A Semantic Question Answering through Heterogeneous Data Source in the Domain of Smart Factory

Orçun Oruç

Technische Universität Dresden, Software Technology Group,
Nöthnitzer Strasse 46, 01187, Dresden

Abstract. Manufacturing technologies have evolved with advancements of Industry 4.0 about big data systems, generation system for linked data from unstructured data sources, and streaming data pools. A heterogeneous data source is still problematic for restricted domain question answering due to the nature of unstructured data in manufacturing companies and data-intensive applications in smart factories. Smart factories have emerged with data-intensive operations that are occurring from manufacturing monitoring systems, hand terminals, mobile tablets, and assembly line controllers. Today, human operators experience an increased complexity of the data-intensive applications in the smart factory. Fetching data from various data sources brings a necessity of decreasing data size and derive an idea regarding what is happening by inductive reasoning at the smart factory. Heterogeneous data source occurs adversity in converting linked data to get answers asked questions by human operators, experts, workers through question answering systems. When dealing with a large amount of linked data, we need to design and implement a software solution that should enhance human operators' and experts' capabilities. In this study, we propose a semantic question answering that connects with heterogeneous data sources from different areas and devices of a smart factory. In the end, we will perform qualitative and quantitative evaluation regarding the semantic question answering that exploits heterogeneous data sources, as well as findings and conclude the main points concerning our research questions.

Keywords: Semantic Web, Web 3.0, Information Retrieval, Natural Language Processing, Industry 4.0.

1 Introduction

Currently, a vast amount of unlabeled data can not be used by applications; therefore, World Wide Web Consortium (W3C) decided to create standardization of Web 3.0 called Semantic Web to apply Linked Open Data [1] concept. In this concept, hypertext ad-hoc documents of the web sites have been connected through links such as Uniform Resource Identifiers (URIs)[2]. As part of this development, Fraunhofer IWU started to organize its smart factories that are capable of generating structured linked data. Smart factories can use real-time data or linked data so as to

[1] `https://lod-cloud.net/`

[2] `https://www.w3.org/DesignIssues/LinkedData.html`

DOI:10.5121/ijnlc.2020.9403

diminish bottlenecks in assembly lines, provide predictive maintenance, enhance human-machine interaction with digitalization.

A smart factory is a highly digitized and connected production facility that relies on smart manufacturing [2]. This concept is one of the key outcomes of Industry 4.0, which intelligently changes manufacturing technologies. Smart manufacturing is a term coined by a set of departments of the United States [3]. The central power of the smart factory is that it makes data collection possible. Additionally, sensors enable the monitoring of specific processes throughout the factory that increases awareness of what is happening on multiple levels [4].

The development of *Industry 4.0* has a significant influence on the manufacturing industry. In the era of smart manufacturing systems, *Industry 4.0* needs to standardize all connection pipelines in smart factories. The primary objectives of *Industry 4.0* are making the manufacturing technologies of factories more capable of handling semantic triples, optimizing the chain of processes, and enhancing the capabilities of communication with each other. Moreover, *Industry 4.0* enforces end-to-end digital integration of engineering throughout the value chain to facilitate highly customized products, thus reducing internal operating costs [5].

The present study introduces a human-machine-interaction concept for smart factories in terms of linked data processing integrated into a question answering. The Semantic Web is a state-of-the-art research area that orchestrates the use of understanding in linked data between humans to machines and machines to machines. You can link data and documents to external data through linked data. In the present day, smart factories equipped with intelligent manufacturing devices, sensors, and actuators create a massive amount of data.

A semantic question answering is used for information retrieval to provide answers to questions through linked data. The proposed semantic question answering can understand complex natural language expressions, and it can respond to the user by answers. Mainly, the semantic question answering system employs unstructured data or structured data. We obtain linked data generated by an OPC-UA Server named *Dynamic Server* and the *eniLINK* [1] streaming data. The empirical analysis indicates the answer return rate and precision; therefore, it evaluates the usability for a human operator, experts, or an end-user web application. The goal of this research is to show an approach of semantic question answering for a smart factory that utilizes the natural language expressions as sentences, questions, or keywords to give a precise and rapid answer to human operators or experts.

With smart factories data pouring in different branches and locations such as assembly line, protocol stacks of the connected devices, and semantic data sources of a factory. Another problem regarding data sources is to consolidate data from disparate structure, unstructured and hybrid-structure is still problematic for a restricted question answering system. In the context with restricted domain question answering, annotated (structured) data is a good choice.

The question answering system is an essential part of human-computer interaction in the manufacturing industry. Human operators navigate a database of produced parts in the manufacturing data and the data with regard to the production line. The problem that we faced is a necessity of an aggregated information extraction tool at a smart factory by utilizing restricted domain linked data. Current researches do not tackle the problem as a whole in industrial manufacturing. We would like to solve the issue that can influence human operators or factory workers who spend a considerable amount of time on operating machines through smart devices. Question answering researchers generally perform research processes on the open-domain question answering. Even if they research restricted-domain question answering, industrial manufacturing and smart factory domain have never been observed before. Because of the amount of data size and semantically untagged streaming data in the manufacturing industry, we emphasize the importance of question answering for human operators and experts who work in different divisions in a smart factory.

The objective of this study is to develop a question answering providing preciseness and accuracy through Industry 4.0 lexicon (Uniform Resource Descriptor-based vocabulary). We would like to perform two major tasks, which are: construction of semantic triples and question answering utilizing the predefined semantic triples. The aspect of the construction of the semantic triples, question answering should use a common linked data format that is underlying semantic web technology. For instance, various data sources have different data types, which leads us to a conversion step to common linked data formats such as Resource Description Framework (RDF) or Ontology Web Language (OWL). In the context of question answering employing the predefined semantic triples, the semantic question answering systems rely on the initiated lexicons. In the case of open-domain questions, lexicons have standards so that a developer can use them without the burden of the conversion between data formats. Due to the fact that restricted domains have no standard question answering system, we will define various benchmarking methods to find answers to our research questions.

This paper has been structured as follows: Section 3 will provide a brief overview of semantic question answering and heterogeneous data sources from different data types such as the *Information Model* and from streaming data to the linked data. Section 4 introduces the theoretical background of natural language understanding and practical implementation of the question answering. Section 2 listed the research questions that we are going to use regarding the semantic question answering aspect of the smart factory constructed by Fraunhofer IWU. In Section 5, we implement an application and we give the implementation details of the present study. As for Section 6, we will explain the test environment; accordingly, we give the results of the semantic question answering. Section 7 explains the state-of-the-art; and then, we answer specified research questions to clarify key points with discussion in Section 8. Finally, we conclude in Section 9.

2 Research Approach

We would like to answer the following research questions throughout this study.

Research Questions:

1. *RQ-1: Can a semantic question answering utilize heterogeneous linked data sources (e.g., OPC UA Information Model, streaming data, static data) in the domain of smart factory?*
2. *RQ-2: What are the requirements of the Semantic Question Answering for smart factories?*
3. *RQ-3: Can we generalize our approach to other plants and how did we contribute to the research area?*

3 Background

3.1 Semantic Question Answering

The heterogeneous dataset can be mixed with unstructured and structured data for question answering systems; however, a common standard dataset has always been a requirement aspect of development for question answering. Linked data through Semantic Web is an abstraction level for manufacturing devices, sensors, and actuators so that designers may decouple physical and semantic layers over a smart factory. Semantic question answering can exploit disparate sources that are both structured and unstructured text. However, information format standardization can affect usability, functionality, and performance, so developers prefer to take data sources as a common structured format. This structured data can be referred to as large-scale knowledge graphs, which can cause complex inferencing in question answering. Needless to say, complex questions need complex inferencing, but we cannot assign easily score to the complexities of questions that human operators or experts asked. To show the complex capabilities of the proposed semantic question answering, we have extensively tested the application in Section 6.

3.2 Heterogeneous Data from OPC Unified Architecture

OPC Unified Architecture one of the advanced platforms that enable us to collect data from various types of devices. In this context, the Information Model of OPC UA can have the data type, data constraints, and semantics of the exposed data. One can define an object-oriented programming model in the OPC UA to bring advanced features to a smart factory application. One of the advanced features can be translating the context-less data into context-aware data-intensive applications. If existing data in a smart factory is context-aware, we can easily use it in question answering systems.

We have followed an approach for context-aware data conversion, which is identifying tree elements of a node by taking namespace indexes. The *namespace index* contains *node ids*. Once a user browses from a node to another, the user needs to know the node identification number. If the user did not scan the total number of references, the application should get all nodes that have references until the algorithm reaches all of the mesh networks. Accumulated nodes are inserted into a list to export an XML format. After obtaining XML structures, the system can convert the elements into linked data such as *Turtle RDF* through *Extensible Stylesheet Language Transformations* (XSLT). XSLT can transform from the XML format to the RDF format by minimizing the nodes without resources called blank nodes. Once the application is converted to RDF/XML format, graph libraries can deal with the conversion process into triple formats. The application takes only care of the uniform locator identifier to do a conversion, and then the application ought to arrange uniform locators by considering different from '*example.org'*.

3.3 Heterogeneous Data from Real Time Data Source

Real-time data sources have intricacies in the use of linked data taken from sensors, actuators, or software logs. In the aspect of smart factories, sensors, and actuators that are the underlying structure of manufacturing machines mostly create continuous streamed data. Fraunhofer IWU collects the real data source by saving it into a time-series database. The major drawback was that when the time series data were taken, the endpoint of a semantic query cannot use the unstructured data without annotating it. Such annotations can be the formulation of triples, insertion of predicates, or serialization from one formal language to another. The proposed architecture provides a real-time semantic data annotator that utilizes to extract triples from time-series data in a database. This work proposes a service named Key-Value Internal Service (KVIN) to perform a SPARQL request against a specified endpoint. This service is based on a combination of the triple store through the Level DB which is a key-value storage library written by the Google company1. It has been used as an RDF4J's extension to create a SPARQL Service. After obtaining time-series data, the data are mapping the SPARQL triples. These graphs contain mapped triples with their time-stamped values so that the researcher can employ values with some complex processes with SPARQL language. Moreover, a federated service replies to the queries determined by users, which reduces the answer return time of a question answering system. The KVIN does not create instantly hard-coded triples or a new language such as C- SPARQL. It only arranges the size of the time window and puts the graphs into the service to present to the end-user.

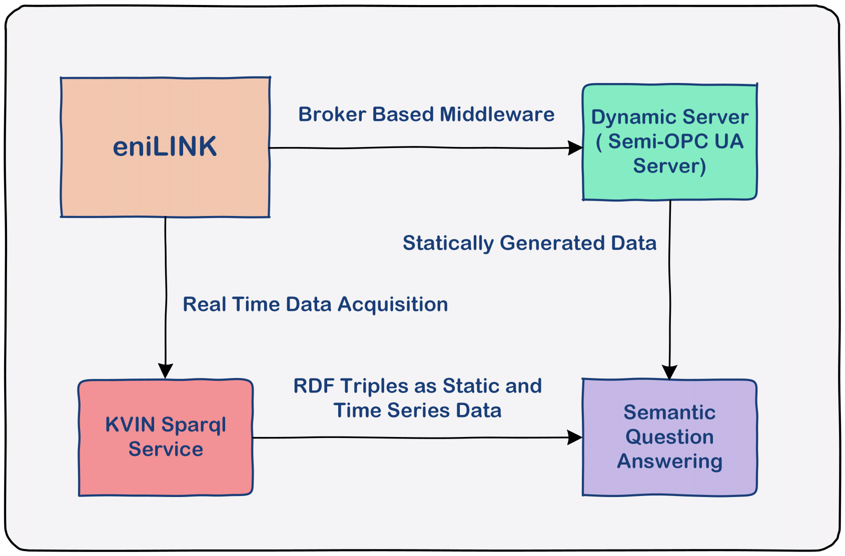
eniLINK
Broker Based Middleware
Dynamic Server (Semi-OPC UA Server)
Statically Generated Data
Real Time Data Acquisition
RDF Triples as Static and Time Series Data
KVIN Sparql Service
Semantic Question Answering

Fig. 1: KVIN Service for Real-time Data Service

3.4 Heterogeneous Data from Textual Semantic Data

This subsection refers to the conversion of textual data into the linked data. In this study, we have used the textual data because we had a system that provides textual data endpoint. Log files, alarm specific information, and daily reports regarding the manufacturing process are always saving into text files. To use kind of files, we should transform data from text files to semantic triples such as subject-predicate-objects. In this study, we have generated small scale semantic triples for hierarchical devices and divisions in the smart factory of Fraunhofer IWU. Using linked data can increase data quality because the data itself is transforming into a structured semantic format that we can validate.

4 Theory of the Natural Language Understanding

In natural language processing, we need to identify the structure of a natural expression to build up for query formulation. To overcome the complexities of natural language, we need to start with preprocessing, which means that cleaning the data for specific tasks that could be the reduction of non-optimized data and discrepancies between the values or removing non-related morphological properties.

As a next step, lemmatization and stemming should be used. Although lemmatization and stemming are similar to each other, while a stemming algorithm is used to find syntactical structures and clears out the morphological structure of suffixes and prefixes, a lemmatization algorithm looks for a semantic structure of a given input. Part-of-speech tagging is a preprocessing step for parse trees to identify item taggers such as verbs, adjectives, or nouns. A sentence consists of a couple of structures including expressions like nouns, verbs, pronouns, prepositions, adverbs, conjunctions, participles, and articles that are the main categories of part-of-speech processing [6].

The approach of parsing is two-fold, which is a rule-based approach, and the probabilistic approach [7]. The rule-based approach is a top-down approach to solve problems via predefined rules such as regex-parsing and character-based parsing. Nevertheless, the rule-based approach could give undesirable results to question answering in a restricted domain so that this will be a time-wasting and error-prone approach for this study. A dependency parser analyzes the grammatical structure of a sentence, and it gives information about the relationship among them; for example, the relationship between dependent words and root words. A constituency (phrase) parser is likely to be known as a phrase parser that has a purpose for checking the grammatical structure of sentences by parsing chunks of morphological structure. The constituency parser may not handle the relationship among language items. The dependency parser examines the grammatical structure of given natural expressions to identify the relationship between root word and dependent words that relate to the root word. In the context of parsing, named-entity recognition is a subtask of information extraction to locate and classify named entities with pre-classified labels, such as names of people, organizations, or locations.

Similarity analysis is important to detect sentence derivation and sentence similarity is used to compare two string inputs to achieve indicative questions like *"Is the system health good?"*. Mainly, this similarity method leverages averaging word vectors such as *word2vec* and *glove* that implement *Euclidean Distance*, *Manhatten Distance*, or *Cosine Similarity* [6]. Question Classification is a part of question processing that can parse the question input and assign it to the correct labels and it should be categorized to get the correct answer. Questions can be grouped with coarse-grained labels, which are *Abbreviation*, *Entity*, *Description*, *Human*, *Location*, and *Numeric.*

5 Implementation

We implement a mixed parsing based approach to define essential elements of a natural query. The major priority is to detect <*subject-predicate-object*> triples and then map the verbs and nouns onto template SPARQL. This template was created according to the requirements of a smart factory. For instance, dynamic queries that fetch information from streaming data possibly need *SUM*, *AVG*, and *MIN* filter statements of SPARQL language.

As for static queries, we have hierarchical triples that contain units of the smart factory and linked data of the *Information Model.* Listing 1.1 and 1.2 show examples regarding hierarchical triples of the smart factory of eniLINK and instantiated linked

Listing 1.1: Sample triples of the eniLINK hierarchical data [9]

```
<http://linkedfactory.iwu.fraunhofer.de/linkedfact
ory/linkedfactory/demofactory/machine10>
factory:contains
<http://linkedfactory.iwu.fraunhofer.de/linkedfact
ory/demofactory/machine10/sensor1>,
```

data of specific domains such as OPC UA. Such predicates <*factory:contains*> should be parsed and they need to be matched with verbs. However, this may lead us to a misconception to match the synonym verb of predicates. Therefore, as illustrated in Figure 2, we inserted an extra step to identify the synonym of verbs.

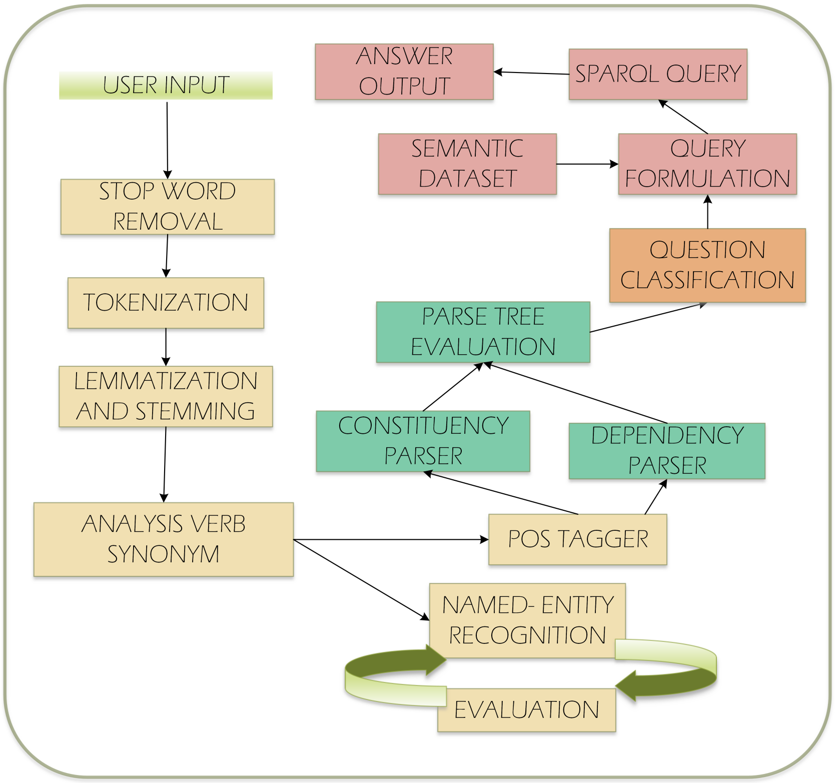

Fig. 2: Natural Language Processing for Question Answering

After taking input from any user, stop-word preprocessing starts to filter unnecessary characters such as question marks, exclamation points, commas, dots, or determiners. Tokenization is the next step to reduce the size of characters to provide optimization in natural language processing and it reduces the complexity of instances of sequence characters. Lemmatization and stemming are fundamental steps before *WordNet* verb analysis since the primary target is to extract verb, nouns, and related chunking to formulate a SPARQL query that can answer.

There is an if-else statement for the named-entity recognition after finding synonyms of the verb. As previously explained, it is a way of extracting the most common entities such as locations or names. A question answering application can face problems in identifying domain-specific names, locations, or organizations. For instance, the *linkedfactory* can be comprehensible for Fraunhofer IWU's smart factory, but another smart factory or different domain may not know what kind of entity this is. Therefore, if the question answering can catch the entity-relationship pair as shown in Figure 3, the question answering system inserts natural expressions into shallow and deep syntactic parsing.

Listing 1.2: Sample triples of the linked OPC UA Data

```
<unknown:namespace#UANodeSet/UAVariable_321> :BrowseName "0:
    ↪ MinSupportedSampleRate" ;
        :DataType "Duration" ;
        :NodeId "i=2017" ;
        :ParentNodeId "i=2013" ;
        :DisplayName <unknown:namespace#UANodeSet/UAVariable_321/
            ↪ DisplayName> .
<unknown:namespace#UANodeSet/UAVariable_321/DisplayName> rdf:value "
    ↪ MinSupportedSampleRate" .
```

For dynamic queries, the question answering system applies a similarity measurement. The similarity flag employs a sentence similarity in the following case. *"Is the system in trouble ?"* is a reasoning query. The system should interpret this query, and the system needs to know exactly the semantic meaning of the sentence. However, the above-mentioned approach is similarity-based identification. When a user asked a question *"Is the system trouble for sensor1 in machine1?"* the semantic question answering can interpret a reasoning question through machine-readable annotations.

The architecture has provided a SPARQL endpoint for local static data, and the Key-Value Internal Service (KVIN) presents a SPARQL Endpoint for time-series data. We are using different techniques for different question types. In case of a given natural language expression as below, we can specify deep and shallow parsing diagram, as depicted in Figure 3:

"Could you tell me which one contains fofab?"

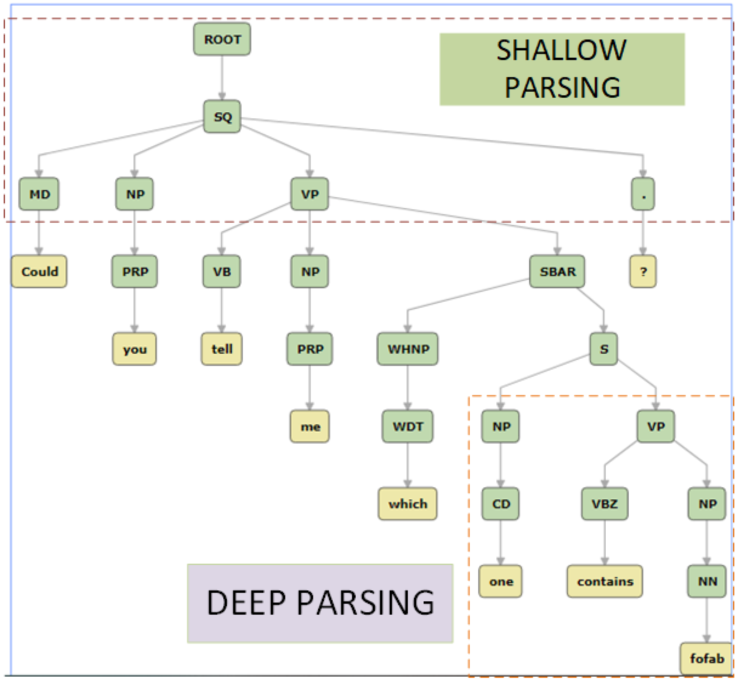

Fig. 3: An example sentence from Stanford CoreNLP [10].

We specified noun and verb phrases at a basic level so that they are using a shallow parsing that can alleviate the constituency-parsing disambiguations. If the system catches the right verb-noun pairs, it should eliminate expressions to reach the origin of the noun or verb. Such expressions may represent determiners, adjectives, or pronouns. The system has two verbs that it needs to map the predicate of triple onto the Turtle RDF data source. If it may find out the similarity level of *'contains'* and *'tell'*, the question answering could say the essential verb to be evaluated. However, the order of a verb is important for direct and indirect questions. Multiple objects have relationships with the head verbs *'tell'* and *'contains'*. Subjects and objects can inverse the order of the SPARQL query. In this case, the system needs to identify universal dependencies [3]. The named-entity recognition can show the types of relationships. A drawback of this identification is a particular keyword can perplex of the identifier, noun, etc. In essence, the question answering system needs more in-depth analyses to solve the perplexities of unique keywords and open-domain words.

[3] https://universaldependencies.org/

6 Evaluation

6.1 Test Environment

In the evaluation phase, the data sources linked data from the OPC UA Server, eniLINK linked data that consist of elements under the linkedfactory [11] and streaming data that resides in eniLINK. As previously detailed in background chapter (3), we have a heterogeneous data source for the semantic question answering. Generated data from OPC UA has no particular namespace definition unless we define it explicitly. However, the user-defined IRIs definition has drawbacks such as collision or non-extendibility. Linked data that has been instantly generated triples makes the structure complex so that two subjects of the list can collide with identical-defined IRIs. In this case, all namespaces are generated with `http://www.example.org/` and "<`unknown_namespace`>". In Table 1, answer return rate means that an answer takes round-trip time after prompting a question or keyword in the system. Querying style indicates the type of queries that we can enter and coverage shows the source of data that has been created. As for the size parameter in Table 1, the size of the dataset that we generated from OPC UA Server has 19,687, which is 2 MB sized Turtle File. The Linkedfactory triples relate to hierarchical triples that have 70 triples as Turtle format and we test the question answering with manually generated questions through *Intel Core i7-2720QM CPU @ 2.20 GHz, 2201 MHz, and x64 based Windows 10 Pro.*

As compared to an open-domain question answering dataset, we have limited semantic triples that can be utilized by the semantic question answering. Even if the size of data is relatively big in manufacturing applications, the quality of data should be annotated and the number of predicates is one of the biggest restrictions in the semantic dataset. This restriction leads us to another restriction, which is a limited vocabulary about industrial automation.

As a result, up-to-dateness supports update statement in SPARQL in a question answering system supports. Lastly, query formulation assistance displays to the end-users about the type of assistant module that is used in a question answering system.

6.2 Result

Evaluation criteria exhibit *Recall*; *Accuracy*, *Precision*, and *F1 Score* of answers against semantic question answering system, as shown in Table 2. General evaluation parameters for a restricted domain question answering are not only limited to responding to questions but also we can assess with speed, user interaction, querying style (keywords, browsing, spell checker, abbreviation recognition). In the following formulas, TP, TN, FN, and FP denote true positive, true negative, false negative, false positive respectively.

$$Prediction = TP/(TP + FP) \tag{1}$$

$$Recall = TP/(TP + FP) \tag{2}$$

$$F1 - Score = 2x(PrecisionxRecall)/(Precision + Recall) \tag{3}$$

$$Accuracy = (TP + TN)/(TP + FP + FN + TN) \tag{4}$$

The precision (1) presents an expected answer that was correctly predicted against the total responses. F1 Score (3) is a balanced weight average between the Recall and Precision. The recall (2) is the proportion of correctly answered questions with respect to the number of questions. The accuracy of the model (4) explains the model that has a ratio of accurately predicted observation to the entire inspection.

Test questions were created with a combination of keywords and elements of sentences, as listed in Table 3. Due to the domain restriction, the generation of test questions has a goal that responds to the questions precisely ranging from keywords to complex natural input. The target data source is a mixed source that combines static and streaming data. In the appendix, readers can observe combinations of test questions to use for further improvements.

Evaluation Parameters	Properties
Answer Return Rate	QA against generated data from OPC UA - 23.25 seconds average QA against static query from RDF file of the eniLINK - 18.92 seconds average QA against dynamic query from streaming data - 17.48 seconds QA against Template Based Open-Domain Questions - 20.55 seconds
Querying Style	Keywords-Based Search and Question-Based Search
Coverage	The eniLINK data, the linkedfactory streaming data
Size	Static data relatively small size Streaming data relatively large size
Up-to-dateness	No update statement provided by SPARQL
Query Formulation Assistance	Voice Input Recognition, Spell Checker

Table 1: The semantic question answering evaluation criterion

Question Answering Parameters	Total Questions
True Positive	34
False Negative	13
False Positive	3
Precision	94.44%
Recall	72.34%
F1 Score	81.92%
Accuracy	68.00%

Table 2: The Evaluation of the Question Answering (QA)

As for the limitation of the evaluation, manually generated test questions have been used for recall, accuracy, precision, and F1-Score. Moreover, the answer return rate is strongly dependent on system performance and web application design principles for the semantic question answering. Types of questions are mostly comprising of wh- questions and listing questions. However, restricted domain why questions (Why-Q) have been considered an irrelevant topic aspect of the semantic question answering but how questions (How-Q) are partly supported as readers can see in the Table 3.

7 Related Work

[Molla, Vicedo 2007] [12] reviewed primary characteristics of question answering in a restricted domain according to the integration of domain-specific information. [Molla, Vicedo 2007] [12]. defined main characteristics of question answering system over limited domains, e.g. circumscription of question answering, the complexity of question answering, and practical usage of question answering. The authors have compared between open-domain and restricted-domain question answering by figuring out key points. [Molla, Vicedo 2007] [12] offers four various aspects such as the *size of data*, *domain context*, *resources*, and *use of domain-specific resources*.

[Ferre 2012] [13] published one of the detailed reports that express common pitfalls of natural language processing and essential points while consolidating SPARQL query and morphological definitions. SQUALL is a solution for querying and updating RDF graphs by exploiting controlled natural language expressions that restrict grammar structures of a sentence to diminish complexities [13]. It has been grouped all substantial features of a morphological language, and the author pointed out what type of features in a natural language harnessed with regarding priorities and orders. The main contribution of SQUALL is categorizing ambiguities of natural expressions and how they turned an advantage out when using a controlled natural language [13].

[Biswas, Sharan, and Malik 2004] [14] proposed an architecture that extracts precise answers for a given question. The authors described the module distinctly and defined the types of questions that can be asked to the question answering. The authors sketched a translation from their intermediate language to SPARQL to gain more accuracy with their system [13]. Template-based solutions were commented on for a restricted domain and open domain question answering systems. [Unger et. al. 2012] [15] proposed a template-based solution that produces a SPARQL template, which directly matches the internal morphological features of the question.

[Chen et. al] [16] present HybridQA, which is a new large scale question answering that contains heterogeneous data sources. HybridQA has the purpose of filling the gap and construct a question answering dataset collecting from heterogeneous data sources [16]. Chiefly, they have collected dataset over tabular and textual data by reasoning and crowdsourcing from annotated Wikipedia dataset.

An application has been proposed by authors [17] that is wide-range services to personal and enterprise clients with regard to personal information, wireless information, Internet histories, and telephone information. They have stated that question sets with 120 questions and the question set were assured that every question will have an answer from the given contents of the corpus [17]. The main idea of the proposed restricted-domain question answering is to try to raise the correct candidates among ranked questions at the 10 best possible questions [17]. Authors of the research introduce some additional knowledge sources which work with geographic information system [17].

Increasing the amount of textual and linked biomedical data has many challenges as created a Biomedical Question Answering (BioQA) and [Wasim, Mahmood, and Ghani 2017] [18]. The authors have found the types of biomedical datasets which are: textual data (scientific article, authentic websites) and linked data(drug, compound, disease, and other types of the medical dataset [18]. In this survey, they have also listed the types of questions as Passage Question, Factoid Question, List Question, Multiple Choice Question, Yes/No Question, Summary Question [18]. In conclusion, they have found BioQA that ensures to exploit heterogeneous sources, perform inference, produce a summarization of answers against a given question.

Finding the right knowledge in terms of end-users due to the lack of uniformity in data sources. The authors [Katz et. al. 2002] [19] propose a solution that integrates heterogeneous data sources through an object-property-value model. Omnibase [19] can be used as a structured query interface that connects with heterogeneous data sources in the World Wide Web.

Manufacturing resources such as machine tools, robots, mobile smart devices generate a large amount of heterogeneous data in the era of Industry 4.0 [20]. Complex event processing makes us create real-time data collection of various manufacturing resources possible. [Wang, Zheng, Hu, and Fan 2018] [20] introduce a system architecture consists of three layers: the device layer, the data processing layer, and the management system layer. For instance, machine tool data is collected by MTConnect protocol, the data collection of industrial robots is coming from OPC UA, and mobile smart devices such as bar code scanner and digital gauges use a special data format [20]. Broadly speaking, we have a heterogeneous data source platform.

Linked data generation is also one of the important parts of this study because the semantic question answering is using generated linked data. [Augenstein, Pado, and Rudolph 2012] [21] propose an RDF Generation Tool that brings tokenization, named-entity recognition, parsing, lemmatization, deep semantic analysis, and word sense disambiguation together. At the end of these steps, the authors generate RDF graphs by defining URIs for the predicate and relation types provided by the system [21]. At the end of this study, they have created a tool that extracts relationships from unstructured text data. in the context with real-time streaming data, event processing is a method that works on streamed event data from various data sources, especially in sensor networks [21].

Cognitive Event Processing systems can identify and exploit contextual elements, such as time, location, domain, task, and goal [22]. The system can understand data from heterogeneous sources, including both structured and unstructured data. Cognitive Event Processing systems can prepare unstructured data to semantically linked data for question answering in the smart factory by using semantic complex event processing [22]. They have concluded that ontology-based semantic annotation is an essential part of a real-time data processing system to exploit third-party applications [22].

Industrial applications that monitor, benchmark, discover trends, and compare physical phenomena like energy, temperature, humidity are very useful for automating process in the manufacturing technology [23]. They have specified some requirements such as providing abstraction, past events detection, and usable on low-end devices [23]. According to the authors' statements in these points, heterogeneity of devices need an abstraction system to decouple the problem space from the solution spaces through semantic web technologies. When a framework turned into a semantically abstracted application, it can detect past events from a set of conditions and decouple the software and hardware without considering the protocol and communication mechanism [23].

Evaluation of a semantic question answering is still a cumbersome and hard problem. Lack of test questions that belong to a specific domain is one of the major problems. [Diekerma, Yilmazel, and D. Liddy 2004] [24] offer different methodologies from an open-domain question answering while evaluating the restricted domain question answering. The authors specify the evaluation methodology as below:

System Performance: Speed and availability **Answers**: Accuracy, Completeness **Display User Interface**: Querying styles, natural language queries, keywords, browsing, and the question formulation assistance (spell checker, abbreviation solver)

The authors stated that the *TREC* style question answering evaluation might not be suited for their restricted domain system so that user-based evaluation can be more viable to evaluate the system [24].

8 Discussion

First, RQ-1 and RQ-2 address distinct architectures for the use of semantic question answering. The proposal is implementing a service called KVIN that employs key-value mapping with windowed time-series data. The time-series data has been windowed with the size of data as well as the extent of the data size. Although the information structure is limited to be mapped onto Turtle triples, it can be useful for rapid prototyping. No cost will arise from designing a new language onto SPARQL or overhead of instant linked data creation from streamed data.

Generating test datasets still is a problematic topic for the restricted domain question answering systems because there could be some bias. For instance, the test dataset for the information technology domain is not valuable for a manufacturing domain, which restricts the testability; however, we have used the parameters of referenced research [24]. One of the findings is that the answer return rate is similar to template-based open-domain question answering [25]. If we want to get an answer relevant to *node id*, *node parent id*, *references*, and connected devices to OPC UA Servers, we need to convert the *Information Model of the OPC UA* to the linked data. Converting from the *root node* to the leaf nodes with namespaces of nodes would be enough to map onto <*subject-predicate-object*> triples. The semantic question answering should give precise answers for dynamic data and list the results of the answer against static data. Previous studies tried to solve the restricted domain question answering problem with template-based solutions by implementing a generic solution. Whereas, we perform a heuristic-based syntactic parsing to a smart factory domain. This heuristic-based approach does not guarantee optimal results in similar statements; however, it can give a high accuracy and F1-Score, as shown in Table 2.

The major problem of this proposal is that the question answering solely depends on the predicates of the data set defined by the smart factory. To solve the dependency problem, *subject-predicate-object* pairs can be recognized by deep learning methods with unstructured data. Correspondingly, the first finding is that the named-entity recognition had shown poor performance compared to the parsing method aspect of identifying noun and verb phrases. The second finding is that complex paragraphs need a complicated mechanism such as co-reference resolution. Speed is another factor that we can infer when it comes to the customization of the semantic question answering. Accordingly, a technical operator or expert cannot get an answer from streaming data within the time-constraint of a mission-critical system. The third finding is the serialization of the OPC UA can be a time-consuming task; moreover, there must be a control script to detect unaltered semantic triples.

We propose the source code [4] so that one could recognize simulation data in OPC UA Server with a script to stave off the repercussion during serializing. The last finding is that the implementation of a generalized algorithm could degrade the precision of answers but increase the scalability at the various departments in a smart factory.

9 Conclusion and Future Roadmap

Heterogeneous data sources can increase the operability and productivity of human operators and experts in smart factories. In this paper, we have introduced a restricted domain semantic question answering that takes heterogeneous data from OPC UA, real-time data streaming, and generated structured text-based data against different types of questions. The proposed application can be used as an example for future applications and it can handle a large scale of linked data from various sources. The significant findings, that are, the proposed novel approach can be used effectively to create a supervisor tool for manufacturing technologies and a synthesized human operator assistant system, which caters to a robust architecture for the aimed platform. The proposed model reduces the complexity of the normalization process and employs state-of-the-art natural language understanding toolkits.

As our future research agenda, we plan to steer our research advanced semantic question answering in time-constraint tasks over the large scale linked data. An advanced instant data generation tool for real-time data, OPC UA Information, and textual data can be developed to embrace data-intensive applications aspect of the single view. Moreover, we can use machine learning methods to train a named-entity recognition concerning the advanced instant data generation tool.

ACKNOWLEDGEMENTS

This work is funded by the German Research (DFG) within the Research Training Group Role-Based Software Infrastructures for continuous-context-sensitive Systems (GRK 1907).

[4] https://github.com/zointblackbriar/QuestionAnswering

References

1. L. D. P. IWU and F., “eniLink,” 2019. [Online]. Available: http://platform.enilink.net/
2. R. Margaret and D. Daniel, “Definition of Smart Factory.” [Online]. Available: https://searcherp.techtarget.com/definition/smart-factory
3. K. D. Thoben, S. A. Wiesner, and T. Wuest, ““Industrie 4.0” and smart manufacturing-a review of research issues and application examples,” 2017.
4. C. Team, “What is the smart factory and its impact on manufacturing?” [Online]. Available: https://ottomotors.com/blog/what-is-the-smart-factorymanufacturing
5. T. D. Oesterreich and F. Teuteberg, “Understanding the implications of digitisation and automation in the context of Industry 4.0: A triangulation approach and elements of a research agenda for the construction industry,” 2016.
6. D. Jurafsky and J. H. Martin, *Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, third edit ed., Stanford University, 2019.
7. J. Perkins, D. Chopra, and N. Hardeniya, *Natural Language Processing : Python and NLTK*. Packt Publishing, 2016.
8. P. Christen, “A comparison of personal name matching: Techniques and practical issues,” in *Proc. - IEEE Int. Conf. Data Mining, ICDM*, 2006.
9. F. IWU, “eniLINK,” 2020. [Online]. Available: http://platform.enilink.net/
10. C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” in *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. Syst. Demonstr.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 55–60. [Online]. Available: http://aclweb.org/anthology/P14-5010
11. F. IWU, “Linkedfactory Intro,” 2018. [Online]. Available: http://linkedfactory.iwu.fraunhofer.de/linkedfactory/view
12. D. Mollá and J. L. Vicedo, “Question Answering in Restricted Domains: An Overview,” *Comput. Linguist.*, vol. 33, no. 1, pp. 41–61, Mar. 2007.
13. S. Ferré, “SQUALL: A Controlled Natural Language for Querying and Updating RDF Graphs,” in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2012, pp. 11–25.
14. P. Biswas, A. Sharan, and N. Malik, “A framework for restricted domain Question Answering System,” in *Proc. 2014 Int. Conf. Issues Challenges Intell. Comput. Tech. ICICT 2014*, 2014.
15. C. Unger, L. Bühmann, J. Lehmann, A. C. N. Ngomo, D. Gerber, and P. Cimiano, “Template-based question answering over RDF data,” in *WWW’12 - Proc. 21st Annu. Conf. World Wide Web*, 2012.
16. W. Chen, H. Zha, Z. yu Chen, W. Xiong, H. Wang, and W. Wang, “Hybridqa: A dataset of multi-hop question answering over tabular and textual data,” *ArXiv*, vol. abs/2004.07347, 2020.
17. H. Doan and L. Kosseim, “Using semantic information to improve the performance of a restricted-domain question-answering system,” 07 2020.
18. M. Wasim, D. W. Mahmood, and D. U. G. Khan, “A survey of datasets for biomedical question answering systems,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 7, 2017. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2017.080767
19. B. Katz, S. Felshin, D. Yuret, A. Ibrahim, J. Lin, G. Marton, A. McFarland, and B. Temelkuran, “Omnibase: Uniform access to heterogeneous data for question answering,” 06 2002, pp. 230–234.
20. Y. Wang, L. Zheng, Y. Hu, and W. Fan, “Multi-source heterogeneous data collection and fusion for manufacturing workshop based on complex event processing,” 01 2019.
21. I. Augenstein, S. Padó, and S. Rudolph, “Lodifier: Generating linked data from unstructured text,” in *The Semantic Web: Research and Applications*, E. Simperl, P. Cimiano, A. Polleres, O. Corcho, and V. Presutti, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 210–224.

22. J. Yang, M. Ma, P. Wang, and L. Liu, "From complex event processing to cognitive event processing: Approaches, challenges, and opportunities," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, 2015, pp. 1432–1438.
23. H. Hossayni, I. Khan, and C. E. Kaed, "Embedded semantic engine for numerical time series data," in *2018 Global Internet of Things Summit (GIoTS)*, 2018, pp. 1–6.
24. A. R. Diekerma and E. D. Liddy, "Evaluation of restricted domain Question- Answering systems," *Cent. Nat. Lang. Process.*, 2004.
25. Machinalis Group, "Quepy Question Answering." [Online]. Available: http://quepy.machinalis.com/

Appendix

Question ID	Sample Questions	Precision	Recall
1	What do linkedfactory,heatmeter, and e3fabrik incorporate exactly ?	0.0	0.0
2	Provide me a combined result for IWU and e3sim	1.0	1.0
3	I want to know which one carries fofab ?	1.0	1.0
4	There is a member named fofab. Please give me all of its members	1.0	1.0
5	I am a customer of this company. Could you tell me please what the value of sensor1 of machine1 is ?	0.0	0.0
6	Could you tell me please what is the current value of sensor2 in machine2 ?	1.0	1.0
7	What POWERMETER holds ?	1.0	1.0
8	What does FOFAB incorporate ?	1.0	1.0
9	What does machine5 HOLD ?	1.0	1.0
10	What does gmx comprise ?	1.0	1.0
11	What comprises karobau?	1.0	1.0
12	System health for sensor2 in machine6	1.0	1.0
13	Tell me the health of system for sensor2 in machine1	0.0	0.0
14	Could you browse generated data ?	1.0	1.0
15	Give me all of the members of gmxspanen4	0.0	0.0
16	What holds coolingwater ?	1.0	1.0
17	What is the hierarchical structure of fofab ?	1.0	1.0
18	What contains IWU?	0.0	0.0
19	Could you give me the members in which contained by versuchsfeld ?	1.0	1.0
20	Could you give me the members in which linkedfactory has ?	1.0	1.0
21	What is the value of sensor1 in machine6 ?	1.0	1.0
22	What is the minimum that we can calculate for sensor1 of machine1 ?	1.0	1.0
23	What is the value of the maximum can be calculated by the sensor1 of machine1 ?	1.0	1.0
24	Could you tell me what the average for sensor3 in machine1 is ?	1.0	1.0
25	I need to learn an average value for sensor5 in machine2	0.0	0.0
26	What is the average of sensor3 in machine3 ?	1.0	1.0
27	Could you get me the references of nodes ?	1.0	1.0
28	Could you browse generated data ?	1.0	1.0
29	Is the E3-Sim member of linkedfactory ?	0.0	0.0
30	Could you take me all members of generated data ?	0.0	0.0
31	Give me all registered node id	1.0	1.0
32	I need to learn parent node id in generated data	0.5	0.5
33	Could you give me parent nodeID in the file of generated data ?	1.0	1.0
34	Give me all data blocks	1.0	1.0
35	Data blocks in generated OPC file	0.0	0.0
36	Give me the name of stations in generated data	0.0	0.0
37	All stations which are in generated data or new data	0.0	0.0
38	Registered node id	0.0	0.0
39	Who is Fofab ?	0.0	0.0
40	How is the system status for sensor1 in machine1?	1.0	1.0

Table 3: 40 Test Questions in order to test the application