buildings

MDPI

Article

Intelligent BIM Searching via Deep Embedding of Geometric, Semantic, and Topological Features

Pin-Hao Huang [1], Sheng-Yu Song [1], Zhen Xu [2], Zhen-Zhong Hu [3,4] and Jia-Rui Lin [1,5,*]

[1] Department of Civil Engineering, Tsinghua University, Beijing 100084, China; hph22@mails.tsinghua.edu.cn (P.-H.H.); scy20@mails.tsinghua.edu.cn (S.-Y.S.)
[2] School of Civil and Resource Engineering, University of Science and Technology Beijing, Beijing 100083, China; xuzhen@ustb.edu.cn
[3] Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China; hu.zhenzhong@sz.tsinghua.edu.cn
[4] Institute for Ocean Engineering, Tsinghua University, Beijing 100084, China
[5] Key Laboratory of Digital Construction and Digital Twin, Ministry of Housing and Urban-Rural Development, Beijing 100084, China
* Correspondence: lin611@tsinghua.edu.cn or jiarui_lin@foxmail.com

Abstract: As a digital representation of buildings, building information models (BIMs) encapsulate geometric, semantic, and topological features (GSTFs), to express the visual and functional characteristics of building components and their connections to create building systems. However, searching for BIMs pays much attention to semantic features, while overlooking geometric and topological features, making it difficult to find and reuse rich knowledge in BIMs. Thus, this study proposes a novel approach to intelligent BIM searching by embedding GSTFs via deep learning (DL). First, algorithms for extracting GSTFs from BIMs and identifying required GSTFs from search queries are developed. Then, different GSTFs are embedded via DL models, creating vector-based representations of BIMs or search queries. Finally, similarity-based ranking is adopted to find BIMs highly related to the queries. Experiments show that the proposed approach demonstrates an efficiency of 780 times greater than manual retrieval methods and 4–6% more efficient than traditional methods. This study advances the field of BIM searching by providing a more comprehensive, accurate, and efficient method for finding and reusing rich knowledge in BIMs, ultimately contributing to better building design and knowledge management.

Keywords: building information model (BIM); model search; feature extraction; similarity calculation; deep learning; feature embedding

Academic Editors: Marcus Sandberg and Ahmed Senouci

Received: 12 February 2025
Revised: 9 March 2025
Accepted: 15 March 2025
Published: 18 March 2025

Citation: Huang, P.-H.; Song, S.-Y.; Xu, Z.; Hu, Z.-Z.; Lin, J.-R. Intelligent BIM Searching via Deep Embedding of Geometric, Semantic, and Topological Features. *Buildings* **2025**, *15*, 951. https://doi.org/10.3390/buildings15060951

1. Introduction

Building Information Model (BIM) has achieved intuitive three-dimensional visualization of design information, providing an efficient solution for cross-disciplinary collaborative design, technical disclosure, and full-process engineering management of construction projects [1]. As a tool for creating and managing data, BIM offers more effective and sustainable options for the entire lifecycle of building design and management [2]. Currently, various online BIMs (Building Information Models hosted on cloud-based platforms) and component libraries are developing rapidly. The rapid development of online BIMs and component libraries presents a new paradigm to reuse design knowledge, while also bringing the challenge for designers to efficiently find suitable BIMs. The vast library of BIM resources and components makes it increasingly difficult for designers to accurately locate

the information they need across numerous models [3]. Therefore, the ability to quickly retrieve required BIMs is an urgent need.

However, the existing BIM searching technologies mainly focus on the searching of internal elements of BIMs or single BIM objects via semantic features, lacking an efficient method for searching an overall design model with multiple BIM objects inside. Referring to the review by Hamed Khademi [4], the existing BIM searching systems include three types based on context, geometry, and content, but they can only realize the retrieval of internal components of BIM. For instance, typical BIM searching systems such as BIMSeek and BIMSeek++ use text and semantic retrieval to search BIM components such as tables, chairs, doors, and windows [3]. Meanwhile, 3D model search methods mainly adopt 3D analysis to construct the geometric shape characteristics of the model as similarity measurement for the retrieval of target models, without consideration of semantic features. Unlike common 3D models, BIM is an engineering data model that contains not only 3D geometric information but also topological relationships and semantic information [5], implying that retrieval based solely on 3D geometry or text is insufficient for searching for BIMs. Furthermore, a study explored the impact of simple topological relationships on BIM search results and found that such relationships do not significantly improve search accuracy [6]. This further underscores the need for a more comprehensive approach that integrates geometric, semantic, and topological features for effective BIM retrieval.

Searching a complex BIM with multiple components needs consideration of not only the semantic features (name, function, type, etc.) but also the geometric features (shape, location, etc.), and even the topological features (connections between spaces, equipment, etc.), it is still an open question to the academia and industry for searching a whole BIM with multiple features. Though some studies have explored the similarity measurement of the overall BIM, they focus on comparing the similarity between BIMs. For example, He et al. introduced a variety of calculation methods for the overall similarity of BIMs from the aspects of attributes, topology, and shape, which can compare the similarity between apartment models [7]. In summary, there is still a lack of efficient methods for searching a complex BIM with integrated consideration of geometric, semantic, and topological features.

Recent advancements in artificial intelligence such as computer vision and natural language processing (NLP) have enabled near or superhuman performance on various tasks by leveraging massive data [8]. Deep learning is also widely used in the field of building and construction. It can construct datasets from massive building data and train corresponding neural networks using supervised or unsupervised methods to complete various machine learning tasks such as classification, regression, and prediction [9]. Notably, NLP—a core deep learning technique—has demonstrated exceptional capabilities in structuring unstructured textual data, enabling automated knowledge extraction and semantic enrichment of BIM metadata [8]. In the field of BIM searching, deep learning also has broad application prospects. It can extract data from BIMs and construct datasets in formats such as text, images, and 3D models to train deep neural networks for feature extraction and matching [10].

Therefore, to enable comprehensive searching of the entire BIM, it is essential to consider not only the semantic information provided in the text but also the topological relationships between various components and spaces, along with the overall shape characteristics of the model. This paper presents a deep learning-based method for intelligent BIM searching by embedding multimodal features such as semantic, topological, and geometric features and demonstrates its effectiveness through experiments and case studies. Through case analysis, it can be seen that this method has achieved advantages in terms of retrieval operation speed, resource consumption, and accuracy.

The rest of the paper is organized as follows: Section 2 reviews related studies on searching for BIM. Section 3 elaborates the methodology for multimodal BIM retrieval based on deep learning, including the method for extracting multimodal features from BIMs using deep neural networks, and the method for calculating BIM similarity and retrieving and recommending BIMs using ensemble learning. Section 4 presents the experiments and results to validate the performance. Section 5 presents the discussion. Section 6 presents the conclusion and future work.

2. Related Work

The development of BIM libraries is currently rapid. Designers often need to spend a lot of time searching for suitable BIMs or components from BIM libraries, such as Autodesk BIM 360 (https://www.autodesk.com/bim-360/ (accessed on 5 February 2025), RevitCity (https://www.revitcity.com/ (accessed on 5 February 2025)), BIMobject (https://www.bimobject.com/ (accessed on 5 February 2025)), or other cloud-based platforms. This process typically involves finding specific building elements, systems, or components that meet design requirements, such as dimensions, materials, or functionality. How to quickly retrieve the required BIMs has become an urgent need [3]. With advances in computer graphics, computer vision, and computer-aided design, search engines for 3D shape models have become increasingly sophisticated [11]. Current mainstream 3D model retrieval methods focus on model-based retrieval using voxels, point clouds, multi-view features, and image-based cross-domain retrieval to overcome the limitations of requiring 3D query models [12].

BIM is an engineering data model based on 3D digital technology. In addition to the 3D geometric information of building components, it also encompasses important data such as topological relationships between components and semantic information of building components. Merely using 3D geometric information for BIM retrieval is insufficient [3]. At present, the mainstream retrieval methods of BIM can be categorized into two types: catalog browsing and text-based matching. The former is time-consuming and requires extensive time to manually collect and organize BIMs in a predefined way. The latter adopts text-based query and semantic expansion for matching BIMs from a large database and is more commonly adopted recently. Generally, this kind of method consists of three parts: (1) information extraction from BIM which extracts useful information for model searching, (2) processing of searching queries that identifies keywords or the intents of users from textual inputs; (3) BIM matching that adopts similarity measures or other metrics to find BIMs highly related to the queries. A detailed review of related works is discussed as follows.

2.1. Information Extraction from BIM

BIMs encapsulate rich semantic attributes, geometric data, and topological relationships through various file formats such as RVT in Revit or Industry Foundation Classes (IFC) [13]. While the RVT format is mainly used for detailed design and collaboration within the Autodesk Revit platform and is suitable for internal use by project teams, the IFC format, as an open standard, is suitable for data exchange and cross-platform collaboration between different BIM software, ensuring information interoperability [14]. A study has demonstrated the critical role of IFC standardization in enabling proactive automated rule checking, particularly through iterative collaboration between developers and end-users to overcome semantic fragmentation in complex regulatory scenarios [15]. Compared to text-based files, it is difficult to use BIM formats such as RVT and IFC for model-searching purposes due to their complex structure. The embedded information in BIM should be first extracted through Revit secondary development (a method of extending Revit's functional-

ity and automating tasks through programming) or an IFC data reader for model searching or other purposes. For instance, Dynamo is utilized to create scripts that directly extract building data from Revit models [16]; while XML standard schema is adopted to extract information pertinent to construction practitioners from BIM, and generate the ontological model for semantic reasoning [17]. An extended IFC schema is also developed to include more entities and attribute sets in BIMs and further adopted for information extraction and sharing [18]. Recent advances leverage graph neural networks (GNNs) to automate topology extraction, such as the SAGE-E algorithm, which processes node attributes and edge features through message-passing mechanisms, capturing implicit spatial dependencies with 79% accuracy in semantic classification tasks [19]. For BIM searching, metadata of BIMs such as keywords, tags, descriptions, etc. [20] are usually used to extract information from BIM and further could be used for keyword-based BIM retrieval [4] or other methods.

In this study, given that the IFC format is widely adopted, IfcOpenShell [21], an open-source (LGPL) software library for working with IFC files, is employed for extracting information from BIMs. IfcOpenShell can use its Python API (https://docs.ifcopenshell.org/ifcopenshell-python.html (accessed on 5 February 2025)) to achieve programmatic access to structural entities and their geometric properties through functions to achieve target entity extraction [22]. And since BIM encompasses diverse information, including entity attributes, materials, geometry, space composition, topology, etc., both semantic, geometric, and topological features are all extracted for further model-searching purposes.

2.2. Query Processing and Expansion

As search queries from users are usually in textual formats, natural text parsing is important for processing user queries in BIM searching. The key idea is to extract key concepts and potential constraints from texts, which could be used for matching relevant BIMs [23,24]. At present, the commonly used natural language parsing methods can be categorized into three types: rule-based parsing, statistics-based parsing, and deep learning-based parsing [25]. However, since the same concepts or objects could be described with different words or phrases, directly matching extracted information from user queries with data extracted from BIM usually generates poor results [26].

To overcome the problem mentioned above, query expansion or semantic expansion is introduced. Ontology modeling or similar approaches are adopted to model domain knowledge, which could be used to expand synonyms, aliases, etc. for better search results. For example, by establishing a text semantic library and forming a semantic-to-knowledge mapping [27], better search results are achieved. In the same way, BIMSeek++ introduces a novel approach to semantic expansion, achieving promising results in searching BIM components [3].

Beyond text-based semantic expansion, recent advancements in multi-modal approaches have enhanced BIM retrieval by integrating geometric and topological features. For instance, a study leverages multi-modal optimization to align semantic attributes with geometric features in BIM retrieval [28]. Similarly, multi-modal deep learning (MMDL) has been applied to fuse graphical and non-graphical BIM data for fine-grained element classification, enabling end-to-end retrieval of semantically and geometrically enriched BIM components [29].

However, the current approaches still pay much attention to the extraction of semantic features from queries and their expansion with the support of domain knowledge, while less attention is paid to the topological relationship between BIM components and the geometric information of the whole building models, limiting its search capabilities to component-level rather than whole building model-level search.

2.3. BIM Matching

The final step is BIM matching, which finds the most relevant BIMs to the user queries. To match keywords extracted from queries with BIMs, methods such as boolean algebra, probability algorithms, vector space models, etc. [30] are usually used. By choosing different methods, one can determine when a property of the BIM component should exactly match the keywords extracted or have a minimum similarity with the keywords. In this process, similarity measures are the basic metrics used for retrieving BIMs or components from databases via text-based queries or finding a BIM similar to the current BIM. For example, Tversky similarity is adopted for the retrieval of BIM components in the BIMSeek++ system [3]. Meanwhile, as BIMs encapsulate multiple features, different metrics may be needed. For instance, the similarity of attributes such as room count and area are measured using cosine distance, while other descriptive attributes can be evaluated using matching rates [7].

Except for the similarity of semantic attributes, topological similarity, and shape similarity could be further considered [31]. In an apartment design case, the Langenhan algorithm, which involves converting the BIM into a decision tree and constructing an adjacency matrix, is introduced to evaluate the topological similarity, and further integrated with shape similarity determined by the maximum overlapping area to find similar floor plans to the current one [7]. In addition to the traditional methods mentioned above, images can also be extracted from the BIM, and image recognition techniques can be utilized to classify the BIM, the best model ResNet50 has an accuracy of 97.92% [32]. With these explorations, it is obvious that taking more features for BIM searching will lead to better search results and enhance the reusing of design knowledge.

2.4. Summary of Research Gap

In summary, searching relevant BIM from libraries or databases is essential for reusing engineering knowledge to create better design models. Recently, quite a few efforts have been devoted to developing various methods for searching BIM components from BIM libraries, such as NLP-based object searching [26], BIMSeek++ [3], etc. In most of these methods, semantic features such as names, types, quantities, and other properties are usually considered when finding the most relevant BIM components from the database. However, geometric features such as shapes, locations, and distances, are omitted, due to the difficulty in considering geometric features in evaluating the similarity of texts and BIM objects. Moreover, since most of the methods focus on searching a single BIM component in a database, relationships between components are not considered. Thus, searching for a BIM with multiple components such as a plan layout of a house, or a whole model of a building is still not easy, and existing methods usually prompt poor results [31]. Although studies like [33] attempt to address component-level dependencies, they remain domain-specific and lack cross-modal fusion of semantic, geometric, and topological features for entire model retrieval. The fusion of multi-model information such as semantic, topological, and geometric information is considered to be the development trend [5], which will further exploit the power of BIM.

3. Methodology

To achieve better searching and reusing of BIM with multiple components, a novel framework extracting and matching geometric, semantic, and topological features (GSTFs) from BIM and search queries via deep learning is proposed in Figure 1. First of all, the GSTFs extraction algorithm is developed to automatically extract semantic (i.e., room names, area, and quantity), topological (i.e., adjacency and connectivity between rooms), and geometric features (i.e., images and contours of floor plans) from IFC-based BIM.

Meanwhile, the search queries were parsed using predefined regular expressions and further extended with certain rules. Then, deep learning-based vectorization is introduced to create embeddings of multimodal features extracted from BIM and also the constraints extracted from search queries. With those embeddings, the similarity between a certain search query and BIMs is easily calculated and then the BIMs are ranked based on similarity. Details of these methods are further discussed below. Finally, by collecting 2D drawings and creating BIMs of rural houses in China, a BIM library containing 110 BIMs of rural houses is developed, and some metrics such as mNDCG and mAP are adopted to validate the performance of the proposed approach.

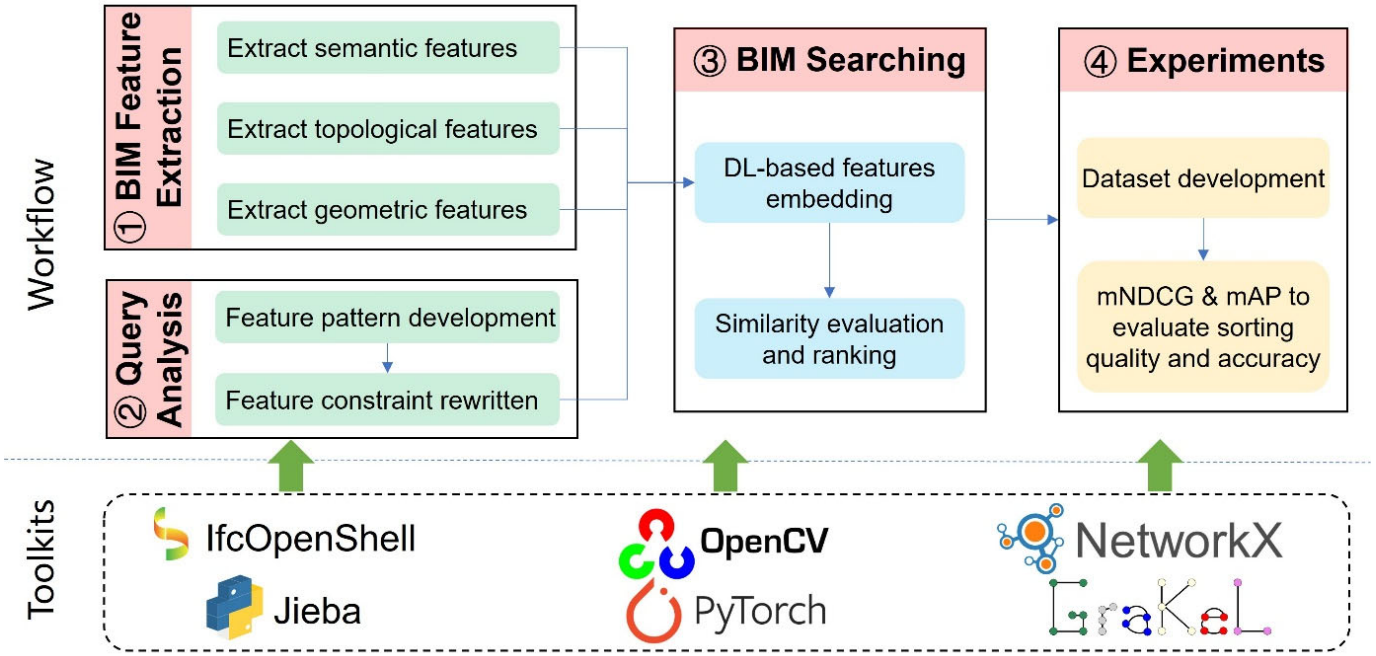

Figure 1. Methodology framework of this research.

3.1. Extracting Multimodal Features from BIM

Since existing search algorithms and deep learning models cannot directly process BIM file formats (e.g., RVT and IFC), the retrieval process requires preprocessing these models by first extracting embedded information —such as semantic text, topology, geometric shapes, and other relevant data. IFC standard is an open-source BIM standard developed by the buildingSMART organization, and various open-source parsing tools can be used to extract information from the BIM model in IFC format. The BIM models in this study were all created using Revit software and exported to IFC4 Reference View (the default IFC format exported by Revit 2020), thus Python scripts could be used for data parsing. Specifically, by utilizing the open source IfcOpenShell tool and employing methods like "open" and "by_type" to iterate through the specified types of components in the model, the geometry of the building plan, the topological connection between the rooms, and the attribute information of components such as walls, doors and windows can be extracted from the BIM model, thus forming an automatic extraction algorithm for the multimodal features of the BIM model, as shown in Figure 2. The following are the sequential steps for extracting semantic, topological, and geometric features from a BIM model.

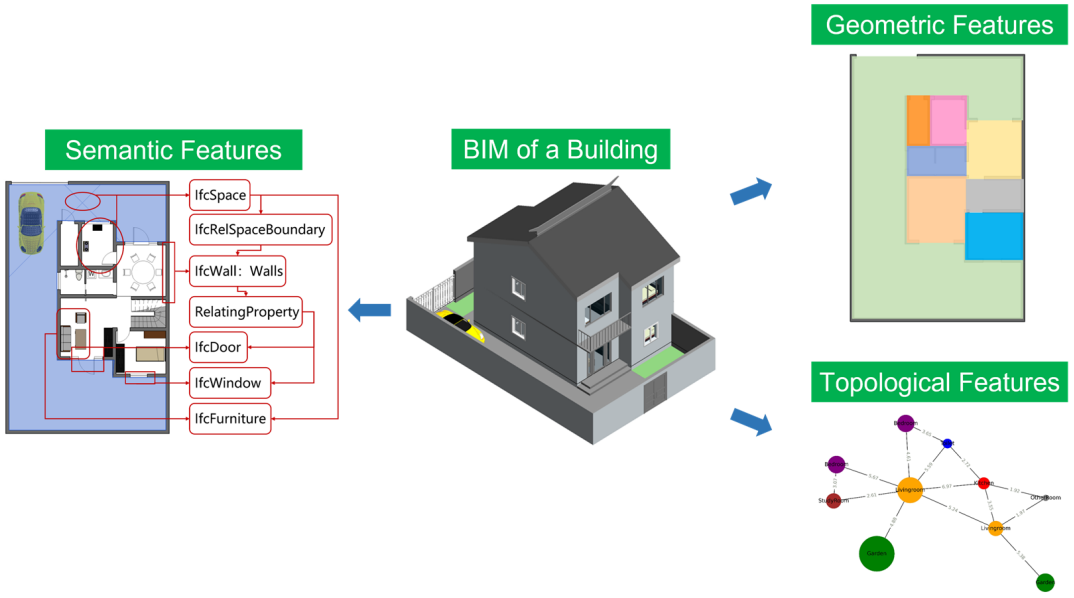

Figure 2. BIM multimodal features extraction.

3.1.1. Semantic Features Extraction

Extracting semantic features involves parsing the IFC file, identifying objects, extracting properties, and generating semantic features that describe the functional and spatial characteristics of the room. In the IFC data structure, each class contains common attributes such as Name and Description inherited from IfcRoot. These attributes can be directly accessed to identify corresponding objects and extract their properties. Building component properties in the IFC class are mostly stored in associated IfcProperty and IfcPropertySet [6]. By using IfcOpenShell, these property sets can be accessed to extract individual properties. For example, to extract properties of a wall, you would first identify the wall object from the IfcWall class, then use IfcRelDefinesByProperties to link to its associated IfcPropertySet, where you can find specific attributes like material type, thickness, and height. Any class inheriting IfcObjectDefinition can use IfcRelDefinesByProperties to extract property information from its associated property set. Relevant association information between IFC classes is stored in the properties, IfcProperty, and IfcPropertySet of the associated relationship entity or associated entity. The extraction of information related to various components within a room, such as identifying a bedroom by the presence of a bed and a closet, is illustrated in the right part of Figure 3.

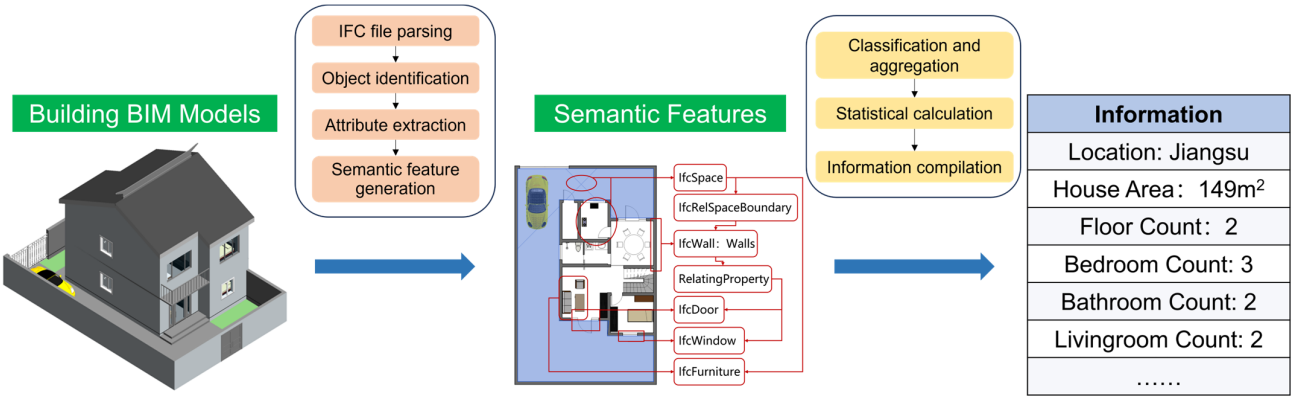

Figure 3. An example of BIM model semantic features extraction.

For example, in the BIM models created in this study, all the information on rooms, walls, doors, and windows are stored in the objects of IfcSpace, IfcWall, IfcDoor, and IfcWindow respectively. For the room as a whole, its corresponding IfcSpace object can be extracted. Basic information such as the room name and area can be obtained through PropertySets. The coordinates of the room boundary outline can be obtained through BoundedBy so that the shape of the room can be obtained. Room separation information can also be obtained in BoundedBy.IfcRelSpaceBoundary. It can be used to determine whether the room has a physical partition and to obtain the relationship between the room and the wall. Similarly, various information such as length, width, and component description can be obtained through IfcRelDefinesByProperties in IfcWall and IfcDoor. After extracting various property information, the property characteristics of the BIM model can be extracted into a dictionary in the form of "{Province: Jiangsu, Area: 149 m^2, Cost: $100,000, Floor Count: 2, Bedroom Count: 3, Bathroom Count: 2, Livingroom Count: 1, Kitchen Count: 1 ...}". Then the information of a single house can be processed into a feature vector according to the agreed parameter method for subsequent house type query and matching.

3.1.2. Topological Features Extraction

The topological features of the building model mainly include adjacency and connectivity relationships between different rooms, which will be considered separately in this study. As shown in Figure 4, if two rooms share a wall of which two sides are each the boundary of two rooms, then the two rooms must be adjacent. If two rooms share a door, then the two rooms must be connected. By using Python scripts and IfcOpenShell, the topological relationship between rooms in a house can also be extracted from the IFC model file. If there is a solid partition wall between two adjacent rooms, the adjacency relationship of the rooms can be obtained from the relationship between IfcSpace, IfcRelSpaceBoundary, and IfcWall. For adjacency, the relationship of the rooms can be ascertained by identifying shared IfcWall elements between IfcSpace instances via their IfcRelSpaceBoundary relationships. If the PhysicalOrVirtualBoundary attribute of the IfcRelSpaceBoundary is set to 'PHYSICAL', it confirms the presence of a solid partition wall, indicating that the rooms are adjacent. For connectivity, the relationship is established by identifying IfcDoor instances, which are associated with an IfcRelVoidsElement that references an IfcOpeningElement in an IfcWall. This IfcOpeningElement represents the void in the wall created by the door. By then checking the IfcRelSpaceBoundary instances, we can find the connected IfcSpace elements on opposite sides of the door, thereby confirming the rooms are connected.

In addition to walls, virtual room separators can also be used to partition rooms. The boundaries of a room can be obtained through the IfcRelSpaceBoundary associated with the corresponding IfcSpace of the room. This object contains the PhysicalOrVirtualBoundary and RelatedBuildingElement properties. When the boundary is a wall, the value of the PhysicalOrVirtualBoundary attribute is PHYSICAL and the value of the RelatedBuildingElement attribute is the corresponding IfcWall. When the boundary is a virtual room divider, the value of the PhysicalOrVirtualBoundary attribute is VIRTUAL and the value of the RelatedBuildingElement attribute is null, indicating that this boundary does not correspond to any physical component. Connectivity can be determined by whether the shortest distance between two virtual boundaries is 0.

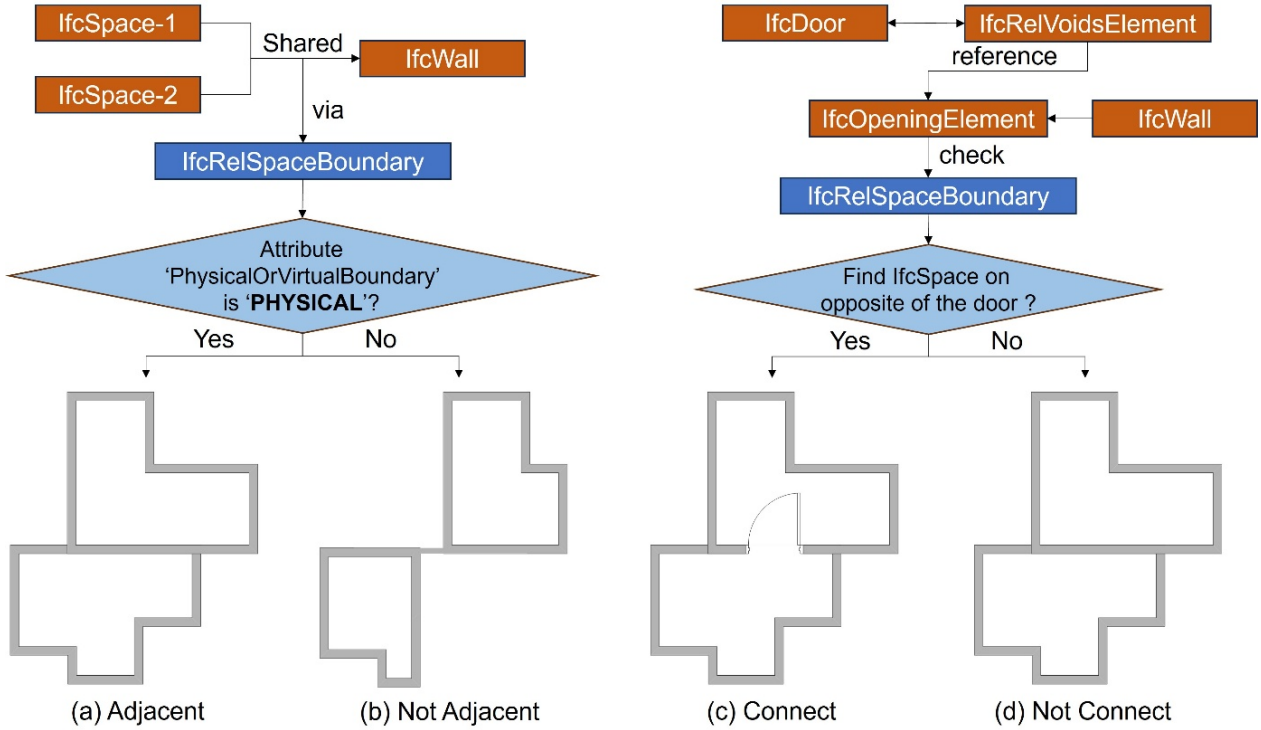

Figure 4. BIM model multimodal features extraction.

After acquiring the spatial topology of rooms, it is transformed into a Python dictionary. Python dictionary is primarily used as a structured data container to store basic topological properties extracted from the BIM model (e.g., room sizes, location coordinates, and adjacency relationships). Then, utilizing Networkx, the topological relationships between rooms on each floor of a building are processed into a topological network graph to form a graph database. Figure 5 illustrates house topology diagrams generated using Networkx (v3.0) and Matplotlib (v3.7.1), where the left part shows a single-story house topology while the right part presents a multi-story house topology. The key distinction lies in their connectivity patterns: in the single-story topology, all nodes can be connected into a single graph, whereas in the multi-story topology, rooms on different floors are represented in separate topological graphs. In both visualizations, different colors represent rooms with varying attribute functions, the node size corresponds to the area size of the corresponding room, and the numerical values on the lines linking the rooms signify the distance between the centroids of neighboring rooms. In this study, the topological relationship within each floor is calculated separately, and the connectivity between houses across floors is temporarily not considered. Therefore, the topological graph of a multi-story building will be processed as multiple separate network graphs, as shown in Figure 5. After establishing the topological network database of household types, machine learning algorithms such as graph neural networks can be easily and practically used to query the graphs for similarity and linkage relationships.

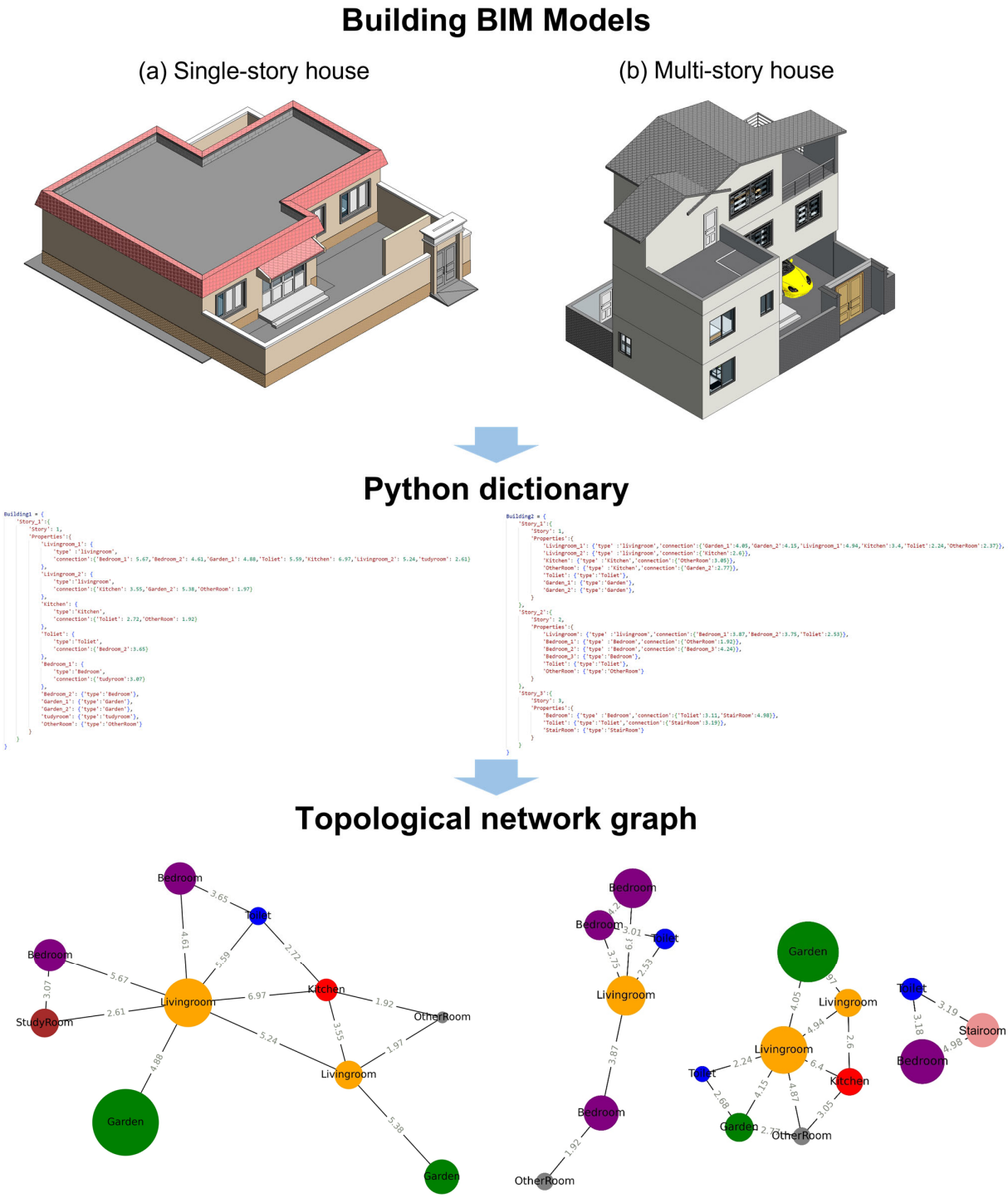

Figure 5. Examples of topology diagrams for the houses' connectivity.

3.1.3. Geometric Features Extraction

Unlike semantic and topological features, extracting geometric features from a BIM model file presents greater challenges. Geometric data requires comprehensive storage, transmission, and visualization, lacks the separability and divisibility of semantic data [34], and its large volume and high heterogeneity further exacerbate these difficulties. Due to the intricacy of 3D geometric features, this study specifically focuses on analyzing the 2D geometric shape features of the BIM of a building. In general, the geometric features of a house can be determined by both the spatial layout and the plan profile information of the building. The former can be directly reflected by the floor plan, while the latter can

be obtained by extracting a list of coordinates representing the contour of the building. They both contain rich information on geometry, so both types of information should be considered in the task of extracting geometric features. Given the objectives of this study and our dataset's current scale (110 BIM models), directly outlining each model's contours provides an efficient means to achieve our research goals. The floor plan information of the house type can be obtained by directly adopting the top view perspective of the BIM of the building, fixing the coordinate orientation of the house type, and intercepting the floor plan in wireframe format.

The coordinate information of the house outline points can be obtained by parsing the BIM model file. Specifically, the relative coordinates of boundary points can be extracted from the spatial separation property information of each IfcSpace of the model and the location information of IfcWall, which can form a description of the contour coordinates of the whole room. At the same time, the spatial location of the household separation can be further verified by combining the starting and ending points of IfcWall and the width of the wall of IfcWall. Finally, the absolute coordinates are normalized to relative coordinates to obtain the coordinate information of household contour points. The results obtained from shape feature extraction are shown in Figure 6, where different colors represent rooms with different attribute functions.

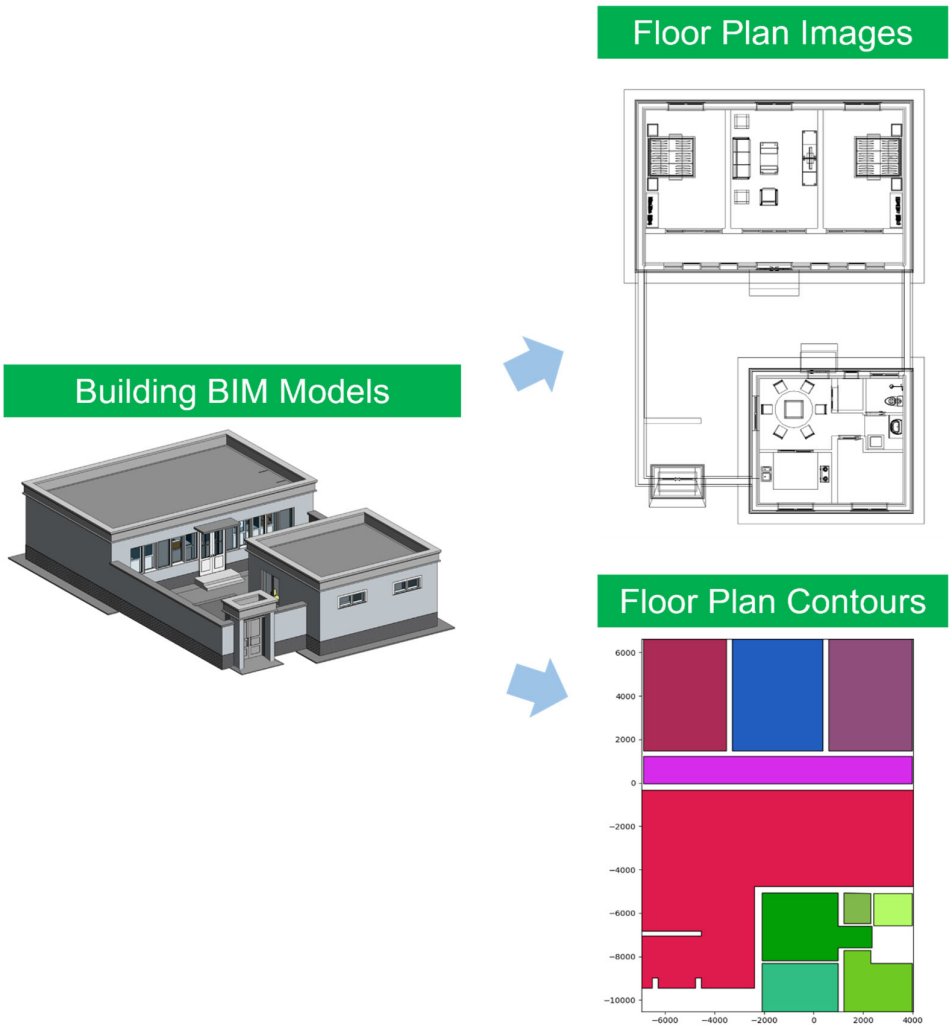

Figure 6. An example of BIM model geometric feature extraction.

3.2. Parsing Multimodal Constraints from Search Queries

The purpose of this step is to obtain multimodal constraints such as semantic attributes, topological relationships of rooms, geometric descriptions, and other information about the desired BIM models from the text-based search queries, to query the target model more efficiently. In this study, domain word lists and the Jieba package (an open-source Chinese segmentation application in Python language) are used to segment the search text to facilitate subsequent attribute and conjunction matching. Considering that the search text primarily consists of the Chinese language, the initial step involves collecting descriptive nouns and commonly used expressions related to the field of housing and real estate. These are compiled to create domain vocabulary for housing and real estate search, which aids in word segmentation and defines the minimum unit of proper noun subdivision. Then, the search engine mode of Jieba is used to cut the search sentences into segmentations with word properties and word order based on the domain word lists.

The segmentations have special keywords and expression vocabulary for the description of rural housing needs and house type information, and the description of semantic, topological, and geometric features in the search intent can be extracted using manually defined regular expressions. Specifically, for the attributes and semantic information of the required BIM model, all kinds of descriptive clauses in the search statement can be extracted, and the nouns, verbs, prepositions, and other keywords in them can be matched according to the syntactic structure, for example, when "area" and "is" appear, the number after "is" is extracted as the area of the house for search. The topological connectivity of rooms can also be extracted in this way, e.g., "living room connects to the kitchen" can extract the keywords "living room" and "kitchen" which represent the names of rooms, and the word "connect" which characterizes their connectivity, to extract the semantic information related to the topological connectivity. For geometry descriptions, labels such as "square" and "narrow" can be extracted to obtain preferences for house geometry in search intent. Following this way, several types of regular expressions and the corresponding text information are shown schematically in Table 1.

Table 1. Several patterns are defined based on regular expressions to extract corresponding information.

Information Type	Regular Expression Patterns	Functions
Description of integers and decimals	re.findall(r'\d+\d*', pseg_cut[i].word) re.findall(r'\d+\.\d+', pseg_cut[i].word)	Extracting the attributes such as area and floor
Description of keywords semantics	re.findall(fr'\b(?:{"|".join(Keywords)})\b', pseg_cut[i].word)	Extracting keyword information such as location, room function, etc.
Description of topological relations	re.findall(fr'\b(?:{"|".join(ConnectionWords)})\b', pseg_cut[i].word), re.findall(fr'\b(?:{"|".join(RoomName)})\b', pseg_cut[i].word)	Extracting the conjunction and connected rooms
Description of Geometric shape	re.findall(fr'\b(?:{"|".join(ShapeWords)})\b', pseg_cut[i].word)	Extracting the description of the shape of the house

An example is given here to illustrate the extraction process of multimodal constraints. The natural sentence "I need a 2-story house in Beijing, about 150 square meters, 4 bedrooms, 2 bathrooms, the master bedroom is with bathroom", is taken as the text to be extracted. Firstly, the Jieba package utilizes its built-in word segmentation algorithm and domain-specific dictionary to segment the input natural sentence and identify its various components. Next, regular expressions are used to extract key information from the segmented text based on pre-defined domain knowledge templated, such as the floor

count is 2, shape is square, location is Beijing, house area is 150, bedrooms count is 4, the master bedroom connects to a bathroom. Subsequently, the extracted information is organized into a hierarchical tree structure with 'House' as the root node, branching into 'Semantic', 'Topology', and 'Geometry'. The key information extracted by regular expressions is allocated to the corresponding child nodes according to attribute categories, forming the tree structure, and different types of attribute information are distinguished by color. Finally, the information from the sentence is compiled into the corresponding columns of a table based on the specific classification of each child node, thereby creating a tabular format of the information content. The overall process is depicted in Figure 7.

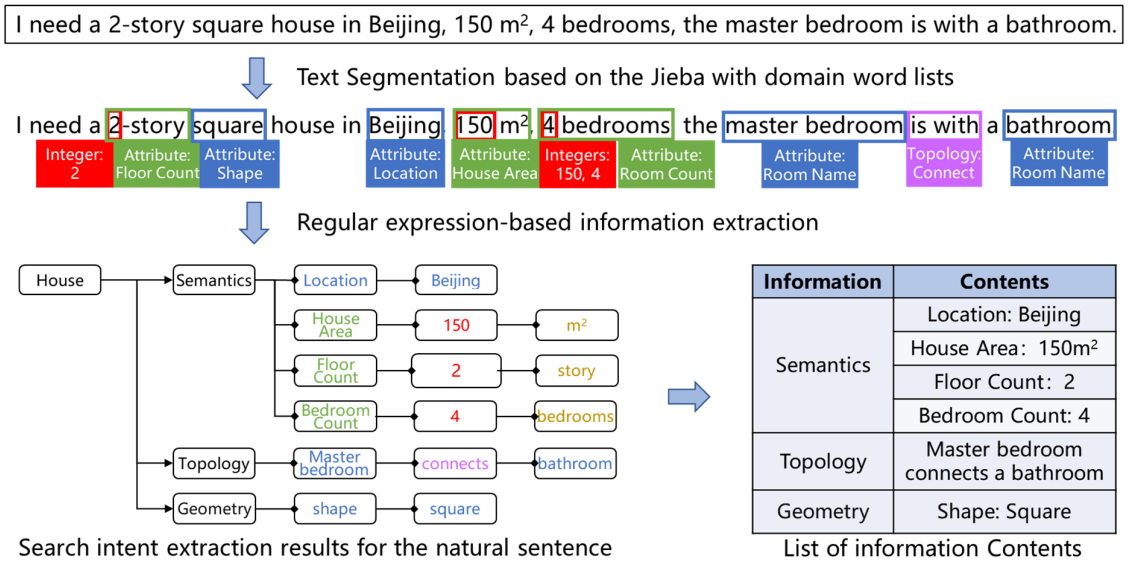

Figure 7. An example of the extraction process of search intent.

3.3. BIM Searching via Deep Embedding of Features

After extracting the multimodal features from the BIM models, the text-to-BIM model search requires the similarity calculation between the search queries in texts and the BIM models. Therefore, the search query is firstly parsed using a combination of natural language phrasing and regular expressions to extract key information from the search query. Secondly, deep learning and feature engineering tools are used to embed the multimodal features from the BIM model and the key information extracted from the search query separately, to transform them into unified feature vectors. Thirdly, a weighted similarity is calculated based on the feature vectors, and the intelligent search and recommendation of BIM models are realized based on the similarity ranking. The workflow of the BIM model similarity searching method is shown in Figure 8. A detailed explanation is as follows.

3.3.1. Multimodal Feature Embedding via Deep Learning-Based Vectorization

After extracting the semantic, topological, and geometric feature information of the BIM model and the search queries, they need to be embedded in a uniform vectorization representation for similarity calculation. As multimodal features such as semantic, topological, and geometric features are considered, deep learning-based vectorization is adopted in this study (Figure 8). The semantic information of each attribute is directly organized into a vector consisting of text and data in a uniform format for the embedding of semantic features. The topological information is stored in the form of Networkx topology graphs, and the graph kernel method is used to implement the topological feature embedding. For the embedding of geometric features, the floor plan images are embedded by the convolutional neural network, the contour information is embedded by feature

parameters of contours, and the description of geometric shapes in the search text are also embedded as feature parameters of house contours. The detailed methods are shown in the following paragraphs.

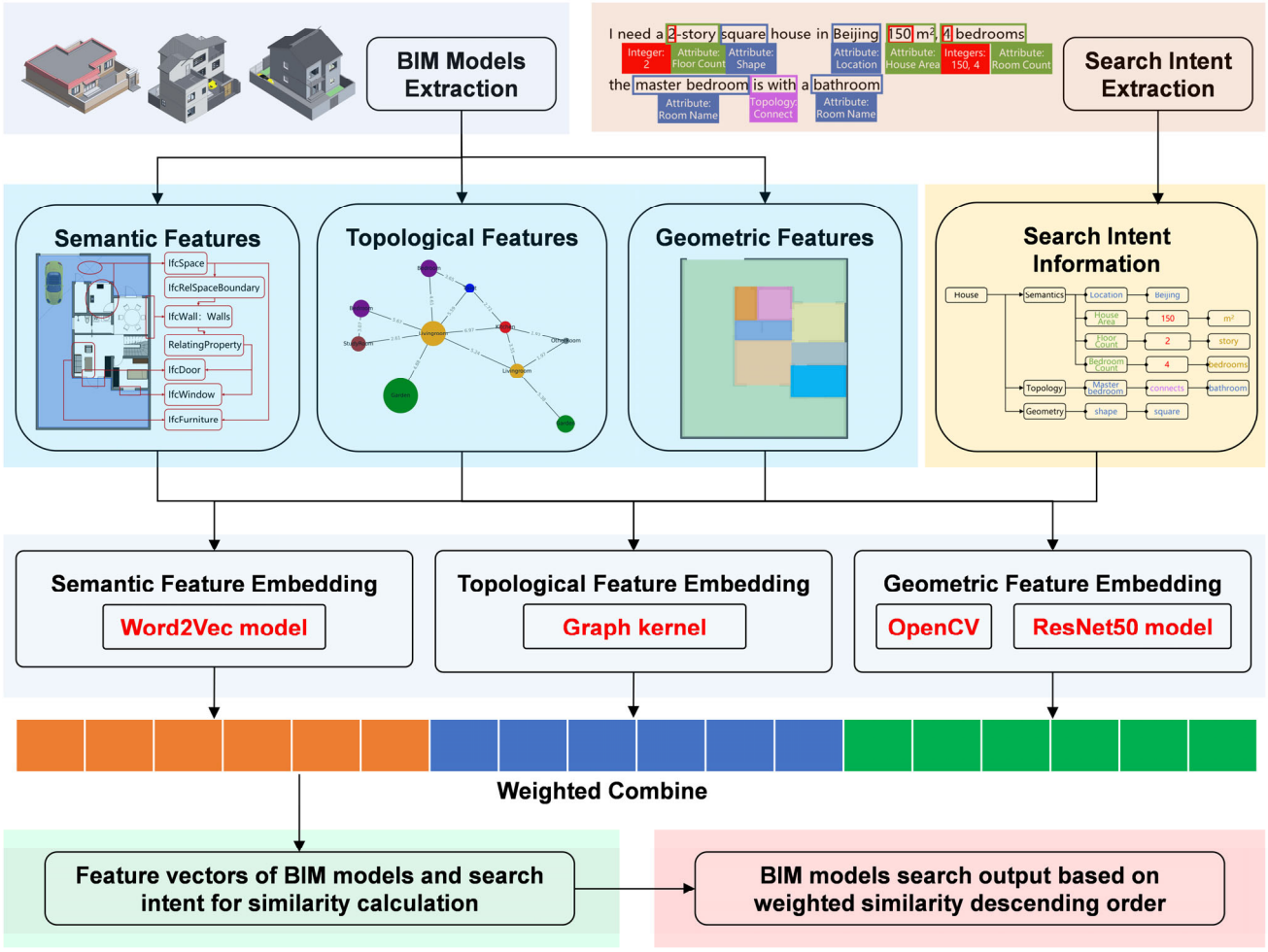

Figure 8. The workflow for Text-to-BIM searching based on deep embedding learning.

Firstly, the embedding of semantic features. The BIM model multimodal features extraction method in Section 3.1.1 can extract the semantic attributes of the BIM model as "{Province: Jiangsu, Area: 149 m^2, Cost: \$100,000, Floor Count: 2, Bedroom Count: 3, Bathroom Count: 2, Livingroom Count: 1, Kitchen Count: 1. . .}", and the extraction method for search intent in Section 3.2 can also yield a dictionary of attributes of the corresponding form. Therefore, manually defined transformation rules can be used to embed the attributes dictionary into a semantic feature vector. Specifically, for near-synonyms that may be widespread in semantic dictionaries, such as "bathroom", "lavatory", "sanitary", "toilet", etc., synonym substitution is performed by manually defining a list of near-synonyms and unifying their descriptions with the same descriptive lexicon. Then, the generic string attributes such as the name of cities are embedded directly as string vectors; the descriptive attributes in text form are embedded directly as word vectors by the Word2Vec model; the data attributes are weighted and normalized to perform feature embedding; Other types of semantic information are split into text and data information according to rules, embedded as string word vectors and data vectors respectively, and then weighted and combined as feature vectors. The corresponding weights are determined manually by experts. The above Word2Vec model is implemented by the Gensim (a popular Python library for natural language processing) package. Based on the list of synonyms, a corresponding text corpus

is constructed as the training set, by which the Word2Vec model is trained with training parameters "vector_size = 100, window = 5, min_count = 1, workers = 4".

Secondly, the embedding of topological features. Section 3.1.2 has proposed a method to extract topological relationships between rooms of the BIM model, and Section 3.2 also proposes a method to extract topological relationships in the search queries. These topological relationships can be stored as NetworkX topology graphs for processing. The graph kernel is an effective machine learning method for graph feature embedding, so the random walk method based on the graph kernel is chosen for the embedding of topological features, which is unsupervised and has better migration and is more suitable for the graph data in this study. The Grakel Kernel method of the Grakel (an open-source Python library that provides a set of tools and algorithms for graph kernel-based machine learning tasks) package is used for program implementation. Specifically, based on the node list, node attribute list, and an adjacency matrix of the Networkx topology graph, the Grakel.Graph method is used to transform the Networkx topology graph into Grakel graph network data. Then, based on the Grakel. GraphKernel method, the Grakel graph network is embedded as a topological feature vector with a random wandering kernel to achieve the embedding of household topological features.

Thirdly, the embedding of geometric features. The method in Section 3.1.3 can extract the contour information of the BIM model with the floor plan, and the method in Section 3.2 can parse the simple description of the shape of the house in the search intent. The geometric information is divided into contour information and floor plan for processing, and the household shape description can be covered in the attributes of the contour for embedding. The contour information is embedded using the OpenCV package to extract feature parameters. Specifically, as shown in Table 2, the embedding of outer contour information is achieved in the form of fitted polygons by the cv2.approxPolyDP method; the image moments of the household contours are calculated by the cv2.moments method, to characterize the center of mass, moments of inertia, and other features of the household from the image moments; The contour fitting ellipse is derived by the cv2.fitEllipse method, and the parameters are extracted and embedded in the form of fixed-length vectors; the average aspect ratio information of each bedroom, living room, kitchen and courtyard are calculated by the cv2.boundingRect method, respectively, to form the aspect ratio features of the four types of subdivision spaces. The above data are embedded as floating point numbers. By the above methods, the contour information can be embedded as a feature vector composed of feature parameters, and the shape description of the household in the search intent can also be embedded as a feature vector with specific feature parameters such as aspect ratio.

To embed floor plan features into convolutional neural networks, we adopt the ImageNet pre-trained ResNet50 model for its proven architectural pattern recognition capabilities and computational efficiency. ResNet50's residual connections mitigate gradient vanishing, enabling stable training of deep networks to capture fine-grained geometric details, particularly in non-orthogonal layouts. Its pre-trained weights accelerate feature extraction on small datasets while preserving critical spatial patterns, such as door and window contours, without requiring fine-tuning. The model's optimized depth, with four residual groups containing 3, 4, 6, and 3 blocks respectively, ensures hierarchical feature extraction within computational limits, maintaining approximately 25.6 million parameters and requiring ~3.8–4.1 GFLOPs per 224 $\times$ 224 input.

Table 2. Some of the parameters used in the contour information embedding.

Information Type	Feature Parameters	Calculation Method of Parameters
Outer Contour	Outer contour fitting polygon	Calculate the list of coordinates of the fitted outer contour polygon after normalization
Shape parameters of Outer Contour	Image moment of the exterior contour	Calculate the center of mass and moment of inertia of the outer contour
	Fitting elliptic parameters	Calculate the fitted ellipse parameters of the outer contour
	External rectangular parameters	Calculate parameters such as the aspect ratio of the house type and the area ratio to the external rectangle
Shape parameters of the internal space	Aspect Ratios of Rooms	Calculate the aspect ratio of the bedroom, living room, courtyard, and other spaces

Input images are standardized to 224 × 224 resolution to leverage pre-trained weight optimizations, balance geometric fidelity, and comply with GPU memory constraints (batch size = 256). The modified architecture removes the final fully connected layer, instead using global average pooling to output 2048-dimensional feature vectors. These continuous vectors are subsequently discretized via similarity sorting, preserving integer-based ranking integrity without quantization loss. Validated on ImageNet and diverse vision tasks, ResNet50's reproducibility and established academic support make it an ideal choice for our application, ensuring both computational efficiency and reliable feature extraction. The model structure is shown in Figure 9.

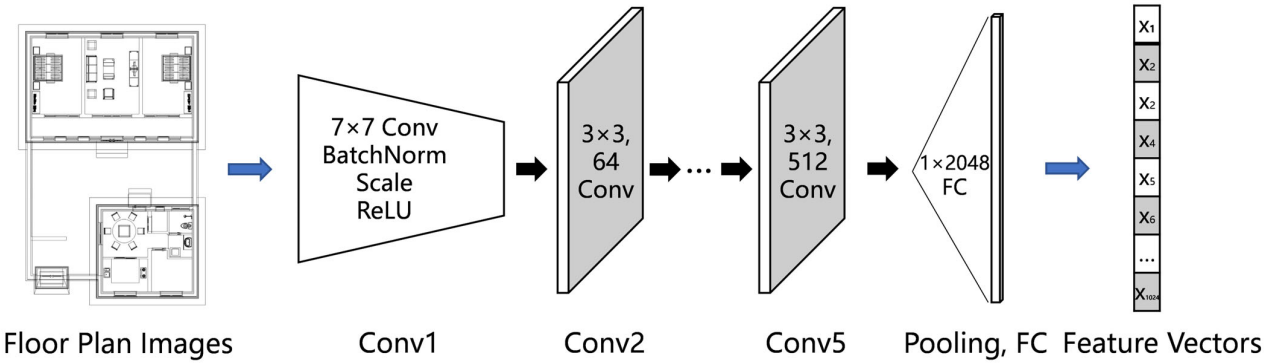

Figure 9. ResNet50-based feature embedding method for floor plan images.

3.3.2. Similarity-Based Ranking BIM Models Searching

As shown in Figure 8, to perform the search from text to BIM models, this subsection performs the integrated similarity calculation based on the feature vectors of the BIM model and search intent and achieves the search of the BIM model by similarity ranking. Similarity-based ranking refers to the process of sorting BIM models based on their comprehensive feature alignment with the query, where higher-ranked models exhibit stronger matches across semantic, topological, and geometric dimensions.

Elements of the feature vectors have different compositions, magnitudes, and meanings, so a more nuanced approach is required for calculating the similarity between different BIM models and between text queries and the BIM model, and then the intelligent retrieval of the BIM model can be realized by similarity ranking. Specifically, for string elements, such as Location, the name of the room, etc., the character matching rate is used as a measure of similarity; for floating-point and integer quantitative elements, such as area and count of bedrooms, relative differences are used to measure similarity; for sub-vector

elements with specific meaning, such as word vectors and fitted rectangular features of topological maps, image moments of shapes, etc., similarity is also measured by taking the cosine similarity of the sub-vectors; for the list of fitted outer contour coordinates in shape features, OpenCV's cv2.matchShapes method is introduced to compare the similarity of the contours by the relative differences of Hu moments of the contours. These tailored similarity calculation methods ensure that each type of feature is compared accurately. The similarity calculation indexes for some typical elements are shown in Table 3.

Table 3. The similarity calculation indexes for some typical elements.

Element Types	Feature	Similarity Indexes
String elements, such as Location, the name of the room, etc.	Semantic	Matching rate
Quantitative elements, such as area and count of bedrooms, etc.	Semantic	Relative differences
Sub-vector elements, such as word vectors and image moments of contours, etc.	Topological	Cosine similarity
Coordinate list of fitted outer contour elements	Geometric	Relative differences in Hu moments

In addition to the similarity index, another major focus in calculating the integrated similarity is to determine the weights of each category of features. Considering the size of the search dataset, this study uses expert manual feedback moderation to establish the weights. The initial weights are firstly determined manually by experts and then sorted in descending order based on the integrated similarity, resulting in a list of BIM model search results. The weight of comprehensive similarity is adjusted by feedback in the way that experts manually evaluate the search results. After 10 rounds of manual alignment, the results of the integrated similarity have satisfied the expert evaluation, and the corresponding weights of each feature element are shown in Table 4. The percentage of semantic feature weight value is about 88%, topological feature weight is about 5%, and geometric feature weight is about 7%. This is consistent with the fact that the input information in the actual BIM model search is mainly semantic attribute information, while the description of topological and shape information is less common. By the weighted feature vector cosine similarity ranking, the corresponding BIM gallery search ranking algorithm is formed, which has the advantages of fast calculation speed and can optimize the search effect by adjusting the weights.

Table 4. The weights of each feature element for similarity calculation.

Feature Name	House Name	House Location	Floor Count	House Area	Room Count	House Topology	Aspect Ratio	Image moment	Other Features
Weights	0.205	0.205	0.023	0.159	0.273	0.045	0.023	0.023	0.045

4. Experiments and Results

4.1. BIM Models and Search Text Datasets

The BIM model dataset used in the BIM model searching experiment covers a total of 110 rural house models, which are derived from the standard design drawings of rural residential houses provided by the governments of Beijing, Nanjing, Suzhou, Lianyungang, Yangzhou, Yancheng, and other cities, with strong typicality and representativeness. All BIM models were built manually using a unified modeling standard and Revit 2016. The IFC export program built into Revit was used to export all 110 Revit models in the "Phase 1"

stage to IFC4 format. The first level of space boundary was set and a two-dimensional room boundary was used to calculate the room volume. Partial models of the dataset are shown in Figure 10.

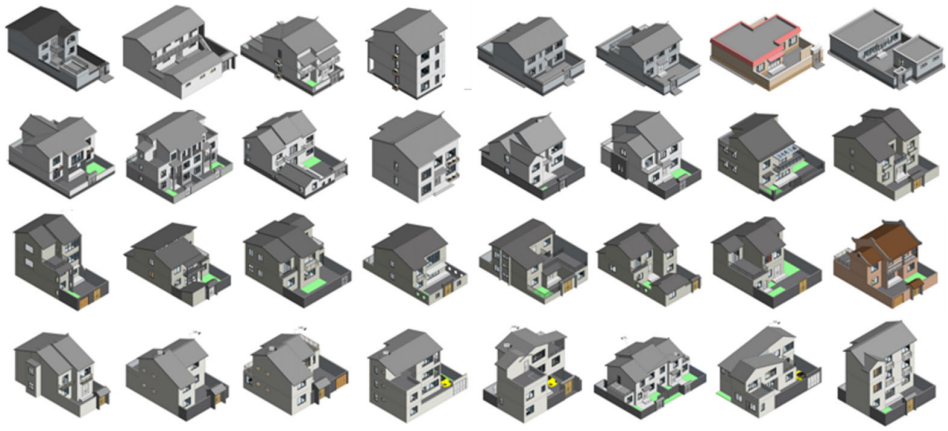

Figure 10. Partial models of the housing BIM model dataset.

By selecting indicators such as source city, floor count, area, and counts of rooms for each BIM model, the statistical overview of the BIM model dataset is shown in Figure 11 and Tables A1 and A2 in the Appendix A. It can be seen that most of the housing types in the dataset used for the search are three-bedroom units with two floors, with areas ranging from 100 to 200 square meters. They commonly have two living rooms, 2–4 bedrooms, 2–3 bathrooms, 1 kitchen, and 1–2 courtyards, which meet the needs of rural life and production.

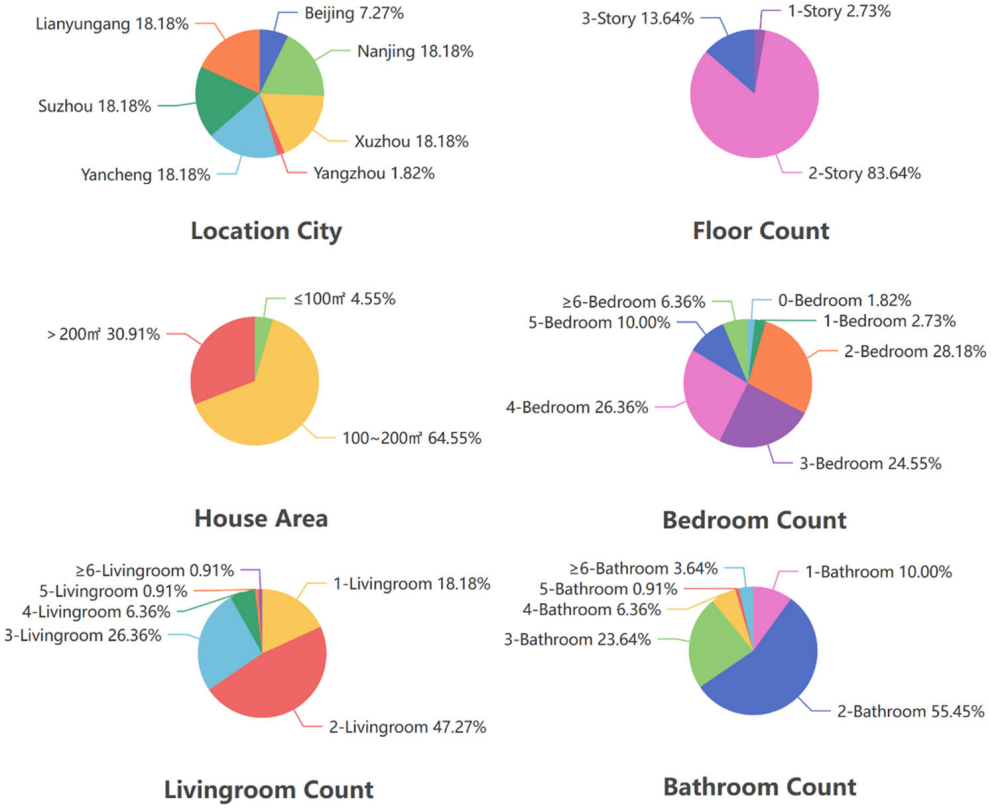

Figure 11. *Cont.*

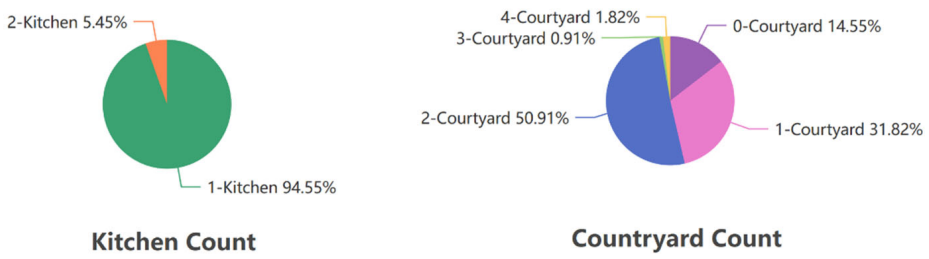

Figure 11. The statistical overview of the BIM models in the housing BIM model dataset.

Meanwhile, a search sentences dataset containing 50 natural sentences was produced to test the computational speed and accuracy of the proposed method, and some sentences from the dataset are shown in Table 5. The statistical overview of the search targets of the search sentences dataset is shown in Figure 12 and Tables A3 and A4 in the Appendix A. On one hand, the search sentences describe the semantic attributes such as location, count of floors, area, count of rooms, etc., and on the other hand, some sentences also describe the geometric features of the house and the topological connections of specific rooms such as master bedroom to bathroom, second bedroom to master bathroom, kitchen to the dining room, etc. The labels of the top 3 correct search results for the search statement in the housing BIM model dataset are tagged through expert manual search. Specifically, for a given search sentence, each of the three experts selects some BIM models that they think are most similar to the search queries and records the time required by the three experts to complete the retrieval of the entire search sentences dataset, which can be easily compared with the calculation time of the proposed automatic model. These top three BIM models chosen by the experts serve as the labels for the top three correct search results for each search sentence, providing a standard for evaluating the accuracy of the intelligent BIM searching method.

Table 5. Some sentence examples of the search sentence datasets.

No.	Search Queries
1	We are looking for a 2-story house in Beijing with a square layout, 3 bedrooms, 1 living room, 1 bathroom and 1 dining room each. The living room can be connected to the garden, and the area should be nearly 200 square meters.
2	I'm looking for a 2-story house with a slender floor plan, 100 square meters, three bedrooms, and one bathroom in Nanjing, Jiangsu this year. The master bedroom is connected to the second bedroom and the living room is connected to the balcony.
3	Looking for a 3-story house in Xuzhou, with a square floor plan, nearly 300 square meters, requiring 5 bedrooms, 4 bathrooms, and 3 baths. The living room is attached to the bathroom and the master bedroom with bathroom.
4	I need a 2-story 120 square meter house in Yangzhou with a square floor plan, including 3 bedrooms, 2 living rooms and 2 bathrooms, and a master bedroom with a bathroom.
5	In Yancheng, I need a two-story house with a slender floor plan, within 200 square meters, with 4 bedrooms, 2 halls, and 3 bathrooms, with the living room connected to the balcony and the bedroom connected to the living room.

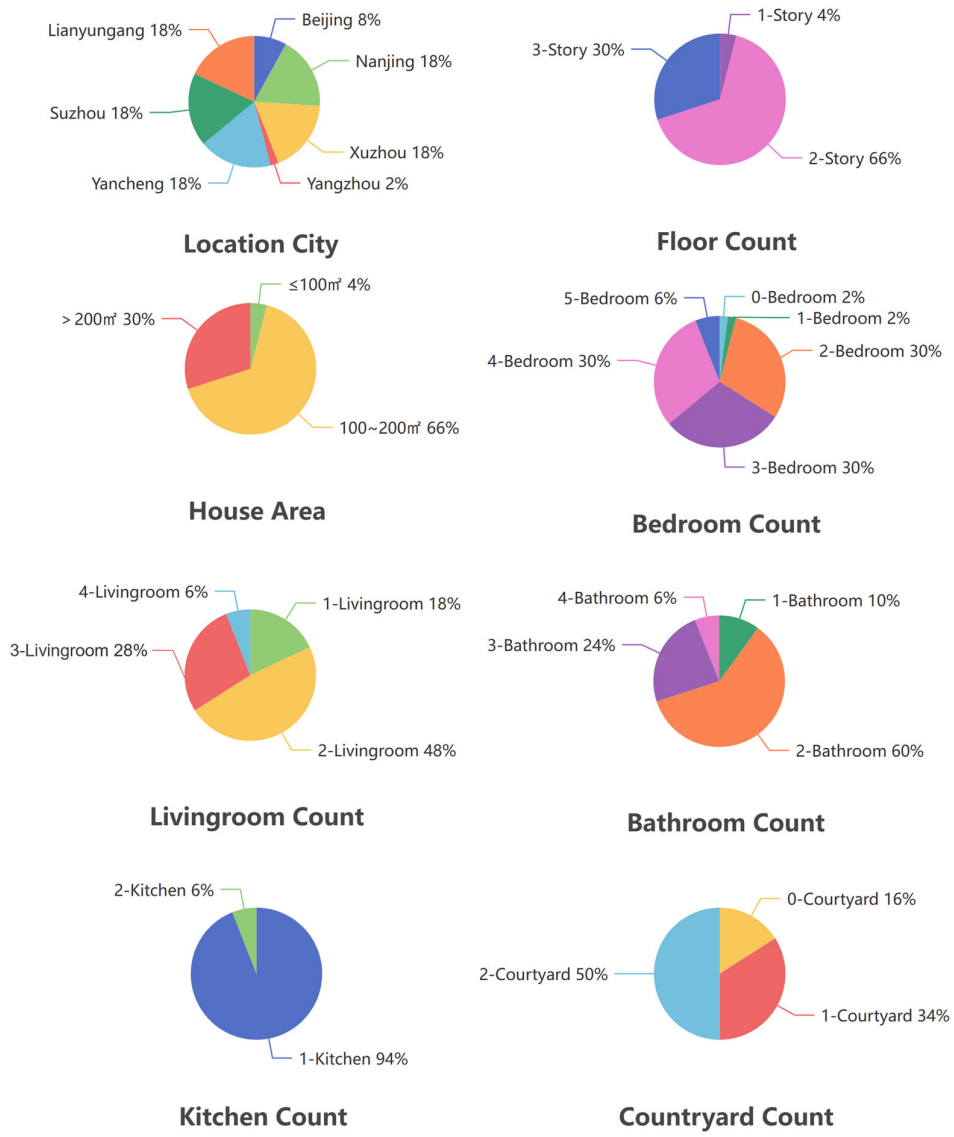

Figure 12. The statistical overview of the search targets of the search queries.

4.2. Experiment Design

To assess the performance of the GSTFs searching method, an experimental design was implemented using a dataset of 50 natural language sentences and a housing BIM model dataset containing 110 BIM models. The efficiency of the method was evaluated by comparing the time taken by three experts to manually retrieve the 50 search statements with the time taken by the search script. The manual retrieval process by experts took an average of 1740 s, whereas the script executed the search in only 2.23 s on an Apple Silicon M1 Pro with 16 GB of memory. This indicated an efficiency improvement of 780 times over manual retrieval and a reduction in total time by nearly three orders of magnitude.

For the accuracy and effectiveness evaluation, the method was benchmarked using the commonly adopted search engine evaluation metrics: mNDCG (mean Normalized Discounted Cumulative Gain), mP (mean Precision), and mAP (mean Average Precision) [35,36]. These metrics were chosen to analyze the first ten query results of the search experiment. The mNDCG@*N* metric reflects the ranking quality of the first *N* search results, with values ranging from 0 to 1. A value closer to 1 indicates better search result ordering. It is calculated by normalizing the DCG (Discounted Cumulative Gain) with the ideal DCG, which represents the best possible ordering of results. The mP@*N* metric reflects the accuracy of the first *N* search results, calculated as the number of relevant results in the top

N divided by N. It provides a measure of how many of the top results are relevant. The mAP@N metric evaluates the average precision of the first N query results, which is the mean of the precision values for each relevant result retrieved. It is calculated by summing up the precisions at each position where a relevant document is retrieved and then dividing by the total number of relevant documents.

Given that only the first 3 search results are labeled in the dataset, mP@N and mAP@N are normalized to the 0 to 1 interval for $N > 3$ and are denoted as mP@N* and mAP@N*. The calculation formulas for mP@N* and mAP@N* are shown in Formulas (1) and (2).

$$mP@N^* = \frac{\text{mP@}N}{\text{mP@3}} \quad (1)$$

$$mAP@N^* = \frac{\text{mAP@}N}{\text{mAP@3}} \quad (2)$$

The evaluation metrics of experimental results are shown in Figure 13 and Table 6.

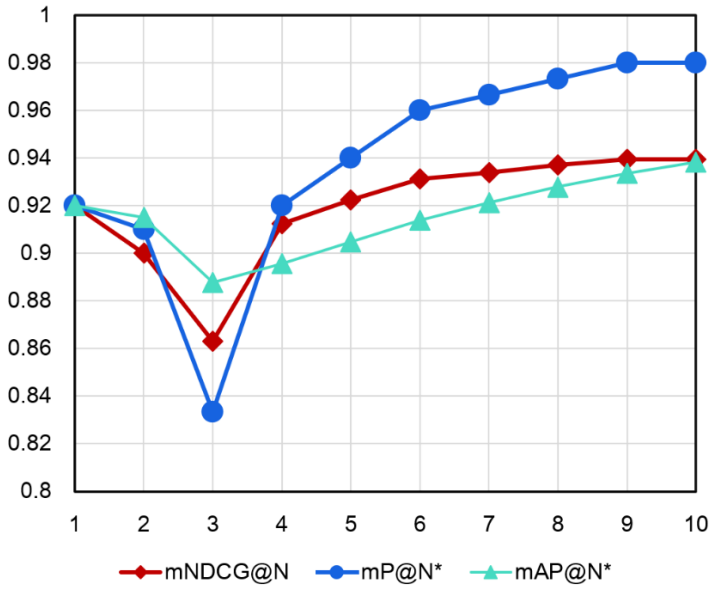

Figure 13. Performance evaluation of the proposed method with different numbers of search results.

Table 6. The values of evaluation metrics of the proposed method.

Metrics	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$	$N = 8$	$N = 9$	$N = 10$
mNDCG@N	0.9200	0.9000	0.8631	0.9126	0.9224	0.9312	0.9339	0.9372	0.9396	0.9396
mP@N*	0.9200	0.9100	0.8333	0.9200	0.9400	0.9600	0.9667	0.9733	0.9800	0.9800
mAP@N*	0.9200	0.9150	0.8878	0.8958	0.9047	0.9139	0.9214	0.9279	0.9337	0.9383

4.3. Result Analysis

The experimental results were recorded and analyzed to evaluate the performance of the GSTFs searching method. The method achieved a mNDCG@3 of 0.8631 and a mAP@3 of 0.8878, indicating high accuracy and effectiveness in search results.

For comparative purposes, a traditional search algorithm that considers only semantic features was chosen for comparison. Here, the traditional search algorithm sets the weights of topology and geometry to 0 for searching, which is used as a comparison between the classic single-feature search method and the multi-feature search method of this study. As shown in Figures 14–16 and Table 7, the comparison revealed that the intelligent BIM searching method outperformed the traditional algorithm, showing a 6.5% improvement in mNDCG@3 and a 4.3% improvement in mAP@3. Here, the 6.5% improvement

in mNDCG@3 indicates that the intelligent method achieved better-ranking quality in the top 3 search results, while the 4.3% improvement in mAP@3 reflects higher average precision, meaning more relevant results were correctly retrieved and ranked within the top 3. This significant enhancement in both accuracy and ranking performance aligns with the understanding that incorporating topological and geometric features broadens the scope of the search, leading to more relevant and accurately ranked results. Although this incremental improvement may seem modest, this approach ensures comprehensive coverage of various search scenarios, especially in edge cases where semantic information alone is insufficient. In addition, we want to make it clear that, our method is designed to retrieve entire building models rather than individual components, as in the case of BIMSeek++. This fundamental difference in scope and application makes direct performance comparisons (e.g., accuracy and speed) less meaningful. Therefore, it is meaningful enough to compare the multi-feature search results with the corresponding semantic feature search results.

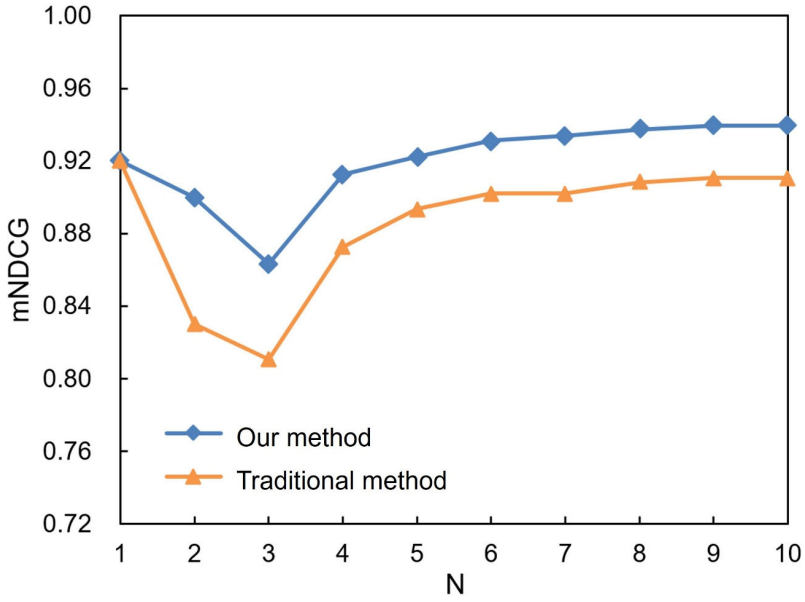

Figure 14. Comparison of mNDCG values of the GSTFs searching method and the traditional method.

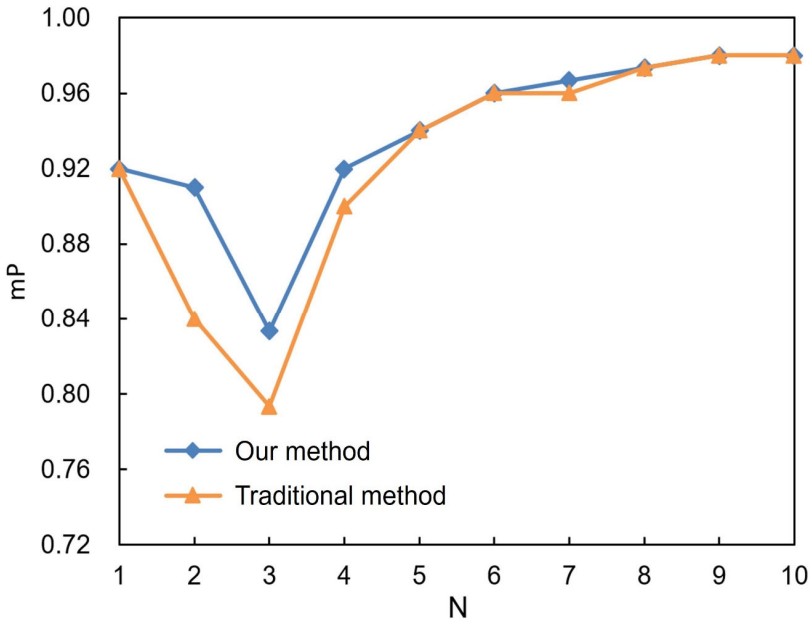

Figure 15. Comparison of mP values of the GSTFs searching method and the traditional method.

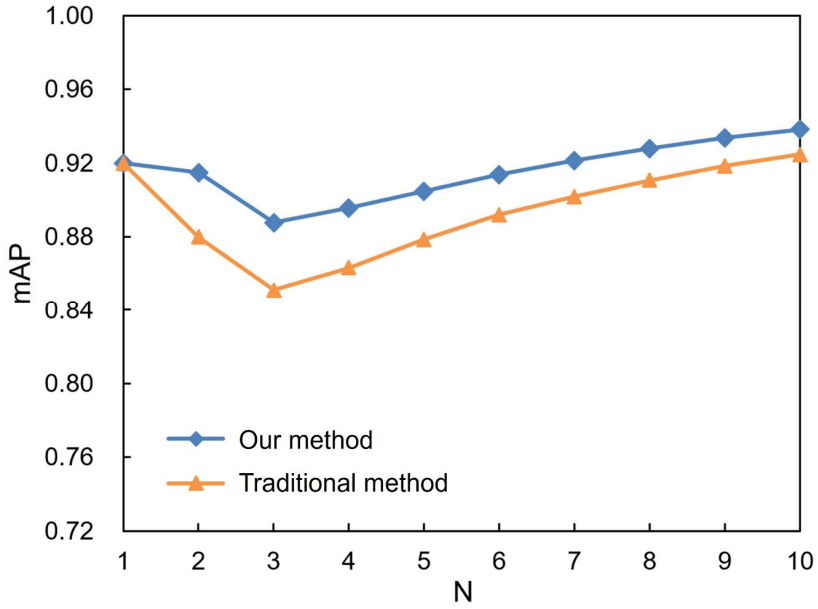

Figure 16. Comparison of mAP values of the GSTFs searching method and the traditional method.

Table 7. The values of evaluation metrics of the traditional algorithm that considers only semantic features.

Metrics	*N* = 1	*N* = 2	*N* = 3	N = 4	*N* = 5	*N* = 6	*N* = 7	*N* = 8	*N* = 9	*N* = 10
mNDCG@*N*	0.9200	0.8300	0.8108	0.8728	0.8935	0.9023	0.9023	0.9082	0.9106	0.9106
mP@*N**	0.9200	0.8400	0.7933	0.9000	0.9400	0.9600	0.9600	0.9733	0.9800	0.9800
mAP@*N**	0.9200	0.8800	0.8511	0.8633	0.8787	0.8922	0.9019	0.9108	0.9185	0.9247

5. Discussion

The proposed GSTFs searching method demonstrated significant improvements in both computational efficiency and search accuracy compared to traditional methods. The experimental results highlight the effectiveness of integrating geometric, semantic, and topological features in the search process, providing a comprehensive and robust approach to BIM model retrieval. The significant advances of our method are summarized as follows.

1. Comprehensive multimodal feature integration. Unlike traditional semantic-focused methods like BIMSeek++ [3] and NLP-based approaches [26] that primarily consider lexical properties, our framework achieves multimodal alignment of geometric, semantic, and topological features. This multimodal approach enables a more thorough understanding and matching of user search queries, thereby enhancing the accuracy and practicality of the search results. By leveraging deep learning techniques, the method transforms multimodal features from the BIM model and key information extracted from the search query into unified feature vectors. This transformation not only ensures consistency in feature representation but also significantly enhances the robustness and flexibility of the search process. The ability to effectively combine and utilize these diverse features is crucial for achieving high-quality search results, especially in complex search scenarios.
2. Excellent computational efficiency. One of the most notable outcomes of the study is the substantial improvement in computational efficiency. The proposed search method executed the search in just 2.23 s, compared to an average of 1740 s required for manual retrieval by three experts. This represents an efficiency improvement of 780 times, reducing the total time by nearly three orders of magnitude. The significant

reduction in search time is crucial for practical applications, especially in scenarios where rapid and accurate retrieval of BIM models is necessary, such as in architectural design, construction management, and facility maintenance.

3. High accuracy and superior ranking quality. The accuracy and ranking quality of the proposed method were evaluated using standard search engine metrics: mNDCG (mean Normalized Discounted Cumulative Gain), mP (mean Precision), and mAP (mean Average Precision). The results showed that the GSTFs searching method achieved high accuracy and superior ranking. Specifically, the method attained an mNDCG@3 of 0.8631 and an mAP@3 of 0.8878, indicating that the top three search results were highly relevant and well-ordered. These values are significantly higher than those of the traditional method, which achieved a mNDCG@3 of 0.8108 and a mAP@3 of 0.8511, representing improvements of 6.5% and 4.3%, respectively.

Our methodological framework demonstrates inherent scalability potential for enterprise-level BIM databases and applications. The BIM model dataset used in this experiment, covering 110 rural house models derived from standardized design drawings provided by municipal governments (e.g., Beijing, Nanjing, Suzhou, Lian-yungang), establishes a robust foundation for extensibility. These models, curated from authoritative regional design standards, exhibit strong typicality and representativeness in architectural patterns, structural systems, and regulatory compliance—key attributes shared by enterprise BIM repositories. The computational efficiency deep reinforcement learning part of the method is ensured by leveraging the pre-trained ResNet50 model, which achieves hierarchical feature extraction with optimized depth (~25.6 million parameters and ~3.8–4.1 GFLOPs per input), enabling rapid and reliable processing of geometric details within GPU memory constraints. Furthermore, the framework's reliance on IFC (Industry Foundation Classes) standardization ensures cross-platform compatibility, allowing seamless integration with diverse BIM software ecosystems (e.g., Autodesk Revit 2023, Graphisoft ArchiCAD 2023) that universally adopt IFC as an open format for model exchange.

Our methodological framework introduces several novel technical contributions and thus has some scientific innovation. Specifically, we integrated geometric, semantic, and topological features (GSTFs) into a unified search framework for the first time, and used deep learning technology to achieve the fusion and representation of multimodal features. This comprehensive method goes beyond the limitations of traditional single-feature matching and provides a new perspective for BIM model retrieval. At the same time, this method has achieved an improvement of nearly three orders of magnitude compared to manual search methods. To operationalize this innovation, we developed a practical BIM searching system, as shown in Figure 17, which demonstrates the real-world applicability of our framework.

The practical implications of the GSTFs searching method are significant. In architectural and construction industries, the ability to quickly and accurately retrieve BIM models can streamline design processes, improve project management, and enhance collaboration among stakeholders. Architects can rapidly find suitable design references, accelerating the conceptual and detailed design phases. Engineers can efficiently locate specific building components, ensuring that structural and mechanical systems are integrated seamlessly. In the context of building renovation or maintenance, the effective retrieval of BIM models through the GSTFs searching method serves as a crucial foundation for successful project execution. This advanced search capability facilitates the rapid identification of pertinent models and historical data, enabling professionals to conduct thorough analyses of the original building's structural integrity, functional characteristics, and performance metrics. The comprehensive understanding derived from these detailed models offers essential

data support for developing precise renovation strategies and implementing effective transformation plans.

Figure 17. Interface Screenshot of the BIM Searching System.

While our method demonstrates theoretical efficacy, its real-world adoption in project management faces challenges stemming from industry-specific constraints, which we formalize below: (1) Data heterogeneity and standardization barriers. As mentioned earlier, our approach relies on a predefined list of domain terms for semantic matching, which is trained on a homogeneous dataset of Chinese rural house models. This limits the generalizability to projects where terminology is not standardized. (2) The current topological feature extraction (e.g., building contours) depends on manual rules based on IFC boundary parsing, which may introduce errors for complex geometries (e.g., curved walls or non-orthogonal structures). For instance, manual rules struggle to accurately capture curvature features in spiral staircases or irregular roof contours, potentially leading to misrepresentation of spatial relationships. Additionally, the efficiency of topological feature extraction decreases with model complexity. (3) Integration with existing project management workflows. Our experiments assume an offline BIM database, while real-time collaboration tools require sub-second synchronization (e.g., clash detection during concurrent design). Construction projects often use vertically segmented software stacks (e.g., Autodesk Revit 2023 for design, Primavera P6 for scheduling), complicating real-time BIM retrieval integration. (4) 3D Complexity and Resource Constraints. Large infrastructure projects (e.g., airports and hospitals) require 3D volumetric feature extraction, which cannot be fully addressed by our current 2D geometry analysis. (5) Human Factors and Training Costs. Our user studies focused on expert annotators and did not account for the different skill levels in real teams. Project managers and field engineers may resist adopting AI-driven search tools due to reliance on traditional workflows or interpretability issues.

To further enhance the proposed method and address its limitations, the following improvements and extensions can be pursued: (1) Expanding the Dataset. The dataset used in this study is limited to rural house models from specific regions in China. Future work should expand the dataset to include a wider variety of building types and geographic locations to ensure the generalizability of the method. This will help validate the method's performance across different contexts and applications. (2) Incorporating 3D Geometric Features. Currently, the method focuses on 2D geometric features. Future research can explore advanced 3D feature extraction techniques to better capture the spatial complexity of buildings. This will enable the method to handle more intricate and detailed search queries, improving the overall accuracy and relevance of the search results. (3) Cross-Modal Alignment for Hybrid Queries. Inspired by CLIP-style multimodal embedding frameworks, integrating cross-modal alignment techniques could unify user sketches, natural language descriptions, and BIM features into a shared latent space. This would enable hybrid "sketch + text" queries, leveraging both visual and textual semantics to enhance retrieval flexibility. (4) Interpretable Attention Mechanisms. Introducing attention-based visualization (e.g., highlighting the contribution of topological connections to similarity scores) aligns with recent advancements in explainable AI for retrieval systems. By exposing feature weights, users can better understand the ranking logic and refine queries iteratively. (5) Incorporating Large-Scale Pre-Trained Models. The use of large-scale pre-trained models, such as those based on deep learning and natural language processing, can significantly enhance the method's ability to understand and handle complex search intents. These models can provide more sophisticated semantic and contextual understanding, leading to more accurate and relevant search results.

6. Conclusions

Existing studies have major limitations in the field of searching BIM models. In this research, we proposed a GSTFs searching method to extract and match geometric, semantic, and topological features from BIM and search queries. We first developed a GSTFs extraction algorithm with the help of IfcOpenShell to automatically extract semantic, topological, and geometric features from IFC-based BIM. Semantic features describe room functionality and spatial characteristics. Topological features capture room relationships through Networkx-generated network diagrams showing inter-room connections. Geometric features are extracted from floor plans and building contour coordinates. The search process employs domain word lists and Jieba for query text segmentation, followed by deep learning and feature engineering to embed both BIM model features and search query information into unified vectors for similarity-based matching. A dataset containing 110 sets of the BIM of building and a search sentences dataset containing 50 natural sentences were created and annotated by expert manual search. The experimental results demonstrate that the proposed method exhibits superior accuracy and efficiency: (1) The method has excellent computational efficiency that is 780 times greater than manual retrieval methods. (2) The method achieves high accuracy with mNDCG@3 of 0.8631, indicating that the top three search results show excellent relevance and ranking order, showing a 6.5% improvement over traditional methods. (3) The method achieves mAP@3 of 0.8878, indicating that 88.78% of the top three retrieved results are highly relevant to the search query, showing a 4.3% improvement over traditional methods.

By integrating the semantic, topological, and geometric features of BIM, this study provides a more comprehensive, accurate, and efficient method for the search and reuse of rich knowledge in BIMs. The proposed method significantly streamlines design processes, improves project management, and enhances collaboration among stakeholders, ultimately contributing to better building design and knowledge management. By facilitating rapid

and accurate retrieval of BIM models, it supports detailed analyses and informed decision-making in architectural design, engineering, and building maintenance.

While the proposed GSTFs searching method has shown significant improvements, several areas can be further explored to enhance its performance and applicability. Future work can optimize the proposed search method by incorporating more BIM models and search sentence types. Additionally, future research can explore the use of large-scale pre-trained models and 3D feature extraction techniques to better understand and handle complex search intents and 3D shapes and topological features, thus achieving more efficient BIM searches.

Author Contributions: Conceptualization, P.-H.H., S.-Y.S., Z.X., Z.-Z.H. and J.-R.L.; Methodology, P.-H.H., S.-Y.S., Z.X., Z.-Z.H. and J.-R.L. Software, S.-Y.S.; Validation, P.-H.H. and S.-Y.S.; Formal analysis, P.-H.H. and S.-Y.S.; Investigation, P.-H.H.; Resources, S.-Y.S.; Data curation, P.-H.H., S.-Y.S., Z.X. and J.-R.L.; Writing—original draft, P.-H.H. and S.-Y.S.; Writing—review & editing, Z.X., Z.-Z.H. and J.-R.L.; Visualization, P.-H.H. and S.-Y.S.; Supervision, Z.X., Z.-Z.H. and J.-R.L.; Project administration, Z.-Z.H. and J.-R.L.; Funding acquisition, J.-R.L. All authors have read and agreed to the published version of the manuscript.

Funding: The authors are grateful for the financial support received from the National Key R&D Program of China (No. 2022YFC3801100, No. 2023YFC3804600) and the National Natural Science Foundation of China (No. 52378306).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Overview of the housing BIM model dataset.

Source City	Beijing	Nanjing	Xuzhou	Yangzhou	Yancheng	Suzhou	Lianyungang	Total
Single-story	0.91%	0.00%	0.00%	0.00%	0.00%	1.82%	0.00%	2.73%
Double-story	6.36%	13.64%	16.36%	1.82%	14.55%	14.55%	16.36%	83.64%
Triple-story	0.00%	4.55%	1.82%	0.00%	3.64%	1.82%	1.82%	13.64%
100 m^2	0.00%	1.82%	0.91%	0.00%	0.00%	1.82%	0.00%	4.55%
100~200 m^2	4.55%	12.73%	15.45%	1.82%	11.82%	10.91%	7.27%	64.55%
>200 m^3	2.73%	3.64%	1.82%	0.00%	6.36%	5.45%	10.91%	30.91%
Total	7.27%	18.18%	18.18%	1.82%	18.18%	18.18%	18.18%	100.00%

Table A2. Number of models that have the corresponding count of zones in the housing BIM model dataset.

Zones Count	0	1	2	3	4	5	≥6
Bedroom	1.82%	2.73%	28.18%	24.55%	26.36%	10.00%	6.36%
Livingroom	0.00%	18.18%	47.27%	26.36%	6.36%	0.91%	0.91%
Bathroom	0.00%	10.00%	55.45%	23.64%	6.36%	0.91%	3.64%
Kitchen	0.00%	94.55%	5.45%	0.00%	0.00%	0.00%	0.00%
Courtyard	14.55%	31.82%	50.91%	0.91%	1.82%	0.00%	0.00%

Table A3. Overview of the search targets of the search sentences dataset.

Source City	Beijing	Nanjing	Xuzhou	Yangzhou	Yancheng	Suzhou	Lianyungang	Total
Single-story	2.00%	0.00%	0.00%	0.00%	0.00%	2.00%	0.00%	4.00%
Double-story	6.00%	12.00%	12.00%	2.00%	12.00%	10.00%	12.00%	66.00%

Table A3. *Cont.*

Source City	Beijing	Nanjing	Xuzhou	Yangzhou	Yancheng	Suzhou	Lianyungang	Total
Triple-story	0.00%	6.00%	6.00%	0.00%	6.00%	6.00%	6.00%	30.00%
100 m^2	2.00%	0.00%	0.00%	0.00%	0.00%	2.00%	0.00%	4.00%
100~200 m^2	6.00%	12.00%	12.00%	2.00%	12.00%	10.00%	12.00%	66.00%
>200 m^3	0.00%	6.00%	6.00%	0.00%	6.00%	6.00%	6.00%	30.00%
Total	8.00%	18.00%	18.00%	2.00%	18.00%	18.00%	18.00%	100.00%

Table A4. Number of search targets that have the corresponding count of zones in the search sentences dataset.

Zones Count	0	1	2	3	4	5
Bedroom	2.00%	2.00%	30.00%	30.00%	30.00%	6.00%
Livingroom	0.00%	18.00%	48.00%	28.00%	6.00%	0.00%
Bathroom	0.00%	10.00%	60.00%	24.00%	6.00%	0.00%
Kitchen	0.00%	94.00%	6.00%	0.00%	0.00%	0.00%
Courtyard	16.00%	34.00%	50.00%	0.00%	0.00%	0.00%

References

1. Zhang, J.; Liu, Q.; Hu, Z.; Lin, J.; Yu, F. A multi-server information-sharing environment for cross-party collaboration on a private cloud. *Autom. Constr.* **2017**, *81*, 180–195. [CrossRef]
2. Wang, T.; Chen, H.M. Integration of building information modeling and project management in construction project life cycle. *Autom. Constr.* **2023**, *150*, 104832. [CrossRef]
3. Li, N.; Li, Q.; Liu, Y.S.; Lu, W.; Wang, W. BIMSeek++: Retrieving BIM components using similarity measurement of attributes. *Comput. Ind.* **2020**, *116*, 103186. [CrossRef]
4. Khademi, H.; Behan, A. A review of approaches to solving the problem of BIM search: Towards intelligence-assisted design. In Proceedings of the CitA BIM Gathering 2017, Croke Park, Dublin, Ireland, 23–24 November 2017.
5. Zhou, Y.W.; Hu, Z.Z.; Lin, J.R.; Zhang, J.P. A review on 3D spatial data analytic for building information models. *Arch. Comput. Methods Eng.* **2020**, *27*, 1449–1463. [CrossRef]
6. Molsa, M.; Demian, P.; Gerges, M. BIM search engine: Effects of object relationships and information standards. *Buildings* **2023**, *13*, 1591. [CrossRef]
7. He, T.; Zhang, J.; Lin, J.; Li, Y. Multiaspect similarity evaluation of BIM-based standard dwelling units for residential design. *J. Comput. Civ. Eng.* **2018**, *32*, 04018032. [CrossRef]
8. Yan, J.K.; Zheng, Z.; Zhou, Y.C.; Lin, J.R.; Deng, Y.C.; Lu, X.Z. Recent Research Progress in Intelligent Construction: A Comparison between China and Developed Countries. *Buildings* **2023**, *13*, 1329. [CrossRef]
9. Zhao, Y.; Deng, X.; Lai, H. Reconstructing BIM from 2D structural drawings for existing buildings. *Autom. Constr.* **2021**, *128*, 103750. [CrossRef]
10. Chen, S.H.; Xue, F. Automatic BIM detailing using deep features of 3D views. *Autom. Constr.* **2023**, *148*, 104780. [CrossRef]
11. Funkhouser, T.; Min, P.; Kazhdan, M.; Chen, J.; Halderman, A.; Dobkin, D.; Jacobs, D. A search engine for 3D models. *ACM Trans. Graph. (TOG)* **2003**, *22*, 83–105. [CrossRef]
12. Fu, X.; Song, D.; Yang, Y.; Zhang, Y.; Wang, B. S^2Mix: Style and Semantic Mix for cross-domain 3D model retrieval. *J. Vis. Commun. Image Represent.* **2025**, *107*, 104390. [CrossRef]
13. BuildingSMART, Industry Foundation Classes (IFC). 2024. Available online: https://technical.buildingsmart.org/standards/ifc (accessed on 11 November 2024).
14. Zabin, A.; González, V.A.; Zou, Y.; Amor, R. Applications of machine learning to BIM: A systematic literature review. *Adv. Eng. Inform.* **2022**, *51*, 101474. [CrossRef]
15. Sobhkhiz, S.; Zhou, Y.-C.; Lin, J.-R.; El-Diraby, T.E. Framing and Evaluating the Best Practices of IFC-Based Automated Rule Checking: A Case Study. *Buildings* **2021**, *11*, 456. [CrossRef]
16. Ignatova, E.; Zotkin, S.; Zotkina, I. The extraction and processing of BIM data. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *365*, 062033. [CrossRef]
17. Nepal, M.P.; Staub-French, S.; Pottinger, R.; Zhang, J. Ontology-based feature modeling for construction information extraction from a building information model. *J. Comput. Civ. Eng.* **2013**, *27*, 555–569. [CrossRef]
18. Zhang, L.; El-Gohary, N.M. Automated IFC-based building information modelling and extraction for supporting value analysis of buildings. *Int. J. Constr. Manag.* **2020**, *20*, 269–288. [CrossRef]

19. Wang, Z.; Sacks, R.; Yeung, T. Exploring graph neural networks for semantic enrichment: Room type classification. *Autom. Constr.* **2022**, *134*, 104039. [CrossRef]
20. Gao, G.; Liu, Y.S.; Wang, M.; Gu, M.; Yong, J.H. A query expansion method for retrieving online BIM resources based on Industry Foundation Classes. *Autom. Constr.* **2015**, *56*, 14–25. [CrossRef]
21. IfcOpenShell 0.8.0 Documentation. Available online: https://docs.ifcopenshell.org/ (accessed on 11 November 2024).
22. Singh, T.; Mahmoodian, M.; Wang, S. Enhancing Open BIM Interoperability: Automated Generation of a Structural Model from an Architectural Model. *Buildings* **2024**, *14*, 2475. [CrossRef]
23. Lin, J.R.; Hu, Z.Z.; Zhang, J.P.; Yu, F.Q. A natural-language-based approach to intelligent data retrieval and representation for cloud BIM. *Comput.-Aided Civ. Infrastruct. Eng.* **2016**, *31*, 18–33. [CrossRef]
24. Zhou, Y.C.; Zheng, Z.; Lin, J.R.; Lu, X.Z. Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking. *Comput. Ind.* **2022**, *142*, 103746. [CrossRef]
25. Chowdhary, K.R. Natural language processing. In *Fundamentals of Artificial Intelligence*; Springer: New Delhi, India, 2020; pp. 603–649.
26. Wu, S.; Shen, Q.; Deng, Y.; Cheng, J. Natural-language-based intelligent retrieval engine for BIM object database. *Comput. Ind.* **2019**, *108*, 73–88. [CrossRef]
27. Kara, S.; Alan, Ö.; Sabuncu, O.; Akpınar, S.; Cicekli, N.K.; Alpaslan, F.N. An ontology-based retrieval system using semantic indexing. *Inf. Syst.* **2012**, *37*, 294–305. [CrossRef]
28. Xue, F.; Lu, W.; Chen, K.; Webster, C.J. BIM reconstruction from 3D point clouds: A semantic registration approach based on multimodal optimization and architectural design knowledge. *Adv. Eng. Inform.* **2019**, *42*, 100965. [CrossRef]
29. Liu, H.; Gan, V.J.; Cheng, J.C.; Zhou, S.A. Automatic Fine-Grained BIM element classification using Multi-Modal deep learning (MMDL). *Adv. Eng. Inform.* **2024**, *61*, 102458. [CrossRef]
30. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
31. He, T. BIM-Based Design Technologies for Industrialized Residential Planning. Ph.D. Thesis, Tsinghua University, Beijing, China, 2017. (In Chinese).
32. Lomio, F.; Farinha, R.; Laasonen, M.; Huttunen, H. Classification of building information model (BIM) structures with deep learning. In Proceedings of the 2018 7th European Workshop on Visual Information Processing (EUVIP), Tampere, Finland, 26–28 November 2018.
33. Xiao, Y.Q.; Li, S.W.; Hu, Z.Z. Automatically generating a MEP logic chain from building information models with identification rules. *Appl. Sci.* **2019**, *9*, 2204. [CrossRef]
34. Hu, Z.Z.; Yuan, S.; Benghi, C.; Zhang, J.P.; Zhang, X.Y.; Li, D.; Kassem, M. Geometric optimization of building information models in MEP projects: Algorithms and techniques for improving storage, transmission and display. *Autom. Constr.* **2019**, *107*, 102941. [CrossRef]
35. Stolee, K.T.; Elbaum, S.; Dwyer, M.B. Code search with input/output queries: Generalizing, ranking, and assessment. *J. Syst. Softw.* **2016**, *116*, 35–48. [CrossRef]
36. Zhao, F.; Huang, Y.; Wang, L.; Tan, T. Deep semantic ranking based hashing for multi-label image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1556–1564.