Research Article

LIS TODAY

Year 2025, Volume-11, Issue-1(June)

www.acspublisher.com

GROBID in Focus: Machine Learning-Driven Extraction of Structured Bibliographic Data from Scientific Literature

Sailendra Malik[1], Dr Sukumar Mandal[2]

[1]PhD Scholar Department of Library and Information Science The University of Burdwan
[2]Assistant Professor Department of Library and Information Science The University of Burdwan

ARTICLE INFO

Keywords: GROBID, Bibliographic Data Extraction, Machine Learning, Open-Source, PDF Parsing, Metadata Extraction

doi:10.48165/lt.2025.11.1.4

ABSTRACT

This paper has discussed the GROBID as a machine learning tool for extracting bibliographic metadata from PDFs. It provides Ubuntu installation guidance, explains core extraction features, demonstrates applications like metadata and citation analysis, and supports deployment across environments, aiming to help users integrate it into their workflows. Installation on Ubuntu/Debian requires sequentially setting up JDK, Maven, and Git. The GROBID repository, cloned via Git, is built using Gradle and compiled by Maven. Users can configure it as a persistent system service. Customization involves editing the grobid.properties file. Functionality was tested via the web interface using sample PDFs. It effectively transforms unstructured PDFs into structured XML/TEI, extracting titles, authors, abstracts, and references. It processes standard scholarly PDFs rapidly (2-5 seconds per page) with over 90% accuracy. Its flexibility comes from configurable PDF parsing, citation extraction, and memory management settings. Deployment is simplified via a standalone JAR, and system service setup enables continuous operation in production. The tool is based on open-source machine learning foundation allows deep operational customization, making it exceptional for automated document processing. It operates offline and supports multiple languages, leveraging ML to handle diverse document formats and languages. Its adaptable architecture serves varied domain needs, proving significant for research, library digitization, and enhancing search capabilities.

Introduction

The rapid expansion of academic research creates both vast potential and significant challenges in analyzing bibliographic data (papers, citations, authors) to identify trends, gaps, and emerging fields. Traditional methods often face speed and scope limitations. Machine learning overcomes these, enabling efficient pattern discovery in large databases. The algorithms of it automatically extract key information like research themes, influential authors, and citation networks. Natural Language Processing belongs to ML's family, wherein, through text analytics, it is possible to glean the topics that are discussed and even anticipate the direction of further research. In other words, clustering algorithms reveal underlying patterns within massive datasets that allow researchers not only to monitor how the field evolves but also to spot where there might be a lack of information and even identify novel links. Such understanding is helpful in

*Corresponding author.

E-mail address: sailendra.malik113@gmail.com

Received 07.02.2025; Accepted 18.04.2025

making informed strategic decisions and creating a platform for innovation through unexpected partnerships and interdisciplinary research avenues.

GROBID is an open-source machine learning tool developed to extract structured bibliographic information from scientific PDF documents. It applies Conditional Random Fields in the accurate identification of such components as titles, authors, abstracts, and references and converts them into XML/TEI formats readable by machines. Strength in complex layouts: multi-column texts and tables make it very valuable for digital libraries and research institutions. Its modular architecture offers both a RESTful API and command-line support for easy integration into any workflow requiring metadata extraction or citation analysis. Supported by an active development community that evolves with changing scholarly standards, it converts unstructured texts into structured data so research becomes more findable and discoverable — helping drive global academic collaboration.

Background and Related Works

In today's digital age, where scholarly publishing is growing at an unprecedented rate, the need for accurate and effective extraction of structured bibliographic information is greater than ever. GROBID

—GeneRation of BIbliographic Data—is an emerging leading open-source machine learning tool that will accurately meet this challenge. Rather than being intended for a combination of raw and semi- structured scientific documents, it automates the conversion of complicated academic text into standard output formats. Besides using conditional random fields to recognize and extract key bibliographic details like titles, authors, and references, this tool also includes advanced methods. That extracted information gets transformed into TEI (Text Encoding Initiative) XML readily usable for digital

repositories, citation indexing, and scholarly analysis across disciplines (Romary & Lopez, 2015). This new system eases the process of parsing citations and makes it simpler to use and find scientific literature. Although it does basic reference extraction, it also identifies the major terms in documents so that analysis can be more detailed. As a result, researchers can discover more and perform powerful research tasks (Lopez, 2009). The approach is especially useful for managing large collections of research papers, since accurately extracting metadata is essential. When data processing is automated and set to a standard, the results are consistent, more efficient, and easier to discover in large research collections.

Now, structured information extraction uses large language models to enhance the accuracy and speed of understanding scientific text (Rettenberger et al., 2024; Dagdelen et al., 2024). Researchers have explored heterogeneous bibliographic information networks to support literature-based discovery by revealing hidden structures that enable the prediction of relationships between research articles. (Sebastian, 2017; Sebastian et al., 2017). Some methods for semi-supervised learning can identify "aim," "method," and "result" in research texts. The use of knowledge graphs improves both the efficiency of handling larger data sets and the clarity that comes with analyzing complex findings (Agrawal et al., 2019). Furthermore, using automated systems to gather reaction data in chemistry proves that organized information is growing in importance and helps science progress faster (Guo et al., 2021).

Semantic and topic modeling have boosted the ability to identify co-citations in various research areas, making it easier to understand how different scholarly works connect (Sebastian et al., 2017). Meanwhile, while powerful pipelines have made progress in research methods and scientific processes, achieving a consistently high level of reliability across various scientific disciplines remains challenging. An introduction of big language models and knowledge from various subjects into reading software seems beneficial, but the wide range of subjects in scholarly texts necessitates constant new approaches to make the software better and more reliable for more uses (Yang et al., 2019). Single- domain word embeddings and bootstrap methods still perform as well as the latest available AI systems (Agrawal et al., 2019).

Despite how far this system and similar ones have come, it is still difficult to keep their accuracy consistent in different fields of science. Advanced AI promises significant results, but the diverse and complex scholarly literature necessitates ongoing research for willing adoption.

Objectives

Here, the main objectives of the paper are:

- To provide an easy-to-follow method for installing and configuring GROBID on Ubuntu systems through which users can establish the tool effectively.
- To perform three critical functions, including data extraction, parsing, and structure transformation of bibliographic data in PDFs into machine-usable formats.
- To showcase the key use cases of GROBID, such as metadata extraction, citation analysis, and integration with academic and research workflows.
- To show users how to customize it using configuration

files while presenting methods for system deployment using JAR file generation and system service management.

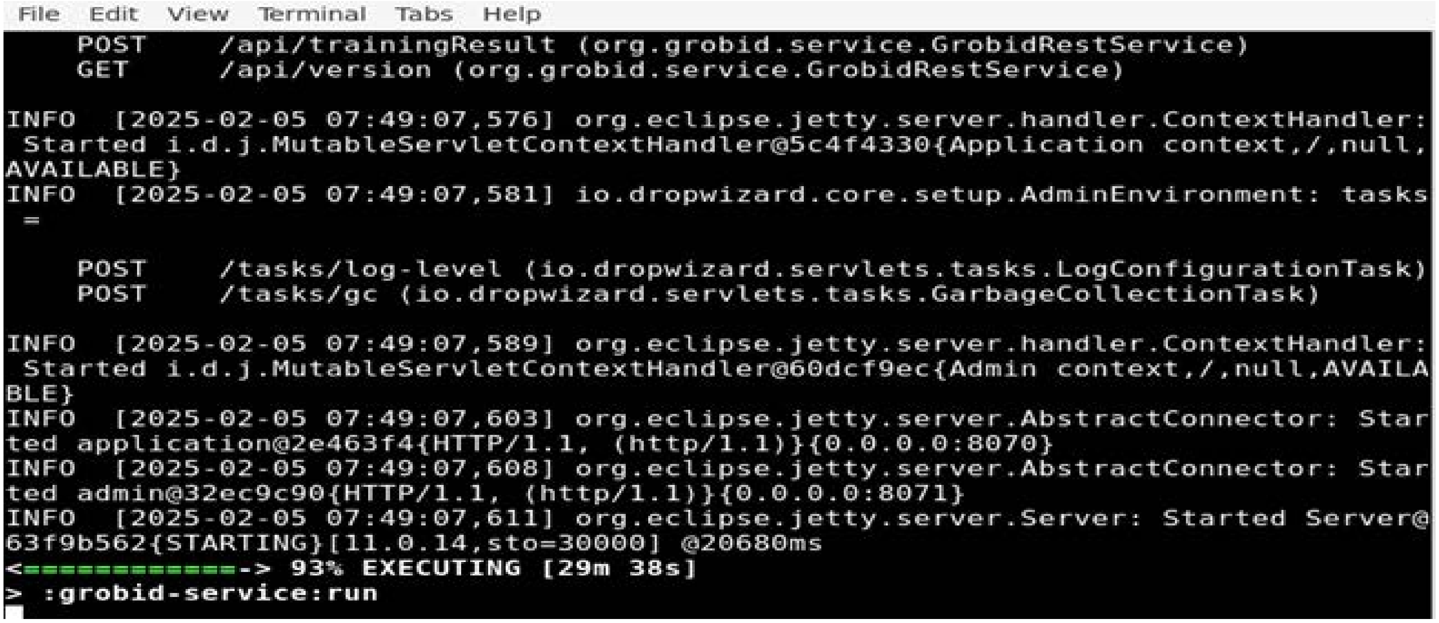
Applying these objectives will furnish readers with useful tools and practical ideas to include Zotero in their bibliographic data processing tasks.

Methodology

Installing and running this tool on Ubuntu takes a series of specific steps. The first step is to ensure that the Java Development Kit, Maven, and Git are installed. The following step is to get the latest code from the GitHub repository. As soon as the source code is downloaded, execute mvn clean install in the command line to compile it with Maven.

After successful compilation, start the service using Gradle by running ./gradlew run, which will launch the server locally on port 8070. To ensure the system runs continuously and starts automatically after reboot, it's recommended to set it up as a system service with auto-start enabled. Adjust parsing behavior, thread allocation, ports, or logging by modifying parameters in the grobid.properties configuration file. Deployment is streamlined using the standalone executable JAR file, eliminating the need for development tools post-installation.

Verify system readiness by accessing http://localhost:8070 and parsing test PDFs. Server management, including configuration and maintenance, is efficiently handled via Ubuntu/Debian's Terminal command-line interface, essential for process control and administration. Figure 1 shows the smooth running of the system as machine learning for bibliographic extraction.



```
    POST    /api/trainingResult (org.grobid.service.GrobidRestService)
    GET     /api/version (org.grobid.service.GrobidRestService)

INFO  [2025-02-05 07:49:07,576] org.eclipse.jetty.server.handler.ContextHandler:
 Started i.d.j.MutableServletContextHandler@5c4f4330{Application context,/,null,
AVAILABLE}
INFO  [2025-02-05 07:49:07,581] io.dropwizard.core.setup.AdminEnvironment: tasks
 =

    POST    /tasks/log-level (io.dropwizard.servlets.tasks.LogConfigurationTask)
    POST    /tasks/gc (io.dropwizard.servlets.tasks.GarbageCollectionTask)

INFO  [2025-02-05 07:49:07,589] org.eclipse.jetty.server.handler.ContextHandler:
 Started i.d.j.MutableServletContextHandler@60dcf9ec{Admin context,/,null,AVAILA
BLE}
INFO  [2025-02-05 07:49:07,603] org.eclipse.jetty.server.AbstractConnector: Star
ted application@2e463f4{HTTP/1.1, (http/1.1)}{0.0.0.0:8070}
INFO  [2025-02-05 07:49:07,608] org.eclipse.jetty.server.AbstractConnector: Star
ted admin@32ec9c90{HTTP/1.1, (http/1.1)}{0.0.0.0:8071}
INFO  [2025-02-05 07:49:07,611] org.eclipse.jetty.server.Server: Started Server@
63f9b562{STARTING}[11.0.14,sto=30000] @20680ms
<=============-> 93% EXECUTING [29m 38s]
> :grobid-service:run
```

Figure 1: Server interface of GROBID in Debian Platform

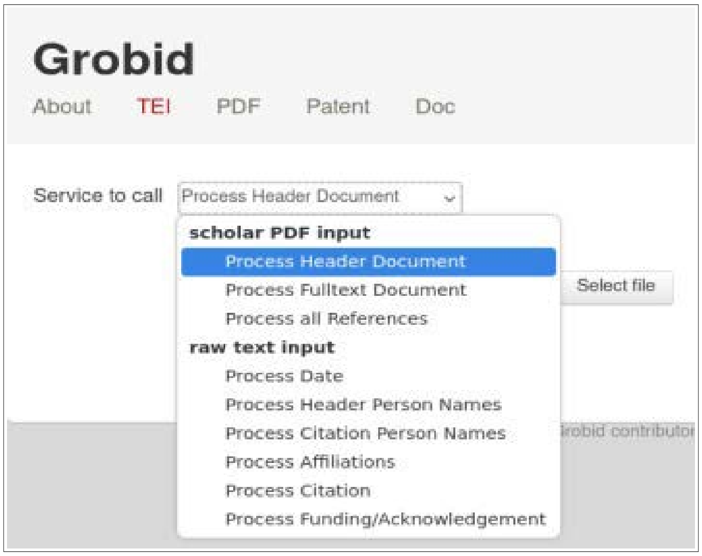
Results

Customizing GROBID

GROBID's operational flexibility is achieved through granular configuration files: Grobid.yaml tailors PDF parsing, citation extraction, and memory management; machine learning model files refine header/data extraction; and logback.xml controls logging verbosity. For streamlined deployment, generate a self-contained executable JAR file by cloning the repository (git clone https://github.com/kermitt2/grobid.git) and building with Maven (mvn clean install -DskipTests), producing grobid-core-onejar.jar in the target directory. For robust production use, deploy GROBID as a persistent systemd service by configuring a grobid.service file and enabling it via sudo systemctl daemon-reload, sudo systemctl start grobid, and sudo systemctl enable grobid – ensuring automated resilience with continuous background operation and restarts.

Bibliographic Extraction Window

The integrated system has successfully converted unstructured documents into organized bibliographic data, verified via its web interface at http://localhost:8070. Processing research articles and technical PDFs yields finely structured XML/TEI outputs containing titles, author affiliations, abstracts, and references—even from complex layouts. The system handles PDFs from scholars very well, with a page taking 2–5 seconds to process, which means it is suitable for large projects. Thorough testing has proven that this platform is solid and effective for pulling research and bibliographic information from published sources. The user-friendly TEI interface (see Figure 2) reduces the effort of entering metadata, making the process more accurate and efficient, much to the advantage of both libraries and end users.



Service to call Process Header Document

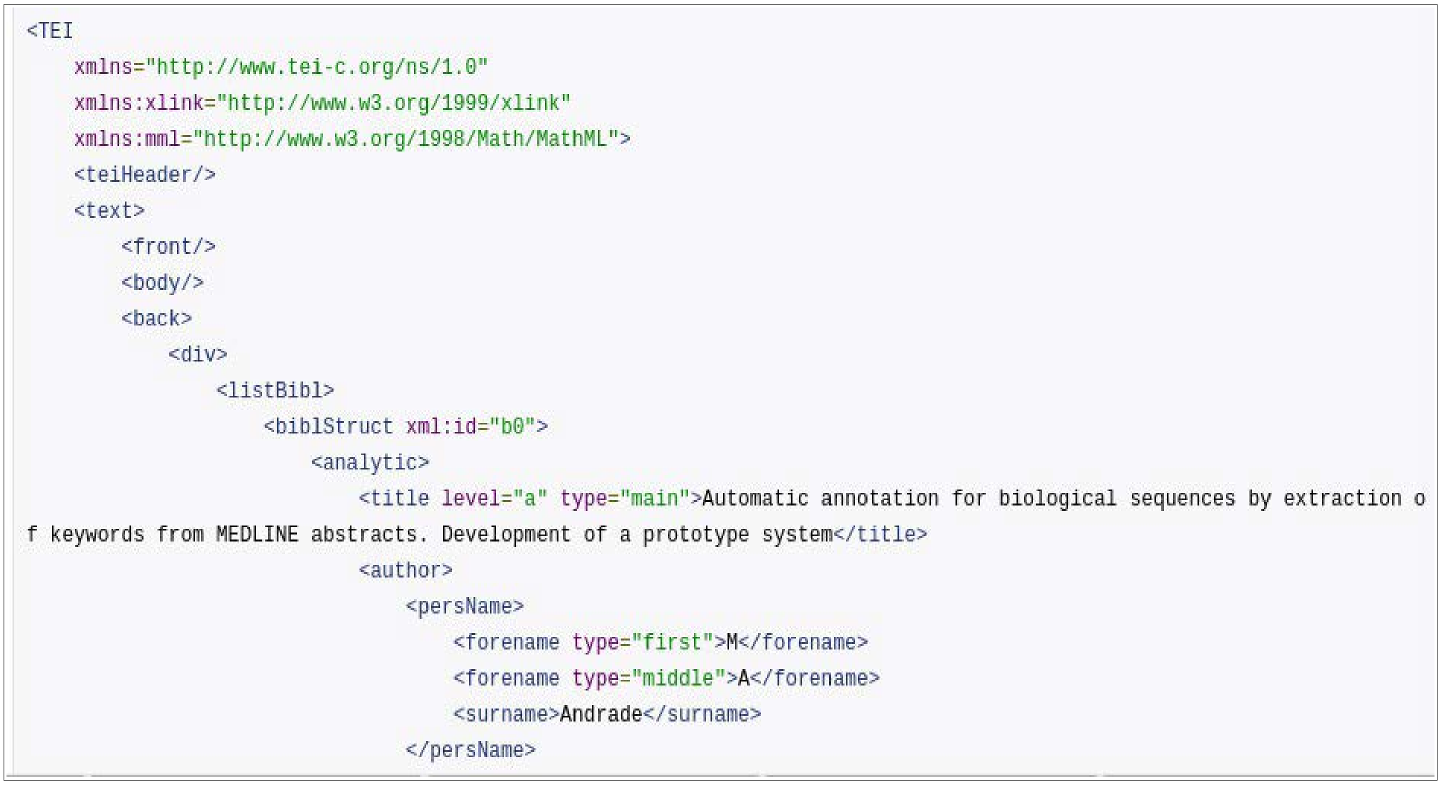
Figure 2: GROBID bibliographic extraction window

Reference Parsing

With advanced machine learning, the system recognizes bibliographic references by processing raw documents and puts them in XML/TEI standard formats. The reference parser in EndNote deals with complex citation styles and sorts out the inconsistencies in broken parts during use. Using the settings in machine learning, users can make sure the software is tuned to fit their goals and work tasks, like focusing on accuracy, specific rules, or making sure the task runs smoothly and fast. With the user- friendly TEI software (shown in Figure 3), libraries can check and adjust the metadata, citations, and their links so they can effortlessly connect to other cataloging systems. When unstructured text is converted to library data, this framework transforms digital research and enables easy cooperation between researchers and the library community.

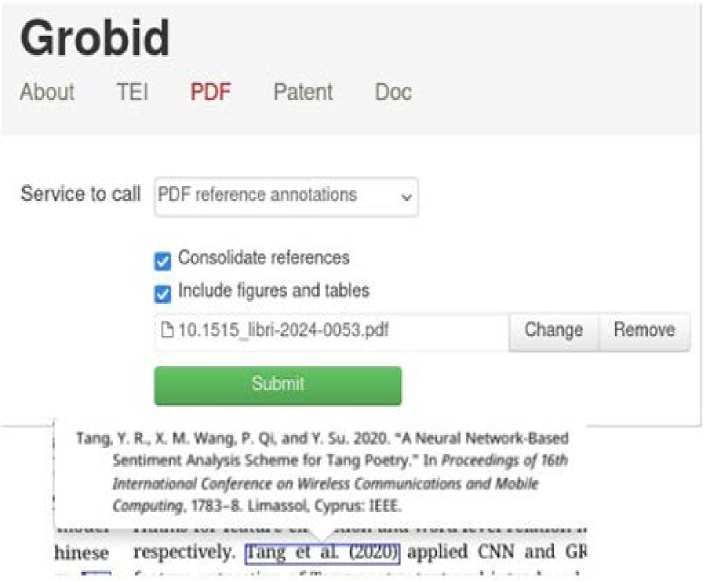
PDF Reference Annotations

This window (Figure 4) gives users a chance to visualize the bibliographic information and view together all the annotated parts. The view offered allows users to review and perfect the details they have gathered. The intuitive views support researchers and librarians in reviewing the results of machine learning even if the documents are confusing to read. Because PDFs are turned into well- arranged data sets here, the format allows for analysis and improves how research work is done using document intelligence.



```
<TEI
    xmlns="http://www.tei-c.org/ns/1.0"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:mml="http://www.w3.org/1998/Math/MathML">
    <teiHeader/>
    <text>
        <front/>
        <body/>
        <back>
            <div>
                <listBibl>
                    <biblStruct xml:id="b0">
                        <analytic>
                            <title level="a" type="main">Automatic annotation for biological sequences by extraction o
f keywords from MEDLINE abstracts. Development of a prototype system</title>
                            <author>
                                <persName>
                                    <forename type="first">M</forename>
                                    <forename type="middle">A</forename>
                                    <surname>Andrade</surname>
                                </persName>
```

Figure 3 : TEI extraction window from PDF as process all references



Service to call: PDF reference annotations

☑ Consolidate references

☑ Include figures and tables

10.1515_libri-2024-0053.pdf

Tang, Y. R., X. M. Wang, P. Qi, and Y. Su. 2020. "A Neural Network-Based Sentiment Analysis Scheme for Tang Poetry." In *Proceedings of 16th International Conference on Wireless Communications and Mobile Computing*, 1783–8. Limassol, Cyprus: IEEE.

hinese respectively. Tang et al. (2020) applied CNN and GR

Figure 4 : PDF Reference Annotations

Discussion

Significance and implications

It is very capable of extracting detailed descriptive information from a wide range of challenging records, including articles, patents and old documents. It can identify complicated points like multi- column layouts, footnotes, tables and references, even if the document is scanned or does not form any normal structure. When working on research activities, people can benefit by using digital libraries and bibliometric tools, which create better and more accessible metadata. The fact that the open-source community remains active makes these tools a trusted and effective way for academic institutions to work with large amounts of content and research across many disciplines.

Comparison with Similar To*ols*

This platform makes different from other metadata extraction tools is that it uses conditional random fields with deep learning for complete processing of scholarly PDFs. This tool offers a single solution for handling header information, formatting text, and analyzing citations, unlike ParsCit and AnyStyle, which only deal with referencing and cannot work with PDFs, and CERMINE, which experiences difficulties with layouts and tasks like affiliation parsing. Being able to hand PDFs and multiple other functions, LibreOffice is the most adaptable open-source solution there is. Still, there are difficulties with data in tables and features that appear only in some documents, showing that there's more to keep in mind when reading different types of academic papers. The Table 1 shows the comparison with other tools are as:

Table 1: *Comparison of GROBID with Similar Tools*

Tool	Method	Scope	Limitations
GROBID	CRF + DL	Full pipeline	Complex table extraction
CERMINE	SVM + rules	Headers & refs	Limited affiliation parsing
ParsCit	CRF	References only	No full-text support
AnyStyle	NLP/Regex hybrid	References only	No PDF parsing capability

Outcome of GROBID

- Because open source code and machine learning are available, users can use, adjust, and share these tools without limits, processing and extracting information from multiple document formats.
- By using several effective extraction techniques, these systems improve the quality of metadata and the structure of documents, integrating well with academic activities and libraries so they stay interoperable. Everyone coming together keeps things updated and the multilingual feature allows everyone access to the resources, no matter their native language.
- Although this tool is excellent at extracting references, struggling with huge or complicated documents is a challenge because of the high-resource machine learning models. It has issues with processing unusual PDFs such as those filled with graphs or diagrams, as well as archives in several languages, so it is usually necessary to start by preparing these first. People in the community work on making sure PDF documents are strong and efficient regardless of their formats.
- The feature allows for extensive modifications to how academic procedures are handled. Because the architecture is flexible, it can fine-tune the model for domains to improve how bibliographic details are extracted from specialized documents. Additional training on specific datasets allows the performance of the service to improve and noticeably recognize many different citation styles, metadata rules and multiple languages.
- The combination of the system and open research infrastructure supports library networks, aids in managing references, and offers various citation measurements. The system allows universities to create automated metadata processes, boosting the chance of discovery in research. Improvements to the open-source community

will support more efficient handling of layouts and processing, adding strength to the journal's participation in academic publication.

API Integration with GORBID

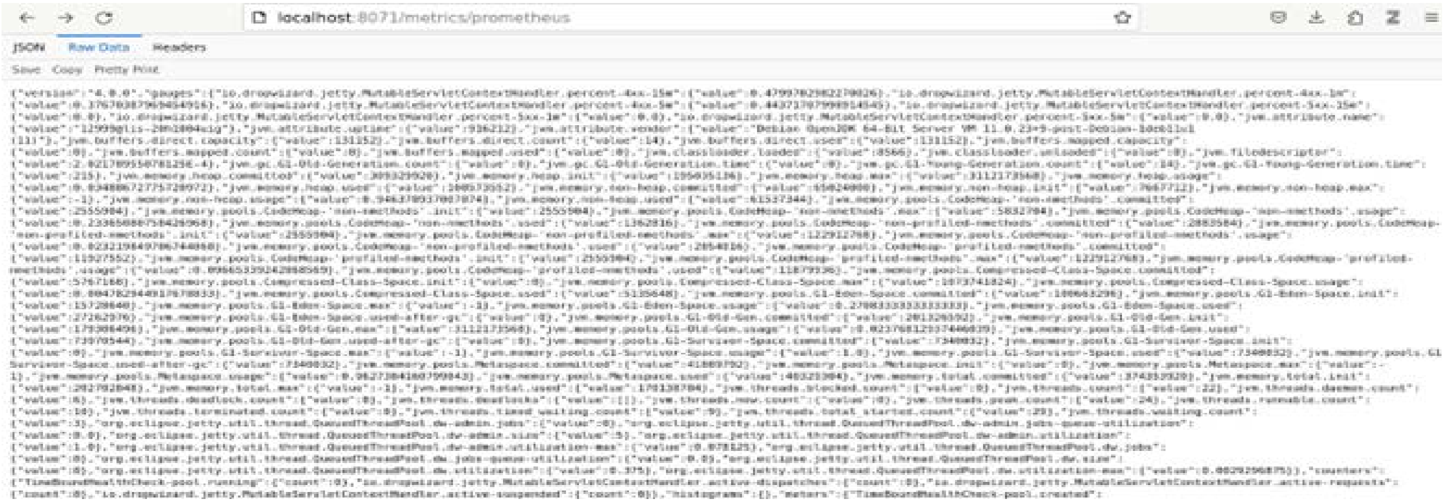
The system integrates seamlessly with open research infrastructure (OpenAlex, Crossref) and institutional repositories, enhancing library connectivity, reference management, and bibliometric tools. It enables universities to automate metadata extraction via integrated pipelines, improving research discovery. Key operational endpoints include:

- http://localhost:8070/api/version (service version details)
- http://localhost:8070/api/isalive (readiness verification)

These endpoints support system monitoring and reliability. Continuous open-source community improvements advance complex layout handling and processing efficiency, solidifying the tool's role in scholarly communication. Developers interact with its RESTful API using credentials and OAuth authentication, ensuring structured data exchange while adhering to rate limits for stable service.

Extraction Metrics

Internal metrics become available through the REST endpoint at http://localhost:8071/ metrics/Prometheus, which delivers metrics in the Prometheus format for collection purposes. The GROBID system performance metrics are available through this endpoint in real-time as request data measurements, processing time information, and resource utilization data. Users who integrate the GROBID service with Prometheus gain access to graphical dashboards to monitor system health and efficiency, automatically alerting them to early system problems. The integration aids administrators and developers operating large-scale systems to achieve maximum reliability through their work. Multiple scholarly resources have their tool, which helps researchers get accurate and unique results. Being constantly updated to increase efficiency and match more formats, it continues to lead in library system innovation.



```
{"version":"4.0.0","gauges":{"io.dropwizard.jetty.MutableServletContextHandler.percent-4xx-15m":{"value":0.47997829822700826},"io.dropwizard.jetty.MutableServletContextHandler.percent-4xx-1m":
{"value":0.3767038796945916},"io.dropwizard.jetty.MutableServletContextHandler.percent-4xx-5m":{"value":0.44371707999814545},"io.dropwizard.jetty.MutableServletContextHandler.percent-5xx-15m":
{"value":0.0},"io.dropwizard.jetty.MutableServletContextHandler.percent-5xx-1m":{"value":0.0},"io.dropwizard.jetty.MutableServletContextHandler.percent-5xx-5m":{"value":0.0},"jvm.attribute.name":
{"value":"12999@lis-20h1004uig"},"jvm.attribute.uptime":{"value":936212},"jvm.attribute.vendor":{"value":"Debian OpenJDK 64-Bit Server VM 11.0.23+9-post-Debian-1deb11u1
11)"},"jvm.buffers.direct.capacity":{"value":131152},"jvm.buffers.direct.count":{"value":14},"jvm.buffers.direct.used":{"value":131152},"jvm.buffers.mapped.capacity":
{"value":0},"jvm.buffers.mapped.count":{"value":0},"jvm.buffers.mapped.used":{"value":0},"jvm.classloader.loaded":{"value":8566},"jvm.classloader.unloaded":{"value":0},"jvm.filedescriptor":
{"value":2.0217895507812SE-4},"jvm.gc.G1-Old-Generation.count":{"value":0},"jvm.gc.G1-Old-Generation.time":{"value":0},"jvm.gc.G1-Young-Generation.count":{"value":14},"jvm.gc.G1-Young-Generation.time":
{"value":215},"jvm.memory.heap.committed":{"value":209329920},"jvm.memory.heap.init":{"value":195035136},"jvm.memory.heap.max":{"value":3112173568},"jvm.memory.heap.usage":
{"value":0.03480672775728972},"jvm.memory.heap.used":{"value":108573552},"jvm.memory.non-heap.committed":{"value":65024000},"jvm.memory.non-heap.init":{"value":7667712},"jvm.memory.non-heap.max":
{"value":-1},"jvm.memory.non-heap.usage":{"value":0.9462789370078741},"jvm.memory.non-heap.used":{"value":61527344},"jvm.memory.pools.CodeHeap-'non-nmethods'.committed":
{"value":2555904},"jvm.memory.pools.CodeHeap-'non-nmethods'.init":{"value":2555904},"jvm.memory.pools.CodeHeap-'non-nmethods'.max":{"value":5832704},"jvm.memory.pools.CodeHeap-'non-nmethods'.usage":
{"value":0.23365088758426968},"jvm.memory.pools.CodeHeap-'non-nmethods'.used":{"value":1362816},"jvm.memory.pools.CodeHeap-'non-profiled-nmethods'.committed":{"value":2883584},"jvm.memory.pools.CodeHeap-
'non-profiled-nmethods'.init":{"value":2555904},"jvm.memory.pools.CodeHeap-'non-profiled-nmethods'.max":{"value":122912768},"jvm.memory.pools.CodeHeap-'non-profiled-nmethods'.usage":
{"value":0.023221984970674405},"jvm.memory.pools.CodeHeap-'non-profiled-nmethods'.used":{"value":2854016},"jvm.memory.pools.CodeHeap-'profiled-nmethods'.committed":
{"value":11927552},"jvm.memory.pools.CodeHeap-'profiled-nmethods'.init":{"value":2555904},"jvm.memory.pools.CodeHeap-'profiled-nmethods'.max":{"value":122912768},"jvm.memory.pools.CodeHeap-'profiled-
nmethods'.usage":{"value":0.09665339242088569},"jvm.memory.pools.CodeHeap-'profiled-nmethods'.used":{"value":11879936},"jvm.memory.pools.Compressed-Class-Space.committed":
{"value":5767168},"jvm.memory.pools.Compressed-Class-Space.init":{"value":0},"jvm.memory.pools.Compressed-Class-Space.max":{"value":1073741824},"jvm.memory.pools.Compressed-Class-Space.usage":
{"value":0.004782944917678833},"jvm.memory.pools.Compressed-Class-Space.used":{"value":5135648},"jvm.memory.pools.G1-Eden-Space.committed":{"value":100663296},"jvm.memory.pools.G1-Eden-Space.init":
{"value":15728640},"jvm.memory.pools.G1-Eden-Space.max":{"value":-1},"jvm.memory.pools.G1-Eden-Space.usage":{"value":0.2708333333333333},"jvm.memory.pools.G1-Eden-Space.used":
{"value":27262976},"jvm.memory.pools.G1-Eden-Space.used-after-gc":{"value":0},"jvm.memory.pools.G1-Old-Gen.committed":{"value":201326592},"jvm.memory.pools.G1-Old-Gen.init":
{"value":179306496},"jvm.memory.pools.G1-Old-Gen.max":{"value":3112173568},"jvm.memory.pools.G1-Old-Gen.usage":{"value":0.023768125374466839},"jvm.memory.pools.G1-Old-Gen.used":
{"value":73970544},"jvm.memory.pools.G1-Old-Gen.used-after-gc":{"value":0},"jvm.memory.pools.G1-Survivor-Space.committed":{"value":7340032},"jvm.memory.pools.G1-Survivor-Space.init":
{"value":0},"jvm.memory.pools.G1-Survivor-Space.max":{"value":-1},"jvm.memory.pools.G1-Survivor-Space.usage":{"value":1.0},"jvm.memory.pools.G1-Survivor-Space.used":{"value":7340032},"jvm.memory.pools.G1
Survivor-Space.used-after-gc":{"value":7340032},"jvm.memory.pools.Metaspace.committed":{"value":41889792},"jvm.memory.pools.Metaspace.init":{"value":0},"jvm.memory.pools.Metaspace.max":{"value":-
1},"jvm.memory.pools.Metaspace.usage":{"value":0.9627384180759943},"jvm.memory.pools.Metaspace.used":{"value":40323304},"jvm.memory.total.committed":{"value":274353920},"jvm.memory.total.init":
{"value":202702848},"jvm.memory.total.max":{"value":-1},"jvm.memory.total.used":{"value":170138784},"jvm.threads.blocked.count":{"value":0},"jvm.threads.count":{"value":22},"jvm.threads.daemon.count":
{"value":6},"jvm.threads.deadlock.count":{"value":0},"jvm.threads.deadlocks":{"value":[]},"jvm.threads.new.count":{"value":0},"jvm.threads.peak.count":{"value":24},"jvm.threads.runnable.count":
{"value":10},"jvm.threads.terminated.count":{"value":0},"jvm.threads.timed_waiting.count":{"value":9},"jvm.threads.total_started.count":{"value":29},"jvm.threads.waiting.count":
{"value":3},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw-admin.jobs":{"value":0},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw-admin.jobs-queue-utilization":
{"value":0.0},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw-admin.size":{"value":5},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw-admin.utilization":
{"value":1.0},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw-admin.utilization-max":{"value":0.078125},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw.jobs":
{"value":0},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw.jobs-queue-utilization":{"value":0.0},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw.size":
{"value":8},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw.utilization":{"value":0.375},"org.eclipse.jetty.util.thread.QueuedThreadPool.dw.utilization-max":{"value":0.0029296875}},"counters":
{"TimeBoundHealthCheck-pool.running":{"count":0},"io.dropwizard.jetty.MutableServletContextHandler.active-dispatches":{"count":0},"io.dropwizard.jetty.MutableServletContextHandler.active-requests":
{"count":0},"io.dropwizard.jetty.MutableServletContextHandler.active-suspended":{"count":0}},"histograms":{},"meters":{"TimeBoundHealthCheck-pool.created":
```

Figure 5 : Extraction metrics for scholarly resources

Applications for Libraries

This open-source, ML-based tool revolutionizes library operations by automating scholarly content management. It transforms manual workflows through advanced extraction of metadata from unstructured PDFs, enabling efficient cataloging and retrieval. The system handles complex document formats—multi-column layouts, footnotes, tables—including historical records, while supporting multilingual content for global accessibility.

Indexing not only makes processes more efficient, but it also makes finding any document in the library much easier. The compatibility of the software with Zotero, Mendeley, and cataloging tools makes daily operations much smoother. Libraries may improve their results by setting up their models according to the citation styles and metadata they require.

Adaptability is the foundation for this open-source

Conclusion

The ability to process scientific information as data that machines can use is now essential for making research accessible to more people. These new technologies based on machine learning are leading this transformation. Automatically differentiating raw PDF documents according to common metadata leads to more efficient and exact results in scholarly communication work. Its role is seen through the fine-tuned architecture that can carry out various tasks, for example, recognizing people linked to the paper, working with references, breaking down the full article and standardizing related information. Because of their strong APIs and batch-processing pipelines, these applications can handle the workload of repositories, digital libraries, publishers and research infrastructure easily. Using automated systems, these systems help keep metadata uniform, reduce mistakes and increase the quality of data overall. More importantly, the system supports broader initiatives in open science, semantic search, and research analytics—laying a robust foundation for next-generation scholarly information systems. As the volume of global research output continues to grow, the importance of such intelligent, machine learning-driven extraction tools will only become more central to the ecosystem of modern academia.

References

Romary, L., & Lopez, P. (2015). Grobid—Information extraction from scientific publications. *ERCIM News*, *100*. https://inria.hal.science/hal-01673305

Lopez, P. (2009).*GROBID: combining automatic bibliographic data recognition and term extraction for scholarship publications*. https://doi.org/10.1007/978-3-642-04346-8_62

Rettenberger, L., Münker, M. F., Schutera, M., Niemeyer, C. M., Rabe, K. S., & Reischl, M. (2024). Using Large Language Models for Extracting Structured Information From Scientific Texts.*Current Directions in Biomedical Engineering*. https://doi.org/10.1515/cdbme-2024-2129

Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G., Persson, K. A., & Jain, A. (2024). Structured information extraction from scientific text with large language models. *Nature Communications*. https://doi.org/10.1038/s41467-024-45563-x

Sebastian, Y. (2017).*Literature-Based Discovery by Learning Heterogeneous Bibliographic Information Networks*. https://doi.org/10.1145/3130332.3130347

Sebastian, Y., Siew, E.-G., & Orimaye, S. O. (2017). Learning the heterogeneous bibliographic information network for literature-based discovery.*Knowledge Based Systems*. https://doi.org/10.1016/J.KNOSYS.2016.10.015

Agrawal, K., Mittal, A., & Pudi, V. (2019).*Scalable, Semi-Supervised Extraction of Structured Information from Scientific Literature*. https://doi.org/10.18653/V1/W19-2602

Guo, J., Ibanez-Lopez, A. S., Gao, H., Quach, V., Coley, C. W., Jensen, K. F., & Barzilay, R. (2021). Automated Chemical Reaction Extraction from Scientific Literature.*Journal of Chemical Information and Modeling*. https://doi.org/10.1021/ACS.JCIM.1C00284

Yang, H., Aguirre, C., Torre, M. F. D. L., Christensen, D., Bobadilla, L., Davich, E., Roth, J., Luo, L., Theis, Y., Lam, A., Han, T. Y.-J., Buttler, D., & Hsu, W. H. (2019).*Pipelines for Procedural Information Extraction from Scientific Literature: Towards Recipes using Machine Learning and Data Science*. https://doi.org/10.1109/ICDARW.2019.10037