buildings

MDPI

Article

A Study on Methods for Parsing Architectural Multi-Modal Data and Extracting Modeling Parameters

Shimei Li [1,†], Weining Song [2,†], Tan Li [1,*], Nanjiang Chen [1], Liefa Liao [3], Xuejun Zhou [3], Fangfang Gao [1] and Runmin Yin [4]

[1] School of Information Engineering, Nanchang University, Nanchang 330031, China; lishimei@ncu.edu.cn (S.L.); chennanjiang@ncu.edu.cn (N.C.); gaofangfang@ncu.edu.cn (F.G.)
[2] School of Information Engineering, East China University of Technology, Nanchang 330013, China; songweining@ecut.edu.cn
[3] School of Information Engineering, Jiangxi Modern Polytechnic College, Nanchang 330095, China; liaolf@xjtu.edu.cn (L.L.); 032003@jxxdxy.edu.cn (X.Z.)
[4] QuikTech Co., Ltd., Beijing 100080, China; yinrunmin@163.com
* Correspondence: litan@ncu.edu.cn
† These authors contributed equally to this work.

Abstract

To address information isolation and incomplete parameter extraction among multi-modal data (e.g., drawings, text, and tables) in the operation and maintenance stage of buildings, this paper proposes a multi-modal data parsing, automatic parameter extraction, and standardized integration method oriented toward 3D modeling. First, by employing vector element parsing and layer semantic analysis, the method enables structured extraction of key component geometry from architectural drawings and improves modeling accuracy via spatial topological relationship analysis. Second, by combining regular expressions, a domain-specific terminology dictionary, and a BiLSTM-CRF deep learning model, the extraction accuracy of unstructured parameters from architectural texts is significantly improved. Third, a multi-scale sliding window and geometric feature analysis are used to achieve automatic detection and parameter extraction from complex nested tables. Regarding the experimental setup: the drawings consist of a large-scale collection of DXF files stratified and randomly split into train/val/test with an approximate 8:1:1 ratio; the text set includes 1550 PDF-derived specification fragments (8:1:1 split); and the tables cover typical door/window, structural, and electrical schedules (also split ~8:1:1). F1 scores use micro-F1 (instance-level aggregation), and 95% confidence intervals and their computation are described in the main text. Experimental results show that the F1 scores for wall line, wall, and column recognition reach 98.1%, 84.9%, and 92.2%, respectively, while the F1 scores for door and window recognition are 74.3% and 76.2%. For text parameter extraction, the proposed PENet model achieves a precision of 83.56% and a recall of 86.91%. For the table task, the parameter extraction recalls for doors/windows and structure are 95.0% and 96.7%, respectively. The proposed method enables efficient parameter extraction and standardization from multi-modal architectural data, demonstrates significant advantages in handling heterogeneous data and improving modeling efficiency, and provides practical technical support for the digital reconstruction and intelligent management of existing buildings.

Keywords: multi-modal data parsing; automatic parameter extraction; BiLSTM-CRF; deep learning model; PENet model

Academic Editor: Derek Clements-Croome

Received: 19 September 2025
Revised: 29 October 2025
Accepted: 7 November 2025
Published: 10 November 2025

Citation: Li, S.; Song, W.; Li, T.; Chen, N.; Liao, L.; Zhou, X.; Gao, F.; Yin, R. A Study on Methods for Parsing Architectural Multi-Modal Data and Extracting Modeling Parameters. *Buildings* **2025**, *15*, 4048. https://doi.org/10.3390/buildings15224048

1. Introduction

The construction industry accounts for approximately 40% of global energy consumption, making it a major source of both carbon emissions and energy use [1]. In particular, the operation and maintenance phase of buildings is characterized by the highest energy consumption and the greatest management complexity [2]. Effectively reducing energy consumption and improving management efficiency during this stage has become one of the key challenges in driving the digital transformation of the construction industry. In recent years, numerous studies have explored ways to reduce carbon emissions and improve the energy efficiency of buildings, especially in areas such as space planning [3], facility management [4], fault detection and diagnosis [5], as well as energy management and optimization [6]. Various methodologies have been proposed to enhance energy efficiency during the operation and maintenance phase. However, the effective implementation of these methods typically requires extensive support from high-quality building-related data [7]. Specifically, data during the operation and maintenance phase can be categorized into two types: as-built documentation and O&M information. As-built documentation refers to project files generated during the design and construction stages, such as construction drawings and as-built drawings; O&M information refers to dynamic records accumulated during building use, such as energy consumption records and equipment maintenance status [8]. Due to the large scale and dispersed sources of these data, reliance on manual organization and utilization can no longer meet the current demands for high-precision and intelligent management [9].

Building Information Modeling (BIM) has gradually become an effective technological approach to meet the aforementioned needs [10]. BIM enables the digital representation of both the physical characteristics and functional attributes of buildings, serving as an information database that stores detailed parameter information about the building. In practice, BIM models have been shown to improve the quality of operation and maintenance management [11]. They also facilitate the integrated management of massive and discrete building information, enabling effective information transfer throughout all stages of a project, from design to operation and maintenance. BIM further supports collaborative work among various disciplines during the operation and maintenance phase, providing owners and managers with more accurate and comprehensive information to assist in rapid decision-making.

During the 3D modeling process, multi-modal data such as drawings, text, and tables serve as key sources of parameters for modeling. Drawings primarily convey the geometric forms and spatial relationships of building components, and the quality of their interpretation directly affects the completeness and accuracy of the model. Although various 3D modeling methods currently exist, significant challenges remain when dealing with legacy building documentation. Firstly, differences in design standards and drafting conventions across different historical periods lead to diverse and complex 2D drawing representations. Traditional rule-based approaches have limited adaptability and generalization capability when faced with such diversity [12]. In addition, information for existing building projects is often stored in a scattered manner across multi-modal formats, such as 2D drawings, textual descriptions, and tables. This results in data heterogeneity, missing parameter information, and significant challenges for automated processing. Therefore, there is an urgent need for efficient data parsing, automatic parameter extraction, and standardized integration technologies to provide reliable data support for 3D BIM model reconstruction.

Early drawing parsing methods were primarily knowledge-driven, relying on layer information, geometric features, and rule-based libraries to achieve component recognition. These approaches perform well on standardized drawings, but their adaptability to

non-standard drawings and resistance to noise interference are relatively weak [13–15]. In recent years, data-driven methods such as deep learning have emerged. For example, object detection algorithms based on YOLO [16] can improve the accuracy of component recognition in complex scenarios. However, such methods depend on large-scale annotated datasets and have limited capabilities in spatial reasoning and semantic completion [17,18].

In addition to drawings, the construction industry also involves a large amount of normative documents, design specifications, and other textual materials. These are mostly semi-structured or unstructured data, containing rich key information such as component attributes, materials, and functions. For automatic extraction of textual parameters, Named Entity Recognition (NER) technology has been widely applied [19–21]. Approaches have evolved from rule-based and dictionary-based methods [22] and statistical learning methods [23] to deep learning methods such as CNN, BiLSTM, and BERT [24–26]. While these methods have achieved good results on standardized texts, their generalization ability and accuracy for entity recognition still need improvement when dealing with as-built documents in the construction industry, which are often freely expressed and lack uniform formatting [27,28].

To enable "Beyond 3D" multi-dimensional BIM integration, our multimodal parameter extraction maps into 4D (schedule), 5D (cost), and 6D (energy/O&M) via a unified model dictionary: drawings provide geometry and topology, text contributes materials, performance, and methods, and tables supply identifiers, specifications, and quantities [29,30]. This integration pathway underpins lifecycle digital management for smart and sustainable buildings.

To align with prevailing lines of multimodal BIM extraction, we offer only a principle-level note without enumerating specific systems: across drawing parsing, text NER, table recognition, and cross-modal fusion, prior approaches span rule-based to learning-based designs and, in some cases, multi-strategy hybrids [31,32]. Our setup follows this trajectory while placing more emphasis on the interplay between hybrid geometric reconstruction and a PENet-style multi-strategy fusion, with pragmatic constraints for reproducibility (e.g., tunable thresholds and implementation notes). Concrete exemplars and quantitative specifics are intentionally omitted here.

2. Materials and Methods

2.1. Information Parsing and Data Extraction from Architectural Drawings

With the widespread adoption of CAD technology, digital storage of architectural drawings has become common. The DXF format, valued for its openness and compatibility, is now a mainstream standard for exchanging and storing drawings. DXF files store graphic entities in a clearly structured text format, enabling easy parsing across platforms. As a result, automatic parsing of formats like DXF is essential for converting 2D data into 3D models. This transformation typically involves four key steps: drawing preprocessing, component information extraction, spatial topology reconstruction, and parameter standardization, as shown in Figure 1. These steps rely on unified geometric rules and threshold settings—such as the pose restoration order (scale → rotation → translation) and angle conventions (θ in radians, positive for counterclockwise)—to ensure robust, transferable parsing across scales (Table 1).

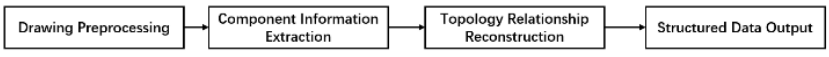

Figure 1. Drawing processing process.

Table 1. Summary of thresholds and geometric rules (for DXF drawing processing).

Category	Item	Rule/Value
General settings	Angle unit	Radians
General settings	Transformation order	Scale → Rotation → Translation
Scale examples	1:100	Endpoint merge threshold = 1 mm; Snapping tolerance = 1 mm
Scale examples	1:200	Endpoint merge threshold = 0.5 mm; Snapping tolerance = 0.5 mm
Guidelines	Tolerances & angles	Endpoint/snapping tolerance ≈ 0.5–1× line width or minimum feature size
Line/Polyline	Endpoint merge	Merge if Euclidean distance ≤ merge threshold
Line/Polyline	Snapping	Snap endpoint to line/circle/arc if shortest distance ≤ snapping tolerance
Line/Polyline	Collinearity	Direction angle ≤ collinearity angle threshold and max deviation to reference line ≤ collinearity distance threshold
Line/Polyline	Parallelism	Direction angle ≤ parallelism angle threshold (overlap not required)
Line/Polyline	De-dup & stitching	If collinear and endpoints touch or gap ≤ merge threshold, stitch; for duplicates, keep the longer segment or higher-priority layer
Circle/Arc	Arc-to-line approximation	If arc length ≤ merge threshold or radius ≫ feature scale, approximate as a line segment
Circle/Arc	Circle merge	Merge if center distance ≤ merge threshold and radius difference ≤ merge threshold
Transform & registration	Scale	Normalize to real-world size using drawing scale (prefer model space)
Transform & registration	Rotation	Rotate by θ (radians) about origin or base point; θ > 0 for CCW
Transform & registration	Translation	Apply (dx, dy); local block references first, then global registration
Layering & semantics	Layer priority	Structural > Architectural > Annotation
Layering & semantics	Semantic filtering	Keep door/window/wall layers; drop dimension/cutline layers

Drawing preprocessing addresses data redundancy, misalignment, and inconsistent layer naming. Floor boundaries are extracted by detecting the drawing frame, and each floor's data are stored separately. Grid and column lines serve as feature references to correct translation and rotation, and matching identically named points aligns multi-floor drawings into a unified global coordinate system. Vector cleanup removes zero-length segments and isolated points to reduce noise in component recognition. To reconcile heterogeneous layer conventions, a mapping table (Table 2) links diverse layer names to their corresponding component types, improving parsing consistency.

Table 2. Example of standardized mapping of layer naming (excerpt).

Non-Standard Layer Name	Normalized Standard Layer
A-WALL	WALLS
WALL_1	WALLS
DOOR_LAYER	DOORS
COLUMN_MAIN	COLUMNS
WINDOW_1	WINDOWS

After completing drawing preprocessing and layer standardization, the next step is to parse and extract detailed information of architectural components. In the structure of a DXF file, the BLOCKS section and the ENTITIES section serve different roles in describing component information. The BLOCKS section stores standardized and reusable components, whose core advantages lie in their global definitions, parametric controls, and nested referencing capabilities. This allows complex components to be reused without repetitive drawing, thereby improving the efficiency and consistency of the drawings.

However, due to these characteristics, the spatial positions of components are not directly stored; instead, they are determined by parameters such as insertion point coordinates, rotation angles, and scaling factors. Therefore, the key to parsing BLOCKS is the computation and interpretation of these parameters. If a BLOCK component is rotated in the drawing, its spatial position must be restored by applying a rotation matrix transformation to process the coordinates. This transformation process is shown in Equation (1).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

where θ is the rotation angle, (x′, y′) are the coordinates after rotation, and (x, y) are the original coordinates. This matrix transformation ensures that the spatial position of BLOCK components in the drawing can be accurately restored. To provide a more intuitive illustration of the BLOCK parsing method, Figure 2 presents a sample DXF code snippet alongside the corresponding JSON file generated after parsing.

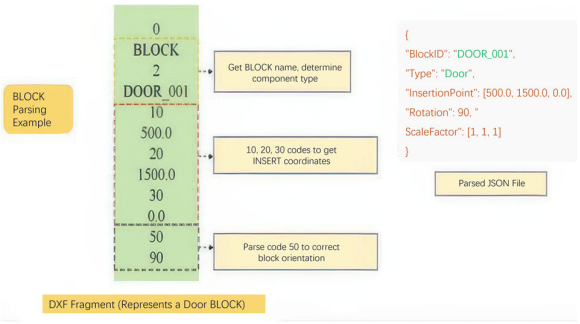

Figure 2. Block parsing example.

The ENTITIES section focuses on recording non-reusable components in the drawing, which are diverse in form and require identification based on layer names, geometric features, and spatial relationships. Nested components such as doors and windows are often stored in the BLOCKS section; however, their spatial positions and opening directions need to be further inferred in conjunction with wall information. Component topology reconstruction is a key step to ensure the integrity and accuracy of automated modeling. Relying solely on geometric information from drawing data is insufficient to accurately restore the spatial positions and associations of components—especially since the attachment relationships among walls, doors, windows, slabs, and other elements are crucial for the completeness of the model's appearance. Therefore, this paper proposes corresponding methods that leverage both spatial distribution features and geometric characteristics to achieve precise extraction of components and their relationships.

In this section, where applicable, authors are required to disclose details of how generative artificial intelligence (GenAI) has been used in this paper (e.g., to generate text, data, or graphics, or to assist in study design, data collection, analysis, or interpretation). The use of GenAI for superficial text editing (e.g., grammar, spelling, punctuation, and formatting) does not need to be declared.

(1) Wall Topology Reconstruction

In drawing data, walls are often represented by multiple line segments or straight lines. However, drawing errors and endpoint offsets may occur, leading to wall discontinuities and chaotic spatial relationships. To address this issue, this paper proposes a strategy that utilizes KDTree clustering analysis and buffer-based methods to repair wall boundary breaks, restore spatial connectivity of walls, and extract wall centerlines. In drawing data, walls are often represented by multiple line segments or straight lines. However, drawing errors and endpoint offsets may occur, leading to

wall discontinuities and chaotic spatial relationships. To address this issue, this paper proposes a strategy that utilizes KDTree clustering analysis and buffer-based methods to repair wall boundary breaks, restore spatial connectivity of walls, and extract wall centerlines. When multiple endpoints have similar distances, we resolve ambiguity via a composite scoring that prioritizes same layer/linetype/thickness, orientation continuity (angle < δ°), and colinearity (overlap ratio > ρ). To prevent spurious merges between adjacent walls, we enforce wall-thickness crossing prohibitions and intersection-direction constraints. A threshold sensitivity analysis was conducted, plotting precision/recall against merge tolerance across scales.

① Wall Boundary Repair
In architectural engineering drawings, wall boundaries often exhibit minor gaps between endpoints due to drafting errors, resulting in discontinuities. To identify and correct such issues, this paper employs Euclidean distance to determine whether walls are closed, as shown in Equation (2). When the distance between two endpoints is less than 1 mm, they are considered closed and are automatically connected; otherwise, adjacent endpoints are merged to repair the break. For straight wall segments, the KDTree algorithm is further utilized for endpoint clustering analysis, where nearest neighbor matching is used to identify endpoint pairs that need to be merged. As shown in Equation (3), if the minimum distance is below a predefined threshold, the merging operation is performed.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2}$$

$$\min_{p_i \in m, p_j \in n} \left\| p_i - p_j \right\| \tag{3}$$

② Reconstruction of Wall Spatial Connectivity
In addition to boundary repair, the overall spatial connectivity of walls is equally important. In this paper, wall connectivity reconstruction is accomplished using the KDTree method. Considering the differences in precision across various drawings, a dynamic threshold mechanism is introduced. For example, in drawings with a scale of 1:100, the merging threshold is set to 1 mm, while for drawings with a scale of 1:200, the threshold is set to 0.5 mm.

③ Extraction of Wall Centerlines
To ensure precise spatial positioning of walls in the 3D model, the buffer method is employed to extract the centerlines of walls, which serve as the basis for locating components such as doors, windows, and slabs.

(2) Reconstruction of Column Component Topology
As critical load-bearing elements in buildings, columns are typically positioned stably, often located within slabs or at wall intersections. The geometric features of columns are relatively regular. For circular columns, the center coordinates are directly extracted to ensure spatial accuracy. For columns represented by polylines, vertex coordinates are extracted and the Polar Angle Sorting method is employed to ensure the contour is closed. Additionally, the vertices must be arranged in a counterclockwise order to form a complete closed polygon, as illustrated in Figure 3.

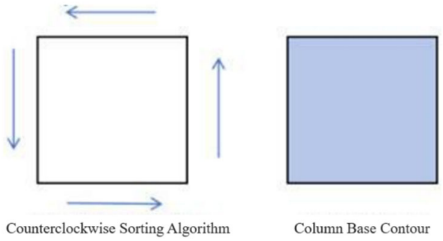

Figure 3. Column members.

In addition, when complex columns are represented as block references, it is necessary to recursively parse their sub-elements and perform merging to reconstruct the complete column geometry. The contour data at the bottom of the slab serves as the basis for determining whether a column is located within the slab, thereby ensuring both the spatial position and structural rationality of the column component.

(3) Reconstruction of Slab Topological Relationships

As a crucial component of each floor in building structures, slabs are typically described by their outer contours, inner ring areas, and wall boundaries. During the reconstruction of slab topology, the bottom contours of peripheral components such as walls, columns, and doors/windows are extracted as important references for defining slab boundaries. Using spatial projection analysis, the spatial relationship between slab contours and wall centerlines is evaluated to determine the degree of alignment between slabs and walls, thereby ensuring the rationality of boundary definitions. For contour merging, the unary_union method from the Shapely library in Python v3.10 is employed to combine all contour lines, ensuring the completeness of the slab's outer contour, as illustrated in Figure 4.

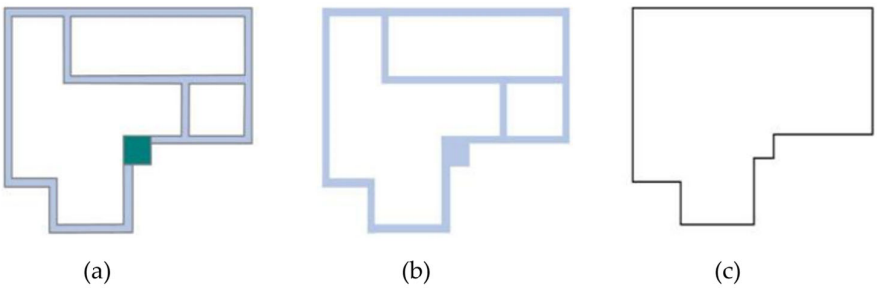

Figure 4. Floor slab components: (**a**) Bottom contours of external structures such as walls and columns; (**b**) Polygon formed by merging the bottom contours of external structures; (**c**) Extraction of the outer boundary of the merged polygon to define the slab area.

(4) Reconstruction of Roof Topological Relationships

The topological reconstruction of roof components is based on the outermost boundary contour as the key feature. The specific approach is as follows: First, all line and polyline data in the roof layer are extracted and combined with their spatial distribution characteristics, elements that conform to roof contour features are selected. Next, the Left Turn Algorithm is used to identify the roof boundary and form a complete polygon. Finally, geometric operations are applied to extract the outer contour of the roof, thereby avoiding contour deviation issues caused by concave structures or irregular shapes, as illustrated in Figure 5.

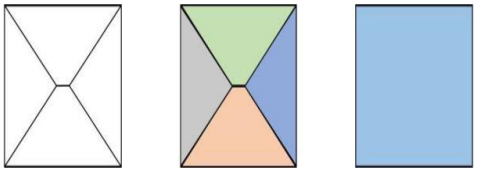

Figure 5. Roof elements.

(5) Reconstruction of Door and Window Topological Relationships
As components embedded within walls, the key to reconstructing the topology of doors and windows lies in identifying the "opening" positions within wall boundaries, as illustrated in Figure 6. The specific method is as follows: First, extract the coordinates of the inserted blocks; second, determine the nesting positions of doors and windows by referencing the wall centerlines and extract the "equivalent lines"; finally, use the projection coordinate system to determine the precise positions of doors and windows as well as their opening directions, thereby ensuring the rational nesting relationship between doors/windows and walls.

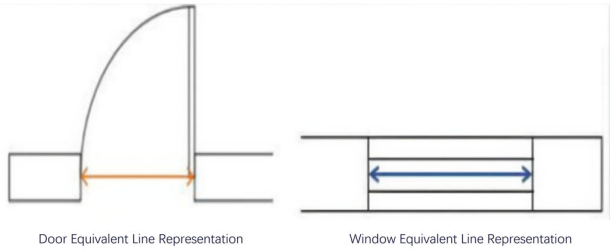

Figure 6. Door and window components.

2.2. Parsing of Architectural Text Information and Parameter Extraction

Architectural text is an important source of non-geometric information in building information modeling (BIM), and plays a crucial role in supplementing construction drawing information. This paper takes PDF-format text as an example to introduce information extraction methods. For standard PDF files, the Python library pdfplumber is used to analyze underlying layout information, identify content, and reconstruct the reading order. For scanned PDF files, OCR technology is introduced: the pdf2image library is used to convert PDFs into images, and the Tesseract engine is employed for text recognition. During image preprocessing, the Canny edge detection algorithm is applied (including steps such as smoothing and gradient calculation, with Gaussian filtering used for noise reduction in the smoothing stage) to improve the accuracy of text recognition. In the image smoothing stage, a Gaussian filter is used to reduce noise interference, and its calculation formula is shown in Equation (4).

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

In the gradient calculation stage, the Sobel operator is used to extract the gradient values of the image in both the horizontal and vertical directions. Based on these gradients, the gradient magnitude map and direction map of the image are constructed. The corresponding calculation formulas are shown in Equation (5).

$$G = \sqrt{G_x^2 + G_y^2}, \theta = arctan\theta \frac{G_y}{G_x} \tag{5}$$

After converting the format of architectural text, issues such as interference from special symbols in the original content, OCR misrecognition, formatting inconsistencies, non-standardized terminology, and confusion in numerical units may still persist. First, irrel-

evant symbols such as ◆, ■, 【】, etc., that appear in the text are detected and removed using a character set comparison method to reduce noise interference, as shown in Table 3. Second, during the OCR process—particularly in scanned documents such as drawings and engineering specifications—the following types of misrecognition may occur:

(1) Confusion between numbers and letters, such as the digit "0" being misrecognized as the letter "O", or the Arabic numeral "1" being misrecognized as the uppercase letter "I";
(2) Confusion between full-width and half-width symbols, such as "(" being misrecognized as "(".

Table 3. Example of clearing special characters in architectural text.

Original Text	Cleaned Text	Processing Method
◆ This project uses C30 concrete\nframe structure • the exterior wall uses stone dry-hanging process	This project uses C30 concrete frame structure, the exterior wall uses stone dry-hanging process	Removal of irrelevant characters (◆, •); removal of line breaks
【Note】 This project mainly adopts brick-concrete+\nprecast frame, \nwith stable structure	This project mainly adopts brick-concrete + precast frame, with stable structure	Removal of special symbols (【】); format standardization

To address the above issues, this paper introduces dictionary matching based on the edit distance algorithm and regular expression methods to correct errors in optical character recognition (OCR) results. The edit operations include insertion, deletion, and substitution, with the calculation formula given in Equation (6).

$$d(i,j) = \begin{cases} max(i,j) \quad if\ min(i,j) = 0 \\ \begin{cases} d(i-1,j)+1 \quad otherwise \\ d(i,j-1)+1 \quad otherwise \\ d(i-1,j-1)+1 \quad (s1[i] \neq s2[j]) \end{cases} \end{cases} \tag{6}$$

Here, d(i, j) denotes the edit distance between the first ii characters of string s1 and the first j characters of string s2. When s1[i] = s2[j] [33], no character substitution is required, and the edit distance inherits from the previous state. If the two characters differ, the operation with the minimal cost is selected from among insertion, deletion, and substitution.

On this basis, to address the issue of non-standardized terminology in architectural texts, this paper constructs a terminology normalization dictionary with reference to the "Code for Fire Protection Design of Buildings", the industry foundation class (IFC) standard, and the "Standard for Terminology Used in Building Engineering" [34–36], as shown in Table 4. The edit distance algorithm is then employed to compare each word in the text with the terms in the dictionary. If the edit distance between a word and a dictionary term is less than or equal to a predefined threshold of 2, it is identified as a potential spelling error. Subsequently, the dictionary term with the smallest edit distance is selected as the correct replacement for the word.

Table 4. Standardized mapping of terminology in the construction field(excerpt).

Original Term	Standardized Term
Frame-shear wall structure	Frame-shear structure
Shear wall structure	Shear wall system
Brick-concrete structure	Brick-concrete system

Finally, the re.sub() function is used to remove redundant spaces, line breaks, and invalid characters, ensuring that the text structure is concise and the parameter information is complete, as shown in Table 5.

Table 5. Examples of architectural text format optimization.

Original Term	Standardized Term	Processing Method
The construction method of this project is cast-in-place cast-in-place, and all beams and columns use H-shaped steel	The construction method of this project is cast-in-place, and all beams and columns use H-shaped steel	Removal of duplicate words
This project adopts a precast concrete frame\n\nstructure, and the construction method is\ncast-in-place system	This project adopts a precast concrete frame structure, and the construction method is cast-in-place system	Merging and removing redundant spaces and line breaks

Although text cleaning removes noisy data, parameter information related to building components still remains unstructured. To convert this into structured data suitable for modeling, this paper applies Named Entity Recognition (NER). As a sequence labeling task, NER assigns predefined entity categories to tokens and extracts key parameters. Considering the characteristics of architectural text and automated modeling needs, nine parameter entity categories (e.g., geometric dimensions, numeric attributes) are defined. During annotation, the BIO scheme is adopted, marking the beginning (B-), inside (I-), and outside (O) segments to distinguish parameter spans from ordinary text. The YEDDA annotation tool is employed to improve both accuracy and efficiency. After annotation, a processing script maps each token to its label, producing standardized, high-quality annotated data, as illustrated in Figure 7.

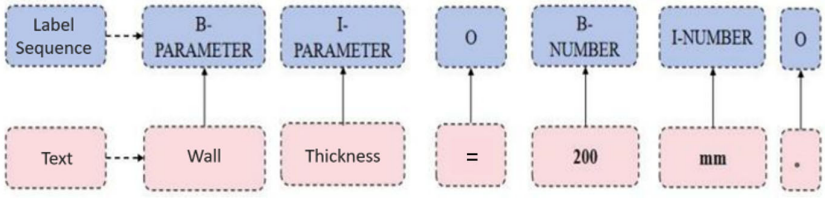

Figure 7. A detailed example of entity labels in the text.

To address the issues of diverse parameter expressions, OCR (Optical Character Recognition) misrecognition, and numerical unit confusion in architectural text data, this paper proposes a parameter extraction model based on a multi-strategy fusion of BiLSTM-CRF, named PENet (Parameter Extraction Network). PENet is designed to automatically extract information such as building components, materials, floors, and parameter values. The overall framework of the model is shown in Figure 8. The core PENet consists of regular expressions, a domain terminology dictionary, and a BiLSTM-CRF. For positioning and robustness evaluation, we explored BERT feature initialization and several variants (e.g., BERT-CRF, BERT-only, BiLSTM-CRF-only) as comparative studies, without changing the definition of PENet or the source of the main results.

Catalogue Name	Door/Window Model Number	Opening Size	Quantity of Doors/Windows	Notes
		Width × Height		
L.J2009 CSZH08 JK365 3618 6X2	C3618	3600×1800	1	Single-frame double-glazed plastic-steel window
L.J2009 CSZH2 JK365 3621 6X2	C3621	3600×2100	14	Single-frame double-glazed plastic-steel window
L.J2009	C3321	3300×2100	5	Single-frame double-glazed plastic-steel window
L.J2009 CSP97 JK365 1521 6X2	C1521	1500×2100	1	Single-frame double-glazed plastic-steel window
L.J2009 CSP28 JK365 1212 6X2	C1212	1200×1200	1	Single-frame double-glazed plastic-steel window
	M3633	3600×3300	2	Steel-wood insulated door with small door
	M3633A	3600×3300	1	Matte white steel door
	M3033	3000×3300	1	Steel-wood insulated door with small door
	M1027	1000×2700	4	Finished solid wood door
	M0927	900×2700	3	Finished solid wood door

Figure 8. The overall framework of the PENet model.

Before input to the model, architectural terms are normalized with a domain dictionary and regular expressions, and numerical parameters are pre cleaned. BERT then generates dynamic character embeddings, which the BiLSTM processes to fuse bidirectional context and derive deep semantic features, producing label score vectors. A subsequent CRF layer performs global optimization over these scores and outputs the final entity label sequence, completing the recognition task.

2.3. Parsing and Parameter Extraction of Architectural Tabular Information

Compared with drawings and narrative text, tables possess clearer structure and allow more efficient data extraction. Excel (v2019)/CSV files are highly structured with standardized parameter fields, whereas some parameters embedded as images in PDFs require OCR, risking recognition errors. In practice, door & window schedules and construction specification tables are common sources: the former underpins extraction of modeling parameters (see Figure 9), while the latter details construction methods. Despite evident structural traits, efficient parameter extraction is still hindered by scattered parameter placement, irregular numbering, and inconsistent measurement units.

Architectural construction drawings contain a large number of complex nested tables, which often exist as discrete combinations of text and line segments. These tables may also exhibit complex features such as merged cells, uneven row and column spans, and broken boundaries. Traditional text parsing methods struggle to effectively extract the relevant structural information. Taking the door and window schedule as a specific example, this paper proposes a table extraction method based on geometric feature analysis and extends it to parameter extraction tasks for other complex nested tables. The core process of the method includes the following three steps:

(1) Table Region Selection

By exploiting distinctive CAD layer attributes, candidate table regions can be rapidly localized. Layer-based filtering removes non-tabular graphics early, shrinking the search space and boosting selection efficiency. To automatically detect table regions in complex drawings, a multi-scale sliding-window region detection strategy is adopted. Inspired by the hierarchical multi-scale search mechanism of Nguyen et al. [37], multiple window sizes are systematically swept across the drawing. Each windowed region is evaluated and pruned using a composite score derived from its geometric, structural, and textual feature indicators to decide whether it constitutes a probable table area. The scoring computation is formalized in Equation (7). The weights ωi in Equation (7) are optimized via grid search on the validation set, with features including line density, text density, long-line ratio, rectangular structural consistency, etc. In the absence of labels, we use heuristic weights (e.g., ωline = 0.35, ωtext = 0.35, ωrect = 0.20, ωconsistency = 0.10) and provide sensitivity and robustness analyses.

$$S(x, y, s) = \sum_{i=1}^{n} \omega_i \cdot f_i(x, y, s) \quad (7)$$

where S(x, y, s) denotes the score of the window at position (x, y) with scale s; fi(x, y, s) represents the value of the i-th feature within the window region (such as line segment density, text density, etc.); wi is the weight parameter for the ii-th feature; and n is the number of features. Given that some door and window schedules have irregular structures or may be partially occluded, this paper incorporates manual assisted verification based on automatic selection, thereby enhancing the reliability of target region extraction.

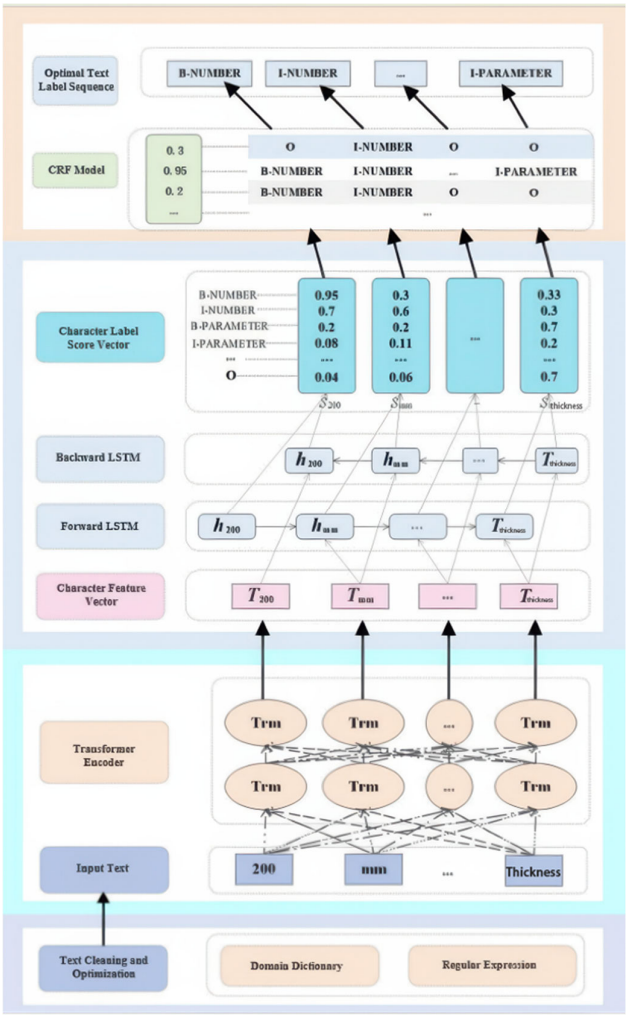

Figure 9. Workflow for table detection and structuring.

(2) Table Structure Reconstruction

Table structure reconstruction refers to extracting the boundary of each cell within the table to achieve row and column segmentation. The specific process is as follows:

① Grouping and Sorting of Parallel Lines

Table elements in the drawing are grouped according to parallel lines. Using the coordinate information from CAD geometric features, the set of line segments is sorted based on their starting coordinates. Parallel line groups in the horizontal direction represent the rows of the table, while those in the vertical direction correspond to the columns.

② Calculation of Table Intersection Points

Based on the grouped sets of parallel lines, the intersection points between horizontal and vertical line segments are calculated sequentially, and the coordinates of each intersection point are recorded. To ensure the correct ordering

of intersection points, the following sorting rules are applied:
Intersection points on horizontal lines are sorted in ascending order of their X-coordinate values, while intersection points on vertical lines are sorted in descending order of their Y-coordinate values.

③ Calculation of Cell Vertex Coordinates
Based on the sorted intersection points, the vertices of each cell are indexed as Pij, and the vertex coordinates are calculated using the following recursive formula (Equation (8)):

$$\begin{cases} x_{ij} = x_{i,j} \\ y_{ij} = y_{i,j} \end{cases} \quad (8)$$

(3) Table Content Recognition
After determining the vertex coordinates of each cell, the boundary of the text information within each cell can be further identified. Specifically, the position of the text object AijAij in the table can be represented by the following coordinate range, as shown in Equation (9).

$$\begin{cases} xi, j_{min} \\ xi + 1, j_{max} \\ yi, j_{min} \\ yi, j + 1_{max}\{ \end{cases} \quad (9)$$

The vertex coordinates of each cell are traversed, and the textual information within their boundaries is extracted. The fields in the table header are used as key parameters for the door and window attribute table and are stored as key-value pairs in a JSON-formatted file. The parameter information corresponding to each column is recorded as attribute data within this JSON structure.

3. Experiments and Analysis

3.1. Experimental Data Sources and Preprocessing

The experimental dataset originates from the architectural design documentation of a large mixed-use building complex in a given region. It integrates multimodal sources—architectural drawings, design specifications, and component (e.g., door, window, structural member) schedules—forming a comprehensive data foundation for transforming 2D design intent into 3D parametric models. Architectural drawings constitute the core layer and include, for example, floor plans and related view types, from which spatial positions, geometric dimensions, elevation relationships, and other component parameters can be extracted. Textual design specifications record functional zoning, material descriptions, performance requirements, and construction methods, providing semantic cues for component classification and for the formulation of modeling rules. Component schedules and quantity tables, expressed in structured tabular form, enumerate identifiers, types, dimensions, material attributes, and sometimes performance indices; these support parameter completion, model semantic enrichment, and subsequent performance analyses. During experimentation, the architectural drawing subset undergoes preprocessing—tiling (to handle large-format sheets), registration/alignment (to reconcile coordinate inconsistencies), and layer standardization (harmonizing heterogeneous layer naming conventions)—following the workflow defined in Section 2.1, ensuring the reliability and consistency of the structured data extracted downstream.

To enhance transparency while avoiding dataset-specific numerical disclosures, we further describe the dataset composition, annotation protocol, and geometry metrics used for evaluation. The multimodal dataset consists of architectural drawings (primarily DXF), native and scanned PDFs, and structured schedules covering typical door/window, struc-

tural, and electrical categories. Text entities are annotated under a BIO scheme using a standardized tooling workflow (e.g., YEDDA), with inter-annotator agreement assessed on overlapped samples using κ-based statistics (Cohen's or Fleiss' κ) to ensure label consistency. For drawings, at the component level we use contour IoU, mean positional error (in millimeters), and endpoint RMSE for internal quality checks; to avoid dispersing the main metric focus, we do not tabulate these in the paper. Where necessary, we provide representative descriptions, with stratification by drawing scale (1:50/1:100/1:200) and layer quality (standard/non-standard/mixed). Tables are evaluated by category, with automatic detection and parameter extraction reported primarily in terms of Precision/Recall/F1, aligned to the same micro-averaged F1 protocol. This specification ensures that modality-specific results are comparable, robust to heterogeneous sources (e.g., mixed scales, varying layer conventions, OCR noise), and reproducible under the train/validation/test splits described in the main text. We compute 95% confidence intervals for micro-F1/Accuracy using an instance-level binomial approximation as $p \pm 1.96 \cdot \mathrm{sqrt}(p(1-p)/N)$, where N is the number of instances, and report representative intervals alongside key results.

3.2. Parameter Extraction and Experimental Analysis of the Data

3.2.1. Extraction of Drawing Component Parameters and Evaluation of Recognition

In this study, the parsing of architectural drawing data is accomplished through vector graphic element analysis and layer semantic analysis. By extracting the geometric information of components such as walls, doors and windows, columns, and roofs, the effectiveness of component extraction is statistically analyzed. The recognition statistics for each component are shown in Table 6. To align with detection metrics, we report Recall in the main text and additionally provide Accuracy to remain consistent with the abstract. The roof sample size is small (30 cases) with large symbol variation; the current result serves as a feasibility check and is not emphasized in the overall conclusions.

Table 6. Statistics on the accuracy of component identification in architectural drawings.

Component Type	Actual Number of Components	Detected Number of Components	Correctly Identified Components	Recall	Precision	Score
Wall Lines	1000	1020	991	99.1%	97.2%	98.1%
Walls	500	560	450	90.0%	80.4%	84.9%
Doors	350	430	290	82.9%	67.4%	74.3%
Windows	280	350	240	85.7%	68.6%	76.2%
Columns	200	210	189	94.5%	90.0%	92.2%
Roofs	30	25	25	83.3%	50.0%	62.5%

Results show high recognition of wall lines, walls, and columns, for example, wall F1 = 84.9% (95% CI: 82.7–86.9%), confirming the effectiveness of the combined vector element and layer semantic analysis for primary components, whereas doors, windows, and roofs perform worse due to geometric complexity and nested door–window relationships; supplementing missing or ambiguous attributes with textual specifications and tabular schedules mitigates drawing-only limitations and improves complex component accuracy.

3.2.2. Recognition of Architectural Text Parameters and Comparative Model Analysis

After preprocessing and structuring architectural semantic texts, we perform NER on core component modeling parameters using 1550 PDF-derived specification fragments (comprehensive complex project) annotated with YEDDA (terminology normalized) and split 8:1:1 (train/val/test), employing BIO → CoNLL encoding; a PyTorch v2.0.1

BERT + BiLSTM + CRF model (max length 128, batch 32, epochs 18, Adam lr 5×10^{-5}, dropout 0.5 before/after BiLSTM, CRF output = 9 entity types) models contextual dependencies and globally decodes labels, with performance evaluated on the test set (parameters in Table 7).

Table 7. Model training hyperparameter settings.

Hyperparameter Name	Parameter Value
SeqLength	128
BatchSize	32
Epochs	18
LearningRate	5×10^{-5}
Dropout	0.5

Using a unified model architecture, hyperparameters, and annotation scheme, we ran four comparative experiments to evaluate how different strategy combinations affect parameter entity recognition in architectural semantic texts. Precision and Recall (Table 8) show that the multi-strategy model combining dictionary and rule-based methods adapts well to complex completion text semantics, supplying more accurate data for modeling parameter extraction.

Table 8. Comparison of text parameter extraction performance.

Model	Precision	Recall
BiLSTM	78.68%	83.60%
BiLSTM-CRF	78.65%	84.38%
BiLSTM-CRF + Dic	82.14%	85.72%
PENet	83.56%	86.91%

3.2.3. Analysis of Table Parameter Extraction and Recognition Performance

In this study, the recognized cell vertex coordinates are used to extract the textual content within each cell. Specifically, text data is extracted based on coordinate intervals and then saved in JSON format files. During the experiments, attribute information was extracted and evaluated for different types of tables, as shown in Table 9.

Table 9. Differences in recognition accuracy and extraction effect among different table types.

Table Type	Actual Number of Parameters	Extracted Number of Parameters	Correctly Identified Parameters	Recall
Door & Window Table	200	220	190	95.0%
Structure Table	150	160	145	96.7%
Electrical Table	100	110	90	90.0%
Table Type	Actual Number of Parameters	Extracted Number of Parameters	Correctly Identified Parameters	Recall
Door & Window Table	200	220	190	95.0%
Structure Table	150	160	145	96.7%

The experimental results show that the recognition accuracy for door and window tables as well as structure tables is relatively high, indicating that this method performs well in processing typical tables in architectural drawings. However, for non-standardized tables such as electrical tables, which feature more flexible text arrangements and more complex cell nesting, the recognition performance is relatively limited, and some information loss still exists. Dominant errors arise from door/window nesting and opening inference, OCR misrecognition of numerals/units in scans, parsing of non-standard merged cells, and

spurious wall merges due to noisy endpoints. Mitigations include stronger topological constraints and disambiguation, layout-understanding models for tables, unit/terminology normalization, and weakly supervised data expansion. Cross-modal integration uses a unified schema (IFC-compatible): geometry from drawings receives GUIDs; text entities and table entries align via IDs/indexes and attach as attribute dictionaries to geometric instances. Attribute conflicts are resolved by source priority (tables > specification text > drawing annotations) and confidence weighting, outputting standardized JSON/IFC for BIM import.

4. Conclusions

To address the bottlenecks of insufficient fusion mechanisms and incomplete component information during the parameter extraction stage of multimodal architectural data, this paper proposes a multimodal feature extraction and parameter standardization method oriented toward 3D modeling. Through vector graphic parsing and layer semantic analysis, the structured extraction of geometric and spatial information from architectural drawings is achieved. By integrating regular expressions, domain-specific dictionaries, and the BiLSTM-CRF model, the precision of automated semantic parameter extraction from textual data is improved. A multi-scale window search and structural reconstruction strategy is employed to enhance the recognition and structuring of tabular attribute information. Experimental results demonstrate that this method can efficiently integrate heterogeneous data from drawings, texts, and tables, enabling automated and standardized extraction of modeling parameters. This provides a feasible technical pathway for the digital reconstruction of existing buildings and lays a solid data foundation for intelligent management and digital transformation in the construction industry.

The proposed multimodal automatic parameter extraction and fusion method is expected to be standardized and applied in large-scale digital renovation and intelligent operation and maintenance of existing buildings. By integrating generative artificial intelligence and knowledge graphs, the level of automation and intelligence in 3D modeling can be further improved, promoting end-to-end digital twin city construction. Future work will focus on the adaptability of the method to diverse building types and complex real-world scenarios, as well as deep integration with industry standard systems, thereby providing more comprehensive technical support for Building Information Modeling (BIM) and related fields.

Author Contributions: T.L. and S.L.; methodology, T.L., S.L. and W.S.; validation, N.C., L.L. and X.Z.; formal analysis, T.L. and R.Y.; investigation, S.L. and N.C.; resources, W.S.; data curation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, T.L., W.S. and R.Y.; visualization, F.G. and X.Z.; supervision, W.S. and L.L.; project administration, W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Project of Jiangxi Provincial Higher Education Society in 2024, Grant No. JY-C-003, and the Project of the Vocational Education Development Center of the Ministry of Education, Grant No. JZJG25093.

Data Availability Statement: The data used in this study are derived from real-world architectural engineering projects, including CAD drawings (in DXF format), PDF documents (both native and scanned), and construction tables such as door/window schedules and specification sheets. These data were collected from existing building documentation and are not publicly available due to restrictions related to project confidentiality and ownership.

Conflicts of Interest: The author Runmin Yin was employed by the company QuikTech Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Natephra, W.; Yabuki, N.; Fukuda, T. Optimizing the evaluation of building envelope design for thermal performance using a BIM-based overall thermal transfer value calculation. *Build. Environ.* **2018**, *136*, 128–145. [CrossRef]
2. Hussain, M.; Zheng, B.W.; Chi, H.L.; Hsu, S.-C.; Chen, J.-H. Automated and continuous BIM-based life cycle carbon assessment for infrastructure design projects. *Resour. Conserv. Recycl.* **2023**, *190*, 106848. [CrossRef]
3. Peng, Y.; Lin, J.R.; Zhang, J.P.; Hu, Z.-Z. A hybrid data mining approach on BIM-based building operation and maintenance. *Build. Environ.* **2017**, *126*, 483–495. [CrossRef]
4. Chen, W.W.; Chen, K.Y.; Cheng, J.C.P.; Wang, Q.; Gan, V.J. BIM-based framework for automatic scheduling of facility maintenance work orders. *Autom. Constr.* **2018**, *91*, 15–30. [CrossRef]
5. Gourabpasi, A.H.; Nik-Bakht, M. BIM-based automated fault detection and diagnostics of HVAC systems in commercial buildings. *J. Build. Eng.* **2024**, *87*, 109022. [CrossRef]
6. Bottaccioli, L.; Aliberti, A.; Ugliotti, F.M.; Patti, E.; Osello, A.; Macii, E.; Acquaviva, A. Building energy modelling and monitoring by integration of IoT devices and building information models. In Proceedings of the IEEE 41st Annual Computer Software and Applications Conference (COMPSAC 2017), Turin, Italy, 4–8 July 2017; pp. 914–922. [CrossRef]
7. Allegrini, J.; Carmeliet, J. Simulations of local heat islands in Zürich with coupled CFD and building energy models. *Urban Clim.* **2018**, *24*, 340–359. [CrossRef]
8. Nagpal, S.; Mueller, C.; Aijazi, A.; Reinhart, C.F. A methodology for auto-calibrating urban building energy models using surrogate modeling techniques. *J. Build. Perform. Simul.* **2019**, *12*, 1–16. [CrossRef]
9. Srivastava, C.; Yang, Z.; Jain, R.K. Understanding the adoption and usage of data analytics and simulation among building energy management professionals: A nationwide survey. *Build. Environ.* **2019**, *157*, 139–164. [CrossRef]
10. Zhang, D.D.; Mui, K.W.; Wong, L.T. Ten questions concerning indoor environmental quality (IEQ) models: The development and applications. *Appl. Sci.* **2023**, *13*, 3343. [CrossRef]
11. Tang, S.; Shelden, D.R.; Eastman, C.M.; Pishdad-Bozorgi, P.; Gao, X. BIM assisted building automation system information exchange using BACnet and IFC. *Autom. Constr.* **2020**, *110*, 103049. [CrossRef]
12. Zhao, Y.F.; Deng, X.Y.; Lai, H.H. A deep learning-based method to detect components from scanned structural drawings for reconstructing 3D models. *Appl. Sci.* **2020**, *10*, 2066. [CrossRef]
13. Liang, X. Research on Recognition Algorithm of Building Units Based on DXF. Master's Thesis, Northeastern University, Shenyang, China, 2013.
14. Li, R. DXF Reading, Recognition and 3D Reconstruction of DXF Architectural Drawings. Master's Thesis, Tianjin University, Tianjin, China, 2017.
15. Pan, R.J.; Gao, X.Y.; Guan, F.L.; Gao, X. 3D modeling of highway based on section plans. *J. Syst. Simul.* **2012**, *24*, 17–19, 39. (In Chinese)
16. Li, P.D. Recognition of CAD Architectural Drawings Based on Deep Learning and Image Recognition. Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2024.
17. Zhang, Y. Generalization of Deep Neural Networks in Supervised Learning, Generative Modeling, and Adaptive Data Analysis. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 2022.
18. Lin, J.R.; Zhou, Y.C.; Zheng, Z. A review on research and application of automatic and intelligent drawing review. *Eng. Mech.* **2023**, *40*, 25–38. (In Chinese)
19. Chen, J.; Yu, F.Q.; Yi, S.K. Design and development of an automatic drawing system for construction detailing based on BIM. *J. Graph.* **2023**, *44*, 801–809. (In Chinese)
20. Jeon, K.; Lee, G.; Yang, S.; Jeong, H.D. Named entity recognition of building construction defect information from text with linguistic noise. *Autom. Constr.* **2022**, *143*, 104543. [CrossRef]
21. Goyal, N.; Singh, N. Named entity recognition and relationship extraction for biomedical text: A comprehensive survey, recent advancements, and future research directions. *Neurocomputing* **2024**, *618*, 129171. [CrossRef]
22. Liu, C.; Yang, S.W. Using text mining to establish knowledge graph from accident/incident reports in risk assessment. *Expert Syst. Appl.* **2022**, *207*, 117991. [CrossRef]
23. Gao, X.A. Research on Multi-Scale Information Retrieval in BIM Models Based on Natural Language Processing. Master's Thesis, Beijing University of Civil Engineering and Architecture, Beijing, China, 2021.
24. Yu, J.Y.; Lu, Y.L.; Zhang, Y.H.; Xie, Y.; Cheng, M.; Yang, G. A unified model for Chinese cyber threat intelligence flat entity and nested entity recognition. *Electronics* **2024**, *13*, 4329. [CrossRef]
25. Tsironi, E.; Barros, P.; Weber, C.; Wermter, S. An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. *Neurocomputing* **2017**, *268*, 76–86. [CrossRef]
26. Moon, S.; Chi, S.; Im, S.B. Automated detection of contractual risk clauses from construction specifications using bidirectional encoder representations from transformers (BERT). *Autom. Constr.* **2022**, *142*, 104465. [CrossRef]

27. Wang, H.X.; Xu, S.; Cui, D.D.; Xu, H.; Luo, H. Information integration of regulation texts and tables for automated construction safety knowledge mapping. *J. Constr. Eng. Manag.* **2024**, *150*, 04024034. [CrossRef]
28. Li, Z.Y.; Liu, Y.M.; Zhang, C. Method for Identifying and Outputting Pipeline Numbers from AutoCAD Process Flow Diagrams. China Patent CN112193375A, 15 January 2021.
29. Kasar, T.; Barlas, P.; Adam, S.; Chatelain, C.; Paquet, T. Learning to detect tables in scanned document images using line information. In Proceedings of the 2013 International Conference on Document Analysis and Recognition (ICDAR 2013), Washington, DC, USA, 25–28 August 2013; pp. 1185–1189. [CrossRef]
30. Prasad, D.; Gadpal, A.; Kapadni, K.; Visave, M.; Sultanpure, K. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2020), Seattle, WA, USA, 14–19 June 2020; pp. 572–573. [CrossRef]
31. Huang, Y.; Yan, Q.; Li, Y.; Chen, Y.; Wang, X.; Gao, L.; Tang, Z. A YOLO-based table detection method. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR 2019), Sydney, Australia, 20–25 September 2019. [CrossRef]
32. Ma, Z.X. Research on Key Technologies for Automatic 3D Model Generation Based on Architectural Drawings. Master's Thesis, Xi'an University of Technology, Xi'an, China, 2020.
33. Wei, W.; Ding, X.X.; Guo, M.X.; Yang, Z.; Liu, H. A review of text similarity calculation methods. *Comput. Eng.* **2024**, *50*, 18–32. (In Chinese)
34. *GB/T 50720-2011*; Standard for Fire Safety of Construction Site. Ministry of Housing and Urban–Rural Development: Beijing, China, 2025.
35. *20252453-T-424*; Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries—Part 1: Data Schema. Standardization Administration of China: Beijing, China, 2025.
36. Ministry of Housing and Urban-Rural Development of the People's Republic of China. *Building and Civil Engineering—Vocabulary—Part 3: Sustainability Terms*; Ministry of Housing and Urban–Rural Development: Beijing, China, 2025.
37. Nguyen, D.D. TableSegNet: A Fully Convolutional Network for Table Detection and Segmentation in Document Images. *Int. J. Doc. Anal. Recognit.* **2022**, *1*, 25. [CrossRef]