Template-based Rules and Grammars [11]. | PDFMiner [2], [11], Publisher-specific Layout Rules [11], Heuristics | Explicitly template-driven; Relies on consistency |

| | | | | |
|---|---|---|---|---|
| | | | (e.g., area, position) [11], Section Detection [11], Reference parsing via sequence numbers [11]. | within a publisher's layout style [11]. |
| **LayoutLMv3 [15],[16]** | Multimodal document understanding with unified text–image layout modeling. | Transformer-based Vision Language Model (LayoutLMv3) [15]. | Masked Language Modeling (MLM) [15], Masked Image Modeling (MIM) [15], Unified Multimodal Pretraining [15], Advanced Layout Embeddings [16], Document Structure Understanding (titles, tables, captions) [15]. | Explicit multimodal layout modeling using visual pixel features + text embeddings + spatial coordinates; Robust for multi-column and complex scientific PDF layouts [15], [16]. |

Table II: Comparative Analysis of Metadata Extraction systems

This suggests that PDFs conveys significant semantic information that pure text processing misses. PDFMiner [2] is a common tool for obtaining initial layout information [1], [11], but often requires refinement [1].
Secondly, the choice of extraction technique involves tradeoffs. Deep learning models like Layout-MetaBERT [1] and SBERT [4] achieve high performance but require substantial training data, which LAME addresses through automatic dataset construction [1] and AutoIE through an online learning loop [4]. SVMs require careful feature design [9] but can be effective with less data [9]. Template-based approaches like PDFDataExtractor prioritize precision over recall [11], suitable for controlled environments but brittle when encountering new layouts [11].
Thirdly, domain specificity is often necessary. AutoIE explicitly targets molecular sieve synthesis [4], while the SVM method uses domain-specific word lists [9]. This suggests that highly accurate extraction often requires tailoring to specific fields or document types.
A significant gap remains in bridging the accuracy of specialized models with the ease of use and general applicability needed for tools like MetaScan. Future work for MetaScan should focus on improving the robustness of its pdf_extractor.py module, perhaps by incorporating elements of the layout analysis techniques seen in LAME [1] or by adopting a more sophisticated classification model like the SVM approach [9] for identifying header fields. Integrating findings from effectiveness studies [13] (e.g., prioritizing accuracy for Title, Description, Subject [13]) could also guide development.
Furthermore, exploring techniques to handle multi-column layouts and more complex structures seen in scientific papers is essential. As suggested in MetaScan's roadmap, moving towards ML-based categorization and potentially integrating LLMs for more semantic understanding are promising directions.

## VII. CONCLUSION

This paper surveyed four distinct approaches to automatic metadata extraction from scientific literature: LAME's layoutaware BERT model [1], AutoIE's domain-specific framework with SBERT [4], Han et al.'s SVM classification method [9], and PDFDataExtractor's template-based system [11]. We compared these techniques, highlighting their strengths in handling layout diversity, leveraging deep learning, applying classical machine learning with feature engineering, and achieving high precision through templates. Framed by our MetaScan project, the analysis reveals a landscape of tools trading off between generalization, precision, complexity, and data requirements. Robust layout analysis and adaptable extraction techniques remain key challenges. Future work should focus on integrating the strengths of these diverse approaches to

create tools that are both powerful and accessible for researchers managing the ever-growing volume of scientific information.

## REFERENCES


[1] J. Choi, H. Kong, H. Yoon, H. Oh, and Y. Jung, ”LAME: Layout-Aware Metadata Extraction Approach for Research Articles,” *Computers, Materials & Continua*, vol. 72, no. 2, pp. 4019–4037, 2022.

[2] ”PDFMiner.six,” [Online]. Available: https://github.com/pdfminer/ pdfminer.six.

[3] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, ”BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, USA, vol. 1, pp. 4171–4186, 2019.

[4] Y. Liu and S. Li, ”AutoIE: An Automated Framework for Information Extraction from Scientific Literature,” *arXiv preprint arXiv:2401.16672*, 2024.

[5] S. Li, et al., ”Vtlayout: Fusion of visual and text features for document layout analysis,” in *PRICAI 2021: Trends in Artificial Intelligence*, Part I, Springer, 2021.

[6] S. Li and Q. Wang, ”A hybrid approach to recognize generic sections in scholarly documents,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 24, no. 4, pp. 339–348, 2021.

[7] M. Eberts and A. Ulges, “Span-based joint entity and relation extraction with transformer pre-training,” *arXiv preprint arXiv:1909.07755*, 2019.

[8] E. Loper and S. Bird,”Nltk: The natural language toolkit,” *arXiv preprint cs/0205028*, 2002.

[9] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox, ”Automatic document metadata extraction using support vector machines,” in *Proc. 2003 Joint Conference on Digital Libraries (JCDL’03)*, Houston, TX, USA, pp. 37–48, 2003.

[10] T. Joachims, ”Making large-scale Support Vector Machine learning practical,” in *Advances in Kernel Methods: Support Vector Machines*, MIT Press, 1998.

[11] M. Zhu and J. M. Cole, ”PDFDataExtractor: A Tool for Reading Scientific Text and Interpreting Metadata from the Typeset Literature in the Portable Document Format,” *Journal of Chemical Information and Modeling*, vol. 62, pp. 1633–1643, 2022.

[12] M. C. Swain and J. M. Cole, ”ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature,” *J. Chem. Inf. Model.*, vol. 56, pp. 1894–1904, 2016.

[13] L. Yang, ”Metadata Effectiveness in Internet Discovery: An Analysis of Digital Collection Metadata Elements and Internet Search Engine Keywords,” *College & Research Libraries*, vol. 77, no. 1, pp. 7–19, 2016.

[14] N. F. Mosha and P. Ngulube, ”Metadata Standard for Continuous Preservation, Discovery, and Reuse of Research Data in Repositories by Higher Education Institutions: A Systematic Review,” *Information*, vol. 14, no. 8, p. 427, 2023.

[15] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei, “LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking,” arXiv preprint arXiv:2204.08317, 2022.

[16] Xu, Y., Li, M., Cui, L., Wei, F., & Zhou, M. “LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding,” ACL 2021.