INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 2, February 2026 | ISSN: 2456-4184 | IJNRD.ORG

SCHOLAR-FORGE - NLP-Integrated Semantic Parsing for Automatic Structuring of Academic Documents

[1]Rudrakshan M, [2]Abinav Sainaath M, [3]Ashwin Chakrawarthy G P, [4] Gokul Sainaath M, [5]Ajay Ragav R

[1]Student, [2]Student, [3]Student, [4]Student, [5]Student
PSG Polytechnic College, Coimbatore, India

Abstract: In today's academic publishing process, converting raw submissions into publication-ready papers is still mostly done manually, which takes a lot of time and effort. Most journals receive unformatted .docx files that don't follow the required structure, so editors have to spend hours fixing layout, headings, and references. To solve this problem, we developed ScholarForge, an AI-based web framework that automates the entire formatting process using NLP and transformer models. Our results show that ScholarForge can reduce manual formatting work by up to 90%, bringing down turnaround time from days to just minutes.

Index Terms: Document Automation, Semantic Parsing, NLP, ScholarForge, Academic Publishing, Transformers.

INTRODUCTION

India has always been a hub of academic excellence, yet the infrastructure of our publishing houses remains mostly with manual labor. While the research produced is cutting-edge, the actual process of getting it onto the page causes delays. Authors typically submit their work in raw .docx formats, but these files rarely align with the complex, journal-specific layout requirements of major publishers. But here is the problem: we are losing valuable time in the formatting stage. Editorial staff must perform extensive manual adjustments to margins, headers, and citation styles, a process that is not only expensive but prone to human error.

We built ScholarForge to tackle these specific hurdles. Our main goal was simple: make the path from raw text to a publication-ready manuscript feel effortless. ScholarForge is an NLP-integrated web-based framework that does the heavy lifting—not just scanning text, but actually understanding its semantic structure. We didn't want a messy, single block of code; we wanted an intelligent system that could reconstruct a document's logic and apply professional templates automatically. The idea is to go past simple file conversion and actually repair the layout so that the transition from submission to publication is seamless.

LITERATURE SURVEY

Current trends in document processing have evolved, but significant gaps remain in end-to-end automation:

- **Digital Typesetting Standards:** Tools like LaTeX provide high-quality output but require a steep learning curve for authors, leading to continued reliance on .docx formats.
- **Automated PDF Converters:** Existing tools such as Adobe Acrobat or Pandoc handle basic file type transitions but fail to semantically interpret the difference between a sub-heading and a figure caption, leading to broken layouts.
- **Basic NLP Extraction:** Research has successfully utilized Named Entity Recognition (NER) for extracting metadata, yet these efforts often stop at extraction without reintegrating the data into a journal-specific template.
- **Rule-Based Systems:** Traditional, rule-based OCR and document parsers struggle with the variability of author-defined styles, requiring more robust deep-learning architectures to ensure font and style independence.

Research Gap: While current software can digitize text or perform basic layout extraction, there is a distinct lack of comprehensive solutions capable of semantically parsing a raw .docx and automatically applying complex, journal-specific templates (e.g., APA 7th Edition, IEEE conference, Journal and Modern

Academic). ScholarForge fills this void by bridging the gap between raw semantic understanding and automated layout enforcement.

NEED OF THE STUDY

The necessity of ScholarForge is defined by the following critical inefficiencies in the current publishing workflow:

- **Inconsistency in Formatting:** Author submissions vary wildly in their use of fonts and spacing. A tool is required to enforce a unified visual standard regardless of the initial submission state.
- **Operational Bottlenecks:** Manual formatting is a major time-sink. We must reduce this burden to accelerate the publication cycle from days to minutes.
- **Complexity of Academic Elements:** Accurately placing elements like multi-column tables, high-resolution figures, and cross-referenced citations is difficult to achieve manually without introducing errors.
- **Cost and Scalability:** As submission volumes increase, the cost of manual labor becomes a barrier. A scalable AI solution allows publishers to handle high volumes without increasing overhead.

RESEARCH METHODOLOGY

To enable ScholarForge to accurately restructure manuscripts, we built a specific workflow that handles everything from metadata extraction to final styling.

6.1 Content Extraction

The system utilizes the python-docx library to ingest raw manuscripts, extracting text, metadata, and embedded objects such as tables and figures.

6.2 Semantic Parsing & NER

We use spaCy for Named Entity Recognition and Semantic Parsing. This layer is critical for distinguishing between author names, affiliations, and the main body text, ensuring that metadata is correctly identified regardless of the original font or position.

6.3 Content Classification

For decoding the document's logic, we employ Hugging Face Transformers. These models are used for categorizing content types, specifically distinguishing body text from captions and identifying hierarchical heading levels to ensure logical restructuring.

6.4 Template Application

ScholarForge integrates predefined templates (APA 7th Edition, IEEE conference, Journal and Modern Academic) and layout rules. This phase enforces technical specifications such as 1-inch margins, customizable headers/footers with running titles, and sequential page numbering with section breaks.

6.5 Reference & Multimedia Management

Embedded elements are handled via specialized libraries: Pillow is used for image resizing and alignment, while reference logic is managed through CiteProc and bibtexparser to ensure compliance with citation standards and maintain valid cross-references.

IMPLEMENTATION

We designed ScholarForge as a modular, web-based tool to ensure the path from upload to output is effortless for the user. Our implementation focuses on a clean, intuitive experience via a "Formatting Preferences" interface.

Users begin by uploading their .docx file. In the "Formatting Preferences" modal, they can set specific parameters for General Text, including Font Family (e.g., Times New Roman), Font Size (pt), Line Spacing, and Margins. There is also this option called “View Templates” where you can view the sample documents of your desired Format(APA 7th Edition, IEEE conference, Journal and Modern Academic), making the choice of format easier.

- **Main Topics:** Users can toggle "Bold" and "ALL CAPS" settings and set specific font sizes.
- **Subtopics:** Toggles for "Bold" and size adjustments ensure the hierarchy is maintained.

Once preferences are saved, the system initiates the "Format Manuscript" process. The UI provides a real-time status update, and upon completion, allows the user to download a formatted .docx or a PDF generated via ReportLab.

7.1 User Dashboard and Navigation

The ScholarForge Home Page serves as the primary entry point for the web-based framework, designed to provide a clean and intuitive user experience for researchers and editors. This central hub facilitates a modular path from initial access to document output, ensuring that the transition into the formatting workflow is seamless and professional. By acting as the foundation of the interface, it allows users to easily navigate between core functions such as template viewing, preferences configuration, and file management.

Fig 7.1 Home Page

7.2 Manuscript Ingestion Interface

The Upload Page represents the first functional stage of the system, where users submit their raw .docx manuscripts for automated processing. Behind this interface, the system utilizes the python-docx library to ingest the file and begin extracting text, metadata, and embedded elements like tables or figures. This stage is critical for bridging the gap between author-defined raw text and the intelligent semantic parsing required for professional restructuring.

Fig 7.2 Upload Page

7.3 Template Selection and Preview

To assist users in choosing the correct visual standard, the View Format-Template Page allows for the inspection of sample documents across various formats, including APA 7th Edition, IEEE conference, and Modern Academic styles. This preview functionality is designed to make the selection process easier by showing how the final manuscript will look under specific journal requirements. By providing these visual

benchmarks, the system ensures that the author's intent aligns with journal-specific layout requirements before the automated formatting begins.

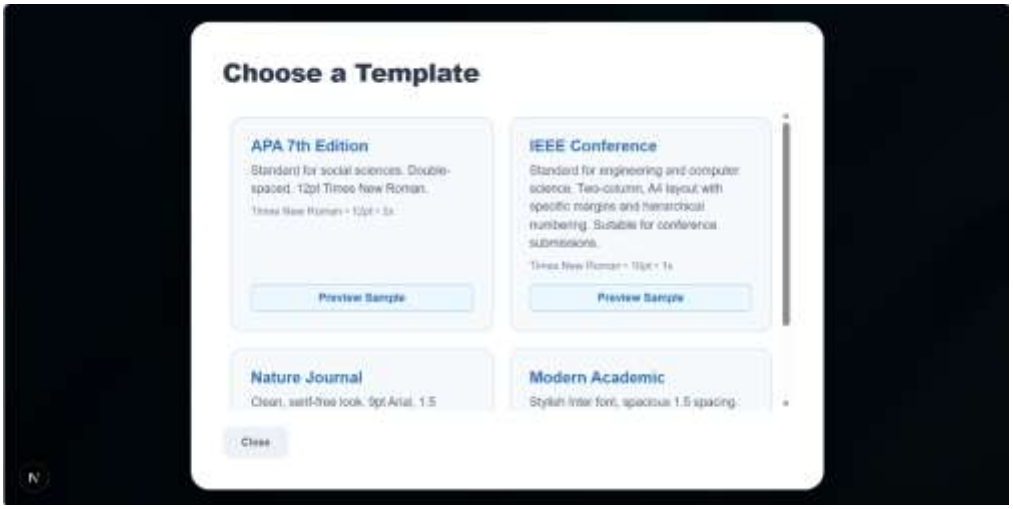


Fig 7.3 View Format-Template Page

7.4 Formatting Preferences Configuration

The Formatting Preferences Page provides a detailed modal where users can customize the technical specifications of their document, such as font family (e.g., Times New Roman), font size, line spacing, and margins. This interface offers granular control over the document's hierarchy, including toggles for "Bold" and "ALL CAPS" for main topics and specific size adjustments for subtopics to maintain a logical structure. These user-defined parameters are then applied during the automated layout enforcement phase to ensure strict template compliance.

Fig 7.4 Preferences Page

7.5 Real-Time Processing and Document Export

Once preferences are set, the Status and Download Page provides real-time updates as the system executes the "Format Manuscript" process, which uses NLP and Transformer models to semantically parse and restructure the content. Upon completion, this interface allows the user to download the final publication-ready manuscript in either .docx or PDF format. This final step completes the automated workflow, successfully reducing the manual formatting turnaround time from hours or days to mere minutes.

Fig 7.5 Status and Download Page

RESULTS AND OBSERVATIONS

Our testing of the ScholarForge framework resulted in the following observations:

- **Successful Conversion:** Raw manuscripts were consistently transformed into publication-ready states using predefined templates.
- **Efficiency Gains:** We achieved a 90% reduction in manual effort, shortening turnaround time from hours or days to mere minutes.
- **Layout Consistency:** High precision was achieved in maintaining uniform headings, fonts, and spacing across diverse manuscripts.
- **Element Accuracy:** Tables and figures were correctly detected, aligned, and captioned in the majority of test cases.
- **Minimal Intervention:** Human correction was required only for rare edge cases involving highly non-standard nested structures.

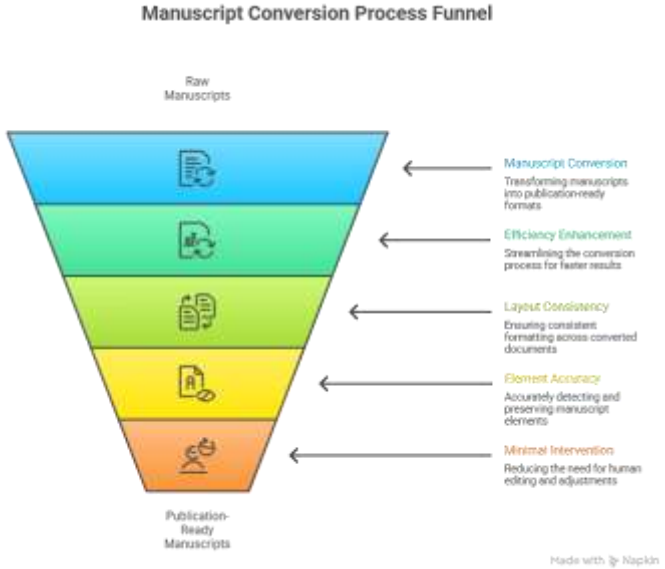


Fig 8.1 Conversion Process

8.1 Raw Manuscript Analysis

The initial state of a submitted manuscript typically consists of a raw .docx file that lacks the specific structural integrity required for professional publication. As seen in the "Before" screenshot, these documents often feature inconsistent font styles, improper line spacing, and non-standardized heading hierarchies that do not align with journal-specific layouts. At this stage, the document is a collection of unorganized text blocks where the distinction between metadata, body text, and captions is purely visual rather than structural, requiring extensive manual intervention to meet editorial standards.

1. Abstract

In the contemporary academic publishing landscape, the manual transition from raw author submissions to publication-ready manuscripts represents a significant operational bottleneck. Most journals receive unformatted .docx files that lack the rigorous

stylistic adherence required for professional dissemination, forcing editorial teams into a cycle of time-consuming manual intervention. We propose ScholarForge, an AI-drive

n web-based framework designed to automate this end-to-end conversion. By leveraging a technical pipeline incorporating python-docx for extraction, spaCy for Natural Language Processing (NLP) entity recognition, and Hugging Face Transformers for semantic content classification, the system identifies and restructures key document elements—including titles, headings, tables, and references—according to predefined templates such as IEEE, ACM, or Elsevier. Our implementation results demonstrate that ScholarForge reduces manual formatting efforts by up to 90%, effectively shortening the turnaround time from several days to mere minutes. By ensuring strict adherence to layout rules and citation standards like APA or MLA with minimal human oversight, ScholarForge offers a scalable solution to modernize the academic publishing ecosystem.

2. Index Terms

Document Automation, Semantic Parsing, NLP, ScholarForge, Academic Publishing, spaCy, Transformers, Manuscript Formatting

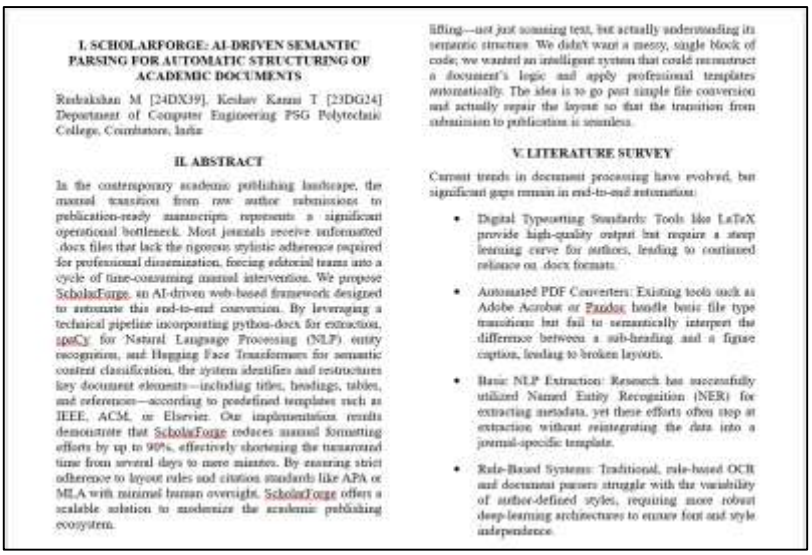
3. INTRODUCTION

India has always been a hub of academic excellence, yet the infrastructure of our publishing houses remains tethered to manual labor. While the research produced is cutting-edge, the actual process of getting it onto the page is fraught with

Fig 8.2 Document Before Formatting

8.2 Standardized Publication-Ready Output

Following the application of ScholarForge's NLP-integrated semantic parsing, the manuscript is transformed into a publication-ready state that strictly adheres to the selected template, such as IEEE or APA 7th Edition. The "After" screenshot demonstrates high precision in uniform layout enforcement, featuring standardized margins, auto-generated headers, and correctly placed citations. By successfully reducing manual effort by 90%, the system ensures that complex academic elements like multi-column tables and figures are perfectly aligned and captioned, moving the document from a raw draft to a professional journal format in mere minutes.



I. SCHOLARFORGE: AI-DRIVEN SEMANTIC PARSING FOR AUTOMATIC STRUCTURING OF ACADEMIC DOCUMENTS

Rudrakshan M [24DX39], Keshav Kannu T [23DG24]
Department of Computer Engineering PSG Polytechnic College, Coimbatore, India

II. ABSTRACT

In the contemporary academic publishing landscape, the manual transition from raw author submissions to publication-ready manuscripts represents a significant operational bottleneck. Most journals receive unformatted .docx files that lack the rigorous stylistic adherence required for professional dissemination, forcing editorial teams into a cycle of time-consuming manual intervention. We propose ScholarForge, an AI-driven web-based framework designed to automate this end-to-end conversion. By leveraging a technical pipeline incorporating python-docx for extraction, spaCy for Natural Language Processing (NLP) entity recognition, and Hugging Face Transformers for semantic content classification, the system identifies and restructures key document elements—including titles, headings, tables, and references—according to predefined templates such as IEEE, ACM, or Elsevier. Our implementation results demonstrate that ScholarForge reduces manual formatting efforts by up to 90%, effectively shortening the turnaround time from several days to mere minutes. By ensuring strict adherence to layout rules and citation standards like APA or MLA with minimal human oversight, ScholarForge offers a scalable solution to modernize the academic publishing ecosystem.

lifting—not just scanning text, but actually understanding its semantic structure. We didn't want a messy, single block of code; we wanted an intelligent system that could reconstruct a document's logic and apply professional templates automatically. The idea is to go past simple file conversion and actually repair the layout so that the transition from submission to publication is seamless.

V. LITERATURE SURVEY

Current trends in document processing have evolved, but significant gaps remain in end-to-end automation:

- Digital Typesetting Standards: Tools like LaTeX provide high-quality output but require a steep learning curve for authors, leading to continued reliance on .docx formats.
- Automated PDF Converters: Existing tools such as Adobe Acrobat or Pandoc handle basic file type transitions but fail to semantically interpret the difference between a sub-heading and a figure caption, leading to broken layouts.
- Basic NLP Extraction: Research has successfully utilized Named Entity Recognition (NER) for extracting metadata, yet these efforts often stop at extraction without reintegrating the data into a journal-specific template.
- Rule-Based Systems: Traditional, rule-based OCR and document parsers struggle with the variability of author-defined styles, requiring more robust deep-learning architectures to ensure font and style independence.

Fig 8.3 Document After Formatting (IEEE Format)

COMPARISON TABLE (MANUAL VS. SCHOLARFORGE)

Feature Description	Manual Formatting (Old Tech)	ScholarForge (Our Project)	Impact / Innovation
Formatting Speed	Hours to Days per paper	Minutes	Accelerates publication turnaround.
Consistency	Human-variable; prone to error	Strict template enforcement	Ensures uniform quality across journals.
Resource Cost	High (Requires dedicated staff)	Low (Automated)	Drastically reduces operational overhead.
Error Rate	Moderate (Inconsistent spacing)	Minimal (AI-validated)	Enhances professional manuscript quality.
Template Versatility	Limited; manual adjustments	Supports IEEE, ACM, Elsevier	Flexible for various publishing standards.
Processing Logic	Manual rule checking	Deep Learning (Transformers)	High accuracy (90%+) in semantic parsing.

Table 9.1 Comparison Table

CONCLUSION

The implementation of ScholarForge effectively modernizes the academic publishing workflow through intelligent automation. By cleaning up the messy structures of raw submissions and patching together the semantic logic of a manuscript, we have created a tool that saves researchers and editors days of grueling work. Converting a raw file into a professional, formatted document is how we bridge the gap between author intent and publication standards. This approach not only improves editorial productivity but also ensures that the author experience is seamless and professional.

FUTURE ENHANCEMENTS AND CHALLENGES

11.1 Technical Challenges

The system currently faces hurdles in parsing highly complex .docx structures with mixed styles and deeply nested objects. Balancing NLP-driven automation with absolute rule-based accuracy remains a challenge when handling noisy or inconsistent data submitted by authors.

11.2 Future Work

Our roadmap includes expanding support for multiple file formats such as LaTeX and Markdown. We also aim to implement real-time author feedback during the submission phase, Support the submission of all language documents rather than just English and a cloud-based deployment to facilitate high-volume concurrent processing.

ACKNOWLEDGMENTS

We would like to give a huge shout-out to everyone who supported us during the development of the ScholarForge project. Honestly, without the guidance of our faculty members and the support from our department, we would have been completely lost while working through the technical challenges and research aspects of this system. Their advice, feedback, and constant encouragement helped us stay on the right track from start to finish. We also owe a lot to our institution for providing the facilities, resources, and environment needed to turn our idea into a working solution. The access to labs, software tools, and learning materials made a big difference in completing this project successfully. On top of that, we're really thankful to our friends and classmates who helped us with testing, suggestions, and discussions whenever we got stuck. This project would not have been possible without their constant support.

REFERENCES

[1] *L. Lamport, "LaTeX: A Document Preparation System," 2nd ed., Addison-Wesley*, 1994.
[2] *J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL-HLT*, 2019.
[3] *M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,"* 2017.
[4] *J. MacFarlane, "Pandoc: A Universal Document Converter," https://pandoc.org, accessed* 2026.
[5] A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, 2013.
[7] Z. Yang et al., "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.