2023 17th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) | 979-8-3503-7091-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/SITIS61268.2023.00019

Neuro-symbolic approach for querying BIM models

REYNAUD Stéphane
B27-AI
Dijon, France
0009-0003-3117-7765

DUMAS Anthony
B27-AI
Dijon, France
0009-0009-1204-5488

ROXIN Ana
Laboratoire d'Informatique de Bourgogne (LIB EA 7534)
University of Burgundy
Dijon, France
0000-0001-9841-0494

***Abstract*— Integrating technologies like the Internet of Things (IoT) and artificial intelligence (AI) has transformed the smart building domain, enabling innovative services. The data gathered throughout a building's lifecycle, including construction and usage, create a comprehensive database for efficient management and improvement, mainly through Building Information Modelling (BIM). However, accessing and utilising these data can be challenging for users, as it often requires specialised query skills. Improving information accessibility involves intuitive and user-friendly query interfaces. Traditional approaches struggle to provide the flexibility and adaptability needed to keep up with evolving building information and user language expressions. To address these challenges, modern and scalable methods from symbolic and numeric artificial intelligence offer a solution. These systems can handle varying queries, adapt to changing data, and provide more user-friendly interactions, aligning better with user expectations in the industry.**

Keywords*— *Building Information Modelling (BIM), knowledge-based question answering (KBQA), ontology, artificial intelligence, neuro-symbolic.

I. Introduction

Since the arrival of technologies such as the Internet of Things (IoT) [1] or artificial intelligence (AI) [2] in the field of smart building, new innovative usages have emerged, and services have been offered to users. The evolution of smart buildings requires, in particular, consideration of innovative approaches for communication between users and buildings, focusing on providing more fluid and broader access to information about buildings [3].

Information concerning the entire life cycle of a building, starting from its design until its demolition, undergoes transformation to be recorded in databases. This information can be further enriched with data relating to the use and condition of the building, collected from various sources such as sensors and technical devices. This data aggregation helps create a rich and accurate representation of the building, providing invaluable insights for its management, maintenance and improvement [4].

Building information modelling (BIM) is the methodology of choice to share digital representation of facilities in the architecture, engineering and construction (AEC) industry [5]. BIM is defined as the "use of a shared digital representation of a built asset to facilitate design, construction and operation processes to form a reliable basis for decisions." [6]

Using these data requires a certain skill level to formulate highly structured queries outside most building users' skills. Therefore, to make these data accessible to as many people as possible, providing users with the ability to submit requests clearly and intuitively becomes crucial to ensuring accessibility and ease of use. For instance, answering the simple question "Find all the doors with height greater than 2.10m." would require the user to have the skills to write a 90-line structured query, the knowledge of the formal objects hidden behind each keyword and their relationships.

Balancing the flexibility required to formulate queries in natural language with adapting to frequently changing building information poses a significant challenge [7]. Conventional approaches, which rely on sets of pre-established rules, face several obstacles in this complex context.

First, the flexibility inherent in formulating queries in natural language is essential to facilitate interaction between users and building information systems. However, traditional methods based on rigid rules struggle to manage this variety of formulations and intentions. Building an exhaustive set of rules for every query becomes tedious and time-consuming. In addition, these rules must be constantly updated to follow the evolution of linguistic expressions linked to the available data.

Furthermore, the field of information related to buildings is constantly evolving. Buildings undergo regular modifications, updates, and additions, directly impacting the associated data. Rules-based approaches prove ill-suited to this dynamic, as they cannot easily adjust to new elements and changes in building characteristics.

These challenges highlight traditional approaches' limitations and the need to adopt more modern and scalable methods [8]. Natural language processing (NLP) systems based on AI, such as language understanding models, can adapt to variations in query wording and learn from constantly changing data. This new approach offers superior performance in terms of accuracy and usability while better aligning with the expectations of users who seek more fluid and natural interactions with building information systems. Incorporating cutting-edge technology into building practices can effectively surmount the restrictions of conventional techniques and cater to the evolving demands of users in the industry.

II. Scientific Background

The need for a structured knowledge representation of the building information stems partly from the abundant laws and regulations surrounding the AEC industry and partly from the existing information systems in use throughout this industry [9].

A. Semantic Web

Semantic Web is a framework aimed at improving the ability of computers to understand web content. Therefore, the main objective of the Semantic Web is to add meaning to the existing web content. This facilitates the comprehension

979-8-3503-7091-1/23/$31.00 ©2023 IEEE
DOI 10.1109/SITIS61268.2023.00019

of the relationships between different pieces of information by machines, enabling them to perform more intelligent tasks [10].

Semantic web technologies provide a standardised way of describing and linking data on the Web, which allows machines to process and reason about information. These technologies include Resource Description Framework (RDF), RDF Schema (RDF-S), Web Ontology Language (OWL), and the query language SPARQL [11].

B. Ontology

An ontology is a formal way of representing knowledge that outlines the concepts, their properties, and the links between them in a specific domain or subject area. This is achieved notably by providing a shared vocabulary for describing and modelling a knowledge domain. The underlying goal of ontologies is to structure information in a manner understandable to humans and machines by ensuring that data are consistently organised and interpreted.

In the Semantic Web context, ontologies are commonly described employing RDF and OWL [12]. OWL is a compelling language for creating ontologies with rich semantics.

C. Knowledge graph

A knowledge graph is a structured representation of knowledge that connects various entities (e.g., people, places, objects, concepts) through relationships. This kind of graph database stores data in a format that makes it easy to explore the connections between different pieces of information [13].

Knowledge graphs help organise and link data from multiple sources, allowing complex queries and data analysis to be performed efficiently. They are essential in many applications, such as knowledge management systems, search engines, and recommendation systems.

D. SPARQL

SPARQL is a query language for manipulating and querying data stored in RDF format. SPARQL is a key Semantic Web component that queries knowledge graphs and other RDF data sources. Indeed, SPARQL is explicitly designed for querying RDF data, a widely used format for representing structured data in the semantic Web. RDF data are made of subject-predicate-object triples, and the core of SPARQL is triple pattern matching. Variables can then be used to represent unknown parts of the triple, making it possible to retrieve data with specific patterns.

To query data from multiple RDF sources in a single query, SPARQL supports federated queries. This feature is essential to integrate data from various distributed sources seamlessly. SPARQL also defines a protocol for querying remote RDF data sources over HTTP.

III. Related Work

A. Information extraction

BIM information extraction (IE) approaches can be grouped into 3 categories.

1) Manual and rule-based IE

The simplest, though most time-consuming and labour-intensive, approach is to have human experts sort through BIM models to extract specific information.

BIM models often contain metadata associated with their elements, allowing the use of metadata extraction techniques to gather additional information.

By implementing systematic semantic tagging using standardised labels and classifications, like Industry Foundation Classes (IFC) [14], tools and scripts can be used further to streamline the extraction of information from BIM models [15].

The approaches in this category suffer from a common drawback: the requirement to have experts in each domain of a project readily available.

2) User interface-based IE

BIM tools and software applications often propose interactive user interfaces (UIs) to select elements coupled with filters to ease the retrieval of specific class or object properties [15].

This approach is convenient and efficient for a domain practitioner seeking information, but the limited capabilities inherent to UIs do not allow for fine-grained fast information extraction. Also, practitioners cannot work outside their domain efficiently, requiring the help of a domain expert.

3) Automated and query-based IE

As a step further from the first two categories mentioned above of approaches, developers can take advantage of the APIs (application programming interfaces) provided by most BIM software platforms to automate information extraction through plugins and scripts [15].

Additionally, BIM domain knowledge can be represented by creating ontologies and knowledge graphs associated with BIM models. Then, querying these knowledge structures can provide the desired specific information.

While this category of approaches provides the most advanced capabilities, they require that almost all project stakeholders master the principles of query languages and have a profound understanding of the underlying knowledge and/or data schema.

4) Natural language-based approach

Using NLP systems can alleviate the limitations of all the approaches mentioned above. Indeed, a natural language interface can effectively hide the underlying mechanisms and complexity required to properly query databases and ontologies, enabling seamless access to BIM model information without needing experts. Despite the recent advances in NLP, aligning a request expressed in natural language with the concepts contained in project-dependent BIM models remains a challenge for the scientific community [16].

B. Knowledge Base Question Answering

Knowledge Base Question Answering (KBQA) systems are a type of NLP application designed to answer questions by querying knowledge base (KB). These systems rely on structured knowledge representations, such as RDF-based knowledge graphs, to deliver precise and accurate answers to user queries. While the earliest work dates back to the 1960s and 1970s, the field of KBQA systems has significantly grown thanks to advances in NLP and the emergence of new artificial intelligence techniques. Indeed, many approaches have been developed since the 1990s and 2000s to locate and extract relevant information from unstructured texts and

improve the understanding of natural language, entity recognition, query generation, and the presentation of responses [17]. The recent advent of neural networks and large language models has made it possible to make significant progress in the precision and robustness of KBQA systems, particularly in the understanding and processing natural language [17].

Interpreting a natural language request goes beyond simply identifying specific entities in text. Semantic interpretation, understanding of syntactic context and qualification concerning a KB are involved. Notably, the KBQA systems have an extremely high overall sensitivity to named entity recognition (NER) quality.

Indeed, when entities are identified in a text, understanding their semantic meaning is critical. This involves assigning a precise meaning to these entities based on the context in which they appear. Semantic interpretation helps associate the correct meaning with the entity, which is crucial for accurately understanding the text.

Additionally, each named entity is often used in a specific context in the text. Understanding this syntactic context is essential to fully grasp the entity's role in discourse, thus helping avoid ambiguities and ensure accurate analysis of information in the text.

Lastly, the retrieved entities often have specific relevance to the underlying KB. By qualifying these entities regarding the KB, a crucial link is established between the information extracted from the text and the database, thus allowing more precise exploitation of the data for subsequent tasks.

KBQA systems work on thematic entities present in their KBs, but they must meet the challenge of processing named entities extracted from natural language questions, which are less rich in domain-specific information. This requires sophisticated extraction, linking and contextual understanding techniques to ensure accurate answers.

A review of the literature [18] outlines mainly two families of approaches in KBQA systems to process natural language queries: semantic analysis (SA) and information retrieval (IR).

1) Semantic analysis

This approach usually relies on Symbolic AI elements and can be summarised in a linear process consisting of 4 steps:

1. Syntactic and semantic understanding
2. Analysis and understanding of logic
3. Instantiation in the KB of the logical form
4. Executing the query on the KB

The SA approach has many advantages for NLP to query a KB.

First is the explainability of reasoning, essential to understanding how a model or system makes decisions. In the context of SA, the reasoning carried out to understand the meaning of sentences and formulate queries is often more transparent. This means it is possible to trace back to the logical steps and semantic connections leading to a particular conclusion or interpretation. Unlike other more opaque approaches, SA makes it possible to follow the path of answer construction, which is crucial in areas or situations where decision-making is critical.

Second is the interpretability of results, which is a challenge rarely addressed. SA aims to produce results that are more easily understandable by human beings. This approach makes results more intuitive by using techniques that highlight semantic relationships and conceptual connections. For example, SA can provide information on the key concepts that led to a classification, thus helping to interpret the decision process and the outcome.

The third is the robustness to question disturbance, which maintains performance and pertinence even when questions are formulated differently or with deviations from what was anticipated during the design. This robustness arises from the ability of SA to capture the overall meaning and semantic relationships between words and concepts rather than relying strictly on exact keyword matches. Therefore, SA can capture the underlying intent and produce coherent answers even if a question is phrased with synonyms or variations.

However, even if the SA approach offers significant advantages in terms of explainability, interpretability and robustness, it has several significant disadvantages which limit its effectiveness in certain situations.

First, one of the significant drawbacks of this approach is its narrow generalisation capacity. This means that SA may have difficulty handling cases outside its predefined operating range. For example, if the system was designed on a limited or narrow data set, it might have difficulty understanding examples containing terms, contexts, or relationships not included. This limitation can hinder this approach's ability to handle varied data and provide accurate answers in less familiar scenarios.

Second, SA requires considerable expertise to build a solid logic for analysis and understanding. This expertise can extend to system design and semantic processing rules' definitions. To obtain reliable and accurate results, in-depth knowledge of the domain language, data structures and semantic relationships is often necessary. This requirement for expertise can make implementation and maintenance of the approach complex, as well as the need to collaborate with highly qualified specialists for any changes to the system, even minor ones.

Third, SA can face scaling problems when processing large amounts of data or responding to many queries in real time. Indeed, complex semantic operations often require intensive calculations, leading to performance issues when available computing resources are limited.

Lastly, another important drawback is the high sensitivity of the approach to the complexity of the questions asked. Queries that are ambiguous, poorly worded, or contain conflicting information can lead to irrelevant or incorrect answers. As a result, the quality of SA results can vary significantly depending on the quality and clarity of the questions asked.

2) Information Retrieval

This approach usually relies on the use of Numeric AI and can be summarised in a process of 4 steps:

1. Thematic analysis for isolation of a subgraph
2. Construction of reasoning instructions by semantic parsing

3. Reasoning on the entities (and their neighbours) of the subgraph
4. Generation of the answer from the classification of candidate answer entities

Some systems repeat steps 2 and 3 iteratively.

The IR approach has specific advantages for NLP, different from SA, to query a KB.

One of the IR approach's main advantages is the possibility of end-to-end training models. This means it is possible to train a model from raw data without requiring an intermediate step of manual data processing. This can speed up the learning process and make it easier to deploy solutions.

Unlike the SA approach, where it is necessary to build complex logic to answer questions, the IR approach avoids this step by relying on correspondence patterns between queries and documents. This can make systems development simpler and less prone to human errors in logic design.

Thanks to the reduction of the necessary reasoning space, this approach shows a low sensitivity to the questions' complexity. Even for complex questions, IR models can extract relevant information by focusing on key terms and matches in the data.

However, despite these advantages of simplicity of training, management of the complexity of questions and lack of need to construct complex logic, the IR approach presents a certain number of disadvantages which can be strongly limiting.

The first major disadvantage of the IR approach lies in the opacity of the reasoning. IR models often do not clearly explain how an answer is generated. This can make understanding the model's decisions difficult, especially in situations where transparency is essential.

Also, IR models may have difficulty handling unforeseen variations in the data. Minor changes in the available data or the context of the questions may result in incorrect or inconsistent answers. This limits their ability to generalise effectively about new or unknown data.

Lastly, IR models require a massive, high-quality training dataset to achieve reliable and accurate performance. This can make their development costly, time-consuming, and complex: collecting and preparing adequate data can be a significant challenge in many specialist areas.

3) Neuro Symbolic Reasoning

The 2 previous approaches, SA and IR, can be supplemented by the neuro-symbolic reasoning (NSR) approach to reduce their limitations and improve their capabilities because NSR enables combining the capacity of neural networks to capture information from unstructured data with the power of symbolic reasoning approaches to formalise and manipulate this information in a logical and structured way [19].

Symbolic systems are designed to manipulate symbols, logical rules, and abstract representations of knowledge. They allow information to be modelled and structured in an organised manner, which leads to a reduction in the search space when solving complex problems. Unlike purely statistical approaches which may require exhaustive exploration of all possibilities, symbolic systems take advantage of the structure of knowledge to eliminate irrelevant branches of research. This allows faster convergence to valid solutions while saving computational resources.

The structured representation of these symbolic systems captures the underlying information precisely and is organised. When solving problems or making decisions, this knowledge structure allows symbolic systems to consider relationships between elements rather than treating each element in isolation. Thus, symbolic systems exploit the rich semantics of symbols to perform more sophisticated inferences and reasoning.

Compositional generalisation is the combining of elements to form new meanings. Symbolic systems are particularly suited to this form of generalisation because they can manipulate symbols and logical rules to construct complex structures from simple elements. Using symbolic representations allows systems to understand how elements fit together to form more significant concepts. This leads to more powerful generalisation, where the system can extrapolate knowledge from limited examples and apply that knowledge to new situations using logical rules and pre-existing relationships [18].

Neural systems are designed to process unstructured and complex data. They can analyse information from different sources, such as text, images, videos and tabular data. Thanks to their architecture and ability to learn patterns from training data, these systems can answer questions that require deep understanding and complex analysis.

Heterogeneous data refers to information from various sources and formats. Neural systems are suited to processing this data type because they can integrate and analyse information from different modes. For example, they can extract textual information from documents, interpret images or videos, and combine these different forms of data to produce coherent and informed responses.

KBs are data structures that store information about entities, their attributes, and their relationships. However, these KBs may be incomplete or lack key relationships. Neural systems can help fill these gaps by extrapolating from existing information and generating responses based on their understanding of the data. They can also identify implicit relationships between different entities, improving the quality and richness of available information [19].

NSR represents an approach in artificial intelligence that combines elements of traditional symbolic reasoning with techniques based on neural networks. This combination aims to take advantage of the advantages of each approach to solve complex and varied problems. The generation of logical forms in SA approaches and the optimal expansion of the reasoning subgraph in IR approaches illustrate how this fusion can be applied in the context of KBQA.

SA is concerned with understanding the meaning of words, sentences, and language in general and NSR can be used to generate logical forms that represent the semantic structure of a text. This involves translating the information contained in the text into formal logical representations, such as predicates, propositions, and relationships between entities.

Neural networks can be leveraged to perform the first step in this transformation by identifying text concepts, relationships, and contexts. For example, neural networks

such as recurrent neural networks or transformers can be used to extract semantic information from texts. Then, symbolic reasoning techniques, such as first-order logic or modal logic, can be applied to formalise this information into logical forms understandable by a computer [18].

One of the challenges in the field of IR is determining how to extend a knowledge graph or set of relevant relationships to answer a given query optimally.

NSR can guide the expansion of the knowledge subgraph using symbolic information and signals from neural networks. For example, when a query is asked, a neural network can extract contextual information from the query itself and relevant documents. Then, this information can be integrated into a symbolic reasoning framework to guide the search and expansion of the knowledge graph to provide more accurate and relevant results [19].

C. Knowledge Base

The KB and its structure constitute the foundation for understanding and interpreting user requests.

One of the international standards for the definition and exchange of data from digital building models is the Industry Foundation Classes (IFC), the current version of which is IFC 4 ADD1. However, the most used version is the IFC 2X3 TC1 version. There are ifcOWL ontologies for the different versions of IFC made available officially using the EXPRESS-to-OWL tool [20].

The ifcOWL ontology is massive and complex, mainly to meet the needs of the industry with the best coverage in breadth and depth. Thus, numerous works have led to ontologies with narrower scopes and/or simplified structures [21]–[23].

IV. Our approach

A. Objectives

Complying with existing and future laws and regulations supports the need to have explainability of reasoning and interpretability of results. Moreover, the targeted users, domain experts, need to be confident in the answers provided by our KBQA system. Therefore, we choose an SA approach.

An ontology-guided SA methodology allows for identifying and qualifying the entities once the target instances are imported into the domain ontologies. Accordingly, augmenting the existing industry ontologies must be done to adapt them to process natural language.

Thankfully, such work has been completed and published [16]: the proposed IFC Natural Language Expression (INLE) ontology, based on the official IFC version 2X3 TC1, is a precious contribution from the authors as it provides the basis used in our approach to ground entities and relations in the KB once extracted from the analysis of the request.

However, SA approaches tend to have narrow generalisations outside anticipated questions and difficulties handling complex questions. To mitigate these side effects, we choose a different approach to the syntactic and semantic understanding step of SA [16] by relying on better-performing neural networks [18], making our approach part of the NSR family, thus naming it NSR-BIM.

B. Design

The NER phase in [16] is done by searching for the presence of all the entities of the ontology in the query, then filtering and disambiguating the terms thus found. We chose to reverse this part by searching only for the terms from the query in the ontology: our approach is more straightforward and faster without presenting any drop in relevance.

Our entity search from the question aims to build triples between subjects and objects of the ontologies linked by a predicate. The dependency tree is thus created using an NLP neural network while guided by the ontology, ensuring syntactic and semantic cohesion associated with the explainability and interpretability of the results. This tree serves as the basis for constructing the query on the KB.

This dependency tree is then traversed in depth because the entities have relationships in depth instead of in width. Then, the dependency tree stores the branches in subtrees starting on the left. Thus, when a sentence includes nodes with longer branches, the latter are stored in subtrees on the right. To capture the relationships of these sentences stored in right-hand subtrees, traversing the tree in depth and to the right is preferable, as opposed to a more conventional traversal to the left.

Also, our entity search is designed to traverse the tree only once, starting from the entity considered at the current iteration and going up the tree towards its root to find the best subject and predicate for this entity, which is the object of this triple. When a node is present in the sentence, our entity search also explores the transverse branches and the possibility of a subject-predicate-object triple in a top-down search, where the subject is the entity considered at this stage. These transversal branches are explored in the subtrees on the left since the tree is traversed iteratively in-depth first towards the subtrees on the right.

Algorithm 1 Query Processing

Input: text question Qt, natural language-oriented knowledge base *Knl*, BIM model-based knowledge base *Kbim*
Output: list of items *Lout* from the *Kbim*
Algorithm:
$Lout \leftarrow \varnothing$
$Traw$ = nlp_parser(Qt) ◊ raw tree from parser
$Tdp \leftarrow \varnothing$◊ dependency tree
for each token in *Traw* **do**
  if token has children **then**
    if token is a noun **then**
      $NP_i \leftarrow$ {token, token_children} ◊ Noun phrase
    else
      $GP_i \leftarrow$ {token, token_children}◊ General phrase
    $T_{dp} \leftarrow \{NP_i, GP_i\}$
$Tkb \leftarrow \varnothing$◊ dependency tree grounded in *Knl*
for each token in *Tdp* **do**
  $Qent_i$ = build_query($token_i$)
  $Lent_i \leftarrow$ search $Qent_i$ in *Knl* ◊ List of entities
  sort $Lent_i$ on text_similarity($token_i$, $entity_i$)
  $Node\_ent_i \leftarrow \{token_i, Lent_i\}$
  $Qrel_j$ = build_query($Lent_{k \in [i-1, 0]}$, predicate, $Lent_i$)
  $Lrel_j \leftarrow$ search $Qrel_j$ in *Knl* ◊ List of relations
  sort $Lrel_j$ on text_similarity($token_j$, predicate) +
    distance($entity_k$, predicate_domain) +
    distance($entity_i$, predicate_range)
  $Node\_rel_j \leftarrow \{token_j, Lrel_j\}$
  $Tkb \leftarrow \{Node\_ent_i, Node\_rel_j\}$
$Qfinal \leftarrow \varnothing$◊ final query to be executed
for each *Node_rel* in *Tkb* **do**
  for each {subject, predicate, object} in *Lrel* **do**
    Q_i = build_query(subject, predicate, object)
    if res $\leftarrow$ search Q_i in *Kbim* **then**
      $Qfinal \leftarrow Qfinal \cup Q_i$
      break
$Lout \leftarrow$ search $Qfinal$ in *Kbim*

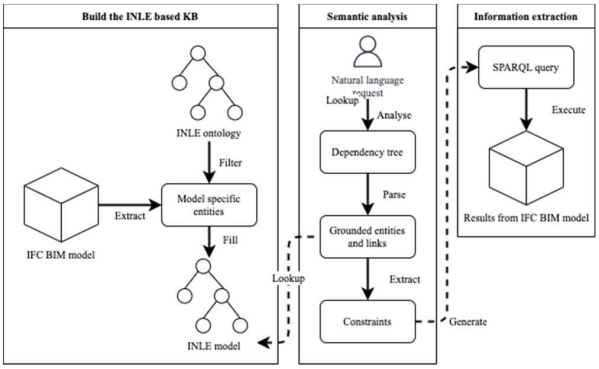

Fig. 1. Overview of our KBQA system

C. Implementation

1) Tools and applications

Protégé [1], open-source software from Stanford University, creates, maintains, and edits ontologies. It provides a graphical interface for modelling ontologies using OWL and allows testing inferences and executing SPARQL queries.

Python is chosen as the programming language for its versatility and wide choice of libraries and frameworks (third-party and native) in the areas explored. The owlready2[2] module (GNU LGPL v3 license) is a Python module used for loading, browsing, and modifying ontologies and searching, inferencing, and executing SPARQL queries.

Conda[3] is the open-source package manager that creates isolated Python environments containing packages and dependencies specific to different tools and experiments.

The minimalist and lightweight Flask[4] framework is used to develop web services in Python. It provides all the basic elements needed to create lightweight web applications.

Project management and sharing are done via GitHub, Microsoft's online platform using the Git version control system.

Microsoft's Visual Studio Code (VS Code) is the source code editor. It is light, fast, and versatile and notably offers good integration of Python, Conda and Git.

2) Natural language processing

For NLP, the main (open-source) library used is spaCy[5]. Available for Python, it is designed to perform various NLP tasks, such as tokenisation, lemmatisation, NER, parsing, or dependency tree construction. spaCy stands out for its high execution speed and precision in text processing.

NLTK[6] (Natural Language Toolkit) is another popular (open source) NLP library that offers identical functionality to spaCy. NLTK is less efficient than spaCy on many tasks but offers a broader range of functionalities, which makes it possible to test and compare other approaches and, above all, to access functionalities missing from spaCy.

Thus, spaCy allows one to carry out fewer tasks but more efficiently than NLTK, while the latter offers functionalities simply absent from spaCy. They are, therefore, an excellent complement to automatic NLP. They extensively use pretrained convolutional neural network (CNN) with different output layers based on the target task (e.g., tokenization, tagging, parsing). spaCy recently introduced pre-trained transformer-based models.

Lastly, the extreme sensitivity of KBQA systems to identifying and qualifying entities has been emphasised [18]. For the very specific and precise task of constructing the dependency tree representing a question, the BeNePar[7] (Berkeley Neural Parser) parser is used because it is quite simply the most efficient [24]. Initially developed by the University of Berkeley, it is designed to perform syntactic analysis of sentences using deep neural networks (pre-trained transformer-based large multilingual model) to improve the accuracy of this analysis. It can capture complex syntactic dependencies, allowing exact analysis in several languages. It is integrated with spaCy and NLTK.

3) Ontologies

The formal knowledge representation language used to define ontologies is OWL 2, which allows the modelling of concepts, properties, and relationships between entities. RDF/XML is the serialisation syntax used to represent data when stored as files.

The INLE ontology [16] reduces the complexity of the IFC ontology and increases access to building elements via natural language elements. It focuses on 19 architectural and structural elements and 2 spatial elements. The geometry elements are discarded. Standard and specialised natural language elements are associated with the elements of the INLE ontology, with variations (synonyms, hyponyms, acronyms, abbreviations). Moreover, the relationships between these elements are expressed by shorter paths than with the IFC ontology, sometimes using SPIN (SPARQL Inferencing Notation) properties to maintain the correspondence with IFC without unnecessarily burdening the representation scheme. To illustrate the lightweight INLE aspect when compared to IFC, Table I shows the (rounded) number of components for both schemas, alone and incorporating objects from a duplex building.

TABLE I. ONTOLOGIES' CONCEPTS & INDIVIDUALS

	IFC		INLE	
	Schema	***Duplex***	***Schema***	***Duplex***
Axioms	18,000	306,000	6,500	9,200
Classes	1,100	1,100	150	150
Predicates	1,500	1,500	60	60
Individuals	0	80,000	0	7,000

The scripts used to populate an INLE RDF/XML file from its IFC counterpart are made publicly available on GitHub[8].

D. Example

The question "Find all the doors with height greater than 2.10." is having the following output at each step of the query processing.

[1] https://protege.stanford.edu/
[2] https://owlready2.readthedocs.io/en/latest/
[3] https://anaconda.org/anaconda/conda
[4] https://pypi.org/project/Flask/
[5] https://spacy.io/
[6] https://www.nltk.org/
[7] https://pypi.org/project/benepar/
[8] https://github.com/sreynaud-b27/IFC2INLE

The raw tree from the NLP parser and the processed dependency tree are in the below table, where each token is shown with its Part of Speech (POS) tag.

TABLE II. TREE VIEWS OF EXAMPLE QUERY

Raw tree from NLP parser	Dependency tree
Find:VB |-doors:NNS | |-all:PDT | |-the:DT | |-with:IN | | |-height:NN | |-greater:JJR | | |-than:IN | | | |-2.10:CD	GP |-Find:VB |-NP | |-all:PDT | |-the:DT | |-doors:NNS | |-GP | | |-with:IN | | |-height:NN | |-GP | | |-greater:JJR | | |-GP | | | |-than:IN | | | |-2.10:CD

The token "doors" has 10 entities in the INLE ontology, with text similarity ranging from 0.29 to 1.0 (on a scale from 0 to 1). The token "height" has 20 entities with text similarity ranging from 0.29 to 1.0.

Linking the tokens "doors" and "height", 79 triples (subject, object, predicate) are considered, with a ranking score between 0.08 and 0.57 (on a scale from 0 to 1).

The representation of the query using the topmost entities and their relations from the INLE ontology is a single branch with 2 predicates:

ifcdoor1—**hascommonproperty**→ifcpsetcustom_Height —**largerthan**→Double

The execution of the built SPARQL query gives 2 results of type IfcDoor whose labels are "M_Single-Glass 1:0813 x 2420mm:0813 x 2420mm:171853" and "M_Single-Glass 1:0813 x 2420mm:0813 x 2420mm:171975"

V. RESULTS

The tests were conducted on minimal questions (12) like "Find external walls on the floor 2.". They show a sensitivity of 95.0%: failure appeared in one request. Sensitivity is here measured as the true positive rate over the set of questions. There isn't any false positive (nothing has been found that wasn't in the ground truth), so the precision is 100%. 2 items haven't been found on a total of 40 across the 12 questions, which gives a miss rate (i.e., false negative rate) of 16.7%

TABLE III. QUERY RESULTS

	TP	FP	FN
Find external walls on the floor 2.	5	0	0
Select the windows with width greater than 4.	2	0	0
Find all the beams with span larger than 5.	4	0	0
Find all the doors with height greater than 2.10.	2	0	0
Find the footing at Level 0.	7	0	0
Select all stairs on the floor 1.	2	0	0
Are there load bearing roofs?	0	0	0
Find windows whose sill height is greater than 1.5	4	0	0
Find all the load bearing beams with slope = 0.	8	0	0
Select the windows on the roof.	2	0	0
Find all the stairs.	2	0	0
Find the exterior doors with width larger than 1.	0	0	2

The ground truth has been established by exploring the BIM model, then the results of the queries have been manually reviewed to double check the results against the BIM model.

In the question that didn't receive any answer, the disambiguation on the "width" property failed to take the most appropriate property amongst the 6 candidates.

The programs were tested on a computer with an Intel ® i5 CPU (2.5 GHz), 16 GB RAM, and the Windows 10 64-bit system.

TABLE IV. QUERY EXECUTION TIME (S)

	MOP-SP [16]	NSR-BIM (Ours)
Data extraction to build the KB	170.33	31.90
Loading RDF BIM data	30	8.52
NLP preprocessing	8.67	0.01
NER and hierarchical pairing Logical relationship detection Semantic relationship extraction Value restriction extraction	10.93	0.09
Automatic code generation	10	0.02
SPARQL query execution	0.6	0.55
Total per single question	30	0.67

Importantly, our KBQA system implements straightforward and limited logical relationship detection, semantic relationship extraction, and value restriction extraction, which may partly account for the low execution time when compared to [16]. Another notable limitation in our current system is that units for measures are assumed to be raw SI (e.g., meter and not millimetre).

Python has great flexibility and ease of use but remains slow to execute. One solution could be to use Cython, the programming language that allows compiling Python code into C or C++. It helps improve the performance of Python programs by translating them into more efficient machine language. Cython also allows you to add static types and integrate calls to C or C++ functions directly into Python code.

VI. CONCLUSION AND FUTURE WORK

Querying BIM models has extensively been addressed by researchers. Given the high complexity of IFC files, extracting pieces of information from them has been a challenge for long.

KBQA systems aim to bridge the gap between human language and structured data, allowing users to ask questions in natural language and receive accurate and relevant answers based on the information stored in the KB. These systems involve several steps, including natural language understanding, query generation, NER, relationship extraction, and answer presentation. Therefore, this type of system is ideally suited to meet the challenges of querying IFC files.

In our approach to building a KBQA system to query BIM models, we use a combination of symbolic and numeric AI techniques, with the question parsing done using neural networks and NER and entity linking done using ontology-guided algorithms.

Our approach does not process compound queries i.e., several sentences, so implementing a method for resolving coreferences between sentences is a potential improvement. Constraint resolution also remains an issue: a potential approach would be to construct alternative branches when parsing the dependency tree, thus making it possible to create alternative SPARQL (sub)queries executed in case the previous ones fail. Besides, logical relationship detection, semantic relationship extraction, and value restriction extraction only straightforward use cases are managed by our approach. Future work will address consolidating the different modules to handle broader use cases. Additionally, we will thoroughly test the accuracy and speed of execution of our approach.

ACKNOWLEDGEMENT

We want to thank B27-AI for providing financial and material support for this study, the Faculty of Science and Technology at the University of Burgundy for their academic support, and the anonymous reviewers for their helpful comments.

REFERENCES

[1] M. Jia, A. Komeily, Y. Wang, and R. S. Srinivasan, 'Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications', *Autom. Constr.*, vol. 101, pp. 111–126, 2019, doi: https://doi.org/10.1016/j.autcon.2019.01.023.

[2] D. Rodríguez-Gracia, M. de las M. Capobianco-Uriarte, E. Terán-Yépez, J. A. Piedra-Fernández, L. Iribarne, and R. Ayala, 'Review of artificial intelligence techniques in green/smart buildings', *Sustain. Comput. Inform. Syst.*, vol. 38, p. 100861, 2023, doi: https://doi.org/10.1016/j.suscom.2023.100861.

[3] N. K. Gamboa-Rosales, L. D. López-Robles, L. B. Furstenau, M. K. Sott, M. J. Cobo, and J. R. López-Robles, 'Determining Technologies Trends and Evolution of Smart Building Technologies by Bibliometric Analysis from 1984 to 2020', in *Handbook of Smart Materials, Technologies, and Devices: Applications of Industry 4.0*, C. M. Hussain and P. Di Sia, Eds., Cham: Springer International Publishing, 2020, pp. 1–33. doi: 10.1007/978-3-030-58675-1_42-1.

[4] A. Watson, 'Digital buildings – Challenges and opportunities', *Adv. Eng. Inform.*, vol. 25, no. 4, pp. 573–581, 2011, doi: https://doi.org/10.1016/j.aei.2011.07.003.

[5] V. Kushwaha, 'Contribution of building information modeling (BIM) to solve problems in architecture, engineering and construction (AEC) industry and addressing barriers to implementation of BIM', *Int Res J Eng Technol*, vol. 3, no. 1, pp. 100–105, 2016.

[6] ISO 19650-1:2018, 'Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles'. 2018.

[7] J.-R. Lin, Z.-Z. Hu, J.-P. Zhang, and F.-Q. Yu, 'A Natural-Language-Based Approach to Intelligent Data Retrieval and Representation for Cloud BIM', *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 31, no. 1, pp. 18–33, 2016, doi: https://doi.org/10.1111/mice.12151.

[8] O. Kolomiyets and M.-F. Moens, 'A survey on question answering technology from an information retrieval perspective', *Inf. Sci.*, vol. 181, no. 24, pp. 5412–5434, 2011, doi: https://doi.org/10.1016/j.ins.2011.07.047.

[9] P. Pauwels, S. Zhang, and Y.-C. Lee, 'Semantic web technologies in AEC industry: A literature overview', *Autom. Constr.*, vol. 73, pp. 145–165, 2017, doi: https://doi.org/10.1016/j.autcon.2016.10.003.

[10] T. Berners-Lee, J. Hendler, and O. Lassila, 'The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities', in *Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web*, 2023, pp. 91–103.

[11] A. Patel and S. Jain, 'Present and future of semantic web technologies: a research statement', *Int. J. Comput. Appl.*, vol. 43, no. 5, pp. 413–422, 2021.

[12] A. Gómez-Pérez and O. Corcho, 'Ontology Specification Languages for the Semantic Web', *IEEE Intell. Syst.*, vol. 17, no. 1, pp. 54–60, Jan. 2002, doi: 10.1109/5254.988453.

[13] D. Fensel *et al.*, 'Introduction: What Is a Knowledge Graph?', in *Knowledge Graphs: Methodology, Tools and Selected Use Cases*, Cham: Springer International Publishing, 2020, pp. 1–10. doi: 10.1007/978-3-030-37439-6_1.

[14] ISO 16739-1:2018, 'Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema'. 2018.

[15] E. Ignatova, S. Zotkin, and I. Zotkina, 'The extraction and processing of BIM data', *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 365, no. 6, p. 062033, Jun. 2018, doi: 10.1088/1757-899X/365/6/062033.

[16] M. Yin, L. Tang, C. Webster, S. Xu, X. Li, and H. Ying, 'An ontology-aided, natural language-based approach for multi-constraint BIM model querying', *J. Build. Eng.*, vol. 76, p. 107066, 2023, doi: https://doi.org/10.1016/j.jobe.2023.107066.

[17] J. Gomes, R. C. de Mello, V. Ströele, and J. F. de Souza, 'A study of approaches to answering complex questions over knowledge bases', *Knowl. Inf. Syst.*, vol. 64, no. 11, pp. 2849–2881, Nov. 2022, doi: 10.1007/s10115-022-01737-x.

[18] Y. Lan, G. He, J. Jiang, J. Jiang, W. X. Zhao, and J.-R. Wen, 'Complex Knowledge Base Question Answering: A Survey', *IEEE Trans. Knowl. Data Eng.*, pp. 1–20, 2022.

[19] J. Zhang, B. Chen, L. Zhang, X. Ke, and H. Ding, 'Neural, symbolic and neural-symbolic reasoning on knowledge graphs', *AI Open*, vol. 2, pp. 14–35, 2021, doi: https://doi.org/10.1016/j.aiopen.2021.03.001.

[20] P. Pauwels and W. Terkaj, 'EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology', *Autom. Constr.*, vol. 63, pp. 100–133, 2016, doi: https://doi.org/10.1016/j.autcon.2015.12.003.

[21] P. Pauwels and A. Roxin, 'SimpleBIM: From full ifcOWL graphs to simplified building graphs', in *Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM)*, 2017, pp. 11–18.

[22] M. Bonduel, A. Wagner, P. Pauwels, M. Vergauwen, and R. Klein, 'Including widespread geometry formats in semantic graphs using RDF literals', in *2019 European Conference on Computing in Construction*, European Council on Computing in Construction, 2019, pp. 341–350.

[23] T. Mendes de Farias, A. Roxin, and C. Nicolle, 'IfcWoD, Semantically Adapting IFC Model Relations into OWL Properties', Oct. 2015.

[24] N. Kitaev and D. Klein, 'Constituency Parsing with a Self-Attentive Encoder', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2676–2686. doi: 10.18653/v1/P18-1249.