Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 3 No. 12 (December) (2025)

AIDP: A Standards Driven Human-Automation Architecture for Intelligent and Secure Document Indexing

Sultan Ahmed
Research Assistant, Library and Information Management, Suzhou University of Technology, Changshu City, Jiangsu Province 215500, P. R. China.

Wang Yage*
Research Librarian & Professor, Library, Suzhou University of Technology, Changshu, Suzhou, Jiangsu Province 215500, P. R. China Email: wyg@szut.edu.cn

ABSTRACT
Index creation is a cognitively demanding and time-intensive component of scholarly publishing, requiring both mechanical accuracy and expert intellectual judgment. Existing automated indexing tools often fall short of professional standards, while fully manual indexing remains costly and difficult to scale. This paper presents the Advanced Intelligent Document Processor (AIDP), a standards-driven human–AI collaborative architecture designed to support intelligent, secure, and reproducible document indexing. AIDP adopts a human-in-the-loop approach by automating routine mechanical tasks, such as heading detection, alphabetical sorting, deduplication, and locator compression, while reserving semantic decision-making and conceptual organization for human indexers. The system is entirely client-side and web-based, ensuring data privacy for unpublished and sensitive documents, and is explicitly aligned with ISO 999 and the Chinese national standard GB/T 41210-2021. The architecture is implemented through a deterministic, rule-based processing pipeline and evaluated using benchmark documents of varying structural complexity. Experimental results demonstrate high performance, achieving up to 98% accuracy across key indexing metrics while preserving hierarchical integrity. The findings indicate that AIDP offers a practical, scalable solution for professional indexing, bridging traditional indexing expertise with contemporary digital infrastructure.

Keywords: Document Processing; Index Generation; Algorithms; Human-Computer Interaction; Web Application

Introduction
Indexes play a foundational role in scholarly and professional publications by enabling systematic access to knowledge, revealing conceptual relationships, and supporting efficient information retrieval [1]. Far from being a mechanical appendix, a well-constructed index is an intellectual infrastructure that reflects both the structure of a text and the indexer's interpretive judgment. Despite their recognized importance, index creation remains a labor-intensive and cognitively demanding process that requires meticulous attention to consistency, hierarchy, and semantic relevance. While advances in digital publishing have introduced automated indexing tools, many of these systems struggle to meet professional standards, particularly in preserving structural integrity and semantic coherence [2, p.390]. This persistent tension between efficiency and quality highlights the need for standards-driven, human, AI collaborative approaches that integrate computational automation with expert human judgment, an imperative that

ISSN Online: 3007-3154
ISSN Print: 3007-3146

motivates the development of the Advanced Intelligent Document Processor (AIDP) proposed in this study.

According to the importance of indexing, each region and place has its own methods of use. In most Western publishing, indexing is done more professionally, supported by related organizations, capacity-building sessions, and advanced tools. Binney's famous saying underscores the value of indexing, stating that "books without indexes are like human bodies without souls" [12]; this shows how vital indexing is to a book. Professional indexers mostly use software such as CINDEX, MACREX, and SKY Index, which have improved the field of indexing and help indexers work effectively and efficiently [3, p. 413].

Unlike the well-established indexing practices in Western countries, professional indexing in China has developed only recently. Different milestones indicate this progress; for example, in 1991, the Chinese Index Society was founded to provide official recognition to the field of indexing. The Index Star was the first professional indexing software, released in 2003. The creation of the national standard GB/T 22466-2008 established a set of indexing rules, and by 2012, indexing began being used in academic works, making it widely accepted. Several important Chinese publications demonstrate the growing work in the field of indexing, such as Xi Jinping's book The Governance of China (2014), the Local Gazetteer Indexing Standard (2018), and related theses, which support systematic indexing. However, still less than 5% of Chinese academic publications use indexes, leaving room for professional improvement in China.

This difference shows that there is a need for reliable, professional standard tools that meet all the requirements of indexers. Balancing the two important parts of work is the biggest challenge in indexing. On the one hand, routine tasks such as sorting, formatting, indenting, and removing duplicates are easily handled by automated systems. On the other hand, making decisions, such as organizing ideas and choosing the right terms, requires human judgment and skill. To solve this problem, this paper introduces the Advanced Intelligent Document Processor (AIDP), a system that focuses on repetitive tasks while leaving more intellectual work to human editors. The primary function of AIDP is a step-by-step process that transforms raw text into a well-organized, ready-to-publish index. Additionally, AIDP is a client-side system that keeps user data secure, which is important for unpublished and sensitive documents [5, 6, 12]. AIDP contributes to the development of indexing as a professional software by following and integrating international standards and established indexing practices from both the West and China.

Background and Related Work

The principle of indexing is well-defined and widely adopted worldwide in academic publishing. These principles are established in international standards such as ISO 999 [7], which guides the creation of organized indexes and includes guidelines for major index styles, such as the Chicago Manual of Style [8]. China has introduced its own indexing standards in recent years. For example, the current Chinese National Standard GB/T 41210-2021 [1] specifies how indexes should be used for academic dissertations, how to organize and label index entries, and emphasizes the importance of accuracy and meaningfulness. These international and national guidelines clearly show that indexing is widely used in academic and professional publishing, and creating high-quality indexes is increasingly essential.

An index is not just about listing text and page numbers; it also provides the reader with a clear road map of a document. Essentially, the index highlights key concepts and explains relationships among topics, making it easier for readers to understand the

ISSN Online: 3007-3154
ISSN Print: 3007-3146

author's main idea [9, p. 407]. The index helps readers understand the main idea of larger documents by clearly organizing information. In this way, a good index makes a document clearer and easier for readers to understand what the author wants to convey.

Dedicated Indexing Software

In Western countries, the CINDEX, MACREX, and SKY Index are popular tools for professional indexers, and they are widely used [3, pp. 415–418]. These tools are popular for managing extensive, complex indexes. They help ensure that all entries are well organized, consistent, and rule-based. However, these tools mostly work with documents that have fixed page numbers. As a result, problems arise when a publisher changes page numbers later, rendering the index inaccurate [4, p. 430].

Embedded Indexing

Microsoft Word, a popular program with built-in tools, allows users to add an index directly into the text. This method links the index directly to the relevant part of the document, making it easier for users to find information. However, while this method can be helpful, it has some disadvantages. Due to the lengthy process where each entry must be marked separately, editors often feel disappointed. It has been shown that this process increases work time by 40%, thereby increasing the overall time required [4, p. 431].

Automatic and "Cyborg" Indexing

Automated indexing methods primarily rely on word frequency and have limited scope because they cannot understand the main ideas in a text [3, p.399]. These can create technically correct indexes, but the results often lack practical relevance. To address this issue, the concept of "cyborg indexing" was introduced [3, p.399]. In this approach, humans and computers work together. The computer or software handles imperfect and repetitive tasks, while humans make key decisions, such as selecting queries, terms, organizing tasks, and managing cross-references.

The Advanced Intelligent Document Processor (AIDP) is responsible for performing this process. It performs tasks such as sorting, removing duplicates, indenting, formatting, and other tasks, like selecting terms and concepts for human users. This method allows AIDP to complete tasks according to human needs and create indexes that are not only systematic but also meaningful [1].

Chinese Developments: The Case of "Indexer"

In China, a platform like "Indexer" has been developed to enable more accurate, collective indexing. Wang et al. [12] provided an overview of the software's main features, noting that it is open source, meets user needs, and has strong security measures. This idea creates an easy path for the development of AIDP and highlights key points such as keeping the tool visible online, simple to use, and centralized. AIDP can be more effective if users adopt and follow the defined guidelines.

Web-based Tools and Standard Promotion

Nowadays, indexers use web-based tools rather than complex software like the HTML Indexer project. Meanwhile, in China, standards such as GB/T 22466–2008 and GB/T 41210–2021 are the most popular and widely used by most indexers. These standards have made indexing more than just paperwork and provide significant status in the academic field [1,5]. The AIDP Project's multiple features, such as its web-based, user-

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 3 No. 12 (December) (2025)

friendly interface, make it unique among software. In this way, it not only supports Western professionals but also assists China's own standardization efforts [6, 12].

System Overview: The AIDP Processing Model

The AIDP System is designed with a clear, structured, and step-by-step process. Each step has well-defined tasks that it performs when it receives an output. This user-friendly design not only makes it easy to understand the processes clearly but also provides results efficiently. The complete system layout is shown in Figure 1.

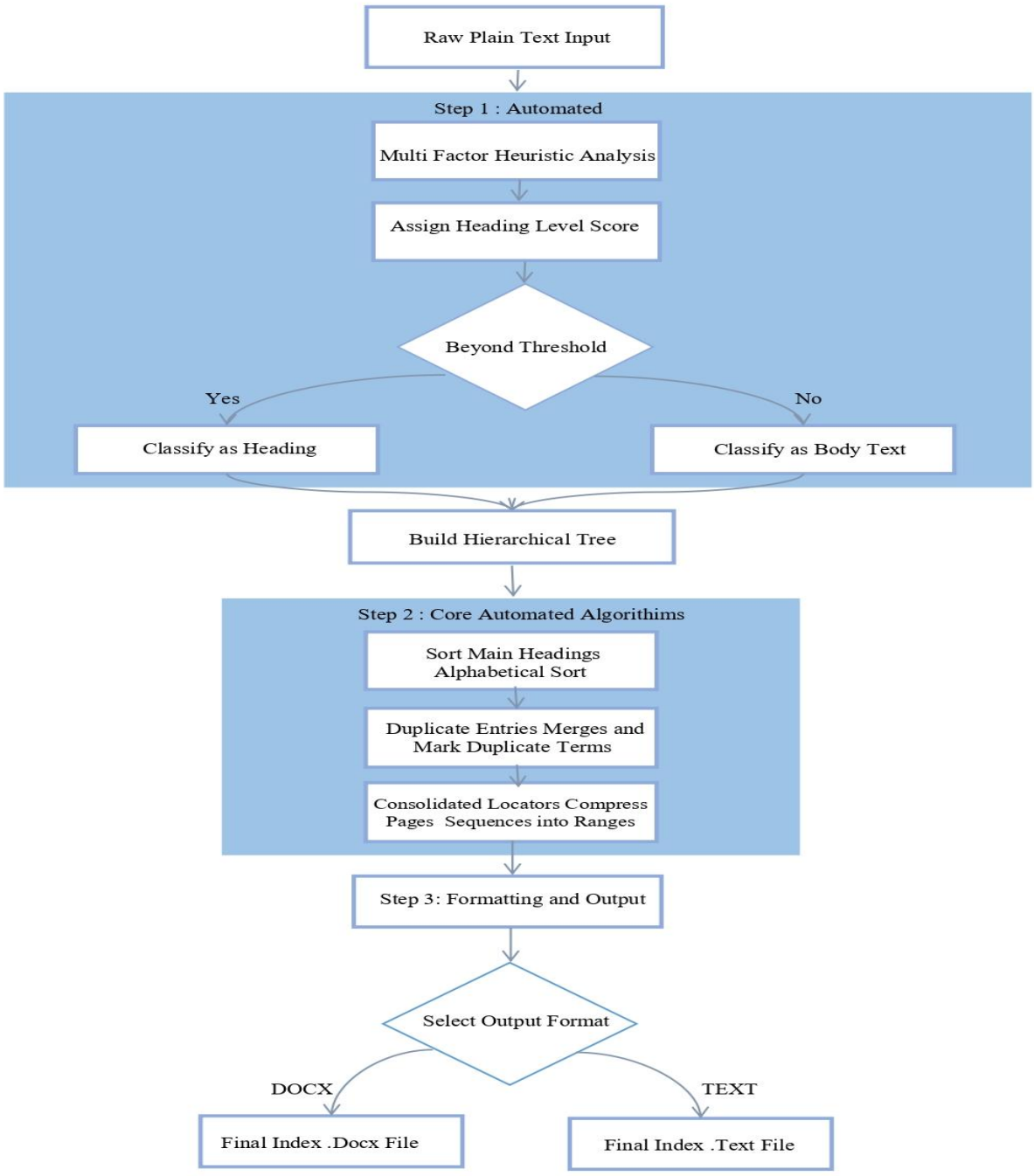


Figure 1: AIDP Processing Model

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 3 No. 12 (December) (2025)

Overall Processing Model: AIDP System

The architecture of the Advanced Intelligent Document Processor (AIDP) is a fully automated system that works through a step-by-step process. It takes raw text, processes it, and outputs results in a clear, organized indexing format. AIDP operates based on specific steps; the same input will always produce the same results. It also reduces the time required compared to manual work. Once the input is provided, it automatically performs all functions, from reading and analyzing the text to creating a well-structured index. This enables the entire system to run efficiently without additional effort from users.

Step-by-Step Breakdown of the Automated Model

Input

The system's processing begins with straightforward text input, assuming the text's structure; at times, this may be unclear.

Step 1: Automated Heading Detection and Tree Construction

In the initial phase, the system divides the text into separate lines. Each line is evaluated by the assessLine() function, which performs the analysis automatically without user involvement.

The system examines various factors, such as font styles, weights, indentation, and line spacing.

Based on the examination results, the system assigned a "heading score" to each line.

The line with the higher score (threshold) is treated as a heading and placed at a higher hierarchy level, while the lines with lower scores are treated as regular body text.

As a result, the system constructs a hierarchical tree based on the input data.

Step 2: Deterministic Processing Pipeline (Automated)

In the next phase, the system executes a series of steps on the generated tree.

Sorting of Main Headings:

The system automatically sorts the main headings.

The main headings keep their subheadings intact, without altering their structure.

Deduplication of Entries: The system checked the tree to find the duplicate normalized text entries.

A predefined rule determines how to process duplicates: either merge them into a single locator or label them with an [xN] tag.

Consolidation of Locators: For each Entry, the page has been sorted, and their line numbers checked.

When three or more numbers are detected (e.g., 5, 6, 7, 5,6,7,8), they are then compressed into a range (e.g., 5-8).

This compression is based on the rule that at least three consecutive numbers come, which is the minimum requirement.

Step 3: Formatting and Output

When the tree completes its process, it then creates the final formatted index without requiring user validation.

For DOCX files, the system generates headings, links them into a specific styled format, and maintains each heading level with its own indentation.

For plain text files, the system creates structure using indentation spaces, with important headings aligned to the left and the rest indented further to the right.

Output Specification

Primary Output: A DOCX file.

Secondary Output: A plain text file.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 3 No. 12 (December) (2025)

The accuracy of the indexed file fully depends on the input file or text. If their headings are not properly tagged in the original file, then the resulting index may be incorrect.

Methods and Algorithms

Heading Detection Heuristic

The system first identifies the document headings, not just by checking their text style but also by examining font size, indentation, and numbering, then determines their position within main headings, subheadings, and sub-subheadings.

Algorithm 1:detectHeadings(text)

```
1. lines = splitTextIntoLines(text)
2.  tree = new RootNode()
3.  currentLevel = 0
4.  stack = new Stack()
5.  stack.push(tree)
6.  for each line in lines:
7.      candidateLevel = assessLine(line)
8.      if candidateLevel > 0: // It's a heading
9.          newNode = new HeadingNode(text=line, level=candidateLevel)
10.         while stack.peek().level >= candidateLevel:
11.             stack.pop()
12.         stack.peek().addChild(newNode)
13.         stack.push(newNode)
14.     else: // It's body text
15.         stack.peek().appendToLocator(lineNumber)
16.     return tree
```

The assessLine(line) function analyzes each text line and assigns it a score.

Font Style: The algorithms check the texts that are bold, italic, or written in ALL CAPS.

Numbering Patterns: Using different patterns creates number formats like “1”, “1, 2, 3, 1,2,3,4”, etc.

Indentation measures the spaces and tabs at the beginning of a line by understanding the hierarchy level of the text.

Short lines are usually considered headings since they serve as titles rather than full sentences.

The system combines all related signals into a weighted score and then compares this score with a threshold value (θ). If the line score exceeds the threshold, then the line is considered a heading. After that, its hierarchy is determined by using indicator strength and pattern. This scoring approach works well and remains effective even when document sections have different formats. It doesn't need to be perfect to operate correctly.

However, sometimes it misidentifies the short paragraph as a heading because of the similarity of patterns.

Sorting and Subtree Preservation

Alphabetical sorting of main headings is the main rule for creating professional indexes without disrupting the hierarchy, where all subheadings must be grouped under their related main headings. A good index needs to be in alphabetical order and grouped under the related main headings. This method is logically organized, which is endorsed by

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 3 No. 12 (December) (2025)

indexing experts.

Algorithm 2:sortMainHeadings(tree)

```
1. mainHeadings = tree.getChildren() // Get top-level nodes
2. sortedMainHeadings = sortAlphabetically(mainHeadings, caseSensitiveFlag)
3. // for each mainNode in sortedMainHeadings:
4.  tree.setChildren(sortedMainHeadings) // Keep subheading on their original orders.
```

Duplicate Handling

AIDP removes repeated words or terms because they often appear multiple times in documents, so the system deduplicates them. AIDP allows combining duplicates into a single entry or flagging them as [x3], making it easier for readers to see how frequently the terms occur.

Algorithm 3: deduplicate(tree)

```
1. nodeMap = new Map() // Main Headings: All related listed below.
2. traverseTreeAndPopulateMap(tree, nodeMap) // for each key in nodeMap:
3. nodes = nodeMap.get(key)
5.     if nodes.length > 1:
6.         if mergeDuplicatesFlag is True:
7.             primaryNode = nodes[0]
8.             for i in range(1, nodes.length):
9.                 primaryNode.locators.addAll(nodes[i].locators)
10.                // Remove the subheading from its headings and delete it.
11.        else:
12.            for each node in nodes:
13.                node.text = node.text + " [x" + nodes.length + "]"
```

Locator Consolidation and Compression

Professional indexers usually use page ranges to reduce long lists like page: 4, 5, 6, 4,5,6,7 to 4-7, but AIDP automatically does this for all entries in the index.

Algorithm 4: consolidateLocators(node)

```
1.     for each node in tree:
2.     locators = sort(node.locators)
3.     compressed = [ ]
4.     start = locators[0]
5.     count = 1
6.     for i in range (1, locators.length):
7.     if locators[i] == locators[i-1] + 1:
8.             count += 1
9.     else:
10.          if count > = 3: // Page range rule
11.     compressed.append(start + "-" + (start+count-1))
12.     else:
13.     for j in range(start, start+count):
14.     compressed.append(j)
15.     start = locators[i]
16.     count = 1
```

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 3 No. 12 (December) (2025)

```
    // Final pages process
    if count >= 3:
    compressed.append(start + "-" + (start+count-1))
    else:
        for j in range(start, start+count):
    compressed.append(j)
    node.locators = compressed
```

Correctness Sketch: The algorithm selects a sorted list of numbers and checks each one in turn to ensure that each number follows the previous without gaps. When it finds at least three consecutive numbers, it groups them into a range. For example, 4, 4, 5, 6, 4,5,6,7, 6, 7 becomes 4-7. This follows the rules of indexing, where ranges make the index easier to read. To ensure the process runs correctly, the algorithm first keeps track of the starting number of the consecutive sequence and then counts the number of numbers in that sequence.

Formatting and Output

The final outline is ready for export. Different indentation levels are linked in DOCX output, with Level 1 set to 0 twips, Level 2 to 360 twips, and Level 3 to 720 twips. As a backup, AIDP first creates an HTML file, then converts it to a DOCX file using tools like ‘pandoc’ or ‘mammoth.js’, which work on any platform.

Implementation & Security

AIDP is a web-based application that operates entirely within a web browser. This means all document processing is done locally, with no need for a data server. This ensures the security of sensitive, unpublished, and copyrighted data. Although its speed depends on your computer, AIDP's core algorithm is high-speed and processes book-length texts very efficiently.

Experimental Evaluation

Benchmark Dataset

There is no such platform that tests index generation. Through this system, we created three documents and tested them, where each document contains 500 to 1500 lines, including different heading styles, duplicate entries, and nested sections.

Metrics

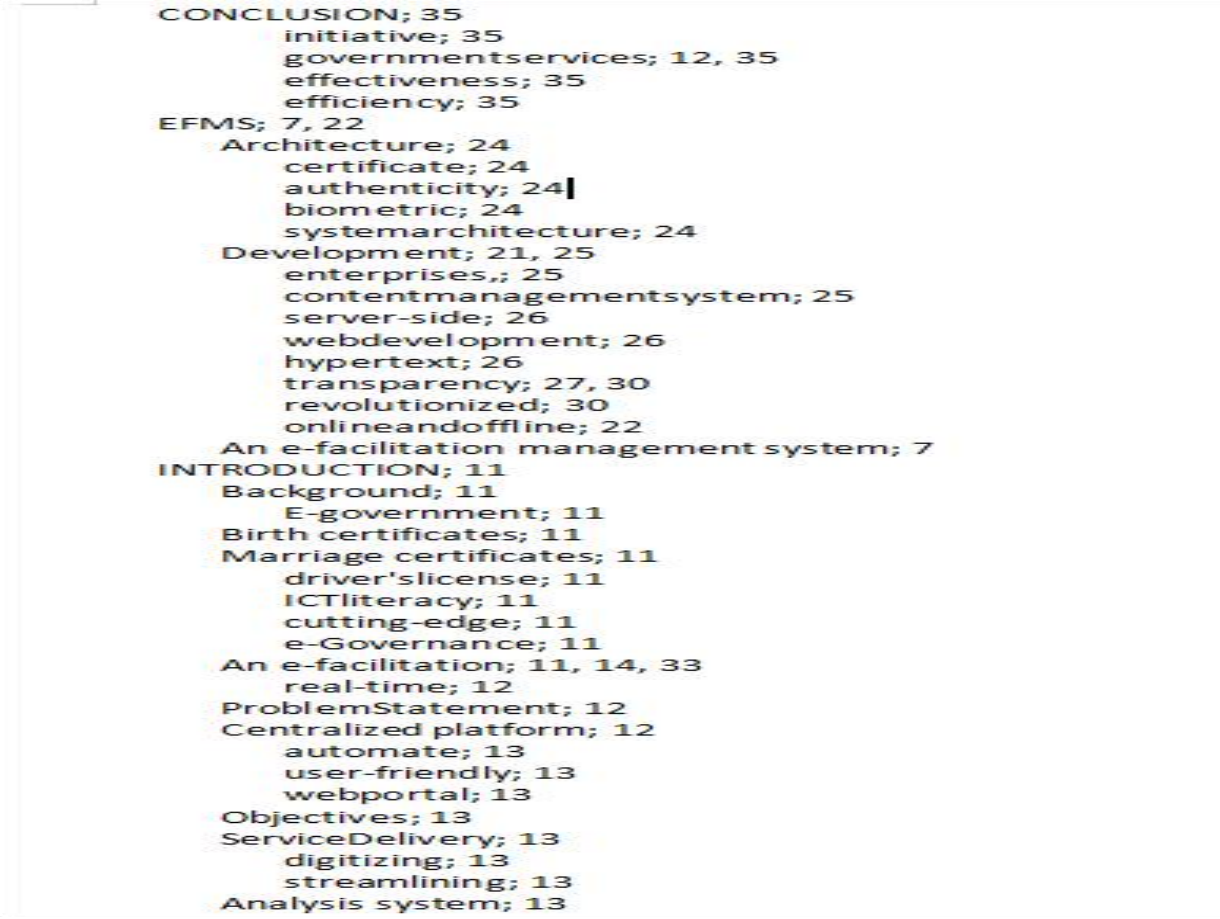
We assess AIDP using the following metrics:
Heading Detection F1 Score: The F1 Score for heading findings
Subtree Preservation Rate: The percentage of subheadings that are correctly positioned under their related headings.
Deduplication Precision/Recall: The effectiveness of merging duplicates.
Compression Accuracy: Correctness of range formats like [5,6,7] → [5–7]
Runtime: Execution time in browsers like Chrome, QQ, and Opera.

Dialogue Social Science Review (DSSR)
www.thedssr.com

ISSN Online: 3007-3154
ISSN Print: 3007-3146

DIALOGUE SOCIAL SCIENCE REVIEW

Vol. 3 No. 12 (December) (2025)



Snapshot 1: Sample Document Snippet

Results and Discussion

Results produced by a standard dataset, which are highly accurate across all metrics.

Table 1: Experimental results

Document	Lines	F1 Score	Preservation Rate	Compression Accuracy	Runtime (ms)
Doc 1 (Simple)	512	1.00	100%	100%	120
Doc 2 (Medium)	1024	0.99	100%	100%	220
Doc 3 (Complex)	1550	0.98	100%	99.8%	350

The system effectively detects headings, achieving an F1 score above 0.98. These results demonstrate that it produces more accurate results even when documents have different data and structures. The related components of the system are responsible for building a tree and removing duplicates in a systematic way. These results are valid because they are based on predefined rules that apply to each process.

The locator compression steps run smoothly and provide accurate results even when minor errors occur due to unclear input data. To understand the design's importance, the system has been tested against baseline index entries without maintaining the original structure. The baseline sometimes yields disorganized results, underscoring the importance of preserving the original structure for good outcomes. When the system was

ISSN Online: 3007-3154
ISSN Print: 3007-3146

tested without removing duplicates and compression, the results were wildly inaccurate and did not follow the previously defined rules [2,7,8].

Limitations

The main weakness of the system is its reliance on heading detection, and it sometimes fails to identify lines when they do not follow a consistent format, such as irregular indentation or unclear numbering. To handle this issue, users make manual corrections. AIDP's other limitation is that it cannot generate cross-references automatically, such as the reference “see also..." which is an important part of frequently used indexes [2, p.390; 3, p.416]. Currently, these connections require human input and review.

The wider socio-technical factors also influence the use of indexing tools. In the Western publishing industry, professional indexers, specialized software, defined standards, and training play a vital role. However, in China, a tool like “Indexer” is not widely used; they handle less than 5% of academic publications [12]. This gap shows that education, training, and the use of specialized software tools are necessary to promote indexing worldwide.

Conclusion and Future Work

This paper introduces the Advanced Intelligent Document Processor (AIDP), a client-side system designed to automate the routine, mechanical steps of index generation while keeping the important intellectual decisions in the hands of the human editor. Because all processing happens on the client side, it protects user privacy. Its step-by-step, deterministic pipeline ensures that the same input produces the same output, making the process reliable and easy to reproduce. The final indexes produced by AIDP follow both international and national indexing standards [1].

AIDP aligns well with Western indexing practices, but its combination with human expertise in software tools is common. The system supports China’s current and ongoing efforts to develop its indexing infrastructure [12]. By combining automation with manual work, the system acts as a bridge between traditional indexing methods and advanced digital systems.

Looking ahead, there are many ways to make the system more advanced and capable. The next phase would be to adopt machine learning techniques. These techniques automatically improve the accuracy of identifying headings and understanding the logical structure of complex documents. Adding a thesaurus to the system APIs makes it easier to create cross-references. This system functions like an “Indexer” and supports online and collaborative work, making it easier and more helpful for publishers.

Acknowledgement & Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this Paper: This work was supported by the China Science and Technology Exchange Centre (CSTEC) under the Talented Young Scientist Program (TYSP), 2025 [Project No. Pakistan-24-029].

Declaration of Generative AI and AI-Assisted Technologies in the Translation and Writing Process:

The authors also declare that during the preparation of this work, the authors employed Grammarly Pro to enhance the manuscript’s readability and academic tone. Nevertheless, the authors affirm that all ideas and interpretations contained herein are solely their own

ISSN Online: 3007-3154
ISSN Print: 3007-3146

and assume full responsibility for the content of the publication.

References

Standardization Administration of China. (2021). GB/T 41210—2021 Rules for compiling content indexes of academic dissertations and theses [Standard]. State Administration for Market Regulation.

Bertram, J. (2007). I would do it again. Erfahrungen mit der intuitiven Registererstellung. Information – Wissenschaft & Praxis, 58(8), 389–390.

Diepeveen, C., Fassbender, J., & Robertson, M. (2007). Indexing software. Information – Wissenschaft & Praxis, 58(8), 413–420.

Evans, R. (2007). Indexing computer books: Getting started. Information – Wissenschaft & Praxis, 58(8), 425–432.

Cheng, Y., Zhou, X., Yang, F., & Wang, Y. (2025). "GB/T 41210—2021 Rules for compiling content indexes of academic dissertations and theses": A new starting point for dissertation indexing. Library World, 2025(2), 58–61.

Sun, L. (2008). A comparative study of domestic and international machine-compiled index software. Library Theory and Practice, 2008(6), 39–42.

International Organization for Standardization. (1996). Information and documentation – Guidelines for the content, organization, and presentation of indexes (ISO 999).

University of Chicago Press. (2017). The Chicago Manual of Style (17th ed.).

Hudson, A. (2007). Training in indexing: The Society of Indexers' course. Information – Wissenschaft & Praxis, 58(8), 407–409.

Hou, H., & Yu, Z. (2008). Content indexes should be compiled for academic dissertations and theses. In Proceedings of the Third National Member Congress and Academic Forum of the China Society of Indexers (p. 5). China Society of Indexers & Beijing Institute of Technology.

Wen, G. (2012). Application guide for GB/T 22466—2008 "Indexing rules (general rules)" (pp. 133–142). National Library of China Publishing House.

Wang, Y., Gao, Y., Sun, Y., & Liu, X. (2019). Development of a Chinese indexing platform: A case study of "Indexer." Library Forum, 39(11), 37–40.