A Semantic Question Answering System for Smart Factories

Orcun Oruc, Adrian Singer, Ken Wenzel, *Fraunhofer IWU, IEEE*

Abstract: The industrial production characterized by the increasing interconnection of machines and devices (e.g. Industrial Internet of Things, IIoT). Smart Factories evolved with its rapid requirements over the past few years. Leveraging the Industry 4.0, smart factories require a machine-to-human or machine-to-machine cooperation in order to present information to experts, users or operators. Semantically created data by production systems, mostly used by the Human Machine Interface (HMI) Devices. Nowadays, smart factories produce large amount of data that need to be apprehensible by HMI Devices. To tackle the data source problem, several proposals are introduced relating to the question answering. Consequently, a semantic question answering placed in a smart factory can use streaming data generated by production systems or statically generated data by OPC UA Servers or Semantic Tools such as eniLINK. As a result, an information model can be extracted from an extensible markup language in order to use with semantic data. Nature of linked data might have different properties from other open domain or closed domain question answering so that we will examine a restricted domain question answering system with linked data through our main architecture. Moreover, we present a semantic question answering with data based on eniLINK and generated data from an OPC UA Server known as Dynamic Server. In addition, we will evaluate our question answering with different measurements according to the requirements of a restricted-domain question answering.

***Index Terms*— Semantic Web, Web 2.0, Web Services, Information Retrieval**

I. INTRODUCTION

THIS work introduces a new concept for smart factories in terms of linked data processing integrated into a semantic question answering. The Linked Data is a new concept ubiquitously covered all around of use. As a result, a great amount of unlabeled data could not be used by applications, so W3C (World Wide Web Consortium) decided to create a standard for Semantic Web in order to create linked open data concept. Fraunhofer IWU also designs its smart factories capable of creating structured linked data. A semantic question answering used for information retrieval to provide answers from questions by means of linked data. Our semantic question answering is able to understand complex natural language inputs and it can respond back to the user by answers. Mainly, a question answering system uses unstructured data or structured data. We take ontology data generated by an OPC UA Server and eniLINK streaming and hierarchical data. The empirical analysis indicates the answer return rate and precision, moreover, it evaluates the usability for a human operator, an experts or an end-user of a web application.
The goal of this research is to show a model of semantic question answering for a smart factory utilizing the natural language inputs as sentences, questions or keywords.

II. REQUIREMENT AND APPROACHES

Requirement 1: Data source creation for the semantic question answering,

Linkedfactory Static and Dynamic Data created by KVIN Service with LevelDB (key-value storage library).
The data source is a key factor for a successful question answering system. Our question answering uses linked data created by eniLINK.

Requirement 2: Continuous streamed linked data mapping through an external service.

Requirement 3: A standard-based interface compatible with RESTful communication.

Requirement 4:

Requirement 5: Evaluation of the predefined parameters for the Semantic Question Answering
The remainder of this paper is organized as follows: Section 2 describes requirement and approaches, Section 3 defines how we can prepare real-time data. In Section 4, we clearly explain the prerequisite methods in natural language processing. Thereafter, Section 5 clearly examines the

proposed architecture. As a result, we conclude in Section 9.

We will answer the following research questions throughout the research in order to clarify key points.

At the end of this section, we might write our research questions.

Research Questions:

1) How can a semantic question answering system integrate to an OPC UA Server?

2) What are the requirements of the Semantic Question Answering aspect of Web Development and Natural Language Processing?

3) What are the characteristics associated with the architecture of the application

4) What are the key components of the approach and how did the research contribute to the research area?

III. SMART FACTORIES AND INDUSTRY 4.0

The definition of smart factories has evolved over the past few years. In the present study, a smart factory has defined an aspect of boosted technologies named Industry 4.0 and Human-Machine Interface. Impact of manufacturing development affected economic growth over the last few decades in Germany. Continuously improvement of Industry 4.0 brought the researchers to find cutting-edge technologies such as Question Answering System, Manufacturing Augmented Reality etc.

A smart factory is a highly digitized and connected production facility that relies on smart manufacturing [1]. This concept one of the key outcome of Industry 4.0, which intelligently changes manufacturing technologies. Smart manufacturing is a term coined by a set of departments of the United States [2]. The central power of the smart factory is making data collection possible. Additionally, sensors enable the monitoring of specific processes throughout the factory that increases awareness of what is happening on multiple levels [3].

The development of Industry 4.0 has a big influence on the manufacturing industry. In the era of smart manufacturing systems, Industry 4.0 is a necessity that needs to standardize all communication structures in smart factories. The primary objective of Industry 4.0 makes the manufacturing technologies of factories more intelligent, optimizing the chain of processes and enhancing capabilities of communication one to another. 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 [4].

IV. DATA PREPARATION

Our main data sources are structured semantic data source. All data source has linked triples regardless of the type of semantic data such as Turtle, RDF or OWL.

A. *Mapping an OPC UA Data into a Semantic Data*

Our main data sources are semantically parsed data from eniLINK [5] and OPC UA Server generated data. In the phase of OPC UA Server generated data, we used an SDK which is published by FreeOPCUA [6]. OPC UA Protocol uses an information model and the information model can be used with metadata to simulate in other OPC UA Server with languages such as XML (Extensible Markup Language). Due to the nature of XML, it is a language a strongly hierarchical and hardly extendable. However, semantic data such as Resource Description Framework (RDF) can employ triples with SPARQL. Different RDF Graphs stored in eniLINK[5] are uniquely identified.

Algorithm 1 Node Extraction

```
function MainFunction()                                   ▷ Starting point
    export = ServerExport(serverurl, filename)
    export.IMPORT NODES(serverurl)
    export.EXPORT FILE(outputFile, namespaces)
function Build Node Tree(nodes)                           ▷ Node Formatting
    client ← GetEndpoint()
    client ← Client(serverurl)
    nodecumulated ←None
    nodeID ←0
    for node < nodes do
        nodecumulated = node.nodeid.Namespaceindex
        for ref < node.getreferences() do
            nodecumulated.extend( ref.nodeid.Namespaceindex)
        nodecumulated = list(set(nodecumulated))          ▷ Clear duplicates
    return nodeID                                         ▷ Return node id list
function Import Nodes(serverurl)                          ▷ Traverse Node
    client = Client(serverurl)
    client.connect()
    for ns < client.getNamespaces() do
        namespaces[client.getNamespaceIndex(ns)] = ns
    root = client.getRootNode()
    child = client.iterateChildNodes()
function Export File(outputFile, namespaces = None)       ▷ Export into XML
    if namespaces != None then
        for node != nodes do
            if node.nodeid.namespaceindex is namespaces
                nodes = [node]
            else
                nodes = list(nodes)

        export = XmlExport(client) then
        export.BUILD NODE(nodes)
        export.appendXML(outputFile)
```

Figure IV-1. OPC UA Address Space Representation in XML [6] [7] **.**

A. *Real Time Data Mapping with KVIN*

Real-time data has intricacies to use as 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 into a time series database. The major drawback is that when a time series data taken, semantic query endpoint cannot use the pure data without annotating. Such annotation could be triple creation, adding of predicates or serialization from one formal language to another.

Our architecture is providing a real-time semantic data annotator KVIN that utilize to extract triples from time-series data in a database.

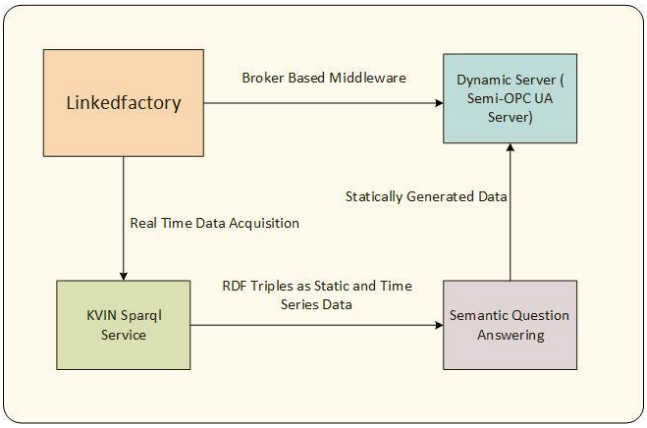

Figure IV-2. KVIN Continuous SPARQL Mapping

This work proposes a service named KVIN to perform a SPARQL request into a specified endpoint. This service is based on a combination of the triple store through Level DB which is a key-value storage library written by Google [1]. It has been used an RDF4J's extension to create a SPARQL Service.

V. NATURAL LANGUAGE UNDERSTANDING FOR THE SEMANTIC QUESTION ANSWERING

In Natural Language Processing, we need to identify the sentence's structure in order to reach the step named Query Formulation of SPARQL.

Preprocessing: Chiefly, all natural language tasks start with preprocessing which means cleaning data for specific tasks. This could be the reduction of undervalue data, reduction of discrepancies between the values or removing non-related morphological properties.

Tokenization: A question answering system should parse all input as tokens. Tokenization is an initial step for Part of Speech tagging to parse into a verb, noun, cardinal numbers, adjective etc.

Lemmatization and Stemming: Lemmatization and Stemming are similar steps to each other with one difference. While stemming used to find syntactical structures, lemmatization looks for a semantic structure. Stemming clear of the structure of suffix and prefixes. In our system, we are supposed to use a lemmatizer and stemmer in order to reduce lexical complexities. A lemmatizer has been used to consider the morphological analysis of verbs, e.g. from "contains" or "contained" to "contain". Then we use this verb mapping into a predicate to construct a SPARQL Query. The lemmatization and stemming are part of the normalization process in terms of linguistic properties.

Part of Speech Tagging: It is a preprocess step for parse tree to identify item taggers such as verbs, adjectives or nouns. A sentence consists of a couple of structure, including words like noun, verb, pronoun, preposition, adverb, conjunction, participle and article that are main categories of part of speech processing [8]. Part of Speech Tagger mostly uses a Markov Model that is a part of statistical natural language understanding. Markov Model stands for a state can depend on a previous step, but there is no dependency on states of historical steps more than one. For instance, a noun or a verb defines its neighbors, e.g. nouns are preceded by determiners, adjectives, verbs [8]. For example, a chess player makes a movement according to the last movement of a rival rather than guessing from the first movement of the rival. In this step, pre-saved corpora which have a million words have to be tagged by POS Taggers.

Parse Tree and Penn Treebank: One of the common list that has an identifier for POS denominated as Penn Treebank. A Treebank used for annotating syntactic and semantic structure of a sentence with a million words of part-of-speech tagged text.

When a natural query is given, a question answering system should understand the grammar behind it. POS tagger is not enough to identify the grammatical structure for complex natural queries. Relationships among noun phrases, adjective phrases, adverb phrases, and verb phrases should be examined in order to map correctly subject-predicate-object triples in linked data. The approach of parsing separated into two main sections, which are the rule-based approach and the probabilistic approach [9]. The rule-based approach is a top-down approach to solve problems via predefined rules such as Regex-parsing. Therefore, a question answering system should define rules precisely to get the correct answer. Open-domain question answering systems use this approach because of the complexity of the bottom-up approach and broadened question types. Nevertheless, the rule-based approach could give undesirable results in restricted-domain question answering or semantic question answering and could be a time-wasting approach. A dependency parser analyzes the grammatical structure of a sentence and it gives the relationship among them. The dependency parser also gives the relationship between general words and root words. Thus, we can identify the center verbs or nouns of complex sentences. This parser utilizes a dependency treebank file and word embedding files. Chiefly, a dependency parser applies the supervised machine learning method to reach a syntactical result. For example, with dependency treebank, data is broken into test and training set, however, word embedding used for the training phase.

Dependency Parser: //Explain
Constituency Parser: //Explain

Named Entity Recognition: It is a subtask of information extraction to locate and classify named entities with pre-classified labels such as names of people, organizations, locations, and quantities etc. Named-entity recognition is a method that identifies the item of a sentence as a domain-

[1] https://github.com/google/leveldb

specific. It identifies all structures mainly as a person, a location, an organization, and an entity.

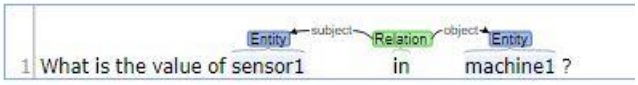

Figure V-1. Natural Language Processing Steps for Question Answering

As shown in Figure V-1, “sensor1” and “machine1” named as an entity and found a relation between each other. After doing a couple of experiments on named-entity recognition, shreds of evidence show us a named-entity recognition is an application-specific task. An NER Method that is created for a different domain may not be reused for another domain. In order to create a named-entity recognition for a smart factory, a model can be trained to satisfy the requirements of a smart factory. In this context, a model can be created by statistical methods or a rule-based model. In context with the rule-based model, the character regex method can identify the structure of a natural query. For instance, a named-entity recognizer can employ a model that contains a combination of “HeatMeter” or “HeatingWater”, which it can assert given items with the character started by “Heat” in a smart factory.

VI. PROPOSED ARCHITECTURE

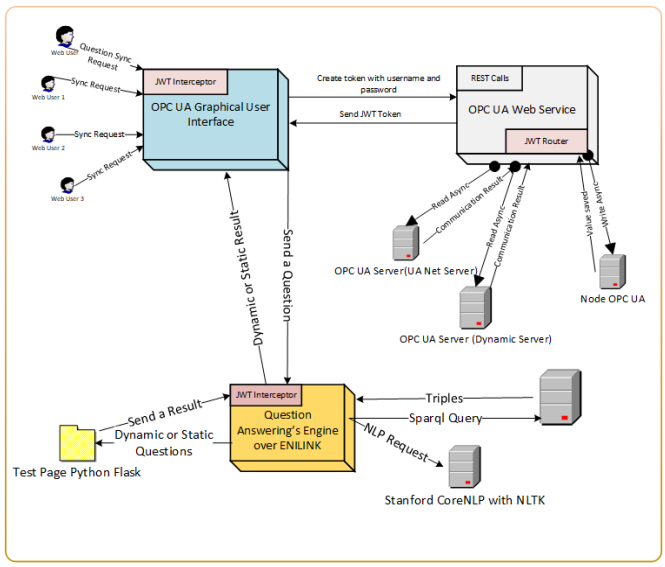

Figure VI-1. A RESTful Architecture for Question Answering

As shown above, in Figure 1, we aimed a RESTful architecture for the semantic question answering as an orange box.

```
[HttpGet("api/authenticate/) Body of Request {username, password}]
```

Listing VI-1

```
[HttpPost("/integratedstaticmessage/{question} Authentication Bearer {JWT}]
```

Listing VI-2

```
[HttpPost("/integrateddynamicmessage/{question} Authentication Bearer {JWT}]
```

Listing VI-3

NLP Requests are sent to Stanford CoreNLP server to parse grammatical structure on given the natural query. Our semantic question answering system handles with token normalization process with Tokenization, Lemmatization, and Stemming

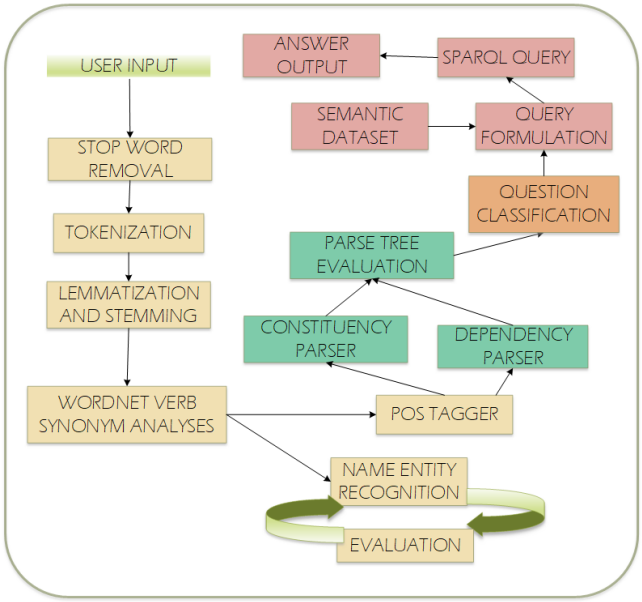

Figure VI-2. Natural Language Processing Steps for Question Answering

As illustrated in Figure VI-2, we have multiple stages to get an answer at the end. Restricted domain question answering suffers from a problem of named-entity recognition when we implement the system.

After taking input from any user, stop-word preprocessing stage start to filter unnecessary characters such as question mark, exclamation point, comma, dot or determiners. Tokenization is the next step in order 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 is an essential step before WordNet Verb analysis because our main target is to extract verb, nouns and related chunking in order to formulate a SPARQL query that can give an answer.

Algorithm 1 Query Formulation

```
function QUERY FORMULATION(a, b)                          ▷ Explain here
    query ← QueryWithPrefixes
    r ← contituent.parse.tree
    indirectdependency ← dependency.parse.tree
    while nodes ≠ leafs.terminal do                ▷ Until leaf nodes(Terminals)
        verbs ← PARSER(nodes)
        nouns ← PARSER(nodes)
        similarityflag ← WORDLATENANALYSIS(verbs)
        if StaticInformation is True then
            indirectdependencyFlag ← DEPENDENCYPARSER(nodes)
            if similarityflag and IndirectDependency is true then
                object ←nouns
                predicate ←verbs
                query += object + predicate + ?subject
            else
                subject ←nouns
                predicate ←verbs
                query += ?object + predicate + subject
        if DynamicInformation is True then
            predicate ← PARSER(nodes)
            object ← PARSER(nodes)
            similarityflag ← SIMILARITYLEVENSHTEIN(input)
            query += object + predicate + ?subject
    return query                                ▷ The last query has been constructed
```

Figure VI-3. Natural Language Processing Steps for Question Answering

The general information about our architecture is in Figure 1.1 below: The RDF data from eniLINK and OPC UA Server (right-hand side). A SPARQL Endpoint has been provided by our architecture for local static data and KVIN also presents a SPARQL Endpoint for time-series data.

VII. EXPERIMENTAL DEVELOPMENT

A. Data Sets

In the evaluation phase, our main data sources are OPC UA Server Generated Data, eniLINK Static, and Dynamic Data. Chapter 3 A. and B. parts are examining in a detailed way how to create and obtain data and convert into a linked data format. OPC UA Generated Data has not specific namespace definition unless we define as we wish. However, user-defined IRIs definition has drawbacks such as collision or non-extendibility. Long listed linked data makes the structure complex so that two subjects of the list can be collapsed because of same-defined IRIs. In our case, all namespaces are generated with http://www.example.org" or "<unknown_namespace>".

One of the poorest performance has been observed in the phase of named-entity recognition because restricted-domain requires different lexicons to train entities.

B. Model Setup

VIII. EVALUATION

Evaluation criteria comprise of question classification and recall, accuracy, precision, F1 score of answers against semantic question answering system. General evaluation parameters for a restricted domain question answering is not only limited with answering of questions but also we can assess with speed, user interaction, querying style (keywords, browsing, spell checker, abbreviation recognition)

QUESTION CLASSIFICATION- QUESTION LABELS

Evaluation Parameters	Properties
Answer Return Rate	Initial Query against the OPC UA Server – 39.88 seconds Consecutive Query against a generated data of OPC UA Server – 12.31 seconds Dynamic Query – 17.48 seconds An Open-Domain Question Answering – 20.55s
Querying Style Coverage	Keyword-Based Search and Semantic Search eniLINK data, linkedfactory streaming data
Size	Static data relatively small size Continuous data relatively large size
Up-to-dateness Query	No update statement provided by SPARQL
Formulation Assistance	Voice Input Recognition, Spell Checker

Listing VIII-1: Additional Parameters to evaluate the Semantic Question Answering

Parameters	Precision	F1	Recall
Newton-cg	%95.55	%95.56	%95.57
Linear SVC	%92.75	%92.76	%92.77
Limited BFGS	%94.21	%94.22	%94.23
Logistic Regression	%95.63	%95.63	%95.64
Linear SVC for Li&Roth Taxonomy	%65	%45.5	%35

Listing VIII-2: The evaluation of the Question Classification

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 of the Model	%68.00

Listing VIII-3: The Evaluation of the Question Answering

IX. RELATED WORKS

[Diego Molla et. al] reviewed a main characteristic of question answering in restricted domains is the integration of domain-specific information that is either developed for question answering or disclosed for other purposes [10]. [Diego Molla, Jose Luis Vicedo] 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[10].

The authors have compared between open-domain and restricted-domain question answering by figuring out key points. [Diego Molla, Jose Luis Vicedo] offers four clear-cut subjects such as the size of data, domain context, resources, and use of domain-specific resources.

[Sebastian Ferre] has published one of the detailed reports that express common pitfalls of natural language processing, essential points while consolidating SPARQL query and morphological definitions [11]. SQUALL is a solution for querying and updating RDF graphs by exploiting a controlled natural language which restricts grammar structures of a sentence in order to diminish complexities [11]. It has grouped all substantial features of a morphological language and pointed out what type of features in a natural language harnesses with regarding priorities and orders. The main contribution of SQUALL is categorizing ambiguities of natural languages and how turned out an advantage when using a controlled natural language [11]. The authors sketched a translation from their intermediate language to SPARQL to gain more accuracy with their system [11].

X. CONCLUSION

Please include a brief summary of the possible question answering implications of the work conducted at Fraunhofer IWU in this section.

//Answer the research questions // to all of them

APPENDIX

Appendixes, if needed, appear before the acknowledgment.

ACKNOWLEDGMENT

Fraunhofer IWU supports this work and we would like to thank the group of HMMI on the account of financial support.

REFERENCES

[1] R. Margaret and D. Daniel, 'Definition of Smart Factory'. [Online]. Available: https://searcherp.techtarget.com/definition/smart-factory. [Accessed: 05-Dec-2018].

[2] K. D. Thoben, S. A. Wiesner, and T. Wuest, '"Industrie 4.0" and smart manufacturing-a review of research issues and application examples', *Int. J. Autom. Technol.*, vol. 11, no. 1, pp. 4–16, 2017.

[3] C. Team, 'What is the smart factory and its impact on manufacturing?', *13 June 2018*. [Online]. Available: https://ottomotors.com/blog/what-is-the-smart-factory-manufacturing. [Accessed: 05-Dec-2018].

[4] 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', *Comput. Ind.*, vol. 83, pp. 121–139, 2016.

[5] L. D. Platform and F. IWU, 'eniLink'. [Online]. Available: http://platform.enilink.net/. [Accessed: 23-Nov-2018].

[6] Pure Python OPC-UA Client and Server, 'Free OPC-UA Library'. [Online]. Available: https://github.com/FreeOpcUa/python-opcua. [Accessed: 22-Nov-2018].

[7] TU Dresden, 'Plt-TUD'. [Online]. Available: https://github.com/plt-tud/opc_ua_xml_export_client. [Accessed: 22-Nov-2018].

[8] D. Jurafsky and J. H. Martin, 'Speech and Language Processing', *Speech Lang. Process. An Introd. to Nat. Lang. Process. Comput. Linguist. Speech Recognit.*, vol. 21, pp. 0–934, 2009.

[9] J. Perkins, D. Chopra, and N. Hardeniya, *Natural Language Processing : Python and NLTK*. 2016.

[10] D. Mollá and J. L. Vicedo, 'Question answering in restricted domains: An overview', *Comput. Linguist.*, vol. 33, no. 1, pp. 41–61, 2007.

[11] S. Ferré, 'SQUALL: A controlled natural language for querying and updating RDF graphs', *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7427 LNAI, pp. 11–25, 2012.

ADDITIONAL DATA

Sample Inputs	Question ID	Precision	F1 Score	Accuracy
Provide me a combined results for IWU and e3sim	1	A	B	C
I am a customer for this company. Could you tell me please what the value of	2	A	B	C
C	3	A	B	C
D	4	A	B	C
E	5	A	B	C
F	6	A	B	C
G	7	A	B	C
H	8	A	B	C
I	9	A	B	C
J	10	A	B	C
K	11	A	B	C
L	12	A	B	C

Listing X-1: Sample Questions with Precision and Recall