



**Universitat Politècnica de Catalunya**  
**BarcelonaTECH**

Department of Computer Science

# Using Natural Language Processing for Question Answering in Closed and Open Domains

A DISSERTATION  
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE  
OF UNIVERSITAT POLITÈCNICA DE CATALUNYA - BARCELONA TECH  
(UPC)  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

**Majid Latifi**

Advisors:

Dr. Horacio Rodríguez Hontoria

Dr. Miquel Sànchez-Marrè

*March 2018, Barcelona*



# ACKNOWLEDGMENTS

---

This dissertation would not have been accomplished without the support of the following people who deserve to be appreciated. First and foremost, I would like to thank my advisor, Dr. Horacio Rodríguez Hontoria and co-advisor Dr. Miquel Sànchez-Marrè. I was very fortunate to have both of you as my advisors who always gave me confidence to take conscious decisions about my thesis. Dr. Rodríguez, your contributions were sizeable. You taught me to see the nuances in everything. I would like to express my sincere gratitude to you for your quick and insightful answers to my questions. I could not have imagined having a better advisor and an inspiring teacher about natural language processing (NLP). My sincere thanks also go to Dr. Sànchez-Marrè without whose precious support, it would not be possible to conduct this research project.

I would like to appreciate the support of Dr. Qun Liu at ADAPT Centre in DCU University in Dublin, Ireland, who gave me the opportunity to develop my thesis project by collaborating with the specialized NLP and Machine Learning group.

It goes without saying that my thesis could not have been completed without the support of my dear friends. I would like to thank my friends in Barcelona, Dublin, and ADAPT Centre for their affection and encouragement. In particular, my Iranian intimate friends who created happy and joyful moments for me while I was in distress and unwell. And I do thank the efforts of influential teachers and professors in Iran in different periods of my education that I will never forget their encouragements and guidances.

Last but not the least; I would especially like to thank my family without whom I would not be where I am today. They gave me support, encouragement, and motivation. Words cannot express how grateful I am to my mother and father for all the sacrifices you have made for me throughout my life to get an education and knowledge. I prefer to do this in my mother tongue in Farsi (Persian) for my family:

تقدیم به پاس محبت ها، حمایت ها، تشویق ها و دعاهاى بی دریغ

مادر دلسوز و مهربانم به زلالی چشمه، پدر عزیز و بردبارم به استواری کوه،

خواهرانم به شبنم های صبحگاهی، برادرهای عزیزم به آرامی ساحل اقیانوس،

همراهان همیشگی و پشتوانه های زندگیم که در سایه درخت پر بار وجودشان بیاسایم و از ریشه آنها شاخ و برگ گیرم و از نسیم روحبخش سخاوت و گذشتشان در راه کسب علم و دانش تلاش نمایم. شما با عاطفه سرشار و گرمای امیدبخش و به خاطر حمایت های همه جانبه به عنوان ارزنده ترین پشتوانه و انگیزه، مایه دلگرمیم در انجام پروژه تحقیقاتی و تکمیل این رساله بودید. قلم و کلمات ناتوان من توانایی آراسته نمودن ورق به آنچه در حق من در این سالها لطف نمودید و رنگ زندگیم را در سال های دور از دیار شاداب و سبزین نگه داشتید را ندارند. پروردگارا سلامت، سربلندی و سعادت را مورد لطف و محبت خود برای آنان ارزانی و مقدر نما.

---\* از همه شما سپاسگزارم \*---



# Abstract

---

Developing computers capable of answering questions on any subject is a long-standing goal of artificial intelligence communities. Due to the growth in the amount of social, environmental, and biomedical information available digitally, there is a growing need for Question Answering (QA) systems that can empower users to master this new wealth of information. Recently, promising progress in the field of QA in Natural Language Processing (NLP) have been made, such that it is now possible to interpret questions and generate the structured queries. Although the quality of interpretation and extraction of the desired answer is not adequate. We believe that striving for higher accuracy in QA systems is subject to on-going research, i.e., it is better to have no answer than wrong answers. QA researchers are attempting to deal with a wide range of factoid and non-factoid question types including wh-, list, definition, opinion, hypothetical, cross-lingual, and semantically constrained questions. Significant progress has been made at answering factoid questions. However, there are diverse queries, which the state of the art QA systems cannot interpret and answer properly. Current research has addressed the factual questions, where we can distinguish between Wh-queries (who, where, what, how many, etc.), and commands (give me, list all, etc.).

The problem of interpreting a question and how to perform a technical know-how of the mapping method in a way that could preserve its syntactic-semantic structure is considered as one of the most important challenges in this area. In the case of exploiting an efficient approach, it leads to generate a more accurate structured query (e.g. SPARQL) so that the precise answer is retrievable from the open and crowdsourced knowledge graphs such as DBpedia. In this work we focus on the problems of semantic-based QA systems and analyzing the effectiveness of NLP techniques, query mapping, and answer inferencing both in closed (first scenario) and open (second scenario) domains. For this purpose, the architecture of Semantic-based closed and open domain Question Answering System (hereafter “ScoQAS”) over ontology resources is presented with two different prototyping: Ontology-based closed domain and an open domain under Linked Open Data (LOD) resource.

In ScoQAS, we address the deployment of the NLP and artificial intelligence techniques to classify questions integrating syntactic and semantic techniques. The ScoQAS is based on NLP techniques combining semantic-based structure-feature patterns (Ss-fP) for question classification and creating a question syntactic-semantic information structure (QSiS). It uses an empirical technique integrating syntactic parsing, lexical meaning (e.g. WordNet) and semantic information (e.g. Ontology) by building constraints to formulate the related terms on syntactic-semantic aspects. The QSiS provides an actual potential to build a question graph (QGraph) which facilitates making inference for getting a precise answer in the closed domain. In addition, our approach provides a convenient method to map the formulated comprehensive information into SPARQL query template to crawl in the LOD resources in the open domain.

The main contributions of this dissertation are as follows:

1. Developing ScoQAS architecture integrated with common and specific components compatible with closed and open domain ontologies.
2. Analysing user's question and building a *question syntactic-semantic information structure (QSiS)*, which is constituted by several processes of the methodology: question classification, Expected Answer Type (EAT) determination, and generated constraints. The QSiS is the basic block of knowledge to obtain the answer to the questions or to transform it to the other formal query.
3. Presenting an empirical *semantic-based structure-feature pattern (Ss-fP)* for question classification and generalizing heuristic constraints to formulate the relations between the features in the recognized pattern in terms of syntactical and semantical.
4. Developing a syntactic-semantic QGraph for representing core components of the question.
5. Presenting an empirical graph-based answer inference in the closed domain.

In a nutshell, a semantic-based QA system is presented which provides some experimental results over the closed and open domains. We employ AI and NLP techniques to interpret question semantically, classifying question, building constraints to formulate the related terms in syntactic-semantic aspects, and making graph-based inference. The empirical evaluation shows the effectiveness and scalability of ScoQAS. The efficiency of the ScoQAS is evaluated using measures such as precision, recall, and F-measure on LOD challenges in the open domain scenario. We focus on quantitative evaluation in the closed domain scenario, its accuracy is analyzed on an Enterprise ontology. The lack of predefined benchmark(s) is one of the major challenges of evaluation in the first scenario. Therefore, we define measures that demonstrate the actual complexity of the problem and the actual efficiency of the solutions. The results of the analysis corroborate the performance and effectiveness of our approach to achieve a reasonable accuracy.

# Table of Contents

---

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Introduction.....	1
1.2	Motivation of QA Systems.....	5
1.3	Roadmap of Question Answering System .....	7
1.4	NL Technologies Involved in QA Systems .....	12
1.5	Dimensions of QA Systems.....	14
1.6	Research Objectives .....	17
1.7	Summary of Contributions .....	18
1.8	Thesis Overview.....	19
<b>2</b>	<b>State of the Art .....</b>	<b>23</b>
2.1	First Steps in QA: Natural Language Interfaces (NLIs).....	23
2.2	Introduction of QA Systems.....	27
2.2.1	ONLI+ - Ontology Natural Language Interaction .....	27
2.2.2	PANTO-Portable nAtural laNguage inTerface to Ontologies.....	28
2.2.3	Aqua Log - An Ontology-driven Question Answering.....	29
2.2.4	QuestIO - Question-based Interface to Ontologies.....	30
2.2.5	QACID - Question Answering System Applied to the Cinema Domain.....	31
2.2.6	FREyA: Feedback, Refinement and Extended Vocabulary Aggregation .....	32
2.2.7	QASYO - Question Answering System for YAGO Ontology .....	33
2.2.8	Pythia: Ontology-based Question Answering on the Semantic Web .....	33
2.2.9	DEQA: Deep Web Extraction for Question Answering.....	34
2.2.10	QAAL.....	35
2.2.11	QAKiS: Question Answering wiKiframework-based System .....	36
2.2.12	PARALEX - Open QA System.....	37
2.2.13	SINA .....	38
2.2.14	DEANNA.....	39
2.3	A Summary of Analysed QA Systems.....	39
<b>3</b>	<b>The Architecture of Semantic-based QA System (ScoQAS) .....</b>	<b>43</b>
3.1	The Components of the ScoQAS Architecture .....	43
3.1.1	Common Components in ScoQAS .....	45
3.1.2	Specific Components for the Closed-Domain Scenario .....	46
3.1.3	Specific Components for the Open Domain Scenario.....	46
3.2	Question Preprocessing.....	47
3.3	Question Representation .....	48
3.4	Rule-Based Question Classifier .....	49
3.4.1	QT - Question Type.....	51
3.4.2	EAT- Expected Answer Type .....	61
3.4.3	RT- Related Terms (Keywords).....	61
3.5	Modelling Constraints .....	62

3.6	Semantic-based QA in a Closed Domain: 1 <sup>st</sup> Scenario.....	67
3.6.1	Mapping Matched Ontology Items .....	69
3.6.2	Expansion of Generating the Constraints.....	72
3.6.3	Generating the Graph for Extracting the Answer .....	77
3.6.4	Inference to Elicit Exact Answer from the QGraph .....	81
3.7	Semantic-based QA in an Open Domain: 2 <sup>nd</sup> Scenario.....	86
3.7.1	Mapping Syntactic-Semantic Information Structure of the Question (QSiS) to Generate SPARQL Formal Query .....	88
3.7.2	Pre-processing Steps.....	90
3.7.3	Constraint-Based Mapping Rules to Build SPARQL Query .....	90
<b>4</b>	<b>The Empirical Evaluation and Results.....</b>	<b>97</b>
4.1	Evaluation Measures.....	97
4.2	The Evaluation of Closed-domain Approach.....	98
4.3	The Evaluation of Open Domain Approach .....	103
4.4	Error Analysis .....	107
<b>5</b>	<b>Conclusions and Future Work.....</b>	<b>111</b>
5.1	Conclusions .....	111
5.2	Future Work .....	113
	<b>Bibliography .....</b>	<b>115</b>
<b>A.</b>	<b>Appendix A: Set of Questions for Closed Domain Empirical Evaluation (1<sup>st</sup> Scenario) .....</b>	<b>122</b>
<b>B.</b>	<b>Appendix B: QALD's Training Data Set for Open Domain (2<sup>nd</sup> Scenario) .....</b>	<b>125</b>
<b>C.</b>	<b>Appendix C: Bounded Variables and Constraints for Q1 in Closed Domain (1<sup>st</sup> Scenario) .....</b>	<b>131</b>
<b>D.</b>	<b>Appendix D: Results of ScoQAS over QALD-2 Test Set for Open Domain (2<sup>nd</sup> Scenario) .....</b>	<b>137</b>
<b>E.</b>	<b>Appendix E: Results of ScoQAS over QALD-3 Test Set for Open Domain (2<sup>nd</sup> Scenario).....</b>	<b>141</b>
<b>F.</b>	<b>Appendix F: Results of ScoQAS over QALD-4 Test Set for Open Domain (2<sup>nd</sup> Scenario) .....</b>	<b>145</b>
<b>G.</b>	<b>Appendix G: Results of ScoQAS over QALD-5 Test Set for Open Domain (2<sup>nd</sup> Scenario) .....</b>	<b>147</b>
<b>H.</b>	<b>Appendix H: Pseudocode of Generating QGraph.....</b>	<b>149</b>
<b>I.</b>	<b>Appendix I: List of the Publications .....</b>	<b>151</b>



# Table of Figures

---

Figure 1.1: The IR-based question answering system .....	3
Figure 2.1: Minipar parse tree for the question “Find 2 vendors who sell enzyme products” .....	28
Figure 2.2: Mapping question into answer in the PARALEX Open QA .....	38
Figure 3.1: Architecture of Semantic-based Question Answering System (ScoQAS) .....	44
Figure 3.2: Basic dependencies in Stanford CoreNLP parser over Q1 .....	47
Figure 3.3: NIF format for Q1 .....	48
Figure 3.4: Schema of relations between ontology items expanding for Where_Person_Action .....	75
Figure 3.5: Related Terms (RTs) and bounded ontology graph for question Q1 .....	77
Figure 3.6: Part of the Nodes format belongs to the QGraph for Q1 .....	79
Figure 3.7: Automatically produced QGraph for Q1 based on its constraints .....	80
Figure 3.8: Part of the Edges format belongs to the QGraph for Q1 .....	81
Figure 3.9: The EAT class and instance with candidate answer format attached to the QGraph for Q1 .....	81
Figure 3.10: The relationships obtained in the inference of the answer between involvedVars and Answer .....	86
Figure 3.11: A flowchart of the process in open domain .....	87
Figure 4.1: Evaluation and error analyzing steps for closed-domain (first scenario) .....	99
Figure 4.2: External error in Stanford parsing over question .....	108



# List of Tables

---

Table 1.1: The dimensions of QA and query and search interfaces in general .....	14
Table 2.1: An example cluster of questions .....	37
Table 2.2: Comparison of QA systems.....	39
Table 3.1: Applying Stanford CoreNLP Parser over question Q1 .....	49
Table 3.2: Set of QT with satisfied constraints in ScoQAS .....	52
Table 3.3: Sample of Expected Answer Type .....	61
Table 3.4: Sample questions with their correct answers for closed domain scenario ....	68
Table 3.5: Thresholds-mean for lemma "manager" and favorable ontology items .....	71
Table 3.6: Syntactic-semantic information structure of the question (QSiS) for Q2 .....	89
Table 4.1: The 2-by-2 contingency table.....	97
Table 4.2: Binding ontology items for Person in pattern QT:Where_Person_Action ....	101
Table 4.3: Accuracy of closed-domain questions in ScoQAS.....	103
Table 4.4: Experimental results over Enterprise ontology for closed domain .....	103
Table 4.5: Evaluation of ScoQAS over QALD benchmark.....	104
Table 4.6: QALD-2 competitions results .....	105
Table 4.7: QALD-3 competitions results for DBpedia test set.....	105
Table 4.8: QALD-4 competitions results in multilingual QA.....	106
Table 4.9: QALD-5 competitions results for multilingual QA.....	106
Table 4.10: The summary of the QALD competitions results in F-Measure .....	107
Table 4.11: Results over QALD-2 Test Set.....	109
Table 4.12: Results over QALD-3 Test Set.....	109
Table 4.13: Results over QALD-4 Test Set.....	110
Table 4.14: Results over QALD-5 Test Set.....	110



# List of Abbreviations

---

CG	Conceptual Graph.
CSTR	Constraint.
EAT	Expected Answer Type.
IR	Information Retrieval.
KB	Knowledge Base.
LOD	Linked Open Data.
NE	Named Entity.
NL	Natural Language.
NLI	Natural Language Interface.
NLIDB	Natural Language Interface to Database.
NLP	Natural Language Processing.
PTB	Penn TreeBank.
QA	Question Answering.
QALD	Question Answering over Linked Data.
QC	Question Classification.
QGraph	Question Graph.
QSiS	Question Syntactic-Semantic Information Structure.
QT	Question Type.
RT	Related Term.
ScoQAS	Semantic-based closed and open domain Question Answering System.
Ss-fp	Semantic-based Structure-Feature Pattern.
TAC	Text Analysis Conference.
TREC	Text REtrieval Conference.



## 1 Introduction

---

### 1.1 Introduction

Given the rapid growth of information on the web, having access to the information and managing the existing bulky data have gained a paramount importance. The ability to answer questions on any subject is a long-standing goal of artificial intelligence. Search engines allow people to filter the entire web to an informal small collection of pages according to keyword-based inquiries. In addition, search engines assume that the answer to a question will be explicitly stated on a single web page. The current search engine model for information access breaks down when responding and return correct expected answers. The current web consisting of documents and the links between documents has been extended by Linked data. Linked data refers to the Web of data in contrast to the Web of documents. DBpedia is one of the central linked data datasets in Linked Open Data (LOD) Project [1][2]. Currently, the Web of interlinked data sources around DBpedia provides approximately 4.7 billion pieces of information and covers domains such as geographic information, people, companies, films, music, genes, drugs, books, and scientific publications. Many medical and healthcare knowledge bases have adopted W3C<sup>1</sup> standards to publish their data online as LOD (e.g. BioPortal<sup>2</sup>, Drug Encyclopedia<sup>3</sup>) [3].

---

<sup>1</sup> <https://www.w3.org>

<sup>2</sup> <https://bioportal.bioontology.org/>

<sup>3</sup> <http://datlowe.org/drug-encyclopedia/datasets.html>

Following the emergence of the Semantic Web, traditional Information Retrieval (IR) approaches are not applicable to linked data structure. Semantic Web technologies bear new benefits to knowledge-based systems. Specifically, ontologies are becoming an axial skeleton to extract domain-specific conceptual knowledge in order to promote the semantic capability of a Question Answering (QA) system. QA is a computer science discipline within the fields of IR and Natural Language Processing (NLP), which is concerned with developing systems to automatically answer questions posed by humans in a natural language (NL). QA can be defined as the task that, given a user information need, expressed as a NL question, provides the correct answer to the user question, not, as usual in IR systems, a set of documents where likely the answer can be found. In QA the user query consists of a question expressed in NL, sometimes, however, limited forms of NL, the so-called Controlled NL, are used instead.

While more and more structured data is published on the web, the question of how typical web users can access this body of knowledge becomes of crucial importance. Over the past years, there is a growing amount of research on interaction paradigms that allow end users to profit from the expressive power of Semantic Web standards while at the same time hiding their complexity. Keyword queries, as common with Web search nowadays, constitute a highly ambiguous and very impoverished representation of an information need. Thus, keyword-based search interfaces (as Swoogle<sup>1</sup>) cannot fully exploit the expressive power of Semantic Web data models and query languages such as SPARQL. Especially, natural language interface (NLI) have received wide attention, as they allow users to express arbitrarily complex information needs in an intuitive fashion and, at least in principle, in their own language[4]. An important challenge for the Semantic Web, but also for NLP communities, is scaling QA approaches to LOD.

Recently, QA on the web has gained a momentum due to the large structured knowledge bases such as DBpedia [1], Freebase<sup>2</sup> and YAGO [5] that regularly collect information from open and ever expanding knowledge resources such as Wikipedia. Lately, a progress in the field of QA in NLP has been made by methods that learn to map questions to logical forms or database queries. QA systems are playing an important role in the current search engine optimization. NLP techniques are mostly implemented in QA systems for analyzing user questions and several steps are followed by conversion of questions to query form for getting an exact answer. By the early 1960s, there were systems implementing the two major modern paradigms of QA to answer questions:

- IR-based QA
- Knowledge-based QA

In the first paradigm, IR-based QA systems typically include a question processing module that determines the type of the question and the type of the expected answer [6]. After the question analysis, the system typically uses several modules that apply increasingly complex NLP techniques on a gradually reduced

---

<sup>1</sup> <http://swoogle.umbc.edu/>

<sup>2</sup> <http://www.freebase.com/>



amount of text. Thus, a document retrieval module uses search engines to identify the documents or passages in the document set that are likely to contain the answer. Subsequently, a filter preselects small text fragments that contain strings of the same type as the expected answer. For example, if the question is "Who invented Penicillin?", the expected answer type is *person*, and therefore, the filter retrieves texts that contain names of people. Finally, an answer extraction module looks for further clues in the text to determine if the answer candidate can indeed answer the question. Figure 1.1 illustrates the typical IR-based QA system. A QA system cannot satisfy the intense need for specialized information experienced by professional analysts if every question is posed in isolation. A major obstacle in building user-friendly QA systems is the need to enable a conversation with the user in which clarifications, follow-up question and context specification are made possible.

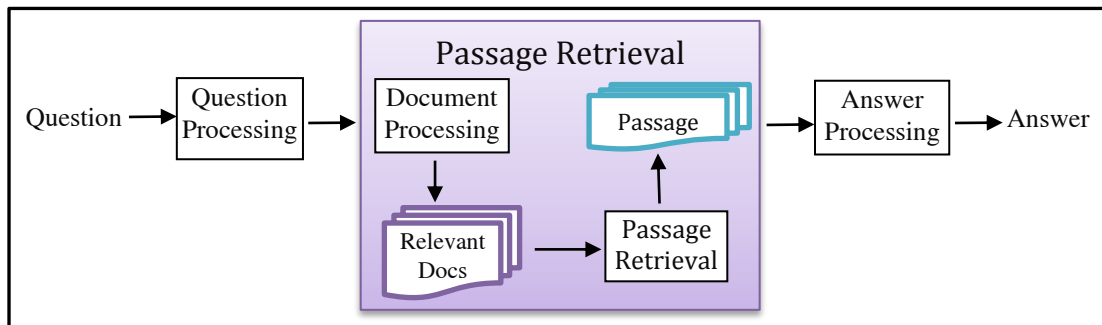


Figure 1.1: The IR-based question answering system

In the second paradigm, a knowledge-based QA system relies on information existing in more structured forms. The structured information can be a full relational database, or simpler structured databases like sets of RDF triples. There is an enormous amount of information, which is available in the semantic repositories as ontology with format such as RDF, OWL, N-Triple, etc. Systems for mapping from a text string to any logical form are called *semantic parsers*. Semantic parsers for QA usually map either to some version of predicate calculus or to a query language like SPARQL or SQL. Although more complex forms of representation have been recently proposed, Bordes et al [7][8][9], for instance embed question and answer space into a common low dimensional vector space, using neural models, QA is then stated as a problem of finding the closest answer candidate to the query. Andres et al [10] go beyond and learn maps from pieces of the question into neural network models that are then combined into a neural network representing the whole query. Finally, this neural network is in charge of finding the answer.

The QA research attempts to deal with a wide range of question types including facts, lists, definitions, Wh- questions, and semantically constrained questions. Most research focus on factual questions, where we can distinguish between Wh-queries (who, what, how many, etc.), commands (name all, give me, etc.) or affirmation (Yes/No) questions. More difficult kinds of questions include those which ask about opinions, like Why or How questions, which require understanding of causality or instrumental relation, what questions which provide little constraint in the answer

type, and definition questions. More complex and challenging questions are those where the complete answer does not occur explicitly in any source and has to be synthesized with pieces of information collected from several resources. For instance in the query “Republican presidents of USA between the two world wars” there is no complete answer likely to be found in a document. In the general consensus, QA systems are categorized based on supporting context factor as follows:

- *Closed-domain QA*: This type of QA deals with questions under a specific domain (for example, medicine or automotive maintenance), and can be seen as an easier task because NLP systems can exploit domain-specific knowledge frequently formalized in ontologies. Alternatively, closed-domain might refer to a situation where only a limited type of questions are accepted, such as questions asking about descriptive rather than procedural information.
- *Open-domain QA*: This type of QA deals with questions about nearly anything, and can only rely on general ontologies and world knowledge. On the other hand, these systems usually have much more data available from which to extract the answer. The system takes a NL question as an input rather than as a set of keywords, for example, "When is the national day of China?". The sentence is then transformed into a query through its logical form.

QA systems have been extended more recently to encompass additional domains of knowledge [11]. For example, systems have been developed to automatically answer temporal and geospatial questions, questions of definition and terminology, biographical questions, multilingual questions, and questions about the content of audio, images, and video. Semantic-based QA system communicates between user's text questions as an input and ontologies or LOD as Knowledge Bases (KBs) in order to find correct answer. Most KBs provide facilities for querying through using some formal language such as SPARQL or SeRQL. However, they have a fairly complex syntax, requiring a well understanding of the data schema and being prone to errors due to the need for typing long and complicated URIs. These languages are homologous to the use of SQL for interrogation of traditional relational databases and should not be seen as an end user tool [12].

Some empirical solutions to the most important problems are proposed in this research area. Due to the need to translate NL questions into a machine-readable format, the first step is that the user questions in natural language should be preprocessed using NLP tools. The question is then transformed into a query through its logical form. Having the input in the form of a NL question makes the system more user-friendly, but harder to implement, as there are various question types and the system will have to identify the correct one in order to give a sensible answer. Classifying questions is an important topic in QA systems, because it compels the answer extraction system to infer the correct expected answers. Assigning a question type to the question and determining answer type are the important tasks in classifying question step. The entire answer extraction process relies on finding the correct question type and hence the correct answer type.

## 1.2 Motivation of QA Systems

We live in a world with possession of modern digital devices to communicate with each other. People, the various segments of society, groups of researchers and specialists in various disciplines, and businesses are communicating globally at incredible speed across the world's languages and over a daily increase of devices in countless volumes. Currently, hopeful evolution in the QA has meant a resurgence of interest in both academic research and social media analysis companies. QA systems have been used in multiple scenarios with increasing number and extended scope of their applications; we sketch out next some of them.

- Application of QA System in Social Welfare:

The goal of QA systems [6] is to allow users to ask questions in NL, using their own terminology, and receive a concise answer. Social information seeking is often materialized in online QA websites such as Yahoo! Answers<sup>1</sup>, Answerbag<sup>2</sup>, WikiAnswers<sup>3</sup> and Twitter<sup>4</sup>. Another goal is to develop systems for supporting such activities, which are driven by a community. Such QA sites have emerged in the past few years as a potential market for the fulfillment of information needs. Social QA or social/community QA, according to Shah et al [13], consists of three components: a mechanism for users to submit questions in natural language, a venue for users to submit answers to questions, and a community built around this exchange.

Some QA systems, as an emerging assistive technology, provide simpler access to information allowing a voice interface. Cognitive computing systems<sup>5</sup> improve over time as they build knowledge and learn a domain - its language and terminology, its processes and its preferred methods of interacting. Unlike expert systems in the past that required rules to be hard coded into a system by a human expert, cognitive computers can process NL and unstructured data and learn by experience, much in the same way humans do.

- Assistance in Biomedical Research and Health Care System:

Researchers can leverage a QA system on biomedical data to update their knowledge about the recent findings in their own field or related fields and thus making interesting discoveries. In healthcare, IBM Watson for Oncology<sup>6</sup> helps oncologists to treat cancer patients with individualized evidence-based treatment options by analyzing patient data against thousands of historical cases. Watson can help doctors narrowing down the options and helping them to pick the best treatments for their patients. Watson is there to make sense of the data and help make

---

<sup>1</sup> <https://answers.yahoo.com/>

<sup>2</sup> <http://www.answerbag.com/>

<sup>3</sup> <http://www.answers.com/>

<sup>4</sup> <https://twitter.com/>

<sup>5</sup> <http://www.research.ibm.com/cognitive-computing>

<sup>6</sup> <http://www.ibm.com/smarterplanet/us/en/ibmwatson/watson-oncology.html>

the process faster and more accurate. IBM's Watson as a language-fluent computer was won in the best human champions at a game of the US TV show Jeopardy. It is being turned into a tool for medical diagnosis. According to the IBM statements, its ability to analyze huge amounts of data is better than that of human doctors, and its deployment through the cloud reduces healthcare costs.

Recently, effective steps have been taken in the field of QA in clinical NLP. Regarding the growth of biomedical information, there is a growing need for QA systems that can help users better utilize the ever-gathering information. In 2014, the United States National Library of Medicine<sup>1</sup> has been received approximately 4,600 health-related questions, which include disease-related questions and drug-related questions from a wide range of consumers around the world. According to an American health<sup>2</sup> survey (Pew Research Center's Internet & American Life Project), 35% of U.S. adults state that they have gone online specifically to try to figure out what medical condition they might have. Report on the accuracy of their initial diagnosis implies that 41% of online diagnoses say a medical professional confirmed their diagnosis while 35% say they did not visit a clinician to get a professional opinion. When asked about the last time they hunted for medical information, 77% of online health seekers say they began at a search engine. Therefore, these users had to filter the numerous results of their queries in order to find needed information. Frequently, Practitioners may need an immediate answer to their questions, which is a crucial point.

- Influence in Educational Technology:

QA has formerly been explored within the educational context to assist learning methods. Google has launched its new technology Google Glass<sup>3</sup>. Now, this new technology will allow the student and teacher to stay connected in an interactive environment featuring Google search, and other online tools. Teachers as well as students can refer to topics related to their studies on the go.

---

<sup>1</sup> <https://www.nlm.nih.gov/>

<sup>2</sup> <http://pewinternet.org/Reports/2013/Health-online.aspx>

<sup>3</sup> <https://eduglasses.com/>

### 1.3 Roadmap of Question Answering System

In 2001 a group of researchers wrote a roadmap of research in QA [14]. The following issues whose main lines continue to be valid nowadays were identified:

- *Question classes*: Different types of questions (e.g., "What is the capital of Liechtenstein?" vs. "Why does a rainbow form?" vs. "Did Marilyn Monroe and Cary Grant ever appear in a movie together?") require different strategies to find the answer. Question classes used to be arranged hierarchically in taxonomies. Question classes range from a simple set (who, where, when, why, Yes/No classes) to fine grained taxonomies as the Webclopedia<sup>1</sup> or Li&Roth classification sets [15].
- *Question processing*: The same information request can be expressed in various ways, some interrogative ("Who is the King Cyrus of Persia?"), some assertive ("King Cyrus of Persia."), and some imperative ("Tell me the name of the King Cyrus of Persia."). A semantic model of question understanding and processing would recognize equivalent questions, regardless of how they are presented. This model would enable the translation of a complex question into a series of simpler questions, identify ambiguities and treat them in context or by interactive clarification.
- *Context and QA*: Questions are usually asked within a context and answers are provided within that specific context. The context can be used to clarify a question, resolve ambiguities or keep track of an investigation performed through a series of questions. For example, the question, "Why did Joe Biden visit Iraq in January 2010?" might denote why Vice President Biden visited and not President Obama, why he went to Iraq and not to Afghanistan or some other country, why he went in January 2010 and not before or after, or what Biden was hoping to accomplish with his visit. If the question is one of a series of related questions, the previous questions and their answers might shed light on the questioner's intent. In our setting, the questions are independent of each other, so the elements of the context have to be extracted (or induced) from the question itself. Frequently, the items selected for forming the context are split into mandatory and optional constraints being the former obligatorily present in the answer context and the later used only for helping in the search of candidate answers. Consider, for instance, the question "All writers having won the Nobel Prize of literature". "Nobel Prize of literature" is a mandatory constraint, while "writer" is optional, because the fact that a "Nobel Prize of literature" is a writer can be inferred using common sense.
- *Data sources for QA*: Before a question can be answered, it must be known what knowledge sources are available and relevant. If the answer to a question is not present in the data sources, no matter how well the question

---

<sup>1</sup> <http://www.isi.edu/natural-language/projects/webclopedia/>

processing, IR and answer extraction is performed, a correct result will not be obtained. For instance, in the previous example, if DBpedia lacks a resource “Nobel prize of literature”, or if existing as a class, there is no property linking it to the instances winners.

- *IR process*: Often an IR process is carried out to select from the sources the most likely documents from which answers could be extracted. Frequently this process is divided into two sub processes: IR of full documents and IR of passages (fragments of documents).
- *Answer extraction*: Answer extraction depends on the complexity of the question, the answer type provided by question processing, the actual data where the answer is searched, the search method and the question focus and context.
- *Answer formulation*: The result of a QA system should be presented as naturally as possible. In some cases, a simple extraction is sufficient. For example, when the QC indicates that the answer type is a name (of a person, organization, shop or disease, etc.), a quantity (monetary value, length, size, distance, etc.) or a date (e.g. the answer to the question, "On what day did Christmas fall in 1989?"). In the case of nonfactual questions, for instance for list questions, the formulation of the answer can be more complex.

First QA systems were devoted to factoid questions where simple question answers are facts. For example, “When was Obama born?” or “Who is the current president of the USA?”. Later, more complex and challenging questions were faced:

- *Real time QA*: There is a need to develop QA systems to be capable of extracting answers from large data sets in a short time, regardless of the complexity, the size and number of the data sources or the ambiguity of the question.
- *Multilingual (or cross-lingual) QA*: The ability to answer a question posed in one language using an answer corpus in another language. This allows users to consult information that they cannot use directly.
- *Interactive QA*: It is often the case that the required information is not well captured by a QA system, as the question processing part may fail to classify properly the question or the information needed for extracting and generating the answer is not easily retrieved. In such cases, the questioner might want not only to reformulate the question, but also to have a dialogue with the system. We can include here the systems where a sequence of related questions are sent, enriching the context for answering the questions at a cost of a higher difficulty of processing the individual questions (having to face, for instance, challenging co-reference processes). The dialog could include clarification replies from the system, narrowing the answer space, formulating related questions, and so on.

Lately, QA system has evolved to complex scenarios whose categorizations are as follows:

- *Advanced reasoning for QA:* Questioners that are more sophisticated expect answers that are outside the scope of written texts or structured databases. To upgrade a QA system with such capabilities, it will be necessary to integrate reasoning components operating on a variety of knowledge bases, encoding world knowledge and common-sense reasoning mechanisms, as well as knowledge specific to a variety of domains. In such systems, the answer can result from the combination, using heavy inference mechanisms of answers to partial questions.
- *Information clustering for QA:* Information clustering for QA systems is a new trend originated to increase the accuracy of QA systems through search space reduction. In recent years, this was widely researched through development of QA systems which support information clustering in their basic flow of process [16].
- *User profiling for QA:* User profile captures data about the questioner, comprising context data, domain of interest, reasoning schemes frequently used by the questioner, common ground established within different dialogues between the system and the user, and so forth. The profile may be represented as a predefined template, where each template slot represents a different profile feature. Profile templates may be nested one within another.

Another line of research in QA, somehow divergent from the above one, is to narrow the search space where the answer is expected to be found. We can find in this line:

- *Domain restricted QA (DRQA):* Where both questions and search space are restricted to a given domain. Many domains have been faced, geographic, tourism, economics, etc. Perhaps the domain object of the most applications is the medical domain. Usually DRQA are applied to specific tasks and use domain specific lexicons, terminologies, knowledge bases, ontologies and other domain restricted lexico-conceptual resources. Search spaces are smaller and so approaches based on the redundancy of answers (as voting techniques) are useless. User's requirements use to be high and system performance is more precision than recall oriented, it is better to have no answer is better than wrong answers. Questions and documents are challenging and frequently contain acronyms, non-textual content (tables, itemized lists, etc.), domain specific jargon, etc. A good reference is the system QACID [17][18], presented in the thesis of Óscar Ferrández centered on the cinematographic domain, within the framework of the European project *QALL-ME*<sup>1</sup>. within the commercial domain, but also related to LOD, we can find the Business to Client (B2C) scenario. Two interesting systems in this scenario are QALM, [19], and SynchroBot, [20].
- *QA for comprehension reading:* Where the questions are related with a document for checking the ability of the user to having understood the document content. Richardson et al, 2013, proposed the MCTest<sup>2</sup> a set of 660

---

<sup>1</sup> <http://qallme.itc.it/>

<sup>2</sup> <http://research.microsoft.com/mct>

stories and associated questions intended for research on the machine comprehension of text. Each question requires the reader to understand different aspects of the story.

- *Community QA (CQA)*: Also termed Q/A social networks, in this scenario a member of the community formulates an initial query (a NL question) that triggers a thread of interventions of the community members that answer, refine, comment, the interventions of previous interventions. Member's interventions can be questions and answers related. CQA have been recently evaluated in the framework of SEMEVAL-2015<sup>1</sup> and SEMEVAL-2016<sup>2</sup>. Both general purpose and topic-specific communities are growing in numbers for posting questions and obtaining direct answers in a short period. Yahoo!Answers<sup>3</sup> (Y!A), for example, provides a broad range of topics whereas Stack-Overflow<sup>4</sup> (SO), and Turbo Tax Live<sup>5</sup> (TT) are quite focused on specific domain. In contrast to the traditional search engines such as Google, CQA services provide an alternative paradigm for seeking targeted information. Zhang et al [21] is a good example of these kind of systems, see also El Adlouni et al [22] and Nakov et al [23]. The approach assumes that questions and answers share some common latent topics and are generated in a “question language” and “answer language” respectively following the topics. Cong et al [24] presents an interesting system for facing the question detection and answer detection problems. Xue et al [25] propose using retrieval models for detecting Q and A in Q&A archives (both FAQ archives and archives generated by CQA web services. The authors use as main source for learning the Wondir<sup>6</sup> collection
- *QA over domain ontologies, Ontology-based QA (ObQA)*: In this case the answers are looked up not in free text documents but in ontologies taking profit not only of the linguistic (terminological) data included into the ontology but also over their relations, properties, and inferential capabilities. An interesting example is Pythia<sup>7</sup> [26]. Pythia is based on an alignment between the question and a vocabulary aligned with the ontology. The process includes the semiautomatic generation of a grammar using LexInfo<sup>8</sup>, a declarative model for lexicon-ontology interface.
- *QA systems using LOD*: In the framework of the Semantic Web, there has been recently a huge growth of available open and closed domain resources. Many of these resources are included into the Linked Open Data (LOD) initiative. The most known and used resources in LOD are FreeBase and DBPedia as open domain LOD and BioPortal (medical and genomic) or

---

<sup>1</sup> <http://alt.qcri.org/semEval2015/>

<sup>2</sup> <http://alt.qcri.org/semEval2016/>

<sup>3</sup> <http://answers.yahoo.com/>

<sup>4</sup> <http://stackoverflow.com/>

<sup>5</sup> <https://ttlc.intuit.com/>

<sup>6</sup> <http://wondir.com>

<sup>7</sup> <http://www.sc.cit-ec.uni-bielefeld.de/pythia>

<sup>8</sup> <http://lexinfo.net>



LinkedGeoData<sup>1</sup> (geographic) as closed domain LOD. QA systems using LOD as search space are referred as QALD. Today we are witnessing the rise of a large volume of RDF data being published as Linked Data. The size of the Web of Data can be estimated based on the data set statistics that are collected by the LOD community in the ESW<sup>2</sup> wiki. The meta-information from CKAN<sup>3</sup> (DataHub<sup>4</sup>) is used to draw the LOD cloud diagram<sup>5</sup> and to maintain the statistics about the size of the Web of Linked Data on the EWS LOD front page. It is likely to contain around 180 datasets altogether having a size of around 20 billion RDF triples in 2010. For example, the DBpedia 2016-04 release consists of 9.5 billion RDF triples.

Current QA research topics include:

- Interactivity — clarification of questions or answers
- Answer reuse or caching
- Knowledge representation and reasoning
- Social media analysis with QA systems
- Sentiment analysis

---

<sup>1</sup> <http://linkedgeo.org>

<sup>2</sup> <https://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics>

<sup>3</sup> <http://ckan.net/>

<sup>4</sup> See <http://datahub.io/> for data set statistics

<sup>5</sup> <http://lod-cloud.net/>

## 1.4 NL Technologies Involved in QA Systems

Many NLP sub-tasks are involved in QA and many approaches can be followed for facing these tasks. Weston et al [27] have recently proposed a framework and a set of synthetic tasks with the goal of helping to develop learning algorithms for text understanding and reasoning and applying them to QA (authors use QA just for allowing a cheap and simple way of evaluating the proposed techniques). The goal is to categorize different kinds of questions into skill sets, which become their tasks<sup>1</sup>. The framework is useful for describing the problems of the QA systems most faces.

The defined tasks are the following:

1. Single Supporting Fact

Example: Mary went to the bathroom, Where is Mary?

2. Two Supporting Facts

Example: John is in the playground. John picked up the football. Where is the football?

3. Three Supporting Facts

Example: John picked up the apple. John went to the office. John went to the kitchen. Where was the apple before the kitchen?

4. Two Argument Relations

Example: The office is north of the bedroom. The bedroom is north of the bathroom. What is the bedroom north of?

5. Three Argument Relations

Example: Mary gave the cake to Fred. Fred gave the cake to Bill. Jeff was given the milk by Bill. Who did Fred give the cake to?

6. Yes/No Questions

Example: John moved to the playground. Is John in the playground?

7. Counting

Example: Daniel picked up the football. Daniel dropped the football. Daniel got the milk. Daniel took the apple. How many objects is Daniel holding?

8. Lists/Sets

Example: Daniel picks up the football. Daniel drops the newspaper. Daniel picks up the milk. What is Daniel holding?

9. Simple Negation

---

<sup>1</sup> The tasks are publicly available at <http://fb.ai/babi>. Source code to generate the tasks is available at <https://github.com/facebook/bAbI-tasks>.

Example: Fred is no longer in the office. Sandra is in the garden. Is Fred in the office?

10. Indefinite Knowledge

Example: John is in either the classroom or the playground. Is John in the classroom?

11. Basic Coreference

Example: Daniel was in the kitchen. Then he went to the studio. Where is Daniel?

12. Conjunction

Example: Mary and Jeff went to the kitchen. Then Jeff went to the park. Where is Mary?

13. Compound Coreference

Example: Daniel and Sandra journeyed to the office. Then they went to the garden. Where is Daniel?

14. Time Reasoning

Example: In the afternoon Julie went to the park. Yesterday Julie was at school. Julie went to the cinema this evening. Where did Julie go after the park?

15. Basic Deduction

Example: Sheep are afraid of wolves. Cats are afraid of dogs. Mice are afraid of cats. Gertrude is a sheep. What is Gertrude afraid of?

16. Basic Induction

Example: Lily is a swan. Lily is white. Greg is a swan. What color is Greg?

17. Positional Reasoning

Example: The triangle is to the right of the blue square. The red square is on top of the blue square. The red sphere is to the right of the blue square. Is the red square to the left of the triangle?

18. Size Reasoning

Example: The football fits in the suitcase. The suitcase fits in the cupboard. The box is smaller than the football. Will the box fit in the suitcase?

19. Path Finding

Example: The kitchen is north of the hallway. The bathroom is west of the bedroom. The den is east of the hallway. The office is south of the bedroom. How do you go from den to kitchen?

20. Agent's Motivations

Example: John is hungry. John goes to the kitchen. John grabbed the apple there. Daniel is hungry. Where does Daniel go?

## 1.5 Dimensions of QA Systems

The QA systems can be classified according to four interlinked dimensions [28] (see Table 1.1):

1. The input or type of questions it is able to accept (facts, dialogs, etc.)
2. How it copes with the traditional intrinsic problems that the search environment imposes in any non-trivial search system (e.g., adaptability and ambiguity)
3. The sources from which it can derive the answers (structured vs. unstructured data)
4. The scope (domain specific vs. domain independent)

Table 1.1: The dimensions of QA and query and search interfaces in general

Dimensions			
Input Types	Search Environment (Traditional Intrinsic Problems)	Sources	Scope
<ul style="list-style-type: none"> <li>▪ Keywords/definitions</li> <li>▪ Factoids (wh-, affirm / negate)</li> <li>▪ Understanding and causality reasoning (why, how)</li> <li>▪ Temporal and spatial reasoning</li> <li>▪ Facts from different sources</li> <li>▪ Common sense reasoning</li> <li>▪ Interactive dialogs</li> </ul>	<ul style="list-style-type: none"> <li>▪ Large scale (scalability)</li> <li>▪ Heterogeneity (mapping, disambiguation)</li> <li>▪ Openness (fusion, ranking)</li> <li>▪ Cross-lingual (multilingual)</li> <li>▪ Trust</li> </ul>	<ul style="list-style-type: none"> <li>▪ Structured (NLIDB)</li> <li>▪ Semi-structured (documents)</li> <li>▪ Textual (TREC, Web)</li> <li>▪ Semantic (ontologies)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Domain dependent (closed-domain)</li> <li>▪ Domain independent (open domain)</li> <li>▪ Proprietary KBs (private)</li> </ul>

At the Input level, the issue is balancing usability and higher expressivity at the level of the query, hiding the complexity of SQL-like query languages, while allowing the user to fully express his/her information [28]. QA systems are classified according to the complexity of the input question and the difficulty of extracting the answer, in several increasingly sophisticated types: systems capable of processing factual questions (factoids), systems enabling reasoning mechanisms, systems that fuse answers from different sources, interactive (dialog) systems and systems capable of deductive reasoning.

QA systems can also be classified according to the different sources used to generate an answer as follows:

- NL interfaces to structured data on databases (NLIDB traced back to the late sixties) [29].

- QA over semi-structured data (e.g., health records, yellow pages, Wikipedia info boxes).
- Open QA over free text, fostered by the open-domain QA track introduced by TREC<sup>1</sup> in 1999 (TREC-8<sup>2</sup>).
- QA over structured semantic data, where the semantics contained in ontologies provide the context needed to solve ambiguities, interpret and answer the user query.

Another distinction between QA systems is whether they are *domain-specific* (*closed-domain*) or *domain-independent* (*open domain*). Ontology-based QA emerged as a combination of ideas of two different research areas - it enhances the scope of closed NLIDB over structured data, by being agnostic to the domain of the ontology that it exploits; and presents complementary affordances to open QA over free text (TREC). The advantage is that it can help with answering questions requiring situation-specific answers, where multiple pieces of information (from one or several sources) need to be assembled to infer the answers at the run time. Nonetheless, most ontology-based QA systems are akin to NLIDB in the sense that they are able to extract precise answers from structured data in a specific domain scenario, instead of retrieving relevant paragraphs of text in an open scenario. Latest proprietary QA systems over structured data, such as TrueKnowledge<sup>3</sup> [30] and Powerset<sup>4</sup>, are open domain which has been limited to their own proprietary sources.

A challenge for domain-independent systems comes from the search environment that can be characterized by large scale, heterogeneity, openness and multilingualism [28]. In order to take a full advantage of the inherent characteristics of the semantic information space to extract the most accurate answers for the users, QA systems need to tackle various traditional intrinsic problems derived from the search environment, such as:

- Mapping the terminology and information needs of the user into the terminology used by the sources, in such a form that: (1) it can be evaluated using standard query processing and inference techniques, (2) it does not affect portability or adaptability of the systems to new domains, and (3) it leads to the accurate answer.
- Disambiguating between all possible interpretations of a user query. Independent of the type of query, any non-trivial NL QA system has to deal with ambiguity. Furthermore, in an open scenario, ambiguity cannot be solved by means of an internal unambiguous knowledge representation, as in domain-restricted scenarios. In open-domain scenarios, systems are involved in the problem of polysemous words, with different meanings according to different domains.

---

<sup>1</sup> <http://trec.nist.gov>

<sup>2</sup> [http://trec.nist.gov/data/qa/t8\\_qadata.html](http://trec.nist.gov/data/qa/t8_qadata.html)

<sup>3</sup> <https://www.evi.com/>

<sup>4</sup> <http://www.bing.com/>

- Applying knowledge fusion and ranking measures to select the best sources, fuse similar answers together, and rank the answers across sources. Because answers may come from different sources, and different sources have varying levels of quality and trust.
- With regard to scalability, there is a compromise between the complexity of the querying process, and the amount of the data systems can be used in response to a user request in a reasonable time.

Multilingualism issues, i.e., the ability to answer a question posed in one language using an answer space in another language, fostered by the Multilingual QA Track at the cross language evaluation forum (CLEF)<sup>1</sup> since 2002 [31], are not reviewed here. This is because in the context of QA in the open domain, challenges such as scalability and heterogeneity need to be tackled first to obtain answers across sources.

NL interfaces are an often-proposed solution in the literature for casual users [32], being particularly appropriate in domains for which there are authoritative and comprehensive databases or resources [33]. As stated in [34], iterative and exploratory search modes are important to the usability of all search systems, to support the user in understanding what is the knowledge of the system and what subset of NL is possible to ask about. Systems also should be able to provide justifications for an answer in an intuitive way (NL generation). This suggests the presence of unrequested but related information, and actively helps the user by recommending searches or proposing alternate paths of exploration. For example, view and form based search can help the user to explore the search space better than keyword-based or NL querying systems, but they become frustrating to use in large spaces and impossible in heterogeneous ones.

---

<sup>1</sup> <http://clef.isti.cnr.it>

## 1.6 Research Objectives

As discussed in Section 1.1, there are two types of QA paradigms. The first is IR-based QA which aims to pull answers from an unstructured collection of natural language documents (containing free text). The second is knowledge-based QA, which answers a NL question by mapping it to a query over a structured database or knowledge sources (Ontology, LD, KBs, etc.) by providing a convenient way to obtain knowledge. In general, QA is a retrieval task more challenging than common search engine tasks because its purpose is to find an accurate and concise answer to a question rather than a relevant document. This thesis focuses on semantic-based QA over ontology-based resources (not over free text) in closed and open domains. Therefore, the adaptations, technical and scientific needs of the NLP tools and methods have also been studied in order to be able to take advantages of the Semantic Web and ontology potentials in this research work.

The difficulty is more acute in tasks such as classifying question to find question type and determining concealed dependency relationship between words and terms in semantic aspect. It semantically addresses classifying questions and determining semantic type of the questions with manual annotations.

With regards to the challenges in semantic-based QA, the following items are the main research questions followed by the works carried out previously:

1. How accurately can we make the semantic QA system technology more coherent infrastructure and hopefully to be able to move it a step closer to a full-fledged QA system?
2. How to provide a set of semantically motivated question types able to be used without launchment to both restricted domain and open domain settings.
3. How can we formulate and generalize the constraints of the complex questions in order to find the grammatical and semantic relations in factoid and non-factoid questions?
4. What kind of structure can be generated in order to facilitate the inference mechanism to extract answer(s)?

There are subsidiary research objectives that are based on the type of QA system (closed domain or open domain). The following items are discussed in this thesis research:

- To develop an empirical method to find more related terms and seek expected answers by traversing the ontology items.
- To present a method in order to map the NL questions to formal query templates (e.g. SPARQL).
- To exploit the large volumes of LOD to answer user queries posed in natural language.

## 1.7 Summary of Contributions

The thesis has made five main contributions in IR technology, within the NLP area of Artificial Intelligence. We accomplished a comprehensive study of several QA systems. This work presents Semantic-based closed and open domain Question Answering System (ScoQAS), a new semantic-based QA system, which is aimed at question interpretation and extracting the answer from ontology/ies (not free text). This thesis does not address the application of free text (document) and the approaches related to its problem(s).

In this work, we present the technical contributions with focus on the challenges of QA systems, NLP techniques, query mapping, and answer inferencing in two different instantiations which can be used over ontology resource: closed domain (first scenario) and open domain (second scenario) under LOD resource. However, ScoQAS is not a complete IR system. The architecture of ScoQAS needs to be improved to deal with more complex questions such as those including anaphora and non-factoid questions like why and how. According to the recent survey on existing challenges in the QA systems in the Semantic Web, we bring here our contributions by considering the issues presented in the research done by Höffner et al [35]. The most significant contributions are summarized below.

The *first contribution* is designing and developing ScoQAS architecture as an end-to-end semantic-based QA system integrated with common and specific components over ontology resources (not free text) with two different prototyping: open and closed domains semantic QA system. This architecture has been developed based on the our initial model [36]. In ScoQAS, some components have been designed and implemented to solve the existing challenges that in the current QA systems have not been addressed. In question interpretation phase, it uses a heuristic method to facilitate the complexity of formulation in the syntactic-semantic relationships between question words approaching the expected precise answer. In the answer retrieval phase is exploited a graph-based inference algorithm.

The *second contribution* is the automatic process of user questions by presenting empirical semantic-based structure-feature pattern (Ss-fP) to classify the question in order to determine Question Type (QT) and Expected Answer Type (EAT). Our approach is the extraction of semantic features using rule-based method and NLP Interchange Format (NIF), which helps the ScoQAS to exploit it in both scenarios towards the interpretation of the question. Given the complexity of both the questions (at least in the first scenario), the rich tag sets have been defined carefully by hand-crafted rules that it seems the only adequate approach where direct learning of a classifier cannot be undertaken. This step represents the skeleton of the formulation of the question, which is carried out syntactically and semantically.

*Third contribution*, ScoQAS is developed by analyzing user's question and building a question syntactic-semantic information structure (QSiS), which is the basic block of knowledge to obtain the answer(s) to the questions. It is obtained after



processing several steps such as classification of the question, determining the EAT, and generating constraints. The constraints located in the main core of the QSiS which is used to formulate the related keywords in terms of syntax and semantics in order to utilize it in downstream steps. In addition, it uses dependency parsing, WordNet and Semantic Web technologies to build and complete the conceptual information of the QSiS. Although the same methodology is applied to implement the QSiS in both scenarios, the supplementary semantic information is added in the ontology-based closed domain. It provides a core dictionary of variables, which bind related terms to corresponding variables, automatically by providing two technical facilities. The first one, the constraints integrate all of the syntactic information from the dependency parsing along with lexical meaning and semantic information from WordNet or ontology. Moreover, the second is that the nature of the constraint indicates the unity formulation for each question type to make a dictionary of variables and relationship between them. This possibility allows providing the information needed to properly inference or even to deal with challenges in mapping question pattern to SPARQL query template.

The *fourth contribution* is defining and generating a question graph (QGraph) for representing core components of the question, further enriched with implicit knowledge (from the ontology in the first scenario). This graph is as a subgraph of the graph representing the domain and generating the graph format is done precisely, coherently and completely using the provided QSiS by the upstream process. The QGraph is used both as a search space for locating the answer and as a resource for enriching the constraints sets and EATs.

The *fifth contribution* is presenting an graph-based inference approach in the first scenario (closed-domain). A graph-based answer inference algorithm is applied to the provided QGraph format in the closed domain scenario. The structure of the QGraph format is analysed to find the relations between all of the involved variables and ontology entities that lead to the EAT ontology items. For many fundamental problems in Artificial Intelligence, adopting a graph-based framework can be straight-forward and very effective. The functionality of proposed algorithm (i.e. empirical technique) to extract precise answer from the whole semantic information generated for question is clearly significant.

## 1.8 Thesis Overview

In this section, we outline the content and organization of the remaining chapters of this document. After this introduction, Chapter 2 gives an extensive review of the state of the art, where the relevant approaches that have been proposed for semantic-based QA system are analyzed and presented. Some presented models of the QA system have been built within a specific domain. Some of them are *independent* or *open domains* such as QuestIO [12], AquaLog [37], DeepQA [38], [39] (IBM Watson<sup>1</sup>), QAKiS [40], SINA [41] and some are *dependent (closed) domains* like as QACID [17], ONLI<sup>+</sup> [42], and Pythia [26]. Moreover, PANTO [43], AquaLog [37]

---

<sup>1</sup> <http://www.ibm.com/smarterplanet/us/en/ibmwatson/>

and QuestIO [12] are systems that act as natural language interface. The framework, tools with combined solutions and techniques are introduced for IR, text mining and QA in NLP. Recently, specialized QA systems have been developed, such as EAGLi<sup>1</sup> for health and life scientists. An increasing number of QA systems use the World Wide Web or LOD as corpus of text and knowledge base.

Chapter 3 contains the general architecture of the proposed ScoQAS consisting of several steps, which are briefly explained below. Roughly, the core of the thesis work is presented in this chapter. The ScoQAS performs over ontologies not over free text and operates on two scenarios, one *Closed-domain*, where Enterprise ontology is in support of domain for questions and answers, and the other *Open domain* where the answers are retrieved from a LOD<sup>2</sup> knowledge base. There are common and specific components, modules, and KBs. Most of them are common and used in both scenarios. Other specific modules are individually described in this chapter. While we use other common components such as Stanford CoreNLP<sup>3</sup> parser, question preprocessing, NLTK WordNet, and SPARQL query engine, the specific components were also designed and implemented in this work such as the question representation, rule-based question classifier, the building constraints component (QSiS), the graph construction component, answer extraction component, pattern to SPARQL mapping, OpenLink Virtuoso, and the SPARQL query construction component.

In the first two steps of the ScoQAS, the initial pre-processing of the user question is done as a NLP parsing. Then the specific format is provided so that its content can be easily used in the next steps. In step 3, a method is introduced to represent the syntactic structure of a question (e.g. morphological analysis and dependency relations) which will be described more in details in Section 3.3. In step 4, the typology of question is presented and a question classifier is built by implementing a semantic-based structure-feature pattern approach. In step 5, the role of the remaining words is specified semantically and syntactically in the question. Thus, constraints are built to create the question syntactic-semantic information structure (QSiS) for associated question type. In order to do that, all of these words have been analyzed and determined in terms of position and their relationship with pattern items. Up to this step, all of the mentioned components operate for both scenarios alike, but there are separate components for each scenario in the downstream steps. To deal with our first scenario, in step 6, an empirical method is presented for creating a question graph that its nodes and edges indicate the dependencies between ontology entities and corresponding question variables. The constraints information is handled to produce the QGraph. This information has been generated at the time of the formation of the QSiS. Finally, in step 7, the inference method over QGraph format is utilized to extract the precise answer. For the 2<sup>nd</sup> scenario, the ScoQAS goes on the process of generating a formal query (SPARQL) through mapping a structural format using information obtained from upstream steps (e.g. the QC and Constraints modules). The structure of the produced SPARQL query templates are

---

<sup>1</sup> <http://bitem.hesge.ch/content/eagli-eagle-eye>

<sup>2</sup> <http://linkeddata.org/>

<sup>3</sup> <https://stanfordnlp.github.io/CoreNLP/index.html>

bound with constraints information. In order to get the answer, in the final step, the generated formal query will be sent to the Virtuoso<sup>1</sup> DBpedia endpoint<sup>2</sup> to crawl in LOD resource.

Chapter 4 provides an empirical evaluation of our implemented ScoQAS system. We evaluate the ScoQAS in both scenarios. Firstly, we analyze the first scenario with a set of questions, which were provided over the Enterprise ontology. Furthermore, in the second scenario, the preliminary results are analyzed and the accuracy of system is tested on QALD<sup>3</sup> training and test sets standard benchmark.

Chapter 5 presents the conclusions and the important aspects of the thesis work over research questions and summarizes the main contributions. At the end, the future research works are also suggested.

The thesis is closed by Appendices A, B, C, D, E, F, G, H, I, and J. Appendix A shows the list of the questions in the first scenario. Appendix B is a table of questions and corresponding question types for our training question chosen from QALD-3 training and test sets which were provided for the 2<sup>nd</sup> scenario. In the Appendix C, we show the details of bounded variables and corresponding constraints for sample question applied in the first scenario. The Appendix D shows the details of results of the ScoQAS over QALD-2 test set, which consists of 99 questions. The Appendix E shows the analysis parameters with results in detail, obtained by ScoQAS over QALD-3 test set. The Appendices F and G illustrate the analysis items in QALD-4 and QALD-5 test set respectively. Other results have been summarized in Section 4.3. Appendix H shows the generalized pseudo code for building QGraph. Appendix I contain our publications during this research work that show the basic and developed framework of the ScoQAS.

---

<sup>1</sup><https://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSSparqlProtocol#SPARQL%20Service%20Endpoint>

<sup>2</sup> <http://dbpedia.org/sparql>

<sup>3</sup> <http://qald.sebastianwalter.org/>



## 2 State of the Art

---

### 2.1 First Steps in QA: Natural Language Interfaces (NLIs)

The 1970s and 1980s witnessed the development of comprehensive theories in computational linguistics, which led to the development of ambitious projects in text comprehension and QA. The development of NLI to databases and other well-defined sources of structured information attracted the interest of the NLP research community. An example of such a system is the Unix Consultant (UC) [44], developed by Robert Wilensky at U.C. Berkeley in the late 1980s. The system answered questions pertaining to the UNIX operating system. It had a comprehensive hand-crafted knowledge base of its domain, and it aimed at phrasing the answer to accommodate various types of users. Another project was LILOG [45], a text-understanding system that operated in the domain of tourism information in a German city. The systems developed in the UC and LILOG projects never progressed to the stage of simple demonstrations, but they helped the development of theories regarding computational linguistics and reasoning.

A natural language interface to a database (NLIDB) [29] is a system that allows the user to have access to information stored in a database by typing requests expressed in a NL (e.g. English). Database query languages can be intimidating to the non-expert, leading to the immense recent popularity for keyword-based search in spite of their significant limitations. Chat-80 [46] is one of the best-known NLIDBs of the early eighties. Chat-80 was implemented entirely in Prolog. It transformed English questions into Prolog expressions, which were evaluated against the Prolog database.

Systems that also appeared in the mid-eighties were Datalog [47], Eufid [48], Ldc [49], Tqa [50], Teli [51], as well as many others. Although some of the numerous NLIDBs developed in the mid-eighties demonstrated successes characteristics in certain application areas, NLIDBs did not gain the rapid and wide commercial acceptance that was expected. The developments of successful alternatives to

NLIDBs, like graphical and form-based interfaces, as well as the intrinsic problems of NLIDBs are probably the main reasons for the unpopularity of NLIDBs. It was said in this time that the worst enemy of NLIDBs is the keyboard. Afterwards, NLIDBs continued to evolve and advance in the general NLP field, exploring architectures that transform NLIDBs into reasoning agents, and integrating language and graphics to exploit the advantages of both modalities, to name some of the lines of current research. The empirical base references such as FraCaS<sup>1</sup> corpus were analyzed. The FraCaS project, undertaken in the mid-1990s, was developed in wide spectrum of resources related to natural language inference and computational semantics [52]. The FraCaS consortium has a FraCaS test suite of NLI problems. It was targeted to collect a broad repository of examples of NLP problems. The problems have been comprised comparatively simple sentences, and the premise and question. There were 346 problems; each problem contains one or more premises and one question. There were a total of 536 premises, or an average of 1.55 premises per problem. The study shows that more challenging topics are quantifiers, plurals, anaphora, ellipsis, adjectives, comparatives, temporal, verbs, and attitudes. For instance, inference patterns involving temporal reference are complicated by the interplay between tense, aspectual information, lexical semantics, defeasible interpretation principles such as narrative progression, rhetorical relations, a theory of action and causation, world knowledge, interaction between plurality, genericity, and temporal/aspectual phenomena etc. Some of the inferences are very basic, some are more involved. The more complex examples give ample illustration of the fact that temporal phenomena are usually discursive phenomena. The NLI provides us more immediate applications, such as semantic search and QA [85]. The NLI is the problem of determining whether a natural language hypothesis “h” can reasonably be inferred from a given premise “p”. Let us consider the below example 1 to clarify the matter as follows:

Example 1: Monotonicity (upwards on first argument)

P1: Every Canadian resident can travel freely within Europe.

P2: Every Canadian resident is a resident of the North American continent.

Q: Can every resident of the North American continent travel freely within Europe?

H: Every resident of the North American continent can travel freely within Europe.

According to the authors, the inferencing tasks are the best way to test the semantic capacity of NLP systems.

The Text REtrieval Conferences (TREC) [53] were a series of workshops focusing on different IR tracks. They were co-sponsored by the National Institute of Standards and Technology (NIST<sup>2</sup>) and the Intelligence Advanced Research Projects Activity that was started in 1992 as a part of the TIPSTER Text Program<sup>3</sup>. Its

---

<sup>1</sup> <https://nlp.stanford.edu/~wcmac/downloads/fracas.xml>

<sup>2</sup> <https://www.nist.gov/>

<sup>3</sup> [http://www.itl.nist.gov/iaui/894.02/related\\_projects/tipster/](http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/)

purpose was to support research within the IR community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. This evaluation effort has grown both in terms of the number of participating systems and in terms of the number of tasks performed. The TREC test collections and software evaluations are available to the retrieval research community at large, so organizations can evaluate their own retrieval systems at any time. TREC has successfully met its goals of improving the state-of-the-art in IR and of facilitating technology transfer. In 2000, an European counterpart focusing on multilingualism was launched, called CLEF<sup>1</sup> (Cross Language Evaluation Forum). Since 2001, QA track has started to achieve more IR than just document retrieval by answering factoid, list and definition-style questions. The IBM research team that built IBM Watson (aka DeepQA, which beat the world's best Jeopardy! players<sup>2</sup>,) used data and systems from TREC's QA<sup>3</sup> track as baseline performance measurements [38]. QA Track of the TREC was holding from TREC-8(1999) to TREC2007 for open domain QA. The track primarily dealt with factual questions, and the answers provided by participants were extracted from a corpus of News articles. While the task evolved to model increasingly realistic information needs, addressing question series, list questions, and even interactive feedback, a major limitation remained: the questions did not directly come from real users, in real time. The LiveQA<sup>4</sup> track has been started since 2015. This track revives and expands the QA track, focusing on "live" QA for real-user questions. Real user questions, extracted from the stream of most recent questions submitted on the Yahoo Answers (YA) site that have not yet been answered by humans.

The Text Analysis Conferences (TAC)<sup>5</sup> are a series of workshops organized to encourage research in NLP and related areas, by providing a large test set collection, common evaluation procedures, and a forum for organizations to share their results. TAC comprises of a number of different tracks, each of which focuses on a particular sub problem of NLP. TAC tracks mainly focus on end-user tasks, but also include component evaluations within the context of end-user tasks. TAC is organized by the retrieval group of the Information Access Division (IAD) in the information technology laboratory at the National Institute of Standards and Technology (NIST). TAC initiated in 2008 and grew out of NIST's Document Understanding Conference (DUC) for text summarization, and the QA Track of the TREC. TAC is sponsored by NIST and other U.S. government agencies and is overseen by an advisory committee consisting of representatives from government, industry, and academia.

QA systems have been extended in recent years to encompass additional domains of knowledge. For example, systems have been developed to automatically answer temporal and geospatial questions, questions of definition and terminology, biographical questions, medical questions, multilingual questions, and questions about the content of audio, images, and video.

---

<sup>1</sup> <http://www.clef-campaign.org/>

<sup>2</sup> <http://www-03.ibm.com/press/us/en/presskit/27297.wss>

<sup>3</sup> <http://trec.nist.gov/data/qamain.html>

<sup>4</sup> <https://sites.google.com/site/trecliveqa2017>

<sup>5</sup> <http://www.nist.gov/tac/>

Comas, 2012, in his thesis presented a modular and flexible factoid QA system (SIBYL) operating over transcripts of speech questions which took advantage of several natural language analyzers, incorporating linguistic information from named entities, syntactic dependencies, and co-reference chains [54]. All of this information is obtained with machine learning tools. SIBYL could be adapted to other domains, or even other languages. Comas presented a method that can overcome part of automatic speech recognition errors using a sound measure of phonetic similarity based on phonetic sequence alignment. It can also be used in combination with traditional document ranking models. His experimental study shows that the use of co-reference resolution helps to increase the coverage of possible answer candidates from automatic transcripts, but the negative effect on the precision is larger, resulting in a general decrease in the overall performance.

Recently, QA on the web gained momentum due to the large structured knowledge bases such as DBpedia, Freebase<sup>1</sup>, YAGO, and YAGO2, that regularly collect information from open and ever-expanding knowledge resources such as Wikipedia [1][5][55][56]. Linked data extends the current Web that consists of documents and the links between documents. Linked data refers to a Web of data in contrast to a Web of documents. In the case of linked data or the Web of data, meaningful links with types between data elements exist, unlike the links in the Web of documents where links are only untyped references in the form of hyperlinks. DBpedia is one of the central linked data datasets in the Linked Open Data (LOD) project [1]. In computing, linked data is a method of publishing structured data so that it can be interlinked and become more useful through semantic queries. It builds upon standard Web technologies such as HTTP, RDF and URIs, but rather than using them to serve web pages for human readers, it extends them to share information in a way that can be read automatically by computers. This enables data from different sources to be connected and queried [2]. It is created by converting infobox information of Wikipedia articles to a RDF data model. The latest version of DBpedia contains more than several billion facts relating to persons, places, organizations, films, music albums, etc. in different languages. Linked open data is linked data that is open content [57][58]. Tim Berners-Lee gives the clearest definition of LOD in differentiation with linked data. He defines linked data by identifying its four components, and then adds a fifth component - open content - to define LOD [59]. Large LOD sets include DBpedia and Freebase.

By using an ontology, we can identify the meanings related to a domain, an enterprise or a society or even determine these meanings within different societies in detail [60]. Accessing structured data such as that encoded in ontologies and knowledge bases can be done using either syntactically complex formal query languages, or complicated form interfaces that require expensive customization to each particular application domain.

---

<sup>1</sup> <http://www.freebase.com/>



## 2.2 Introduction of QA Systems

QA has been studied for at least fifty years, with the earliest research beginning in the 1960s [61], [62]. Researchers have demonstrated that with enough manual effort, it is possible to engineer a QA system that can answer questions for a particular topic. However, there is no single system that can answer complex questions across many domains. The difficulty in creating such a QA system can be traced to two problems:

1. *Question Interpretation*: How does a QA system map questions to queries over its knowledge? Question interpretation involves inferring the information need of a question and then formulating a plan to obtain an answer.
2. *Knowledge Acquisition*: How does a QA system acquire and represent the knowledge needed to answer questions?

Here we address the most important QA systems with their strengths and weaknesses. We then present a selected set of recent and relevant QA systems.

### 2.2.1 ONLI<sup>+</sup> - Ontology Natural Language Interaction

ONLI<sup>+</sup> (Ontology Natural Language Interaction) [42] is a NL QA system used as the front-end to the RACER reasoner and to nRQL, RACER's query language [63]. The nRQL augments and extends Racer's functional API for querying a knowledge base as simply a T-box/A-box tuple (T, A). For instance, Racer provides a query function for retrieving all individuals mentioned in an A-box that are instances of a given query concept. ONLI<sup>+</sup> assumes that the user is familiar with the ontology domain and works through transforming the user's NL queries into nRQL. ONLI<sup>+</sup> takes user input in NL, translates the input into nRQL query format, submits the query to RACER, and presents the RACER output to the user after transforming it into NL. The architecture of ONLI<sup>+</sup> consists of Syntactic Analysis, Ontology Mapping, and Query Interface to RACER. ONLI<sup>+</sup> can handle three types of queries with quantifiers and number restrictions:

1. Unary concept queries with quantifier – e.g. “Find 5 fungi”
2. Binary role queries with quantifier – e.g. “Find 5 fungi that have been reported to have Pectinase”
3. Binary role queries with number restriction – e.g. “Find all fungi that have been described to have more than 3 enzymes”

To deal with quantifiers in unary and binary queries, ONLI<sup>+</sup> extracts the quantifier value from the user query and post-processes query results based on it. The system needs to recognize quantifiers from the parsed tree during the syntactic analysis phase. To do this, they analyzed a corpus of 36 questions and identified that all quantifiers are attached to an argument (noun phrase), and this argument is linked to a predicate (verb). For example, for the binary query with a quantifier “Find 2 vendors who sell enzyme products”, the generated parse tree is shown in Figure 2.1.

In the parse tree, the cardinality value 2 (noun) is syntactically related to the noun “vendor” and this noun is itself related to the verb “find” (the same relations hold true for unary queries with quantifiers). As a result, the system will accept this value as a quantifier.

To identify quantifiers, several syntactic clues are used, notably numerals and determiners. Then ONLI<sup>+</sup> uses the quantifier value 2 to prune the list of query results. If the ontology does not have enough instance names that match the query, then the system will show all available instances.

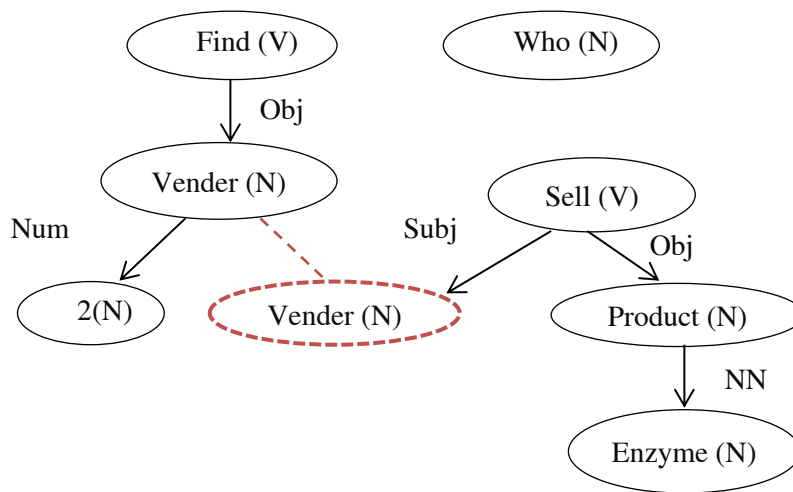


Figure 2.1: Minipar parse tree for the question “Find 2 vendors who sell enzyme products”

ONLI<sup>+</sup> extends the binary role query atoms handled by its predecessor, ONLI, with number restrictions. For example, “Find all fungi that have been described to have more than 3 enzymes”. In this example, they are imposing a constraint or number restrictions (more than 3) on the range of the roles ascribed to the enzyme. They considered nine types of number restrictions: *less than*, *less than equal to*, *more than*, *greater than*, *greater than equal to*, *at least*, *at most*, and *exactly* with a cardinality value. To deal with number restrictions on binary roles, ONLI<sup>+</sup> first analyzes the parsed question in the syntactic analysis phase. Then, it extracts the number restriction used in the query and its associated cardinality value and represents it as a triple structure:

<Modifier, Cardinality, Object>

Where the modifier specifies number restriction type, cardinality describes the cardinal value and the object describes which argument is modified by the number restriction.

### 2.2.2 PANTO-Portable nAatural laNguage inTerface to Ontologies

PANTO [43] models a Portable nAatural laNguage inTerface to Ontologies which accepts input NL forms and outputs SPARQL queries. It is based on a triple model in which a parse tree is constructed for the data model using the Stanford parser. Logic

rules are applied for NL queries such as negation, comparative and superlative form. For mapping, WordNet and string metric algorithms are used. The parse tree forms the intermediate representation as query triples form. Then, it converts the query triples form into OntoTriples form, which is represented as entities in ontology. OntoTriples are finally interpreted as SPARQL forms. A fully portable and widely used system for formalized knowledge bases is still unavailable. In [64] a major obstacle is mentioned. The ambiguity and complexity make it difficult for a machine to understand arbitrary natural language. The NLP community continues efforts to improve in this area. The state-of-the-art statistical parsers<sup>1</sup> can reach about 90% in terms of precision and recall. So PANTO utilizes a statistical parser (the Stanford Parser) to deal with this major obstacle [65].

### 2.2.3 Aqua Log - An Ontology-driven Question Answering

AquaLog [37] is a portable QA system which takes queries expressed in NL and ontology as input, and returns answers drawn from one or more KBs. AquaLog is capable of learning the user's jargon in order to improve his/her experience in time. In this system, two major models are used as the linguistic component, which is used to convert the NL questions into query triple format and Relation Similarity Service (RSS) which converts query triple form into onto-triple form. The data model consists of RDF triples.

AquaLog was implemented in Java as a modular web application, using client-server architecture. When a query is asked, the linguistic component's task is to translate the NL into the triple format used to query the ontology (Query-Triples). This preprocessing step helps towards the accurate classification of the query and its components by using standard research tools and query classification. Classification identifies the type of question, and as a result, the kind of answer required.

So far, AquaLog<sup>2</sup> is able to handle 16 main categories of queries based on pure linguistic criteria. In addition, inside each category, an analysis is done to detect different subcategories or different ways to process the queries, depending not only on linguistic but also on ontology information. The ontology helps us to reformulate and understand the query in terms of concepts, instances, values and relations between them. Meanwhile AquaLog is able to deal with some spelling mistakes. Queries in AquaLog are classified as follows:

- *Basic Queries*: They are basic, because they generate only one triple as an intermediate representation.
- *WH-3 Term Queries*: The linguistic representation is a ternary relation that may be translated into two onto triples.
- *Patterns or Combination of Two Basic Queries*: The queries, which belong to this category, may present NL ambiguities. Different methods are used to deal with ambiguities. The ambiguity can be resolved through the use of types and

---

<sup>1</sup> <http://nlp.stanford.edu/index.shtml>

<sup>2</sup> <http://technologies.kmi.open.ac.uk/aqualog/>

taxonomy in the ontology, the generic of a concept vs. an instance or user's feedback.

At average 63.5% of successive answers are retrieved from ontology with a closed domain environment [37].

The emergence of LOD initiatives has increased the number of large datasets available on the Semantic Web, and at the same time has brought additional challenges that PowerAqua [66], evolved from AquaLog, has addressed to support users in querying and exploring the current Semantic Web.

The query translation refers to how to transform the input query to a formal query. The existing approaches to deal with this transformation are as follows:

- Document-based (e.g., Swoogle [67], Watson [68])
- Entity-centric (e.g., SWSE [69])
- Question-Answering approaches (PowerAqua [66])

Using a QA approach, PowerAqua can automatically combine information from multiple knowledge bases at runtime. The input is a NL query and the output is a list of relevant entities. PowerAqua lacks a deep linguistic analysis and cannot handle complex queries.

### 2.2.4 QuestIO - Question-based Interface to Ontologies

QuestIO [12] system has a NLI for accessing structured information that is domain independent and easy to use without training. It brings the simplicity of Google's search interface to conceptual retrieval by automatically converting short conceptual queries into formal ones, which can then be executed against any semantic repository. The QuestIO application is open-domain (or customizable to new domains with very little cost), with the vocabulary not being predefined but rather automatically derived from the data existing in the knowledge base. The system works by converting NL queries into formal queries in SeRQL. It was developed specifically to be robust with regard to linguistic ambiguities, and incomplete or syntactically ill-formed queries, by harnessing the structure of ontologies, fuzzy string matching, and ontologically motivated similarity metrics. It works by leveraging the lexical information already present in the existing ontologies in the form of labels, comments and property values.

In [70] the CLOnE system, the predecessor of QuestIO, is presented. It provides a textual interface for editing a knowledge base through the use of an open-vocabulary, general purpose controlled language. It was designed as an interface for manual intervention in the process of generating ontological data from either structured information through direct mapping, or from unstructured text, through semantic annotation. QuestIO [12] uses Sesame<sup>1</sup> as a knowledge store. The system is configured to generate SeRQL as a query language. When a query is received, some of the contained words will match ontology concepts, while the textual segments that

---

<sup>1</sup> Sesame is an open-source RDF repository, <http://www.openrdf.org/>

remain unmatched can be used to predict property names and for disambiguation. The system performs using the following steps:

A) Initialization of the System:

When the system is initialized, it processes the domain ontology. Of great importance to the functioning of this system is the ability to recognize textual references to resources as being from the ontology or the knowledge base. This is done by automatically creating a gazetteer when initializing the system with a given knowledge base.

B) Run-Time Operation:

When a query is received, the system performs the following steps:

- a. Linguistic analysis.
- b. Ontological gazetteer lookup.
- c. Iterative transformation until a SeRQL query is obtained.
- d. Executing the query against the knowledge base and displaying the results.

C) Identifying Implicit Relations:

One of the main functions performed in the transformation step is to identify relations, which are not explicitly stated in the input query. After the ontological gazetteer is used to locate explicit references to ontology entities, the remaining text segments between those references are used to infer relations. Relations between query terms are essentially homologous with object properties in OWL, so the system attempts to match snippets from the input query to properties in the ontology. The list of applicable properties is used to generate candidate interpretations that are scored using the following metrics:

1. String Similarity Score
2. Specificity Score
3. Distance Score

D) Creating Queries:

After the implicit relations have been identified, the final interpretation of the input query is presented as a list of explicit references to ontology resources interspersed with references to properties. Using this list of interpretation elements, a formal SeRQL query is dynamically created. References to properties, either explicit or inferred, are used to impose restrictions with regard to relations between the various query elements, either instance URIs or variables.

### 2.2.5 QACID - Question Answering System Applied to the Cinema Domain

The QACID [17] system relies on five main components: the ontology, the data, the lexicon, the collections of user queries and the entailment engine. This approach has been applied and tested on Spanish language and using an ontology modeling the cinema domain. QACID is built on a collection of queries from a given domain, which are analyzed and grouped as clusters, and manually annotated using SPARQL

queries. Its domain is specific and its performance depends on the types of questions collected in the domain.

- *Ontology*: Ontology-based QA needs a formal representation of the information in the domain. Consequently, the ontology is one of the main entries to the system.
- *Data*: The ontology has been populated with information about the domain. This information has been provided by LaNetro, a company that provides leisure guides and updated tourism information all over Spain. The Perez Dolset family founded the company in Spain in 1996. The family launched the web portal LaNetro, whose objective was to offer interactive leisure and entertainment content accessible from any device with access to the internet.
- *Lexicon*: It was designed for establishing a mapping between words in NL queries and ontology instances. Entities in the ontology, including classes, properties and instances, are directly extracted from the data.
- *Collections of user queries*: These queries were automatically processed in order to derive a representative set of query patterns which define the user's requirements. These patterns were associated with their corresponding SPARQL queries that permit accessing the information required in the question from the RDF database. As a result, it obtains a set of pairs {question pattern, SPARQL query} that represent what we know as user query formulation database.
- *Entailment engine*: The core of the system is based on an entailment engine. This module uses entailment techniques to infer semantic deductions between a user query and the query patterns included in the user query formulation database previously obtained.
- *Question analysis*: The question analysis phase processes a NL query in order to obtain a formal representation (a.k.a. pattern) to be compliant with the user query formulation database.
- *Answer retrieval*: At first, the input query pattern is analyzed by the entailment engine, and then it returns a SPARQL query that permits the expected answer to be obtained. Then, the SPARQL generator substitutes the ontology concepts with the data instances, which exist in the NL question in order to launch the provided SPARQL query over the RDF database.

### 2.2.6 FREyA: Feedback, Refinement and Extended Vocabulary Aggregation

FREyA [71] is the successor of QuestIO, providing improvements with respect to a deeper understanding of a question's semantic meaning, to better handle ambiguities when ontologies are spanning diverse domains. FREyA allows users to enter queries in any form. Therefore, to identify the answer type of the question and present a concise answer to the user, a syntactic parse tree is generated using the Stanford parser. In addition, FREyA assists the user to formulate a query through the generation of clarification dialogs; the user's selections are saved and used for training the system in order to improve its performance over time for all users.

Similar to AquaLog's learning mechanism, FREyA uses ontology reasoning to learn more generic rules, which could then be reused for the questions with similar contexts (e.g., for the super classes of the involved classes). Given a user query, the process starts with finding ontology-based annotations in the query. In the case that there are ambiguous annotations, which are not resolved by reasoning over the context of the query (e.g., "Mississippi" can be a river or a state) then the user is engaged in a dialog scenario. The quality of the annotations depends on the ontology-based gazetteer *OntoRoot*, which is the component responsible for creating the annotations. The suggestions presented to the user in the clarification dialogs have an initial ranking based on synonym detection and string similarity. Each time a suggestion is selected by the user, the system learns to place the correct suggestions at the top for any similar question. These dialogs also allow translating any additional semantics into the relevant operations. Triples are generated from the ontological mappings taking into account the domain and range of the properties. The last step is generating a SPARQL query by combining the set of triples.

### **2.2.7 QASYO - Question Answering System for YAGO Ontology**

QASYO [72] is a sentence level QA system that integrates NLP, ontologies and IR technologies in a unified framework. It accepts queries expressed in NL and YAGO [5] ontology as inputs and provides answers drawn from the available semantic resources. Semantic analysis of questions is performed in order to extract keywords used in the retrieval queries and to detect the expected answer type. In the QASYO model, there are 4 phases: question classifier, linguistic component, query generator and query processor, which characterize its architecture as a waterfall model.

In fact, the NL query is translated into a set of intermediates, triple-based representations, and query-triples, these are translated into ontology-compatible triples. The whole QA process is composed of two consecutive steps: question analysis and answer retrieval. This model requires both an evaluation of its query answering ability. Another extension is to provide information about the nature and complexity of the possible changes required for the ontology and the linguistic component.

### **2.2.8 Pythia: Ontology-based Question Answering on the Semantic Web**

Pythia [26] compositionally constructs meaning representations using a vocabulary aligned to the vocabulary of a given ontology. In doing so, it relies on a deep linguistic analysis, which allows for constructing formal queries even for complex NL questions (e.g. involving quantification and superlatives). It is based on the following two main ideas: first, it uses principled linguistic representations in order to compositionally construct general meaning representations that can subsequently be translated into formal queries. Second, it relies on a specification of the lexicon-ontology interface that explicates possible linguistic realizations of ontology concepts. This allows the building of meaning representations that use a

vocabulary aligned to the vocabulary of a given ontology, thereby ensuring a precise and correct mapping of NL terms to corresponding ontology concepts.

In Pythia, NL expressions are parsed and interpreted with respect to a grammar, which they assume to be composed of two parts: an ontology-specific part and an ontology-independent part. The ontology-specific part contains lexical entries that refer to individuals, concepts, and properties of the underlying ontology. It is generated automatically from an ontology-lexicon model. The ontology-independent part comprises of functional expressions like auxiliary verbs, determiners, wh-words and so on. They assume grammar entries to be pairs of syntactic and semantic representation. As syntactic representation, they take trees from Lexicalized Tree Adjoining Grammar (LTAG) [73]. As semantic representations, they take DUDES [74], a kind of Underspecified Discourse Representation Structures (UDRS) augmented with information that allows for a flexible semantic composition. They used LexInfo<sup>1</sup> [75] framework, which offers a general frame for creating a declarative specification of the lexicon-ontology interface by connecting concepts of the ontology to information about their linguistic realization, i.e. word forms, morphology, sub-categorization frames and the way syntactic and semantic arguments correspond to each other.

### 2.2.9 DEQA: Deep Web Extraction for Question Answering

DEQA [76] is a framework for deep web QA approaching the problem as a combination of three research areas: (1) Web data extraction to obtain offers from real estate websites, where no structured interface for the data is available (which happens to be the case for all Oxford real estate agencies). (2) Data integration to interlink the extracted data with background knowledge, such as geo-spatial information on relevant points of interest. (3) QA to supply the user with a NLI, capable of understanding even complex queries. For example, a query like “find me a flat to rent close to Oxford University with a garden” can be answered by DEQA.

DEQA focuses on extracting answers to such questions. It achieves this by mapping NL questions to SPARQL patterns. These patterns are then evaluated on an RDF database of current real estate offers. The offers are obtained using OXPath [77] [78], a state-of-the-art data extraction system, on the major estate agencies in the Oxford area and linked through LIMES [79] to background knowledge such as the location of supermarkets. The TBSL approach [80] is employed for translating NL questions into SPARQL queries. TBSL disambiguates entities in the queries and then maps them to templates, which capture the semantic structure of the NL question. This enables the understanding of even complex natural language containing, e.g., comparatives such as *higher than* and *more than* and superlatives like *the highest* in contrast to most other QA systems that map NL input to purely triple-based representations.

---

<sup>1</sup> <http://lexinfo.net>



### 2.2.10 QAAL

Kalaivani and Duraiswamy [81] survey different types of QA systems based on ontology and Semantic Web model with different query formats. For comparison, the types of input, query processing method, input and output format of each system and the performance metrics with its limitations were analyzed and discussed. In line with previous QA system challenges, QAAL [81] is presented for implementing factoid-based question types. Basic terms in factoid model includes who, whom, why, what, where, when, what, which, i.e., “Wh- Questions”. The template-based approach is used for fast retrieval of answers. If the question is already asked in that system, the retrieval takes place within the question template table, otherwise it is performed using the graph matching algorithm and the spread activation algorithm for query matching with the ontology. The main modules are explained briefly below:

#### A) Graph Matching in Ontology:

Conceptual Graph (CG) acts as an intermediate language for mapping NL questions and assertions to a relational database. CG contains concepts, concept relations, and arguments. QAAL system was implemented as a Semantic Web concept that can be represented by RDF. Information resources are commonly represented as Uniform Resource Identifiers (URIs). URI's content is described by RDF relations. RDF triples are visualized as a directed labeled graph in which subject and objects are represented as nodes, and predicates as arcs. RDF graph matching is practically implemented by using SPARQL language in ontology domain to modelling semantic search.

#### B) Spread Activation:

Spread Activation [82] is a process for searching the nodes in ontology. It looks and finds relations between nodes in ontology. Nodes may be terms, class, property, etc. and relations are labeled in a directed or weighted manner. SA algorithm creates initial nodes that are related to the content of the user's query and assign weights to them. After that, nodes will activate with different nodes on ontology guided by a set of rules.

#### C) Question Classification Methods:

QAAL is used for implementing factoid based question types such as wh-questions. There are three types of question classification methods include machine learning approaches, knowledge-based approaches and template-based approaches. In the QAAL system, the template-based approach is used for extracting the answers. If the question has been already asked in that system, then the retrieval task finds the associated query from question template table, otherwise the matching algorithm is exploited. In the QAAL, question templates are produced in a particular domain. Depending on the user's question, initially the matching is searched with the question template and the answer is retrieved from it when the matching process is successful. Otherwise, semantic searching is processed by implementing the query reformulation strategy.

### 2.2.11 QAKiS: Question Answering wiKiframework-based System

QAKiS [40] is a system for open domain QA over linked data. It addresses the problem of question interpretation as a relation-based match, where fragments of the question are matched to binary relations of the triple store, using relational textual patterns automatically collected. This system allows end users to submit a query to an RDF triple store in English and obtain the answer in the same language, hiding the complexity of the non-intuitive formal query languages involved in the resolution process. QAKiS addresses the task of QA over structured KBs (e.g. DBpedia) where the relevant information is expressed also in unstructured form (e.g. Wikipedia pages). Its major novelty is to implement a relation-based match for question interpretation, to convert the user question into a query language (e.g. SPARQL). It tries first to establish a matching between fragments of the question and relational textual patterns automatically collected from Wikipedia. The underlying intuition is that a relation-based matching would provide more precision with respect to matching on single tokens.

QAKiS performs two main steps: a) the query generator takes the user question as input, generates the typed questions, and then generates the SPARQL queries from the retrieved patterns; b) the pattern matcher takes as input a typed question, and retrieves the patterns matching it with the highest similarity. The main modules of this system are:

- *Expected Answer Type (EAT) and Named Entity (NE) identification*: The target of the question is identified with a NER tool. The Stanford Core NLP NE Recognizer is applied with a set of strategies based on the comparison with the labels of the instances in the DBpedia ontology. Simple heuristics are applied to conclude the EAT from the question keyword, e.g. if the question starts with “When”, the EAT is [Date] or [Time], with “Who”, the EAT is [Person] or [Organization] and so on.
- *Typed questions generation*: Generating a typed question by replacing the question keywords and the NE by the types and super types. In a given question like “Who is the husband of Amanda Palmer?” Nine typed questions are generated, since i) both [Person] or [Organization] (subclasses of [owl: Thing]) are considered as EAT, and ii) [Musical Artist], [Artist] and [owl: Thing] are the types of the NE Amanda Palmer.
- *QAKiS based on wiKiframework*: WikiFramework establishes a 4-step methodology to collect relational patterns - automatically extracted from Wikipedia and collected in the WikiFramework repository [83] - in several languages for the DBpedia ontology relations: i) a DBpedia relation is mapped with all the Wikipedia pages in which such relation is reported in the Infobox; ii) in such pages are collected all the sentences containing both the domain and the range of the relation; iii) all sentences for a given relation are extracted and the domain and range are replaced by the corresponding DBpedia ontology classes; iv) the patterns for each relation are clustered according to the lemmas between the domain and the range.

- *Query selector*: A set of patterns is retrieved by the pattern matcher component for each typed question, and sorted by decreasing matching score. For each of them, one or two SPARQL queries are generated. Such queries are then sent to the SPARQL endpoint for answer retrieval. If the query produces no results, it tries with the next pattern, until an adequate query is found or no more patterns are retrieved.

### 2.2.12 PARALEX - Open QA System

One of the desired properties of a QA system is to be robust to the variations in NL questions in order to provide a natural, declarative interface. Anthony Fader, in his thesis, [84] presented an open knowledge intensive QA system that maps questions onto simple queries against open IE extractions, by learning paraphrases from a large monolingual parallel corpus, and performing a single paraphrasing step.

Table 2.1: An example cluster of questions

1	Can US citizens gamble online?
2	Can you gamble online in America?
3	Internet gambling is legal in the US?
4	Is betting online legal in US?
5	Is online casino legal in the US?
6	Is online gambling forbidden in the US?
7	Online gambling in US is legal?

For example, a QA system should be able to infer that all of the questions in Table 2.1 different ways of asking “Is online gambling legal in the US?” and route them to the same answer. PARALEX [85] is the system that performs open QA over an extracted knowledge base. It uses paraphrases from WikiAnswers<sup>1</sup> to learn a function from questions to knowledge-base queries. Fader uses a PARALEX system that learns a robust question-interpretation function from the paraphrase data available on WikiAnswers. PARALEX is a large monolingual parallel corpus, containing 18 million pairs of question paraphrases from WikiAnswers, which were tagged as having the same meaning by users.

Figure 2.2 shows an example of how open QA maps the question “How can you tell if you have the flu?” to the answer “chills” over four steps.

<sup>1</sup> <http://wiki.answers.com/>

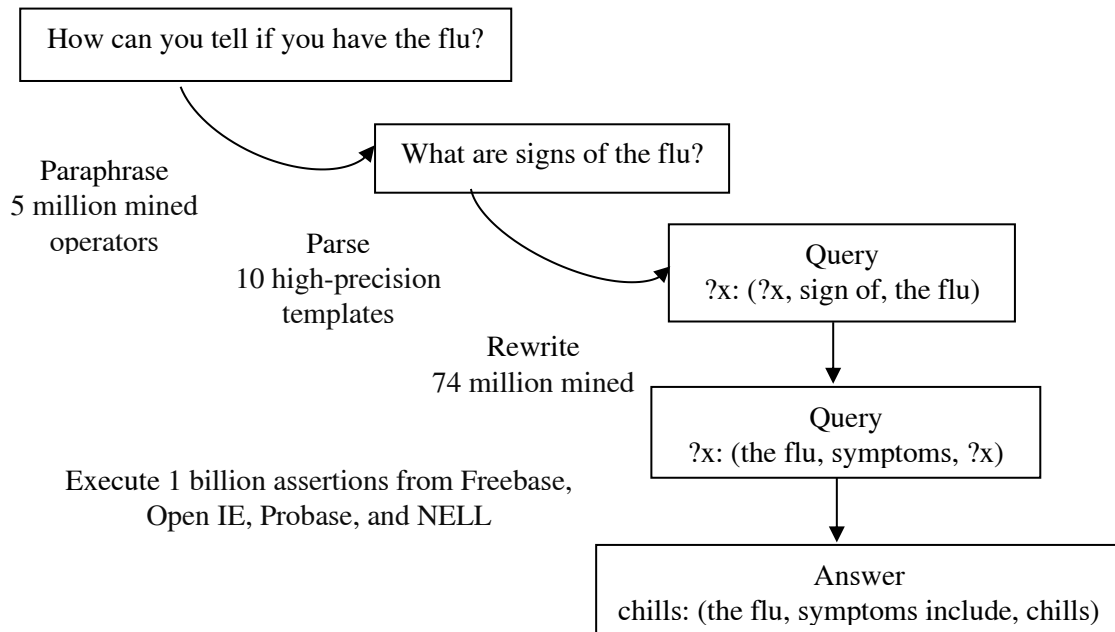


Figure 2.2: Mapping question into answer in the PARALEX Open QA

### 2.2.13 SINA

Shekarpour et al [86] presents SINA, a scalable keyword search system that can answer user queries by transforming user-supplied keywords or NL queries into conjunctive SPARQL queries over a set of interlinked data sources. SINA uses a Hidden Markov Model (HMM) to determine the most suitable resources for a user-supplied query from different datasets. Moreover, the framework is able to construct federated queries by using the disambiguated resources and leveraging the link structure underlying the datasets to query. An approach presented by Shekarpour et al [87] utilizes a set of predefined basic graph pattern templates for generating adequate interpretations of user queries. This is achieved by obtaining ranked lists of candidate resource identifiers for the supplied keywords and then injecting these identifiers into suitable positions in the graph pattern templates. The main advantages of this approach are that it is completely agnostic of the underlying knowledge base and ontology schema, that it scales to large knowledge bases and is simple to use. For learning DBpedia graph pattern templates, a set of 1,000 distinct query logs from DBpedia endpoint<sup>1</sup> has been analyzed. The article evaluated all 17 possible valid graph pattern templates by measuring their precision and recall on 53 queries against DBpedia.

<sup>1</sup> The DBpedia SPARQL endpoint is available at: <http://dbpedia.org/sparql/> and the query log excerpt at: <ftp://download.openlinksw.com/support/dbpedia/>

### 2.2.14 DEANNA

Yahya et al [88] proposed the system DEANNA in 2012 and then the improved in [89]. To my knowledge, it is the only system approaching QALD from integer linear program (ILP) perspective. DEANNA extends the classical subject-predicate-object (SPO) triples:

- KeithRichards composed Angie.
- Womack & Womack performed Angie.
- Angie type Music.

It uses SPOX quadruples (qads) where a textual context is added, for example:

- Angie type Music {"...a ballad which tells of the end of a romance ..."}.

The authors have formalized the problem and designed an ILP for jointly resolving the question decomposition and disambiguation problems. Their model couples the selection of phrases and their mapping onto semantic targets. They have introduced constraints that ensure that phrases are selected in a way that preserves their phrase dependencies in the image of the mapping onto semantic targets. Two kinds of pre-computed weights were used: (i)  $S(i, j)$  denotes a prior score for phrase  $P_i$  mapping to semantic target  $S_j$ , regardless of the context, and (ii)  $R(k, l)$  denotes the semantic relatedness between semantic target items  $k$  and  $l$ , based on co-occurrences in the underlying data and knowledge sources (Yago, DBpedia, Wikidata<sup>1</sup>, Wikipedia), to integrate the question context in scoring. A set of 14 constraint templates is used for the ILP program.

The query process is carried out by the following step:

1. Query generation
2. Query extension from SPO to SPOX
3. Query relaxation when executing the query leads to no results
4. Ranking when relaxation produces many results

## 2.3 A Summary of Analysed QA Systems

Table 2.2 analyzes the main characteristics of the QA systems, which we have presented.

Table 2.2: Comparison of QA systems

QA System	Features	Limitations
ONLI [42]	<ul style="list-style-type: none"> <li>▪ NL queries into nRQL</li> <li>▪ Syntactic analysis</li> <li>▪ Ontology mapping</li> <li>▪ Query interface to RACER</li> </ul>	<ul style="list-style-type: none"> <li>▪ Dependent domain</li> <li>▪ Limited Question Type</li> </ul>

<sup>1</sup> [https://www.wikidata.org/wiki/Wikidata:Main\\_Page](https://www.wikidata.org/wiki/Wikidata:Main_Page)

<p><b>PANTO</b> [43]</p>	<ul style="list-style-type: none"> <li>▪ Using WordNet and String metric algorithms for mapping</li> <li>▪ Input as NL and the output is in SPARQL query.</li> <li>▪ QueryTriples as an intermediate representation</li> <li>▪ Converts query triples form into OntoTriples</li> </ul>	<ul style="list-style-type: none"> <li>▪ Scalability: work with small ontology</li> <li>▪ No database indexing technique</li> <li>▪ Restrictions on query scope (cannot totally interpret semantics)</li> <li>▪ Weakness in user interaction</li> </ul>
<p><b>AquaLog</b> [37]</p>	<ul style="list-style-type: none"> <li>▪ Grammars are domain independent</li> <li>▪ Queries in NL</li> <li>▪ Using string metric algorithm</li> <li>▪ Use of the GATE NLP platform, WordNet</li> <li>▪ Ontology-based Relation Similarity Service (RSS) from triples to answers</li> </ul>	<ul style="list-style-type: none"> <li>▪ Lack of appropriate reasoning services defined by ontology</li> <li>▪ Does not understand queries formed with “How much”.</li> <li>▪ Does not support questions implying anaphora resolution.</li> <li>▪ Does not exploit quantifier scoping (“each”, “all”, and “some”).</li> </ul>
<p><b>QuestIO</b> [12]</p>	<ul style="list-style-type: none"> <li>▪ The application is open-domain</li> <li>▪ Translates a NL or a keyword-based question into SPARQL linguistic analysis</li> <li>▪ Ontological gazetteer lookup</li> <li>▪ Iterative transformation until a SeRQL query is obtained</li> </ul>	<ul style="list-style-type: none"> <li>▪ Is not session-based interaction</li> <li>▪ Is not able to disambiguate keyword terms</li> </ul>
<p><b>QACID</b> [17]</p>	<ul style="list-style-type: none"> <li>▪ Tested on the Spanish language in the Cinema domain</li> <li>▪ Collection of queries as clusters</li> <li>▪ Mapping NL queries into knowledge base by using String Distance Metrics</li> </ul>	<ul style="list-style-type: none"> <li>▪ Costly because of domain dependence</li> <li>▪ Can only be applied with limited coverage</li> <li>▪ Lack of temporal and spatial context aware capabilities</li> </ul>
<p><b>FREyA</b> [71]</p>	<ul style="list-style-type: none"> <li>▪ Identification and verification of ontology concepts</li> <li>▪ Open domain</li> <li>▪ Generating SPARQL</li> <li>▪ Answer Type Identification</li> <li>▪ Reinforcement Learning to improve ranking of suggestions</li> <li>▪ Session based interaction</li> </ul>	<ul style="list-style-type: none"> <li>▪ Needs to test with large datasets</li> <li>▪ Evaluation is not user-centric</li> </ul>

<b>QASYO</b> [69]	<ul style="list-style-type: none"> <li>▪ NL query</li> <li>▪ YAGO ontology (input)</li> <li>▪ NL query gets translated into a set of intermediate, triple-based representations, query-triples</li> <li>▪ Translated into ontology-compatible triples.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Lack of information about the nature and complexity of the possible changes required for the ontology and the linguistic component</li> </ul>
<b>Pythia</b> [26]	<ul style="list-style-type: none"> <li>▪ Handling a wide range of linguistically complex queries, involving quantifiers, numerals, comparisons and superlatives, negation, ...</li> <li>▪ Mapping correctly NL terms to corresponding ontology concepts, even if they are superficially different.</li> <li>▪ Domain-specific lexicon is built automatically from a specification of linguistic realizations of ontology concepts.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Portability (requires the creation of a new LexInfo model for a new domain)</li> <li>▪ Requires non-negligible effort for larger domains (e.g. DBpedia)</li> </ul>
<b>DEQA</b> [76]	<ul style="list-style-type: none"> <li>▪ Applied for web of open data</li> <li>▪ Using TBSL algorithm</li> <li>▪ Comprehensive deep web QA system</li> <li>▪ Web Extraction with XPath</li> <li>▪ Using LIMES for computing complex link specifications</li> </ul>	<ul style="list-style-type: none"> <li>▪ Needs to cover more question types</li> <li>▪ Does not support complex operators</li> <li>▪ Does not support Multilingualism</li> </ul>
<b>QAAL</b> [81]	<ul style="list-style-type: none"> <li>▪ Conceptual Graph matching</li> <li>▪ Using SPARQL language</li> <li>▪ Using NLP to analyze Q &amp; A</li> <li>▪ Using Spread Activation Algorithm</li> </ul>	<ul style="list-style-type: none"> <li>▪ Normal keyword search model</li> <li>▪ Cannot answer complex questions if ambiguity occurs</li> <li>▪ Specific domain</li> </ul>
<b>QAKiS</b> [40]	<ul style="list-style-type: none"> <li>▪ Open domain</li> <li>▪ QA over structured knowledge base (e.g. DBpedia)</li> <li>▪ Relevant information in unstructured form (e.g. Wikipedia).</li> </ul>	<ul style="list-style-type: none"> <li>▪ Not able to deal with Boolean and n-relation questions.</li> <li>▪ Not able to analyze Procedural, Temporal or Spatial</li> </ul>
<b>PARALEX</b> [84]	<ul style="list-style-type: none"> <li>▪ Open domain</li> <li>▪ Transforming text to tuple</li> <li>▪ Paraphrase-Driven learning to interpret questions</li> <li>▪ No manual templates have to be created</li> </ul>	<ul style="list-style-type: none"> <li>▪ Lack of answerability</li> <li>▪ Not able to analyze complex questions</li> </ul>

<p><b>SINA</b> [86]</p>	<ul style="list-style-type: none"> <li>▪ Open domain</li> <li>▪ Lexical gap</li> <li>▪ Using Hidden Markov Model (HMM)</li> <li>▪ Template-based</li> </ul>	<ul style="list-style-type: none"> <li>▪ Not able to analyze Procedural, Temporal or Spatial</li> <li>▪ Does not support complex operators</li> <li>▪ Does not support Multilingualism</li> </ul>
<p><b>DEANNA</b> [88][89]</p>	<ul style="list-style-type: none"> <li>▪ Approaching integer linear program (ILP) perspective</li> <li>▪ Query extension from SPO to SPOX</li> <li>▪ Query relaxation when executing the query leads to no results</li> </ul>	<ul style="list-style-type: none"> <li>▪ Not able to analyze Procedural, Temporal or Spatial</li> <li>▪ Does not support complex operators</li> <li>▪ Does not support Multilingualism</li> <li>▪ Does not use Templates</li> </ul>



---

## **3 The Architecture of Semantic-based QA System (ScoQAS<sup>1</sup>)**

---

### **3.1 The Components of the ScoQAS Architecture**

In Figure 3.1, the architecture of the semantic-based QA system is shown. We, Latifi et al [90], have recently introduced a Semantic-based closed and open domain Question Answering System (ScoQAS) that is based on NLP techniques using semantic-based structure-feature patterns for question classification. It uses a technique integrating syntactic parsing, lexical meaning (e.g. WordNet) and semantic information (e.g. ontology) by building constraints to formulate the related terms on syntactic-semantic aspects. The ScoQAS was developed in the following our initial model, Latifi et al [36], which we briefly introduced the state of the art QA systems. In [36] we presented a framework of the semantic QA for enterprise domain. The significant roles of NLP techniques and ontologies to develop the current search engines and implement a semantic QA system were addressed. The research questions were raised ahead to interpret the input textual question and how to find the answer for the corresponding question. The initial model was a representation of the idea based on our previous knowledge and it was a basic approach to make the bridge in order to map textual queries to formal queries. The ScoQAS was developed in the current thesis work to overcome the weaknesses raised from different semantic QA models. Although there are differences in the number and type of the modules with the initial model but the most of our focus was on how to interpret the questions to be closer a step to the precise expected answer. In this chapter, with regard to the lack of standard implemented modules, the complexity of such systems is evident. For this reason, our research has been conducted to model and implement components in an integrated manner in order to approach the desirable standards in end-to-end semantic QA system. Most of our research activities have been focused

---

<sup>1</sup> Semantic-based closed and open domain Question Answering System

on modelling and developing of constraints and exploiting techniques based on ontology and Semantic Web technology in order to formulation of user's questions. This proposed model attempted to provide an effective empirical method to resolve the challenges posed by the inference of the expected answer (in the first scenario). Its effectiveness could be confirmed by introducing another module on how to map questions into formal semantic queries (in the second scenario). Aligning with other existing state-of-the-art QA systems at this stage, it represents a considerable impact of the introduced ScoQAS system to infer the precise answer.

The ScoQAS is based on NLP techniques by creating the question syntactic-semantic information structure (QSiS). It applies a graph-based inference algorithm by providing heuristic constraints in order to extract the expected answer. The ScoQAS performs over ontologies, not over free text, and operates on two scenarios: the first scenario is *Closed-domain* QA system, where the domain is restricted by a human-made ontology, and the second scenario is *Open-domain* QA system where the answers are retrieved from a LOD knowledge base.

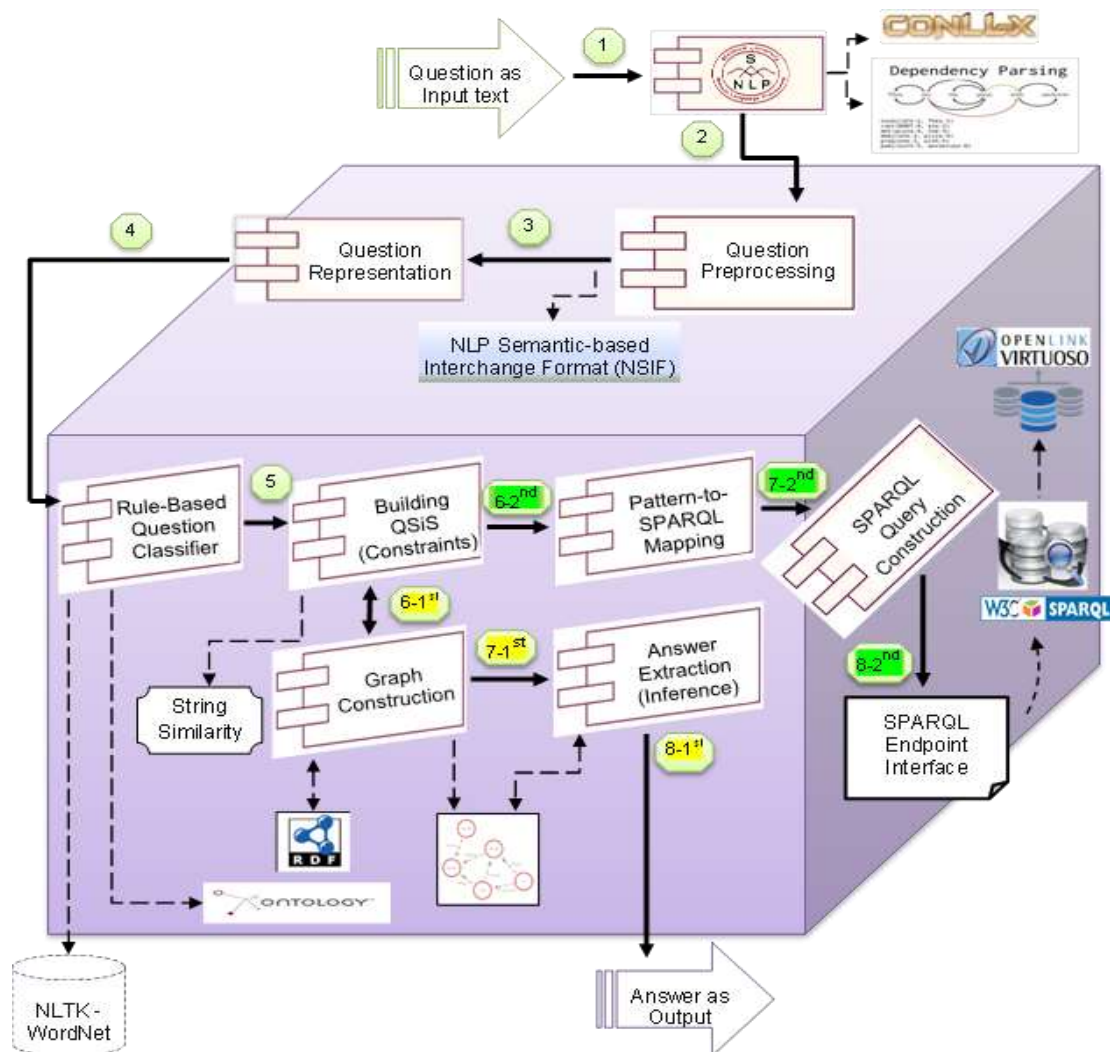


Figure 3.1: Architecture of Semantic-based Question Answering System (ScoQAS)

The purpose of these scenarios for this system is to sketch out the specific behavior in two different settings with the aim of uniting the pros and cons for designing and implementing the integrated approach, in order to demonstrate the adaptabilities of our approach to achieve the desired results in a semantic-based QA system. There are common and specific components, modules and KBs where most of them are used in both scenarios. A number of these modules have been adapted from other systems and redesigned to integrate into ScoQAS. In overall, both of the scenarios use a same approach for interpreting the question by providing formulated constraints only with a few minor differences in implementation.

In the following sections each of these modules belonging to the corresponding domain are explained and discussed.

### 3.1.1 Common Components in ScoQAS

As shown in Figure 3.1, those of the components, which are outside of the cube, are reused as external tools by ScoQAS. The common components are *Stanford CoreNLP<sup>1</sup> parser* and *NLTK WordNet*. They are used, as needed, in part of the process of interpreting the question or extracting the answer. Hence, the others, specific components, were individually developed in this work including *Question Preprocessing*, *Question Representation*, *Rule-based Question Classifier*, *Building QSiS (Constraints)*, *Graph Construction*, *Pattern-to-SPARQL Mapping*, *SPARQL Query Construction*, *SPARQL Endpoint Interface (OpenLink Virtuoso<sup>2</sup>)* and *Answer Extraction (Inference)*. These specific components are described separately in the next sections in terms of exploiting them in closed or open domains. In order to conduct our experiments in two scenarios, the ScoQAS has been split into two contiguous phases. The first phase belongs to steps 1 to 5 and the second phase pertaining to steps 6- $\{1^{st}/2^{nd}\}$  to 8- $\{1^{st}/2^{nd}\}$  where the both scenarios are separated in two branches and integrated in the same framework to fulfill the end to end QA process. We have categorized the first phase, steps 1 to 5, as common components for both scenarios.

In the step 1, the ScoQAS get the input text question and uses the Stanford CoreNLP and its dependency parsing tools to produce the POS, lemma, morphological analysis, syntactic dependencies, etc. in the specialized structured format. The question preprocessing phase, step 2, a NLP Interchange Format (NIF) has been defined for representing syntactic information extracted from the question. In step 3, the goal of this component is to providing a capability to access and control the existing preprocessing data in the NIF and its relations with each of the tokens such that lexico-semantic information (e.g. ontology item) can be simply manipulate in the structured memory of ScoQAS. Step 4 aims to classify a question in order to bind Question Type (QT) and determine the Expected Answer Type (EAT) using rule-based approach. The last common component, building QSiS, analyzes and extracts the further constraint information and formulating the syntactic-semantic

---

<sup>1</sup> <https://stanfordnlp.github.io/CoreNLP/index.html>

<sup>2</sup> <https://virtuoso.openlinksw.com/>

structure of question in a way that automatically can be utilized in downstream process.

### 3.1.2 Specific Components for the Closed-Domain Scenario

As shown in Figure 3.1, the closed-domain scenario was highlighted by yellow color from step 6 while the open domain was determined by bright green color. In the Closed-domain scenario being ontology-guided, the approach of the building constraints (steps 6-1<sup>st</sup>) is carried out with implementation of ontology traversing to enrich the semantic aspect as well as syntactic parsing information. The Graph Construction (step 7-1<sup>st</sup>) has a task to generate a QGraph based on formulated constraints. The Answer Extraction (step 8-1<sup>st</sup>) task is responsible to make inference in order to find the candidate answers.

### 3.1.3 Specific Components for the Open Domain Scenario

In contrast to the first scenario, there are specific components for the second scenario include Pattern-to-SPARQL mapping (i.e. mapping method applied to generate SPARQL format to fetch information from well-formed constraints). In this step, specialized SPARQL template for a recognized QT and pattern are constructed. Given the SPARQL template(s), the system bind the SPARQL Endpoint Interface (OpenLink Virtuoso<sup>1</sup>) to crawl in the LOD resources, and finally list of resources are returned as candidate answers. These components have been specifically dedicated for the Open domain which they are executed in 6-2<sup>nd</sup>, 7-2<sup>nd</sup>, and 8-2<sup>nd</sup> steps, as shown in Figure 3.1. We can distinguish between different components on the answer extraction step where the first scenario uses a search mechanism over a provided QGraph produced by information exist in the QSiS in order to do answer inference while the second scenario performs a mapping to SPARQL queries to crawl in the LOD resources to return a list of candidate answers.

---

<sup>1</sup> <https://virtuoso.openlinksw.com/>

### 3.2 Question Preprocessing

In order to achieve the aims of the semantic-based QA system, the ScoQAS system attempts to figure out the structure of the question syntactically and semantically through carrying out the NLP tasks. If it succeeds, then the next steps in the process will be carried out based on this information. To this end, in the preprocessing step of the ScoQAS architecture, an NIF of the question is built by applying the most recent version (Ver 3.8) of Stanford CoreNLP Parser with DependencyParseAnnotator information. This parser has been chosen because, despite its limitations, it has provided the better results compared with other state of the art available alternatives including SENNA<sup>1</sup>, SyntaxNet<sup>2</sup>. NIF has been defined for representing syntactic parsing results from the question. It is generated in order to use it as an enriched representation of the question in the mapping or in the answer retrieval process. The NIF contains both lexical and relational information from the dependency parsing results for each question. The primary information of the NIF consists of tokenization and morphological analysis such as lemmatization, POS tagging, and named entity recognition (NER) for each question. Despite of similarity between lemmatization and stemming, the lemmatization depends on correctly identifying the intended part of speech and meaning of a word in a sentence. However, the two words differ in their flavor, stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

In addition, the NIF uses dependency parsing information in order to enrich the syntactic-semantic information of the question. The NIF helps the ScoQAS exploit it in both scenarios and the same methodology is applied to produce its content. This information is the foundation to initiate the structure of the QSiS. For example, the parsing information for Q1 (in the first scenario): “Where is the manager of ITC working in the organization?” is shown as a graphical schema in Figure 3.2 and the initial content of the NIF format as shown in Figure 3.3.

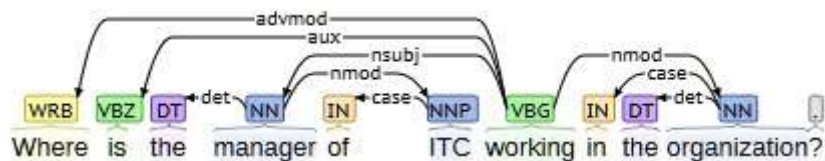


Figure 3.2: Basic dependencies in Stanford CoreNLP parser over Q1

<sup>1</sup> <https://ronan.collobert.com/senna/>

<sup>2</sup> <https://github.com/tensorflow/models/tree/master/research/syntaxnet>

1	Where	where	WRB	O	_	7	advmod	_	_
2	is	be	VBZ	O	_	7	aux	_	_
3	the	the	DT	O	_	4	det	_	_
4	manager	manager	NN	O	_	7	nsubj	_	_
5	of	of	IN	O	_	6	case	_	_
6	ITC	ITC	NNP	ORGANIZATION				4	nmod _ _
7	working	work	VBG	O	_	0	root	_	_
8	in	in	IN	O	_	10	case	_	_
9	the	the	DT	O	_	10	det	_	_
10	organization	organization	NN	O				7	nmod _ _
11	?	?	.	O	_	7	punct	_	_

Figure 3.3: NIF format for Q1

Due to some limitations found in Stanford’s performance when dealing with questions, we carry out some automatic pre-processes (normalizing some of the questions) and post-processes (modifying the output in some cases) for solving the greatest issues (see step 2 in Figure 3.1). In this phase, as shown in Figure 3.1, the input is the Stanford parsing information, which has been generated, in the first step. Preprocessing action on this provided information is carried out to build the NIF content as shown in Figure 3.3.

### 3.3 Question Representation

The main goal of this component is to map the results of previous analysis into classes, which will be accessible and presentable during the question interpretation. Obviously, ScoQAS should be able to access, control, and manage all of the extracted data of NIF format and preprocessing information. Therefore, we have implemented a module containing all of the required classes. Basically, the defined representation classes play main axes to maintain a sequence of tokens so that each one containing the word form, lemma, Part of Speech (POS), and if appropriate, Named Entity (NE) information. Moreover, they contain the semantic information such as related term in ontology item corresponding to the token in the system memory. For example, the result of this binding for the output details in Q1 is shown in Table 3.1 for this step.

The dependency tree information is bound to related class and this information is available to use in the next steps. The result of this dependency tree for Q1 (see Figure 3.2) is shown as follows:

Dependencies: [(6, 0, 'advmod'), (6, 1, 'aux'), (3, 2, 'det'), (6, 3, 'nsubj'), (5, 4, 'case'), (3, 5, nmod), (9, 7, 'case'), (-207, 6, 'root'), (9, 8, 'det'), (6, 9, 'nmod'), (6, 10, 'punct')]

Table 3.1: Applying Stanford CoreNLP Parser over question Q1

Question (Q1)	Where	is	the	manager	of	ITC	working	in	the	organization	?
Token	0	1	2	3	4	5	6	7	8	9	10
Lemma	where	be	the	manager	of	ITC	work	in	the	organization	?
POS	WRB	VBZ	DT	NN	IN	NNP	VBG	IN	DT	NN	.
NE	nil	nil	nil	nil	nil	ORG	nil	nil	nil	nil	nil

### 3.4 Rule-Based Question Classifier

One of the essential components in QA is Question Classification (QC), aiming to extracting question type and the expected answer type needed for the extraction of expected correct answers. The important stage in QC involve analyzing the question to a degree that allows the system to determine the “type” of the sought-after answer, i.e., the question type (QT) of the query. The QC task is approached mainly using manually built rules or Machine Learning techniques by means of sets of lexical, semantic or syntactic features. The research findings show that AI and machine learning approach is really promising particularly in terms of fluency and scalability [15] [91] [92] [93] [94]. Li and Roth [15] have defined the QC to be the task that, given a question, maps it to one of  $k$  classes, which provide a semantic constraint on the sought-after answer. The aim is that this classification, potentially with other constraints on the answer, will be used by a downstream process, which selects a correct answer among several candidates.

A QC module in a ScoQAS provides information that downstream processes use it in formulating the QSiS information in the path of determination of a certain range of answers that could be specific for an answer type, rather than uniform across all types. For example, given the question “Who was the first woman killed in the Vietnam War?” we do not want to test every noun phrase in a document to see whether it provides an answer. At the very least, we would like to know that the target of this question is a person; thereby it would be significantly reduced the number of possible answers.

In [15] a two-layered taxonomy was defined, which represents a natural semantic classification for typical answers in the TREC<sup>1</sup> task. There is a hierarchy categorization that contains 6 coarse classes (*Abbreviation*, *Entity*, *Description*, *Human*, *Location* and *Numeric Value*) and 50 finer ones. Each coarse class contains a non-overlapping set of finer classes. A machine learning approach was used for QC and a hierarchical classifier was developed by a layered semantic hierarchy of answers types, and used it to classify questions into fine-grained classes.

<sup>1</sup> <http://trec.nist.gov/>

One of the most important investigations related to the QC task was presented in thesis research by Håkan Sundblad [91] where focused on five different machine learning algorithms. The algorithms that have been analyzed include k nearest neighbors, naive bayes, decision tree learning, sparse network of winnows (SNoW), and support vector machines (SVM). Håkan uses two corpora where it has been achieved good results by SVM approach in biased corpus while the performance for applying naïve bayes, decision tree learning, and SVM had the same level of functionality in a novel corpus. Moreover, the thesis with focus on problematic questions tried to analyze the errors that had challenges in learning algorithms.

With respect to the recent research in [92] which shows that the additional features, high-frequency words and dependency relations, for question classification have been improved the performance in a SVM-based approach. It is also referred to the widely used UIUC hierarchical classification standard corpus and Li & Roth [15] [93] which has been presented a machine learning approach to categorize questions into semantic classes by developing a hierarchical model with performance to 85.0% in accuracy under fine categorization. Dell Zhang [94] proposed an SVM-based approach with various features, including bag-of-words and bag-of-ngrams to classify the given question. Their approach achieved 80.2% and 90.0% in accuracy under fine and coarse setting respectively.

However, there has been a considerable amount of research done with machine learning approaches operating better than rule-based to classify questions. The research supporting this thesis has not explicitly addressed the question classification problem. With regards to other related works in QC, in a way that is easily perceived there are machine learning approaches, which might operate better than rule-based to classify questions. The main point in this thesis is due to the lack of semantic features that we required to extract for Enterprise ontology domain questions. The main contribution of the thesis is not for focusing on classifying the questions synthetically but more than this in the semantically aspect to look in domain ontology. The concentration of this work in QC is to classify questions semantically and determine semantic type of the questions with manually annotations. Then, all features were extracted for domain questions. The main idea behind the QC is to adapt the coarse and fine classification together and find the nature of the question type and expected answer type (EAT) in order to figure out the grammatical and semantic structure of the question.

To the best of our knowledge, the existing methods in the machine learning approach (e.g. SVM) would non-compliance with domain specific tagsets as the provided Enterprise domain question classification, because the number of training examples (questions) should be very high to get reliable classification models. These ML approaches, however, cannot be used in our case because our problem is not only classifying into QTs but extract the mandatory components of our semantically rich tags. For instance, for a QT “Where\_Person\_Action” we need not only classifying the question into this question type but also extract the involved Action and Person. Therefore, our solution strives to classifying questions by defining the semantic-based structure-feature patterns (Ss-fP). The Ss-fP consists of features that are extracted by hand-crafted rules and then assigned to the associated QT. The



advantage of this method was simplicity in implementation and testing the performance of the provided features. While with having these features, they can be simply used in any machine learning approach or other AI techniques as well.

In order to construct a conceptual structure, we built a prototype implementing a basic version of the ScoQAS system. This system was iteratively tested, and then reworked as necessary until an acceptable prototype was finally achieved. In addition to the classification task, this module analyzes the question and extracts all the information, which will be necessary for the rest of the question interpretation process. We define QSiS which is the basic block of knowledge to obtain the answer or to transform its content to SPARQL query. The definition of QSiS set is:

< Q, QT, EAT, RT, CSTR >

Where:

Q identifies the current Question

QT indicates the Question Type (see the tag set in next section)

EAT shows Expected Answer Type of the question as a list of the most probable EATs

RT is a set of Related Terms extracted from the ontology or other knowledge base resources

CSTR is the list of Constraints over the RT and the dependencies between RTs

### 3.4.1 QT - Question Type

Srihari and Li [95] has shown that it is important to classify questions with respect to their answer types. Hermjakob [96] refers to such answer types as Qtargets. The Qtargets are based on the analysis of 18,000 on-line questions where the number of distinct Qtargets is 122 computed based on a list of 276 hand-written rules. Let us to see an example, “*When did Ferraro run for vice president?*”, the answer could be one of three or more distinct types of Qtargets, such as date, temp-loc-with-year; temp-loc for the temporary question. It has a complex Qtarget, giving first preference to a date or a temporal location with a year and second preference to a general temporal location.

Our tag set consists of 75 QTs shown in Table 3.2. For each QT, a set of rules has been built (usually only one or two rules). If one of the existing rules is matched, then the classifier assigns the Question Type (QT) pattern, e.g. <Where-Person-Action> in the first row (No. 1), as shown in Table 3.2. The “Example” column shows the sample question related to each row in one of the scenarios. ScoQAS uses semantic-based structure-feature pattern (Ss-fP) matching combined with a rule-based system. A list of rules is attached to each QT. Each rule consists of a set of

*Conditions* and set of *Actions*. When all of the Conditions for question are satisfied then Actions rules are executed. The rule conditions of the example can be paraphrased as following: If the token "Where" starts the sentence and there is either a token being a Named Entity (NE) of class PERSON or a token able to be mapped into a node in the ontology being a subclass of PERSON, and there is a token in the question being a verb or a verbal nominalization, then the QT "Where-Person-Action" is extracted. Our aim on using this rich tag set is disposing of a powerful tool able to be used in whatever scenario. This objective has been reached in our two scenarios.

In contrast with most tag set used in QA systems (including the popular ones, Li & Roth [15]), our proposal includes semantic aspects within the tag, that have to be instantiated for each question. In this way, the same tag set can be used for different scenarios. For instance, the QT "Where\_Person\_Action" can be applied to whatever query involving a person performing an action. The EAT can be a location or a place. Q1 in the first scenario is instantiated with a person (an instance of manager) performing the action work. In the second scenario the question "When was Mozart born?", the person is Mozart and the action to be born.

To exploit rule-based method capabilities and in order to increase the accuracy of classifying, we have designed and implemented a learning method to convert the questions into the form of semantic-based structure-feature patterns. ScoQAS uses pattern matching combined with rule-based systems. Obtained syntactic parsing features and semantic information from WordNet [97] are used as a source for synonyms, hyponyms and hypernyms to help increase of the accuracy in the classification step. Applying the rule-based with Ss-fP approach in classifying will be more effective.

Table 3.2: Set of QT with satisfied constraints in ScoQAS

NO	QT	Domain Scenario	EAT	Pattern Description	Example
1	Where_Person_Action	1 <sup>st</sup>	The prefix of QT denotes the EAT. Prefix= Where (Location)	The place(s) which/where a person is engaged in conduct or involved in an activity.	Where is the manager of ITC working in the organization?
2	Where_CompoundProperties	2 <sup>nd</sup>	Location	Place(s) that satisfy a set of related constraints	In which UK city are the headquarters of the MI6?

3	Where_CompoundProperties_Entity_Action	2 <sup>nd</sup>	Location	The same but having performed in action for entity	Which countries in the European Union adopted the Euro?
4	Where_CompoundProperties_GEO	2 <sup>nd</sup>	Location	Place(s) that satisfy a set of related constraints involving location	Which countries have places with more than two caves?
5	Where_CompoundProperties_Person	2 <sup>nd</sup>	Location	The same but involving person or people	Which German cities have more than 250000 inhabitants?
6	Where_Entity_Action	2 <sup>nd</sup>	Location	Place(s) of the type of question having performed by entity	Which U.S. states possess gold minerals?
7	Where_GEO	2 <sup>nd</sup>	Location	A place located in a specific location	In which country is the Limerick Lake?
8	Where_GEO_Action	2 <sup>nd</sup>	Location	Place(s) of the type of question involving an action on a specific location	Which river does the Brooklyn Bridge cross?
9	Where_GEO_Member	2 <sup>nd</sup>	Location	Finding place(s) of the type of question which is part of specific location.	List all Frisian islands that belong to the Netherlands.
10	Where_Properties_Action_TimeRelation	2 <sup>nd</sup>	Location	Place(s) that satisfy a set of independent constraints performing an action over time or at a certain time	Give me all world heritage sites designated within the past five years.
11	Where_Properties_Entity	2 <sup>nd</sup>	Location	Place(s) involving entity that satisfy a set of independent constraints	Which European countries have a constitutional monarchy?
12	Where_Properties	2 <sup>nd</sup>	Location	Place(s) that satisfy a set of independent constraints	Which state of the USA has the highest population density?
13	Where_Properties_Person_Action	2 <sup>nd</sup>	Location	Place(s) which a person that satisfy a set of independent constraints but having performed in action	In which city was the former Dutch queen Juliana buried?
14	Where_Synonym	2 <sup>nd</sup>	Location	Place(s) which has another equivalent	Which U.S. state has the abbreviation MN?

### 3 The Architecture of Semantic-based QA System (ScoQAS)

				name	
15	Who_Properties_Person	1 <sup>st</sup>	Prefix = Who EAT: Person	Person(s) that satisfy a set of independent constraints	Who is executive director?
16	Who_Compound Properties_Person_Action	1 <sup>st</sup>	Person	Person satisfying a set of related constraints involving other resources which are affected by an action	Who are controlled by the executive chief of the organization?
17	Who_Member_CompoundProperties	1 <sup>st</sup>	Person	Person; member of a group	Who are the members of committee Frontier of IT?
18	Who_Properties	2 <sup>nd</sup>	Person	Person satisfying a set of independent constraints	Give me all female Russian astronauts.
19	Who_Properties_GEO	2 <sup>nd</sup>	Person	The same but involving a location	Give me all professional skateboarders from Sweden.
20	Who_CompoundProperties	2 <sup>nd</sup>	Person	Person satisfying a set of related constraints involving other resources	Who is the husband of Amanda Palmer?
21	Who_CompoundProperties_GEO	2 <sup>nd</sup>	Person	The same but involving a location	Who is the mayor of Berlin?
22	Who_Action	2 <sup>nd</sup>	Person	Person(s) who has/have performed an action	Who developed Minecraft?
23	Who_Action_Entity	2 <sup>nd</sup>	Person	The same but involving an entity	Who produces Orangina?
24	Who_Synonym	2 <sup>nd</sup>	Person	Another name of a person	Who was called Scarface?
25	Who_Member	2 <sup>nd</sup>	Person	Person; member of a group	Give me all members of Prodigy.
26	Who_Action_Entity_Person	2 <sup>nd</sup>	Person	Person perform an action for another person	Who composed the music for Harold and Maude?
27	Who_Action_Entity_Properties	2 <sup>nd</sup>	Person	The same but involving a set of	Who wrote the book Les Piliers de la terre?

				independent constraints on entity	
28	Who_Action_GEO	2 <sup>nd</sup>	Person	Person(s) who (has/ have) performed an action involving a location	Which professional surfers were born on the Philippines?
29	Who_CompoundProperties_Action	2 <sup>nd</sup>	Person	Person(s) who perform an action that satisfy a set of related constraints	Who is the daughter of Ingrid Bergman married to?
30	Who_CompoundProperties_Action_GEO	2 <sup>nd</sup>	Person	The same but involving a location	Who painted The Storm on the Sea of Galilee?
31	Who_CompoundProperties_ORG	2 <sup>nd</sup>	Person	Person(s) who satisfy related constraints involving organization	Who is the owner of Universal Studios?
32	Who_CompoundProperties_Person	2 <sup>nd</sup>	Person	Person(s) who satisfy a related constraint involving person	Who was the successor of John F. Kennedy?
33	Who_Properties_Action_GEO	2 <sup>nd</sup>	Person	Person who perform an action involving a location	Which daughters of British earls died in the same place they were born in?
34	What_Action_Properties_Status	1 <sup>st</sup>	Prefix= What EAT: Asking for information	Asking for information specifying something (has/ have) performed an action satisfying a related status	What are the activities that are listed in running condition?
35	What_Properties_Entity	1 <sup>st</sup>	Asking for information	Asking for information which has specific features or traits	Which division or sub division has economic goals?
36	What_CompoundProperties_Entity	1 <sup>st</sup>	Asking for information	Asking for information specifying something that satisfy a related constraints for entity	What are the responsibilities of committee Frontier of IT?
37	What_Action_GEO	2 <sup>nd</sup>	Asking for information	Asking for information specifying something (has/ have) performed an action involving a	Which languages are spoken in Estonia?

### 3 The Architecture of Semantic-based QA System (ScoQAS)

				location	
38	What_CompoundProperties_Action	2 <sup>nd</sup>	Asking for information	The same but performing an action	Which other weapons did the designer of the Uzi develop?
39	What_CompoundProperties_GEO	2 <sup>nd</sup>	Asking for information	The same but involving a location	Which telecommunications organizations are located in Belgium?
40	What_GEO	2 <sup>nd</sup>	Asking for information	Asking for information involving a location	What is the area code of Berlin?
41	What_Person_Action	2 <sup>nd</sup>	Asking for information	Asking type of question that person who (has/ have/is) perform(ed/ing) an action	Which instruments did John Lennon play?
42	What_Person_Action_TimeRelation	2 <sup>nd</sup>	Asking for information	The same but limited to a specific period of time	Show me all songs from Bruce Springsteen released between 1980 and 1990.
43	What_Properties	2 <sup>nd</sup>	Asking for information	Asking for information satisfying a set of independent constraints	Give me all Canadian Grunge record labels.
44	What_Properties_Action	2 <sup>nd</sup>	Asking for information	Asking type of question that who has performed in action satisfying a set of independent constraints	Which music albums contain the song Last Christmas?
45	What_Properties_Person_Action	2 <sup>nd</sup>	Asking for information	The same but person involving in action	Which books by Kerouac were published by Viking Press?
46	What_SubType	2 <sup>nd</sup>	Asking for information	Asking type of question such that a super type entity or person is shared similar characteristics with other entity type that contains attributes which are commons to its subtype.	Give me all animals that are extinct.

47	What_Synonym	2 <sup>nd</sup>	Asking for information	Asking for other equivalent or alias name(s)	What are the nicknames of San Francisco?
48	Howmuch_Properties_Person	1 <sup>st</sup>	Prefix= Howmuch EAT: Amount or Price	Counting type of question satisfying related constraints involving a person	How much is the insurance premium deductions for Ali?
49	Howmuch_CompoundProperties	1 <sup>st</sup>	Prefix= Howmuch EAT: Amount or price	Counting type of question (value) satisfying a set of related constraints	How much is the value of laptop?
50	Howmany_Action_GEO_Properties	2 <sup>nd</sup>	Prefix= Howmany EAT: Count or Number	Counting type of question is performed by an action involving a location	How many official languages are spoken on the Seychelles?
51	Howmany_CompoundProperties_Action_Entity	2 <sup>nd</sup>	Count or number	Counting entity that perform an action which satisfy a related constraints	What is the total amount of men and women serving in the FDNY?
52	Howmany_CompoundProperties_GEO	2 <sup>nd</sup>	Count or Number	Counting type of question (students) satisfying related constraints involving a location	How many students does the Free University in Amsterdam have?
53	Howmany_Entity	2 <sup>nd</sup>	Count or Number	Counting type of question (employees) involving entity	How many employees does Google have?
54	Howmany_GEO	2 <sup>nd</sup>	Count or Number	The same but involving a location	How many inhabitants does Maribor have?
55	Howmany_Person	2 <sup>nd</sup>	Count or Number	The same but involving a person	How many children did Benjamin Franklin have?
56	Howmany_Person_Action	2 <sup>nd</sup>	Count or Number	Counting type of question that a person performing an action	How many films did Hal Roach produce?
57	Howmany_Properties	2 <sup>nd</sup>	Count or Number	Counting type of question that satisfy a related constraints	How many space missions have there been?

### 3 The Architecture of Semantic-based QA System (ScoQAS)

58	Howmany_Properties_GEO	2 <sup>nd</sup>	Count or Number	The same but involving a location	How many monarchical countries are there in Europe?
59	When_Person_Properties	1 <sup>st</sup>	Count or Number	Asking the time for specific type of question which involving a person	When is the start date for employee Mehdi?
60	When_Action_CompoundProperties	2 <sup>nd</sup>	Prefix= When EAT: Time or Date	Asking the time performing an action that satisfy a set of related constraints	When was the Statue of Liberty built?
61	When_Action_CompoundProperties_Entity	2 <sup>nd</sup>	Time or Date	The same but involving an entity	What is the founding year of the brewery that produces Pilsner Urquell?
62	When_Action_Entity	2 <sup>nd</sup>	Time or Date	Asking the time which is performed an action involving an entity	When was Capcom founded?
63	When_CompoundProperties	2 <sup>nd</sup>	Time or Date	Asking the time that satisfy a set of related constraints	When was the Battle of Gettysburg?
64	When_GEO_Action	2 <sup>nd</sup>	Time or Date	Asking the time which performs an action involving a location	When did Latvia join the EU ?
65	When_Person_Action	2 <sup>nd</sup>	Time or Date	The same but involving a person	How often was Michael Jordan divorced?
66	Quantifier_GEO	2 <sup>nd</sup>	Prefix= Quantifier EAT: Quantity	Determining countable and uncountable values of location	How high is the Mount Everest?
67	Quantifier_Person	2 <sup>nd</sup>	Quantity	The same but involving a person	How tall is Claudia Schiffer?
68	Quantifier_Person_Action	2 <sup>nd</sup>	Quantity	The same that a person perform an action	How often did Nicole Kidman marry?
69	YNo_CompoundProperties_Person_Action	2 <sup>nd</sup>	Prefix= YNo EAT: Yes or No	Expected answer is Yes or NO where person who having performed in action that satisfy a set of	Did Tesla win a nobel prize in physics?



				related constraints	
70	YNo_ CompoundProperties _Synonym	2 <sup>nd</sup>	Yes or No	Expected answer is Yes or NO for responding the similarity between a set of related constraints and entity (Person)	Is the wife of president Obama called Michelle?
71	YNo_Equal	2 <sup>nd</sup>	Yes or No	Expected answer is Yes or NO for responding the equality of two entities or combined names	Is Egypt's largest city also its capital?
72	YNo_Person_Action	2 <sup>nd</sup>	Yes or No	Expected answer is Yes or NO for person(s) who having performed in action	Did Socrates influence Aristotle?
73	YNo_Person_Action _GEO	2 <sup>nd</sup>	Yes or No	The same but involving a location	Was Natalie Portman born in the United States?
74	YNo_Person_Status	2 <sup>nd</sup>	Yes or No	Expected answer is Yes or NO to get the status of persons	Is Frank Herbert still alive?
75	YNo_SubType	2 <sup>nd</sup>	Yes or No	Expected answer is Yes or NO where a super type entity (e.g. chemist or person) implies that similar characteristics with other entity type contain attributes, which are commons to its subtype.	Was Margaret Thatcher a chemist?

Creating a QT Rule for a question class QT is described in the Algorithm 3.1.

---

**Algorithm 3.1** Creating a QT Rule

---

**procedure** RULE\_INITIALIZATION(Rule\_id)

1. Assign a new instance of the general Rule class with a new identification of question type and determining tag set, e.g. QT\_TagSet= "Where\_Person\_Action", so the databaseRules is defined as dictionary of rules as follows:

```
databaseRules['Rule_id']=QTclassrule('Rule_id', 'QT_TagSet')
```

2. Define all of the Conditions that should be satisfied for each QT\_TagSet items. There are specific conditions for each associated 'Rule\_id', so:

**for** *i* **in** TagSet items **do**

```
CProc_i=ConditionProcedure_i(Q, Rule_id)
```

```
Cond[i]=QTclassCondition(CProc_i)
```

```
databaseRules['Rule_id'].addConditions(Cond[i])
```

**end for**

3. Define all of the Actions that are needed for initializing (binding) the value(s) for corresponding Conditions which have been satisfied for associated 'Rule\_id', i.e. the task will find the token or set of tokens of the question involved for tag set item such as Person or Action in QT\_TagSet, then bind the token(s) index to the defined variable, so:

**for** *i* **in** TagSet items **do**

```
AProc_i=Action_binding_Procedure_i(Q, Rule_id)
```

```
Action[i]=QTclassAction(AProc_i)
```

```
databaseRules['Rule_id'].addActions(Action[i])
```

**end for**

**end procedure**

---

For instance, some kinds of conditions for "Where\_Action\_Person" are isWhere(), isPerson() and isAction() and other kind of actions are bindWhere(), bindPerson() and bindAction(). The result of running conditions and actions for classifying question Q1 on QT ('Where\_Person\_Action') could be:

```
{'tk_ACT': 6, 'tk_PER': 3, 'ont_PER': 'Manager', 'ont_ACT': None}
```

Here 'tk\_ACT' has been bound with token number 6 ("working") for Action directly because working is a verb, and 'tk\_PER' with token number 3 ("manager")

for Person indirectly through ontology because manager is a subclass of PERSON class. The 'ont\_PER' value is “Manager” that shows the class name in the Enterprise ontology after traversing this ontology, and finally 'ont\_ACT' has no value.

To classify a question as “Where\_Person\_Action”, we are looking for a location (EAT) where a specified person performs a specified action i.e. the rule tries to locate an interrogative adverb (“where”), an action (a verb or a verbal nominalization) and a person (a Named Entity of type PERSON). In this case, the rule fails because of the third condition. There is, however, another rule that satisfies the person condition if a token in the question can be mapped into a subclass of the class “i\_en\_proper\_person” in the ontology. This rule gets fired because ‘manager’ occurs in the ontology as descendant of ‘i\_en\_proper\_person’ and produces the two predicates above. Therefore, after the execution of the rule, two variables have been set,  $X_1$  and  $X_2$ , which are constrained by the two predicates (tk\_PER and tk\_ACT). The rest of this chapter explains how the ScoQAS starts to make constraints with these preliminary values.

### 3.4.2 EAT- Expected Answer Type

Besides its obvious main objective of classifying the question into the set of predefined classes (QT), as shown in Table 3.3, other useful information for answering the question has to be extracted from the question. A QC module has to determine EAT. In this stage, we provided techniques for learning and classifying questions using rule-based methods, Semantic Web technologies like ontology traversing and a rich set of lexical, syntactic part-of-speech tags and syntactic chunks and word similarity measures in order to find the appropriate EAT (see Table 3.3)

Table 3.3: Sample of Expected Answer Type

Question	QT	EAT
Where is the manager of ITC working in the organization?	Where_Person_Action	Place/Location

### 3.4.3 RT- Related Terms (Keywords)

Given the QT, a set of related terms (RT) is created for each of its feature (item). The stop words are not considered as RT. For instance, the RT {organization, manager, work} is assigned to Where, Person and Action respectively in Q1 such that it has been satisfied based on our defined QT rules. This question is ambiguous because the user has under-specified in what kind of place the manager should work. She/he either could work in a physical location or could have been sent by some other entity like the organization where he/she works in part or division of organization. The next section will address to find more items looking in the ontology in order to add new items to set of RT as an evolutionary process when the constraints are built.

### 3.5 Modelling Constraints

In order to be able to identify more specific and appropriate answers, we need to have some level of understanding of the content of question. This does not mean, however, that we need to develop specific techniques for processing every possible type of question and answer. Instead, we need to develop more general approaches to identify as many constraints as possible on the answers for questions. The constraints are simply relations that can hold between the keywords in the target space. The constraints units include predicates and variables acting the former as restrictors over the values of variables. The constraints hold syntactic or semantic relations between the question keywords and its produced QT's.

The constraints can be classified as *mandatory* and *optional*. The QT defines the set of mandatory constraints (MC) that have to be satisfied by the answer, and optional constraints (OC) which simply increases the answer credibility score when satisfied for a question. For instance, for Q1, where the QT is 'Where\_Person\_Action', both the 'Person' and the 'Action' should be constrained (i.e. a Person and an Action should be located, moreover as (6, 3, nsubj) has been placed in MC the located person and action should be constrained to have an "nsubj" relation between them). Generation of MC is placed within the action part of the rules (if the conditions are satisfied then MC is generated). MC depends basically on the QT and is derived from the mentions associated with the variables of the QT and their dependency relations. In the example Q1, both 'tk\_PER' and 'tk\_ACT' are mandatory and should contain the indices of the involved tokens, which can be a single token, as in this case, or a list of tokens. In this instance, two variables  $X_2$  and  $X_3$  are introduced and the corresponding mentions are placed into MC: tk\_PER(3,  $X_1$ ) and tk\_ACTION(6,  $X_2$ ). 'ont\_PER' and 'ont\_ACT' are only filled if the rule accesses to the Enterprise ontology. Here 'manager' was found in the ontology but 'working' was simply located in the sentence as a verb, so the 'ont\_ACT' is set to 'None'.

Constructing such constraints provides more information about the nature of questions and helps to analyze the question semantically and, finally, it leads to obtaining a precise answer. The basic units of modeling of constraint-based framework in ScoQAS are constraints and variables; a constraint is an entity that restricts the values of variables. In order to clarify, consider a sample rule dealing with QT: "Where\_Person\_Action" (i.e. we are looking for a specific place, the EAT, where a PERSON performs some ACTION). For Q1 then, the module should return a dictionary with the following content:

```
{  
  'tk_PER': 3  
  'tk_ACT': 6  
  'tk_Type': 0  
  'ont_PER': Manager  
  'ont_ACT': None  
}
```

Where tk\_PER is the token or list of tokens referring to the PERSON, tk\_ACT is the token for ACTION, and tk\_Type is the token indicating the type of question that refers to the specific location. The ont\_PER and ont\_ACT are a list of ontology entities (Classes, Instances, and Relations) associated with Person and Action respectively. Templates that are used to make constraints are shown in Eq. 3-1-A and Eq. 3-1-B where argument can be thought of as the value that is assigned to associated variable.

Eq. 3-1-A: Predicate (Argument, Variable)

Eq. 3-1-B: Predicate (Variable1, Variable2)

This preliminary information derived from QC is initialized to constraints as Eq. 3-1-A:

tk\_Type ('0', X<sub>1</sub>) : where  
tk\_ACT ('6', X<sub>2</sub>) : work  
tk\_PER ('3', X<sub>3</sub>) : manager

As shown in Eq. 3-1-A and Eq. 3-1-B, there are two types of constraints that can be applied to every QT. Each of these formulated constraints is used based on the QT in appropriate circumstances. The constraints in type of Eq. 3-1-A is used when a kind of single argument constraints is required to be made (e.g. token as an argument). Hence, a new predicate is defined with its new variable (e.g. X<sub>1</sub>) with an assumed argument (e.g. token "0"). The argument can be assigned as a value or constant to which the corresponding variable has been considered. To dedicate the new variable, we follow our method such that one number is added to the most recent index of variable. During the execution of the process, an incremental counter is assigned as a new index of the variable. For example, variable X<sub>2</sub> after defining variable X<sub>1</sub> in constraints tk\_ACT ('6', X<sub>2</sub>). The note is that the last index of variable at each step will also be used to extend the semantic relationships found in the ontology in related procedures. To this end, a structured process has been defined to maintain consistency, non-limitation, and without interference. For all of the QT tags, the new variable is initiated by the associated token number that have been achieved in the question classification step, so other preliminary variables are bound to the corresponding constraints at the same way. For the Eq. 3-1-B, is used another procedure for supposed predicate with involved two related arguments where each of these arguments has been already bound to the associated variables before, so in this case, no new variable is defined and the existing variables are bound to the new constraints. For example, in the dependence (6, 0, advmod) for both of the tokens 6 and 0 have been already assigned the variables X<sub>2</sub> and X<sub>1</sub>, respectively. Therefore, the constraints would be advmod (X<sub>1</sub>, X<sub>2</sub>).

Besides the information provided by question classifier module S: < Q, QT, EAT, RT, CSTR >, the additional mandatory semantic constraints should be satisfied at the

answer extraction step. In order to extend the constraint with the information explicitly stated in the question, we look, into the dependency tree for all of the variables detected so far and their dependencies in any direction. There are three dependencies in tk\_PER ('3', X<sub>3</sub>) (see the dependency tree in Figure 3.2):

1. (3, 5, nmod)
2. (3, 2, det)
3. (6, 3, nsubj)

For the dependence (3, 5, nmod), the new constraint “nmod” is defined. Given the token 5 was not allocated to any variable yet, so the new variable X<sub>4</sub> is defined and then new constraints tk('5', X<sub>4</sub>) is created. For the dependence (3, 2, det), the new constraint “det” is defined. Obviously, the variable has not yet been defined for token 2 while the token 3 has been bound for X<sub>3</sub>. In this case, exceptionally, the new variable is not required. The new constraint is det(X<sub>3</sub>). This process continues to define all of the constraints for other preliminary variables (X<sub>1</sub>, X<sub>2</sub>, and X<sub>3</sub>) in order to find all of the syntax relation(s) provided by dependency tree. Thus, other new variables are initialized if they were not bound until now (see Appendix C for all generated constraints). Some of the constraints are listed as follows:

tk('5', X<sub>4</sub>)  
advmod(X<sub>1</sub>, X<sub>2</sub>)  
tk('1', X<sub>5</sub>)  
aux(X<sub>5</sub>, X<sub>2</sub>)  
nsubj(X<sub>3</sub>, X<sub>2</sub>)  
tk(9, X<sub>6</sub>)  
nmod(X<sub>6</sub>, X<sub>2</sub>)  
det(X<sub>3</sub>)  
nmod(X<sub>4</sub>, X<sub>3</sub>)  
tk('4', X<sub>7</sub>)  
case(X<sub>7</sub>, X<sub>4</sub>)  
tk('7', X<sub>8</sub>)  
case(X<sub>8</sub>, X<sub>6</sub>)  
det(X<sub>6</sub>)  
...

**Algorithm 3.2** Generation of Constraints

**procedure** OBTAIN\_CONSTRAINTS(QTrule, Q)

- 1- Let  $Q$  be an input text question such that  $L=length(Q)$  is the number of tokens.
- 2- Let  $QT$  has a set of tags as  $TS_{itm}$  where  $itm=1..k$  indicates the index of tag obtained from question classification step. The QT is representative of the type of question. For example, QT: Where\_Person\_Action can be shown as QT:  $TS_1-TS_2-TS_3$ .
- 3- Let  $tk_t$  be a set of the token index in the  $Q$  such that  $t$  indicate the location of token  $tk$  and  $t = \{0, 1, \dots, L-1\}$ .
- 4- Let  $P$  be a set of predicates,  $X$  be a set of a variables, and  $Y$  be a set of arguments. Then we define the constraints function  $C_{ij}$  for  $Q$  as follows:

$$C_{ij} = \begin{cases} P(X_i, Y_j) & \text{if } i = 1..m \text{ and } j = 1..n \\ P(X_i) & \text{if } i = 1..m \text{ and } j = 0 \end{cases}$$

Notably  $\forall y \in Y_j$  and  $j \neq 0$  is bound to the  $X_i$

- 5- Assume  $MC$  and  $OC$  are a set of mandatory and optional constraints respectively, which are generally in the form  $C_{ij}$
- 6- The initial value of  $MC$  will be:
 
$$(MC)_{qr} = P(X_q, Y_r), Y_r \in t \text{ and } q=itm$$
- 7- After initialization of the MC, the other constraints are obtained by analyzing the dependency parsing results using NIF. Note that the  $P$  for these constraints is assigned the same values from standard dependency relations lables between two tokens (e.g. prep\_of, nsubj, etc.).
- 8- The dependency among the variables in the constraints of the MC (direct or indirect) is extracted by addDepList() function. From this step, a new index  $s=\max(q)$  is defined. The process of binding the new arguments to the variables corresponding to the MC are carried out as follows:

**while** each token in  $Y_r$  include new dependencies

- The argument " $a$ "  $\in Y_r$  assigned to  $X_q$  in  $(MC)_{qr}$ , there is a direct dependency between " $a$ " and a set of pairs  $Dep\_a = \{(a, b_1, dep_1), (a, b_2, dep_2), (a, b_3, dep_3), \dots, (a, b_w, dep_w)\}$ .
- The dep4tk() function analyze each pair of the  $Dep\_a$  set and checks whether or not a variable associated with token  $b_k$ ;  $k=1..w$  exists in MC. i.e., it would not be defined a new variable where the associated token with its variable has already been assigned.
- If there is no corresponding variable for token  $b_k$ ;  $k=1..w$  in the  $(MC)_{pq}$  then the new variable with new index  $s=s+1$  is added and linked to token  $b_k$ .
- The new MC is added according to the generalized function as:

$$(MC)_{st} = P(X_s, Y_t) = Dep\_a$$

where  $Y_t = b_k$ ;  $k = 1..w$

**end while**

**end procedure**

Algorithm 3.2 shows how the constraints module proceeds to build a generalized approach to deal with formulation of constraints for any type of questions. Let us consider again the example Q1 (in the first scenario): “Where is the manager of ITC working in the organization?”. Our vision of building the constraints is to develop an algorithm in order to generalize the approach that operates like particular case (e.g. Q1). A process template of building constraints for QT, “Where\_Person\_Action” corresponding to the Q1, is as follows:

1. Add a variable ‘ $X_1$ ’ for representing the type of question
2. Add a constraint  $tk\_Type('0', X_1)$  linking the variable with the token ‘0’
3. Add a variable ‘ $X_2$ ’ for representing the Action
4. Add a constraint  $tk\_ACT('6', X_2)$  binding the variable with the token ‘6’
5. Add a variable ‘ $X_3$ ’ for representing the Person
6. Add a constraint  $tk\_PER('3', X_3)$  linking the variable  $X_3$  with the token ‘3’

We look, then, to the dependencies involving the variables  $X_2$  and  $X_3$  (direct or not). For  $X_3$ , there are three dependencies: (3, 2, det), (3, 5, nmod), and (6, 3, nsubj). The first one simply states that  $X_3$  is definite and singular, so a new constraint is added. Later, a new constraint “nsubj” should be added and linked to token 9 (organization).

7. Add a new constraint  $det(X_3)$
8. Add a constraint  $nsubj(X_3, X_2)$
9. Add a variable ‘ $X_4$ ’ for representing the token ‘5’
10. Add a constraint  $nmod(X_4, X_3)$

For  $X_2$  (working, 6) there are four dependencies (see Figure 3.2):

- a. (6, 0, advmod)
- b. (6, 1, aux)
- c. (6, 3, nsubj)
- d. (-207, 6, root)
- e. (6, 9, nmod)
- f. (6, 10, punct)

Some of the dependence labels are considered mandatory such as nsubj, nmod. For non-mandatory as *a* and *b* we look to the tokens and it would be added a variable  $X_5$  representing token 1. Only *c* and *e* are useful as mandatory constraints. The former represents  $X_3$ , the latter introduces a variable  $X_6$  representing the token 6 (“organization”) and the corresponding constraints are as follows:

11. Add a variable ‘ $X_5$ ’ for representing token ‘1’
12. Add a constraint  $tk('1', X_5)$  linking the variable with the token ‘1’



13. Add a variable 'X<sub>6</sub>' for representing organization
14. Add a constraint tk('9', X<sub>6</sub>) linking the variable with the token '9'
15. Add a constraint nmod(X<sub>6</sub>, X<sub>2</sub>)

Given that the constraint “nsubj” with variable X<sub>3</sub> and X<sub>2</sub> has already been defined in the step 8, so, in this case, the new constraint for this dependency is not repeated and not defined again. The variable X<sub>4</sub> includes the following dependencies: (5, 4, case) and (3, 5, nmod). Thus, the following constraints are added:

16. Add a variable 'X<sub>7</sub>' for token 4
17. Add a constraint tk(4, X<sub>7</sub>) linking the variable with the token '4'
18. Add a constraint case(X<sub>7</sub>, X<sub>4</sub>)
19. Add a constraint nmod (X<sub>4</sub>, X<sub>3</sub>)

The variable X<sub>6</sub> includes the following dependencies:

(9, 8, det), (6, 9, nmod), and (9, 7, case)

Thus, the following constraints are added:

20. Add a constraint det(X<sub>6</sub>)
21. Add a constraint nmod (X<sub>6</sub>, X<sub>2</sub>)
22. Add a variable 'X<sub>8</sub>' for token 7
23. Add a constraint tk(7, X<sub>8</sub>) linking the variable with the token '7'
24. Add a constraint case(X<sub>8</sub>, X<sub>6</sub>)

In the same way, the remaining variables are set. The generalized algorithm of the constraints creation process is presented in the next section specifically for the first scenario.

### **3.6 Semantic-based QA in a Closed Domain: 1<sup>st</sup> Scenario**

In this section, we discuss our approach using an ontology-based closed domain in order to deal with some of the technical challenges in the semantic-based closed domain QA. The Enterprise ontology is used as a case study to work with the factoid questions in this domain. In Section 3.7, we present our empirical approach using Linked Data in open domain. The basic idea behind the closed domain is to devise a semantics-aware inference approach, which distinguishes its method from open domain to deal with answer retrieval problem.

In this regard, an existing Enterprise ontology was restructured aligned with the previous presented work [98] as a case study. The improved Enterprise ontology is used as knowledge base to exploit it in the ScoQAS steps. It was applied to ScoQAS project where the ontology entity includes 190 classes, 230 slots and 500 instances designed in Protégé software. In addition to this, the organization's needs and

challenges in this area were analyzed in accordance with the state of the art factoid questions [52][93]. Our initial model to implement semantic-based QA and automatic information inferences for the enterprise's operational knowledge management was presented in 2013 [36]. For an ontology-based experiment (first scenario) we extracted 40 questions (See Appendix A) such that there exists answer for all of the provided questions. In Table 3.4 we present five sample questions from those questions provided for closed domain scenario. In these samples there are different type of questions such as Yes/No (e.g. question No. 1), List (e.g. question No. 2), Where (e.g. question No. 3), Who (e.g. question No. 4), and What (e.g. question No. 5) questions.

The basic idea behind the closed domain (first scenario) is to devise an inference mechanism performing over a question graph (QGraph) which is built from the QSiS enriched with ontology information. The QGraph contains nodes representing the entities referred in MC, edges corresponding to predicate elements in the ontology (classes, instances, and relations) as well as the connected components in the ontology.

Table 3.4: Sample questions with their correct answers for closed domain scenario

No	Question	Answer
1.	Is there any manager who is controlled by the executive director?	Yes
2.	Give me a list of departments in the organization?	<ul style="list-style-type: none"> <li>- Engineering Dep.</li> <li>- Planning Dep.</li> <li>- Administrative Dep.</li> <li>- Cultured Dep.</li> <li>- Educational Dep.</li> <li>- Research and Development Dep.</li> <li>- Telecommunication Dep.</li> <li>- Marketing Dep.</li> </ul>
3.	In which city of Iran is the University of Bu-Ali Sina?	Hamedan
4.	Who are controlled by executive chief of organization?	<ul style="list-style-type: none"> <li>- Head of Development</li> <li>- Educational evaluation expert</li> <li>- Development of private schools and public participation expert</li> </ul>
5.	What are the responsibilities of IT manager?	<ul style="list-style-type: none"> <li>- Review the adequacy and allocation of IT resources in terms of funding, personnel, equipment, and service</li> <li>- Approve and monitor major projects, IT budgets, priorities, standards, procedures, and IT performance</li> </ul>

There are several challenges, which should be addressed:

- Building the QSiS for each QT.
- Exploitation of the graph representation to model the relationships mentions in the question and the corresponding nodes in the ontology.
- Building an inference engine to extract answer(s) from the graph produced during the question processing.

The ways of facing these issues are described in the following sections.

### 3.6.1 Mapping Matched Ontology Items

However, after carrying out the process on query space for getting QT, EAT, RT (Keywords) and CSTR, the process is moved to the ontology space to first map the variables to the ontology entities (classes, slots, and instances) and then expanded upwards to the classes and going from classes to instances. Thus, the tasks mentioned in Algorithm 3.3 should be carried out during this stage of the first scenario. The mechanism of operations is described in continue in order to clarify the steps of this general tasks in Algorithm 3.3.

In step 1 of the Algorithm 3.3, we have to map RTs in the question to elements in the ontology. For this purpose, the action is carried out according to the NIF content so that looking in the ontology to find the corresponding item (e.g. classes, properties, and instances) in ontology for each of the tokens (not stop words) in NIF. This task is enriched using WN synonyms to cover more semantic similarities. We use a thresholding mechanism over the application of Levenshtein Distance (LsD) [99] for string similarity in order to match question tokens with ontology items such as Class, Slot, and Instance. The threshold-mean of LsD has been determined as the following conditions:

Condition (1): *if* LsD (Class) > 0.75 *then* select Class

Condition (2): *if* LsD (Slot) > 0.45 *then* select Slot

Condition (3): *if* LsD (Instance) > 0.30 *then* select Instance

Condition (4): *if*  $0.1 < \text{LsD (Instance)} < 0.3$  *then* select Class, not Instance or Slot

We have obtained these thresholds by applying LsD while traversing the Enterprise ontology for all of the training questions. Table 3.5 shows how the term “manager” can be mapped into elements of the ontology. We proceed to determine the threshold-mean for the lemma “manager” as an example in Q1. This lemma can approximately match a class, a slot, and instances in the ontology overcoming the pending thresholds. If the thresholds are met, the dictionary of the ontology items (bound classes, bound slots, and bound instances) is generated. The content of this dictionary consists of the names of the ontology items and the specific list that indicates the token’s number with special header that is dedicated to the indices.

---

**Algorithm 3.3** General tasks to map ontology items and building structured dictionary

---

**procedure** OBTAIN\_ONTOLOGYITEM\_FOR\_RT(Q)

1. Initialize the ontology-purpose<sup>1</sup> variables by executing the four functions such as *allclasses4Sentence()*, *allslots4Sentence()*, *slots4Classes()*, and *allinstances4Sentence()*. The task of these functions is to traverse the RDF/RDFS ontology and winnowing the content by approximate matching in a way such that matched scores are calculated between keywords and ontology items. The provided ontology-purpose variables from these functions are as follows:  
boundedClass={}, boundedSlot={}, boundedSlot4Class={}, boundedInstance={ }
2. The set of ontology-tagset<sup>2</sup> variables as a type of dictionary is defined in form of fixed prefix joint with the corresponding QT tag set. This is a sort of grouping of the ontology information obtained in the first stage, so that they can be controlled and further expanded in later stages. The samples are shown in the following format:

“boundedClass” + tagset={ }

“boundedSlot” + tagset ={ }

“boundedInstance” + tagset ={ }

3. Given the ontology-tagset variable and its index in dictionary, a specific process should analyze the derived index in order to expand or not the class and the slot. In addition, the value-type of slot is checked, it will be extended if the value-type is Instance-type. i.e., allowed classes for slots of type Instance are often called *range* of a slot, which can be followed, based on its *domain*. This mechanism should provide a possibility such that the assigned ontology items can be exploited in downstream steps. Therefore, the following variable can be added to deal with this type of slot accordingly:

“boundedSlotType” + tagset ={ }

**end procedure**

---

---

<sup>1</sup> The type of these variables is dictionary in Python so that keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

<sup>2</sup> The ontology-tagset is a dictionary variable that each tagset has set of variables corresponding to the class, slot, and instance of ontology such that simply identifiable and trackable.

In this regard, the ontology item vectors created through this way as follows:

- It is not dedicated a header with letter character such as “I” or “S” in the case that the condition (1) is satisfied for being matched class item and token. The distinction parameter would be only the index that refers to a location of tokens in the question.
- The header with label “I” is used for the instances that has been satisfied the conditions (3) or (4) for associated tokens.
- The header with label “S” is used for the slots that has been satisfied the condition (2) for associated tokens
- The classes of these assigned items are listed with the combination of one of these labels, “I” or “S”, and its token number. For example: (‘I5’, 0) has been assigned to the class “i\_en\_proper\_company” shows that this class belongs to an instance corresponding to token in location 5 with a distinction index 0 in Q1 and (‘S6’, 0) is related to the class “Employee” which indicates a matched slot associated with token 6.

Table 3.5: Thresholds-mean for lemma "manager" and favorable ontology items

Lemma = “manager”						
Matched Class(es) with Condition (1)		Matched Slot(s) with Condition (2)		Matched Instance(es) with Condition (3)		Matched Instances with Condition (4)
Manager	LsD =1	Manager: manager_title	LsD = 0.54	Manager: manager_title: Financial manager	LsD =0.41	
				Manager: manager_title: Organization manager	LsD =0.35	
				Manager: manager_title: Library manager	LsD =0.46	
				Manager: manager_title: ITC manager	LsD =0.63	
				Manager: manager_title: Product manager	LsD =0.46	
				Manager: manager_title: Sales manager	LsD =0.54	

For doing the step 2 in this algorithm, we adopted a method that can simply be developed for a variety type of questions, i.e. each QT with a tag set should be adapted with just a few improvements. In accordance with the rule and its associated QT, the ScoQAS system can automatically select those variables that its QT's tag set are matched with second part of dictionary variables, which have been already defined. This approach is applied for assigning the EAT values as well. For example, in QT: "Where\_Person\_Action" the defined variables are:

boundedClassPer={}; it indicates for all classes that matched for Person tag

boundedSlotAct={}; it indicates for all slots that matched for Action tag

boundedInstanceWhere={}; it refers to all instances that matched for Where tag that are considered as EAT

The task of step 3 is to provide a formal approach to expand the ontology items (e.g. class, slot, and instance) that are bound to each ontology-tagset variables and then simply to access for handling them in downstream process. In this regard, there is needed a procedure to analyze the ontology-tagset variables in terms of expandability or no need for expansion. This procedure considers the name of the variable and the value of its formulated index to find out the matching condition as mentioned in step 1. Due to the defined paradigm in indexing the ontology-tagset variables in step 1 and 2, the process should apply a decryption solution to follow how and which of these dictionary variables should be extended. Thus, there are specific procedures in order to expand them. For example, consider that the Condition (1) and Condition (3) to assign a class have been determined in different way to distinguish them because the class matched in Condition (1) should be expanded for all of the subclasses and includes all of the instances for whole associated slots. In front of this, when the Condition (3) has been satisfied then the character "I" was attached to header of the indexing value of ontology-tagset variable for bounded class. Thus, it means that this class would not be expanded more and it is related just to the matched instances.

In the closed-domain scenario, the EAT should be expanded with ontology elements where the answer(s) are sought, so the EAT variables are expanded based on tk\_Type value in a similar way that mentioned for other tag set. Determining the values of these variables play an important role in limitation of the scope of expected answers. This task is an essential step for completing the constraints process.

### 3.6.2 Expansion of Generating the Constraints

In Section 3.5 we described how to initialize variables for formulating the constraints and the generalized algorithm was introduced to deal with both scenarios. We now continue to describe the process of establishing other constraints coming from the ontology in the closed domain scenario. In the target space, the ontology in this case, the nodes mapped to the tokens in the query corresponding to the Person

(the manager of ITC) and the Action (working) should be constrained by the relation holding between them (subject in this case in the (6, 3, nsubj)).

In Section 3.6.1, the basic assignment for initializing the structured variables was presented to utilize them to extend more restriction information in the closed domain scenario. In this development, more entities (variables) and relations can be extracted from the ontology and placed into MC (see Appendix C).

During the question classification step, the primary MC set was determined and the other related information was formulated in the previous section. In Algorithm 3.2, the process of building the MC constraints was presented. In order to mature this process and get full results from this module, the constraints should be extended to complete the syntactic-semantic information structure of the question in generalized way. The MC constraints are extended for each of the Classes, Slots, and Instances, which have already been extracted based on string similarities algorithm. In Section 3.6.1 we described how these ontology items are mapped into ontology-tag set variables. So Algorithm 3.4 shows the process of extending and formulating of MC and expresses how this process comes up with the issues related to expanding the MC in closed domain scenario.

As seen in Algorithm 3.4, the process of extending the constraints in first scenario consists of three main steps. In the step 1, we describe how this algorithm has been adapted from Algorithm 3.2 in order to obtain ontology constraints. In step 2, we figure out the generalized procedure to demonstrate the way of adding new constraints to MC. Given the importance of determining the expected answer at this stage of the ScoQAS, our approach is explicated to clarifying the scope of the EAT and expanding the ontology elements as possible in step 3.

In Appendix C we summarize all of the MC constraints with assigned predicates, variables, and tokens for Q1 with QT: (“Where\_Person\_Action”). In addition, the whole ontology entities associated with each of the RT and the EAT entities have been shown. In order to become clearer the concepts that mentioned in the steps of Algorithm 3.4, we bring an example with more details at the following section.

---

**Algorithm 3.4** Extending the constraints in closed domain scenario

---

**Procedure** OBTAINCONSTRAINTSQT(Q)

1. Here again, we apply all the assumptions and variables discussed in algorithm 3.2 (e.g. QT,  $TS_{itm}$ ,  $P(X_i, Y_j)$ , MC, ... ). All of the obtained ontology-tagset variables corresponding to the tag set achieved in Section 3.6.1 are analyzed in order to bind new constraints with their variables. It should be noted that before going to step 2 in the first scenario, all of the steps in algorithm 3.2 is carried out as well. The process will continue from step 2 to fulfill the ontology constraints.
2. In this step, a new index  $u=\max(s)$  is defined and the process of binding the new constraints corresponding to the MC is demonstrated. The pseudo code according to the generalized function is defined as follows:

**for**  $BVars$  **in**  $boundedClassTS_{itm}$  **do**

$$(MC)_{uv} = P(X_u, Y_v) = \text{"class\_"} + TS_{itm} + \text{"\_"} + \text{classIdx};$$

$$(MC)_{uv} = P(X_u, Y_v) = \text{"ont\_"} + TS_{itm} + \text{"\_"} + \text{classIdx};$$

Where  $Y_v$  is class item

$$u=u+1$$

**end for**

**for BVars in boundedSlotTS<sub>itm</sub> do**

$$(MC)_{uv} = P(X_u, Y_v) = \text{"slot\_"} + TS_{itm} + \text{"\_"} + \text{slotIdx};$$

Where  $Y_v$  is slot item

$$u=u+1$$

**end for**

**for BVars in boundedSlotTS<sub>itm</sub> do**

$$(MC)_{uv} = P(X_u, Y_v) = \text{"instance\_"} + TS_{itm} + \text{"\_"} + \text{instanceIdx};$$

Where  $Y_v$  is instance item

$$u=u+1$$

**end for**

3. We assume the first tag of the QT,  $TS_1$ , as a EAT tag and building the constraints for EAT is carried out as follows:

**for BVars in boundedClassTS<sub>1</sub> do**

$$(MC)_{uv} = P(X_u, Y_v) = \text{"EAT\_class\_"} + \text{EATclassIdx};$$

Where  $Y_v$  is class item

$$(MC)_{uv} = P(X_u, X_s) = \text{"ont\_TS<sub>1</sub>\_"} + \text{EATclassIdx};$$

$$u=u+1$$

**end for**

**for BVars in boundedSlotTS<sub>1</sub> do**

$$(MC)_{uv} = P(X_u, Y_v) = \text{"EAT\_slot\_"} + \text{EATslotIdx};$$

Where  $Y_v$  is slot item

$$(MC)_{uv} = P(X_u, X_s) = \text{"Slot\_"} + \text{EATslotIdx};$$

$$u=u+1$$

**end for**

**for BVars in boundedInstanceTS<sub>1</sub> do**

$$(MC)_{uv} = P(X_u, Y_v) = \text{"EAT\_inst\_"} + \text{EATinstancetIdx};$$

Where  $Y_v$  is instance item

$$(MC)_{uv} = P(X_u, X_s) = \text{"Inst\_"} + \text{EATinstanceIdx};$$

$$u=u+1$$

**end for**

**end procedure**

---



In the example Q1, "manager " (token 3) is found in the ontology as an instance of class "Manager" (i.e. ITC manager). In the ontology, we have the relations is\_a(Manager, Employee) and is\_a(Employee, i\_en\_proper\_person) (see top left of Figure 3.4), then these constraints are produced:

```
class_Person_1(Manager, X8)
ont_Person_1(X3, X8)
```

As MC has grown, a new iteration on the dependency tree is attempted, in this case, adding the nmod([X<sub>6</sub>, X<sub>2</sub>]) is carried out, and so forth.

- X<sub>3</sub> for representing manger (mapped to class Manager) are expanded upwards until reaching the class i\_en\_proper\_person (left part of Figure 3.4), because the QT involves a PERSON.
- X<sub>7</sub> (mapped to class i\_en\_proper\_company) is expanded through its class (right part of Figure 3.4), because the ITC, which has the dependence with manager as (3, 5, nmod), is an instance of the i\_en\_proper\_company. Thus, this syntactic relation between ITC and manager leads to involving with a person who restricted by sub class of i\_en\_proper\_organization.

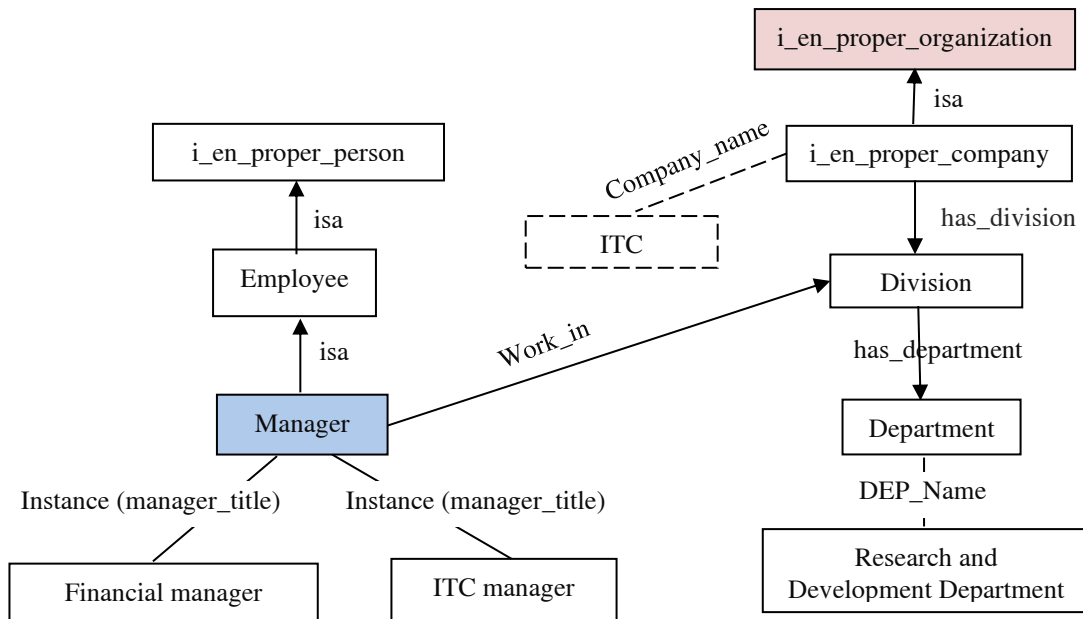


Figure 3.4: Schema of relations between ontology items expanding for Where\_Person\_Action

Then, the ontology entities are expanded greedily until they reach the other entities. New constraints are added with binding the ontology items as follows (see Appendix C with more generated details):

1. Add a variable 'X<sub>8</sub>' for representing PERSON class existing in ontology entities
2. Add a constraint related to ontology entity class\_PER\_1 ('Manager', X<sub>8</sub>)

3. Add a constraint related to ontology entity and PERSON token  
 $\text{ont\_PER\_1}(X_3, X_8)$
4. Add a variable 'X<sub>13</sub>' for representing PERSON slot
5. Add a constraint  $\text{slot\_PER\_2}(\text{'manager\_title'}, X_{13})$
6. Add a constraint  $\text{Slot\_2}(X_8, X_{13})$
7. Add a variable 'X<sub>28</sub>' for representing the PERSON instance
8. Add a constraint  $\text{instance\_PER\_9}(X_{28}, \text{ITC manager})$
9. Add a constraint  $\text{Inst\_9}(X_{13}, X_{28})$

After that, looking in the ontology continues until reaching the ACTION entities (see Figure 3.4) but there are no matched ACTION entities in the ontology.

Let X<sub>1</sub> be the variable with constraints "tk\_Type", which has to be mapped to the ontology. As seen in our example the QT is a "where" question that the EAT is a Place, un-constrained (a Location) or, as in this case, constrained ("organization"). So in this case X<sub>1</sub> has to be mapped to a part-of Organization.

A new variable is added, corresponding to the EAT:

10. Add a variable 'X<sub>66</sub>' for representing the EAT class (Department)

The following constraints are added as well:

11. Add a constraint  $\text{EAT\_Class\_3}(\text{'Department'}, X_{66})$
12. Add a constraint  $\text{ont\_Where\_3}(X_1, X_{66})$
13. Add a constraint  $\text{EAT\_slot\_0}(\text{'DEP\_Name'}, X_{67})$
14. Add a constraint  $\text{Slot\_0}(X_{66}, X_{67})$
15.  $\text{EAT\_inst\_10}(\text{'Research and Development Dep.'}, X_{81})$

In the steps described in the preceding paragraphs, we have ignored many of the constraints and focused on the most relevant ones that lead to the expected answer results.

Now nothing else can be obtained from the question and from its dependency tree. In Figure 3.5, the part of the produced constraints is shown for question Q1 with bounded ontology items for each of the involved tokens after formulating the constraints.

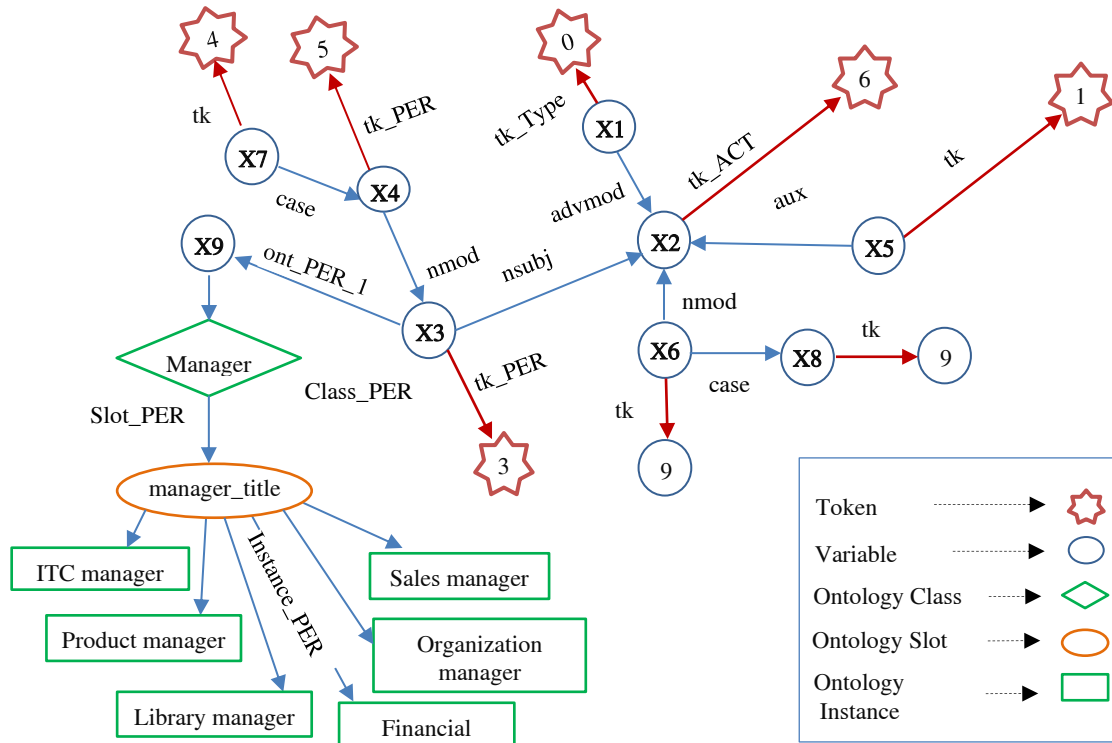


Figure 3.5: Related Terms (RTs) and bounded ontology graph for question Q1

### 3.6.3 Generating the Graph for Extracting the Answer

Moving from words to concepts is important for solving data sparsity issues and promises appealing solutions to polysemy and homonymy by finding unambiguous concepts within a domain [100]. In the case of the first scenario, the ontology provides the resource (semantic space) where answers can be found and extracted. As we mentioned in the previous steps, for the set of different types of questions the QSiS are automatically created. During this process, traversing the Enterprise domain ontology enriches the quality of the classification and the conceptual structure of the QSiS. All of the variables (e.g. ontology-tagset variables) have already been mapped to ontology entities (classes, slots, instances) during the upstream steps (e.g. the case of “manager” in the Q1 example). Up to this stage, the ScoQAS has been able to collect all the syntactic and semantic information of the question along with the expected answers in the form of a coherent dictionary structure. Each question with its corresponding QT pattern is represented as a directed graph, and so the conceptual graph of the question (QGraph) is created by fetching the embedded information in QSiS. However, we define the QGraph as a subgraph of the graph representing the ontology. In the other words, the QGraph is used both as a search space for locating the answer and as a resource for enriching the constraints sets, i.e. for constraining more the search space. The context of the QGraph is analyzed to find the relations between the variables, arguments, ontology classes, and ontology instances corresponding to the variables, EAT classes and EAT

instances. In fact, the ScoQAS is targeting the creation of integrated QGraph at all upstream stages of the evolution process. The QGraph is built based on the QT and the provided constraints.

The challenge we had for implementing this module was how we can properly analyze all the information gathered in relation to the question dependency parsing information and ontology terms in the QSiS and then constructing a well-formed conceptual graph structure. The technical point we needed to taking into account was that the structure of graph should be in a format such that all the nodes and edges can be correctly extracted from graph-based inferencing algorithms. Therefore, we have two main algorithms in this module. The task of the first one is that to analyze the variables and constraints to make the graph elements (nodes and edges). For each of the corresponding QT there is a specific way to check its associated variables and constraints. All of the variables with its arguments are considered as nodes, while constraints are checked with their predicates. Thus, we generate a QGraph file that its content includes “Node” and “Edge” with distinguished edge labels where this format simply can be exploited in the second procedure. We have organized the predicate based on its prefix such as class\_, slot\_, instance\_, ont\_, EAT\_class\_, EAT\_slot\_, EAT\_inst\_, EEAT\_class\_, EEAT\_slot\_, EEAT\_inst\_, and Answer\_. The postfix of the predicate could be tag set or other index. With respect to each of these values, this algorithm controls the variables and arguments of the associated constraints and defines the corresponding edge in the QGraph accordingly. It should be noted that based on the type of predicate’s prefix of the constraints, specific functions have been implemented to add the edge to the QGraph. For example, add\_Edge\_Var2tk(), add\_Edge\_Var2Class(), add\_Edge\_Var2Slot(), add\_Edge\_Var2Instance(), etc.

The second algorithm aims to get the provided QGraph file that includes nodes and edges in order to generate a specific format which processable with Prolog inference algorithm. The output of this algorithm is produced in our well-defined format which mainly consists of five main distinguish features. These features are “Node”, “Edge”, “EAT\_class”, “EAT\_instance”, and “Answer”.

During the generation of the QGraph, we distinguish its context in two parts: one is that we produce all of the dependency and ontology variables and arguments for the corresponding pattern. The second part is dedicated to determine all of the EAT items as EAT Class, EAT Instance. For clarifying the latter, we need to look for a specific location, tk\_Type:’0’ (Where in the example Q1) and a retrieving class, slot and instances related to location (organization) which involved in the ontology. Thus, the feature should be bound to the EAT\_Class and EAT\_Instance. The generated variables, constraints, and EATs associated with Q1 are shown in Appendix C.

The Appendix H shows a related pseudo code that how the ScoQAS generates the QGraph which its format is usable in answer inference algorithm. This algorithm was implemented in general form that the input is the constraint-based QSiS file produced in the Algorithm 3.1. The portions of produced QGraph are depicted in Figure 3.6, Figure 3.8, and Figure 3.9. We brought some samples of each 5 mentioned features which make up the QGraph in our well-defined format.

Node	X1	:	{	Var :	X1	}
Node	0	:	{	Arg :	50	}
Node	X2	:	{	Var :	X2	}
Node	6	:	{	Arg :	100	}
Node	X3	:	{	Var :	X3	}
Node	3	:	{	Arg :	150	}
Node	X4	:	{	Var :	X4	}
Node	5	:	{	Arg :	200	}
Node	X5	:	{	Var :	X5	}
Node	1	:	{	Arg :	250	}
Node	X6	:	{	Var :	X6	}
Node	9	:	{	Arg :	300	}
Node	X7	:	{	Var :	X7	}
Node	i_en_proper_company	:	{	Arg :	350	}
Node	X8	:	{	Var :	X8	}
Node	Manager:	:	{	Arg :	400	}
Node	X9	:	{	Var :	X9	}
Node	has_authority	:	{	Arg :	450	}
Node	X10	:	{	Var :	X10	}
Node	Authority	:	{	Arg :	500	}
Node	X11	:	{	Var :	X11	}
Node	control	:	{	Arg :	550	}
...						
Node	X67	:	{	Var :	X67	}
Node	DEP_Name	:	{	Arg :	3350	}
Node	X81	:	{	Var :	X81	}
Node	Research and Development Dep.	:	{	Arg :	4050	}
...						

Figure 3.6: Part of the Nodes format belongs to the QGraph for Q1

As seen in Figure 3.6, the outstanding pair green rows indicate the nodes corresponding to variable (e.g., X3) and its real value (e.g., token number 3, “manager” in Q1) with bound arguments (e.g., 150).

**Note:** In our assumption, each variable (e.g. X3) as a node has one real value (e.g., 3) where this value plays a role as a leaf node for this variable. It is supposed that each variable can be connected to any other variables in the graph (i.e., more than one variable such as (X3, X2) with label “nsubj” and (X3, X8) with label “ont\_PER\_1”).

As seen in Figure 3.6 the node corresponding to X8 with node “Manager” is also sample of bound ontology item. Taking into account that the real value of the variable has its own argument as an index in order to simply control and access it in the QSiS and the QGraph as well (see Figure 3.8). In Figure 3.7, part of the automatically produced QGraph is shown and the relations between tokens and their related ontology items with bound variables are seen.

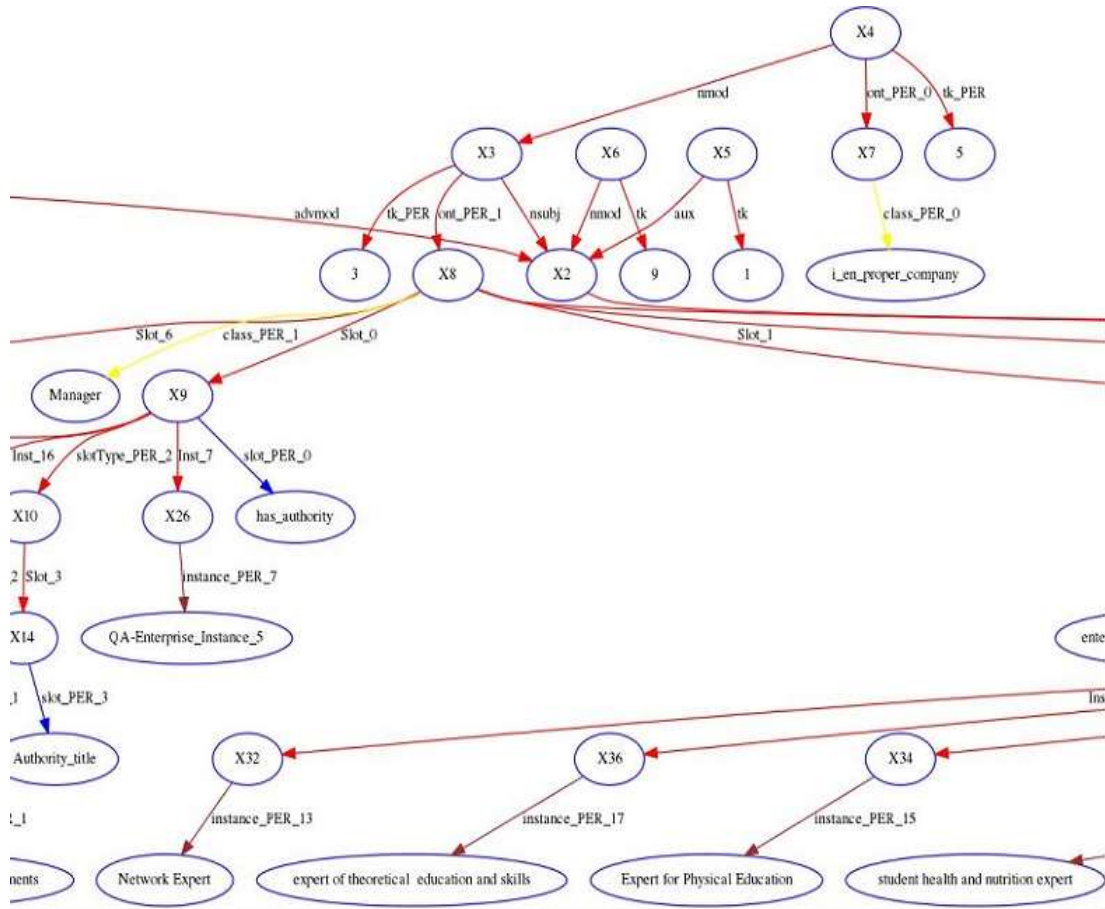


Figure 3.7: Automatically produced QGraph for Q1 based on its constraints

As shown in Figure 3.6 and Figure 3.8 the highlighted pink rows, “DEP\_Name” and “Research and Development Dep.,” also refer to the ontology items with associated arguments. There is an edge between X67 and X81 with instance label (e.g. Inst\_10). Thus, it can be interpreted that the variable X81 is an instance of the variable X67, and so on.

In order to provide the information needed to make a proper inference for extracting the final answer, the final part of the QGraph is for ontological EAT items (e.g., EAT class and instance highlighted yellow rows). As shown in Figure 3.9, the values of ontology items are presented and these are extracted from QSiS information.

Edge	('X1', 0) :	{	tk_Type	}	
Edge	('X2', 6) :	{	tk_ACT	}	
Edge	('X3', 3) :	{	tk_PER	}	
Edge	('X4', 5) :	{	tk_PER	}	
Edge	('X1', 'X2')	:	{	advmod	}
Edge	('X5', 1) :	{	tk	}	
Edge	('X5', 'X2')	:	{	aux	}
Edge	('X3', 'X2')	:	{	nsubj	}
Edge	('X6', 9) :	{	tk	}	
Edge	('X6', 'X2')	:	{	nmod	}
...					
Edge	('X8', 'Manager')	:	{	class_PER_1	}
Edge	('X3', 'X8')	:	{	ont_PER_1	}
Edge	('X9', 'has_authority')	:	{	slot_PER_0	}
Edge	('X8', 'X9')	:	{	Slot_0	}
Edge	('X10', 'Authority')	:	{	class_PER_2	}
Edge	('X9', 'X10')	:	{	slotType_PER_2	}
Edge	('X11', 'control')	:	{	slot_PER_1	}
Edge	('X8', 'X11')	:	{	Slot_1	}
Edge	('X12', 'Expert')	:	{	class_PER_3	}
Edge	('X11', 'X12')	:	{	slotType_PER_3	}
Edge	('X13', 'manager_title')	:	{	slot_PER_2	}
Edge	('X8', 'X13')	:	{	Slot_2	}
...					
Edge	('X81', 'Research and Development Dep.')	:	{	EAT_inst_10	}
Edge	('X67', 'X81')	:	{	Inst_10	}
...					

Figure 3.8: Part of the Edges format belongs to the QGraph for Q1

EAT_class	Employeeel	i_en_proper_personl	Recordl	Departmentl
EAT_instance	Planning Dep.l	1372/10/10l	Engineering	Dep.l
	Telecommunication Dep.l	1380/10/02l	1380/1/1l	Marketing Dep.l
	http://protege.stanford.edu/rdfenterprise_Class130068l		103l	
	http://protege.stanford.edu/rdfenterprise_Class50001l		Research and Development Dep.l	
	1375/02/02l	1378//01/01l	104l	1381/12/1l
	http://protege.stanford.edu/rdfenterprise_Class30008l	100l	105l	1382/03/05l
	1378/10/13l	Administrative Dep.l	1375/11/13l	Educational Dep.l
	Cultured Dep.l	101l	1385/11/10l	107l
	1372/12/1l	102l	1379/01/01l	106l
Answer	Research and Development Dep.l			

Figure 3.9: The EAT class and instance with candidate answer format attached to the QGraph for Q1

### 3.6.4 Inference to Elicit Exact Answer from the QGraph

In the first scenario, the EAT is a set of classes belonging to the ontology (in fact nodes of the QGraph). Depending on its multiplicity the answer has to be one or more instances of one of these classes. The searching process consists of navigating over the QGraph looking for nodes X satisfying the constraints. In practice, the

Algorithm 3.5<sup>1</sup> shows the extraction of the answer (if possible) as a pseudocode, which uses the produced QGraph format.

---

**Algorithm 3.5** Searching process over the QGraph

---

```
procedure INFERENCE_ANSWER_QGRAPH(QGraphfile QG)
1  EAT_Class=findEAT_classes_Node(QG)
2  MC = findMandatoryConstraints_Nodes(QG)
3  Involved_Vars = findInvolved_Variables(QG)
4  for Cls in EAT_Class do
    EAT_InstanceC = findInstance(Cls)
    EAT_Instance = AddtoEAT_Instance(EAT_InstanceC)
  end for
5  for Ins in EAT_Instance do
    CAND_Answer = AddtoCandidateAnswer(Ins)
    for X in MC do
      if (findConnection_Edge(X, CAND_Answer) and ConnectType != "is_a") then
        NodesToAnswers = AddconnectedNode(X)
      end if
    end for
    Expected_to_Connected = Involved_Vars
6  for Y in Involved_Vars do
    for Z in NodesToAnswers do
      DC=directlyConnected(Z)
      AD=extendConnected(Y, DC, Z)
      if (indirectConnection_Edge(Y, AD, Z) and ConnectType != "is_a") then
        Remove_Node(Expected_to_Connected, Y)
      else
        ModifyNodesConnectedToAnswer(AD)
      end if
    end for
  end for
7  if Len(Expected_to_Connected) == 0) then
    Print "The Result is OK!!"
  end if
end procedure
```

---

---

<sup>1</sup> This pseudocode has been implemented with Prolog programming code, which tries to demonstrate in high level algorithm to evaluate the answer. In order to explicitly expressing the algorithm, further details are discarded.



In the following, given the pseudo-code in Algorithm 3.5, we elaborate the pseudo-code operation step-by-step with referring to example Q1. The process of analyzing the nodes, edges and EATs in the produced format to do answer inference is carried out as follows:

- 1) All the expected answer classes are collected from QG for our case in Q1 (Number 1 in Algorithm 3.5):

eatClass={ 'Employee', 'i\_en\_proper\_person', 'Record', 'Department' }.

- 2) For each expected answer class, all the instances are collected and 31 instances are found between them (for loop at number 4 in Algorithm 3.5), so in this case:

**eatInstance('Planning Dep.').**

eatInstance('1372/10/10').

**eatInstance('Engineering Dep.').**

**eatInstance('Telecommunication Dep.').**

eatInstance('1380/10/02').

eatInstance('1380/1/1').

**eatInstance('Marketing Dep.').**

eatInstance('http://protege.stanford.edu/rdfenterprise\_Class130068').

eatInstance('103').

eatInstance('http://protege.stanford.edu/rdfenterprise\_Class50001').

**eatInstance('Research and Development Dep.').**

eatInstance('1375/02/02').

eatInstance('1378//01/01').

eatInstance('104').

eatInstance('1381/12/1').

eatInstance('http://protege.stanford.edu/rdfenterprise\_Class30008').

eatInstance('100').

eatInstance('105').

eatInstance('1382/03/05').

eatInstance('1378/10/13').

eatInstance('Administrative Dep.').

eatInstance('1375/11/13').

eatInstance('Educational Dep.').

eatInstance('Cultured Dep.').

eatInstance('101').

eatInstance('1385/11/10').

eatInstance('107').

eatInstance('1379/01/01').

eatInstance('106').

eatInstance('1372/12/1').

eatInstance('102').

- 3) For each candidate (eatInstance) the following cases might be occurred after the whole process 4-6:

Obviously three cases can occur:

- a) No solution is obtained and none of them cannot be found.

It is failed.

- b) Only one candidate is found, i.e. one solution is obtained.

Perhaps the solution is the correct one, then it should be checked.

- c) More than one solution is obtained.

Perhaps one of the proposed solutions is the good one.

- 4) In this case only the following instance satisfies the constraints:

answer('Research and Development Dep.'). For each of the candidates (eatInstance) the process is the following:

Get the set of nodes that can be reached from the candidate answer following edges (excluding “is\_a” edges and using both directions of the edges), in this case, we obtain the following:

197 nodes connected to answer are found that listed as follows:

- 5) Get the set of involved variables, so in this case:

involvedVars('X2').

involvedVars('X3').

involvedVars('X1').

involvedVars('X6').

involvedVars('X4').

involvedVars('X5').

- 6) Checking whether the involved variables belong to the set obtained in 4. All these nodes should be connected to the candidate as is the case here:

6 nodes expected to be connected

X2, X3, X1, X6, X5, X4

The number of 0 nodes pending to be connected, as 0 nodes are pending, so this means that the result is OK!

Now the paths can be found from the *involvedVars* to the answer.

edge(4,'X1','X2').

edge(7,'X3','X2').

edge(10,'X4','X3').

edge(6,'X5','X2').

edge(9,'X6','X2').

edge(14,'X3','X8').

edge(124,'X1','X63').

edge(130,'X1','X66').

edge(132,'X66','X67').

edge(160,'X67','X81').

edge(159,'X81','4050').

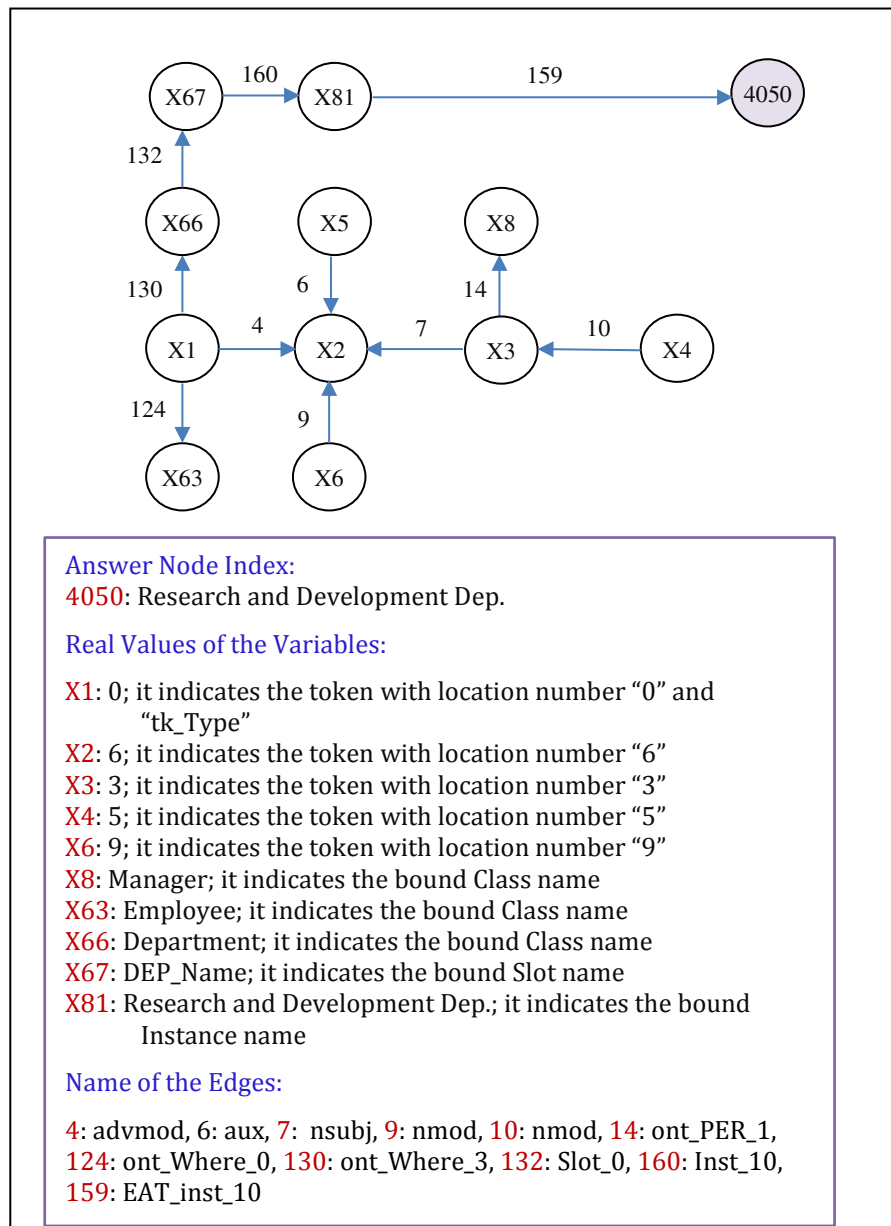


Figure 3.10: The relationships obtained in the inference of the answer between involvedVars and Answer

Figure 3.10 shows the found paths from involvedVars nodes to Answer node in the provided QGraph. In fact getting the paths is not needed for getting the answer. We include the figure for the sake of clarification.

### 3.7 Semantic-based QA in an Open Domain: 2<sup>nd</sup> Scenario

The constant growth of Linked Open Data (LOD) on the Web opens new challenges pertaining to querying such massive amounts of publicly available data.

LOD datasets are accessed through various interfaces, such as SPARQL endpoints, data dumps, and triple pattern fragments. Recently, there have been projects, which have taken a crowdsourcing approach, where knowledge is collected by a web community. For example, DBpedia is the center of the LOD initiative and compiles the information from Wikipedia infoboxes into a single source [1], [101], [102]. Another example is YAGO [5] which combines Wikipedia infoboxes with WordNet. In this way, researchers are presenting QA systems that use these curated KBs to obtain an answer. Questions have been interpreted as formal queries (e.g. SPARQL) over curated KBs like DBpedia, YAGO, etc. Moreover, various sources produce streaming data. Querying on these types of sources is of central significance for the scalability of Linked Data and Semantic Web technologies in order to exploit the massive amount of LOD data to its full potential, so users will be capable to query and combine this data easily and effectively.

In the case of the second scenario, the steps from one to five are carried out very similar to those of the first scenario so that in the fifth step the QSiS almost has been completed (see ScoQAS architecture in Figure 3.1). Many common modules for all the processes are similarly applied to the second scenario but in the answer retrieval step, the resources are different and the answer extraction approaches follow their own specific procedures. There is an exception that we are not considering the rules in QC which performing conditions and actions invoking the domain ontology in order to interpret question. As no constraints regarding domain ontology exist in this scenario, the set of MC uses to be smaller and the ambiguity of the answer candidates is higher. In other words, the provided constraints are only used in order to interpret the question. The specific process for second scenario is shown in Figure 3.11 as a sub part of our general architecture as described in Section 3.1.

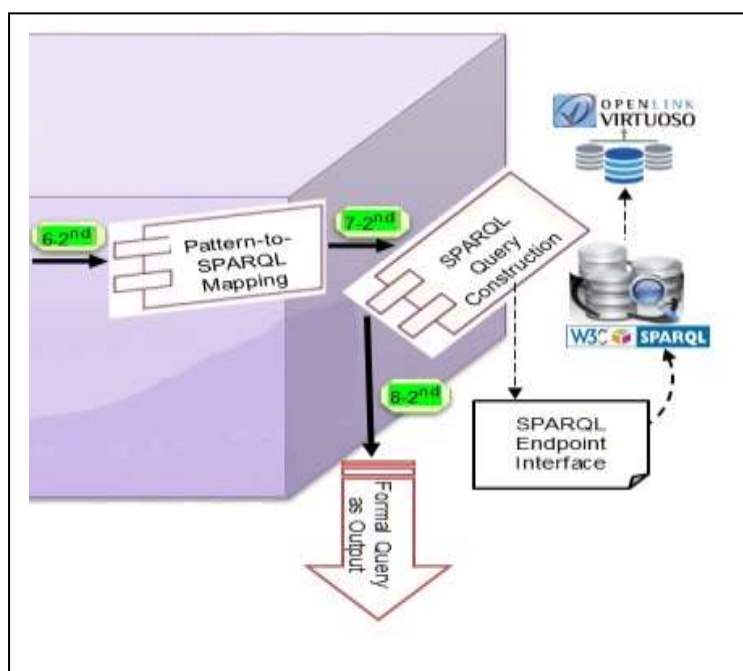


Figure 3.11: A flowchart of the process in open domain

To deal with the challenges of mapping NL to SPARQL query format, we have implemented a heuristic method to generate the SPARQL query associated with the pattern and its constraints without using any predefined templates. For the main task, the dependence on DBpedia is not so important because of domain independence but can sometimes be necessary as well. Consider, for instance, the question “Who is the mayor of Berlin?”; answering this question requires mapping “Berlin” to a DBpedia resource and “mayor” to a DBpedia property. Also a disambiguation of the former (“Berlin” can be a city but also a person) should be considered.

The core of SPARQL queries are basic graph patterns, which can be viewed as a query graph. Our approach aims to have high precision and recall by being able to generate SPARQL queries from NLP and machine learning techniques. We will now discuss the details of these specific steps.

#### 3.7.1 Mapping Syntactic-Semantic Information Structure of the Question (QSiS) to Generate SPARQL Formal Query

Up to this step, the question syntactic-semantic information structure (QSiS) formed by the Question Type (QT), Related Terms (RT), Expected Answer Type (EAT), and Constraints (CSTR) corresponding to the question has been obtained and is available to be used for building a formal query in the second scenario. To this end, all of the extracted information is integrated as shown in Table 3.6 and in the next step; this data will be used to build SPARQL queries to obtain the desired answer. We used a combination of the QALD standard benchmark<sup>1</sup> (QALD-2, QALD-3, and QALD-4) training sets as our training questions in the second scenario of the ScoQAS. The training data comprises of 188 NL questions for English DBpedia, annotated with keywords as well as corresponding SPARQL queries and the answers that these queries retrieve. The questions are of different complexity and are available in six languages: English, Spanish, German, Italian, French, and Dutch. We only extracted English questions, which were used, in our training set.

Table 3.6 shows the analysis of the example Q2 from our training data set “Which countries are connected by the Rhine?”. The first column is the question index, the second is the question (Q), the third is question type (QT), the fourth is EAT, the fifth column is the quantifier (Quant), and the last is dictionary of constraints (cS) which have been satisfied. As can be seen in the cS column, the bounded value of ‘tk\_Type’ is the token with index ‘1’ (country), i.e. the domain of EAT (Location) is limited to all of the countries. In ScoQAS, more than one QT is obtained for some questions. As a result, it may match with more than one SPARQL query. This table presents the QSiS, which obtained before the step 6 (i.e. current step).

The objective of this module is to map QSiS format, previously processed by the steps 1 to 5 in the ScoQAS architecture (Figure 3.1), into the SPARQL template queries that will be next sent to the Virtuoso DBpedia endpoint<sup>2</sup> in order to get the final answer for the question.

---

<sup>1</sup> <http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/>

<sup>2</sup> <http://dbpedia.org/sparql>

Mapping QSiS into SPARQL is not an easy task. Several problems arise and have to be faced:

- Different namespaces coexist in DBpedia, some of them belonging to DBpedia itself, and others corresponding to links from DBpedia to other ontologies, such as Yago.
  - For instance, looking for the generic term 'Mountain' we find 217 categories in the Yago namespace, e.g.:  
<http://dbpedia.org/class/yago/Mountain109359803>)
  - 10 DBpedia properties, e.g.:  
<http://dbpedia.org/ontology/highestMountain>)
  - 6 DBpedia ontology categories, e.g.:  
<http://dbpedia.org/ontology/Mountain>)

Table 3.6: Syntactic-semantic information structure of the question (QSiS) for Q2

ID	Question Q2	QT	EAT	Quant	Constraints (cS)					
					Pred_1	Idx	Pred_2	Idx	Pred_3	Idx
1	Which countries are connected by the Rhine?	Where_GEO_Action	Location	-	tk_ACT	3	tk_GEO	6	tk_Type	1
Token			country	all	connect		Rhine		country	

- There is a lack of coherence in the nomenclature used for naming DBpedia entries (classes, properties and instances). The use of lower/upper case, singular/plural forms, abbreviations, and the order of simple components of the multi-word expressions, the inclusion of parenthesis, underscores, and other orthographic marks is rather arbitrary or at least not followed and difficult to interpret. For instance, the following properties (among many others) were found in DBpedia when looking for “number of members”:
  1. <http://dbpedia.org/property/memberNo>
  2. <http://dbpedia.org/property/members>
  3. <http://dbpedia.org/property/member>
  4. <http://dbpedia.org/property/numMembers>
  5. <http://dbpedia.org/property/membersNumbers>
  6. <http://dbpedia.org/property/noOfMembers>
- The habitual clash when mapping terms of the NL expression of the question to terms of the ontology is obviously present. The habitual problems of

polysemy (a term of the question can be mapped to many terms of the ontology: classes, properties, and instances) and synonymy (an ontology term can be referred to by different question terms) frequently occur.

- The directionality of the relations in the ontology is not always clear. For instance, it is not clear whether <http://dbpedia.org/property/mayor/> links a city to a person or a person to a city.
- Depending on the Question Type (QT), the EAT and the complexity of the question, partially reflected in the constraints provided by the question processing module, resolving the mapping can be difficult.

### 3.7.2 Pre-processing Steps

We have performed two pre-processing steps for this module, one general and the other applied to each of the different datasets to be processed.

#### A) General Pre-processing

We have indexed all the Yago classes, DBpedia classes, and DBpedia properties. In addition to this, we have built an index for all the simple word forms contained in the previously indexed multi-word entries. For instance, from the property [http://dbpedia.org/property/ u.s.SeniorNationalTeamMember](http://dbpedia.org/property/u.s.SeniorNationalTeamMember), the set {'u.s.', 'Senior', 'National', 'Team', 'Member'} has been extracted and indexed. In contrast to this, access to instances is carried out online.

#### B) Dataset Pre-processing

We collected all the actions occurring in the question dataset (all the tokens referred to within the constraint set under the key 'tk\_ACT'). For each action we obtained its lemma (using NLTK's WN-lemmatizer), the set of all its variants (using NLTK's WN<sup>1</sup> tools). The initial set of variants was later enriched with the orthographic variants (upper/lower case, singular/plural, and in the case of multiword expressions, different component separators, ' ','\_'). In this way, we built an action/actor dictionary.

For all the classes corresponding to EAT categories, both generic, a location, date, or person, and specific, a city, astronaut, film, or year, i.e., those referred to within the constraints set under the key 'tk\_Type', we collect the set of their upper classes in Yago and DBpedia ontologies.

### 3.7.3 Constraint-Based Mapping Rules to Build SPARQL Query

Here, the goal is to construct queries (i.e. SPARQL queries) for a given set of constraints and variables presented in Section 3.6.2. We generate queries using all bounded variables and their corresponding QT, EAT, RT, and CSTR as discussed in Section 3.4 and Section 3.5. As shown in Table 3.2 we have defined a set of 75 question types (QT) so that for each QT one or more mapping rules have been manually built. In total 127 rules form our rule set (an average of 1.5 per QT). The

---

<sup>1</sup> <http://www.nltk.org/howto/wordnet.html>



rules attached to each QT are sorted by their accuracy, which is calculated through their application to the training dataset. Table 3.2 presents QT's whose EAT's in different types (person, location, quantifier, date/time...).

The input for the application of each mapping rule coming from the question processing step consists of:

- The question type, QT
- The expected answer type, EAT
- The quantifier, Quant
- The dictionary of constraints (cS) that have to be satisfied. cS constraints are indexed by 'tk\_GEO', geographic tokens occurring in the question, 'tk\_PER', persons, 'tk\_ORG', organizations, 'tk\_ACT', actions, 'tk\_Type', type of the EAT, and some others.

For instance, let us consider the input question with ID number 1 in Appendix B, hereafter Q2, “In which country does the Nile start?” The following information was extracted from the question:

- *QT*: Where\_Action\_GEO
- *EAT*: Country
- *Quant*: None
- *cS*: {'tk\_ACT': '6' ('start'), 'tk\_GEO': '5' ('Nile'), 'tk\_Type': '2' ('country' )}

The rules are also able to access the information resulting from the linguistic processing of the question (tokens, named entities (NE), dependency parse tree). Most of the questions, however, can be solved using only the superficial word-level information. Some general procedures on the form of building the rules are as follows:

- All the tokens corresponding to content categories (i.e. non-function words) should constrain the search if possible. Such tokens in the previous example would be ‘country’, ‘Nile’, and ‘start’. In fact, the only mandatory tokens are those referred to within the cS but, if possible, we try to extend the coverage of cS items to include as many content tokens as possible. For this task the parsed dependency tree of the question is used in a way that tokens are dependent (through 'dep', 'mod', 'nn', or 'prep\_of' labeled dependencies) on those referred to the entry of the governors are incorporated into these entries.
- The EAT has to be set for the cases not solved previously (i.e. in the cases where the EAT was undefined). This mechanism is very simple. For instance, for “Where” questions, if 'tk\_Type' exists in the cS we check whether the corresponding token can be mapped to a class existing in DBpedia placed under the set of classes attached to location. If this is the case, the EAT is set to this class, otherwise it is set to the generic 'location'. In the example above the EAT was set to 'Country' from the information in 'tk\_Type' in the cS.

- The EAT should be mapped to at least one DBpedia or Yago class. If this mapping cannot be set, the rule fails.
- As previously mentioned, all the constraints are mandatory and, thus, have to be mapped to DBpedia or Yago classes or DBpedia resources or properties, depending on the type of constraint. If any constraint remains unsolved, the rule fails.
- In general, when looking in the ontology for a specific token (simple or multiword) all the variants are used, including WordNet (WN) synonyms, morphological derivations and orthographic variations, as described above.
- Most of these tasks are general and are applied in a straightforward manner to any question. The only difference between rules is how the sets of ontology elements obtained by expanding the constraints in cS are combined. The key point is the size of these sets. If more than one set has more than one element, all the combinations (i.e. the Cartesian product) have to be considered. If the number of the combinations is huge, the least likely combinations (using string distance as a metric) are filtered out (we used a threshold of 10). For instance, for question 13 (“Who produces Orangina?”), 50 properties were selected (clearly above the threshold) by the corresponding rule ('production', 'build', 'producer', 'author', etc.). From these, only the most likely were maintained ('produce', 'production', 'producer', etc.).
- In the case of 'tk\_ACT' constraints, for getting the candidates, the action, its lemma and the nominalizations included in the action/actors dictionary, as described in Section 3.4.1, are used as seed terms. The rest of the process is the same as described above.
- In the case of properties, the sets used to be large. We restrict the sets to two types of candidates: i) those containing just a single variant in the predicate form (i.e. the term is a single word expression corresponding to the target), and ii) those satisfying two of the targets (i.e. those where the term is a multiword expression containing two single components, both corresponding to the target. For instance, in question with ID number 0 in Appendix B, hereafter Q3, (“Give me all female Russian astronauts.”) we looked for 'female' and we found that 420 Yago categories included this term as well as 14 DBpedia properties. The set of 420 Yago categories was reduced to only one ('<http://dbpedia.org/class/yago/Female109619168>') applying the first constraint. In the same example, looking for 'Russian', 'astronaut' (and its synonym, using WN, 'cosmonaut') and applying the second constraint the set S is obtained:

S={<http://dbpedia.org/class/yago/FemaleAstronauts>,  
<http://dbpedia.org/class/yago/RussianCosmonauts> }

We will now present two examples of rule application with a detailed explanation of their performance.

**First Example:**

Question Q3 (“Give me all female Russian astronauts.”)

QT: Who\_Properties

EAT: ‘?’

Constraints: {tk\_Quant: 0 (‘all’), tk\_Props: [1, 2] (‘female’, ‘Russian’), tk\_Type: 3 (‘astronauts’)}

First the EAT is set to ‘astronaut’ using the WN lemmatizer. Then the set of keywords, including tk\_Props and tk\_Type is expanded using WN NLTK tools, resulting in {'Russians', 'astronauts', 'Astronaut', 'female', 'females', 'Distaffs', 'Females', 'russians', 'distaff', 'astronaut', 'Russian', 'distaffs', 'Cosmonaut', 'Spaceman', 'Spacemen', 'Distaff', 'russian', 'Cosmonauts', 'cosmonaut', 'spaceman', 'spacemen', 'Astronauts', 'Female', 'cosmonauts'}

A lot of classes, properties, and instances (resources) are found, for instance, for ‘Russian’, 531 Yago classes and 1 DBpedia class are found. For ‘Astronaut’, 2 Yago classes, 2 DBpedia properties, and 1 DBpedia class are found.

As the QT Who\_Properties deals with independent constraints, we try to collect Yago or DBpedia classes that could be related to target by means of an rdf:type relation.

The ScoQAS tries to select classes covering at least two of the three keywords. In this way we obtained the set {<http://dbpedia.org/class/yago/FemaleAstronauts>, <http://dbpedia.org/class/yago/RussianCosmonauts>} where each of the classes cover two of our keywords and the both covers all three keywords.

Using these two constraints the following SPARQL query, clearly correct, is finally built:

```
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbp: <http://dbpedia.org/property/>
Select DISTINCT ?x WHERE {
    ?x rdf:type <http://dbpedia.org/class/yago/RussianCosmonauts>
    ?x rdf:type <http://dbpedia.org/class/yago/FemaleAstronauts> }
```

**Second Example:**

Question Q4: (“Who was the successor of John F. Kennedy?”)

QT: Who\_CompoundProperties\_Person

EAT: ‘?’

*Constraints:* {tk\_PER: [5, 6, 7] ('John F. Kennedy'), tk\_CmpProp: [3] ('successor')}

In the example Q4, at first the EAT is set to 'Person' ('successor' was not found to be a person in DBpedia). As the QT is 'Who\_CompoundProperties\_Person', a person has to be located. In this case, 'John\_F.\_Kennedy' is found as an instance (resource) in DBpedia. In this case, as the constraints are related (not independent as in the previous example Q3) we have to look for the possible relations using the dependency tree of the question. The dependency tree is in this case:

[(1, 0, 'attr'), (-23, 1, 'root'), (3, 2, 'det'), (1, 3, 'nsubj'), (7, 5, 'nn'), (7, 6, 'nn'), (3, 7, 'prep\_of'), (1, 8, 'punct')]

We can see that there is an 'nn' (nominal composition) between the tokens 'John', 'F.', and 'Kennedy' and a 'prep\_of' relation between 'Kennedy' and 'successor'. Therefore, the triple should be <'Kennedy', 'successor', ?x>. We have already located the person 'Kennedy' corresponding to the subject of the triple, and the object of the triple is our target. Therefore, we need only to recover the property or the set of properties that are most likely to correspond to 'successor'.

We recover 19 DBpedia properties containing 'successor' or its expansions, i.e.

<http://dbpedia.org/property/successors>  
<http://dbpedia.org/property/heir>  
<http://dbpedia.org/property/spiritualSuccessor>  
<http://dbpedia.org/property/replacement>  
<http://dbpedia.org/property/officeSuccessor>  
<http://dbpedia.org/property/successorTo>  
<http://dbpedia.org/property/competitorSuccessor>  
<http://dbpedia.org/property/successorCo>  
<http://dbpedia.org/property/successor>  
<http://dbpedia.org/ontology/successor>

Applying the first constraint defined above, we restrict this list to the properties containing only a simple term. The list is reduced to the following one:

<http://dbpedia.org/property/successors>  
<http://dbpedia.org/property/heir>  
<http://dbpedia.org/property/replacement>  
<http://dbpedia.org/ontology/heir>  
<http://dbpedia.org/property/successor>  
<http://dbpedia.org/ontology/successor>

As the initial word form included in the question, 'successor', corresponds to only some of these properties, we remove the others and build the final SPARQL query that is also correct:

```
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbp: <http://dbpedia.org/property/>
Select DISTINCT ?x WHERE {
    { res:John_F._Kennedy <http://dbpedia.org/property/successors>?x. }
  UNION
    { res:John_F._Kennedy <http://dbpedia.org/property/successor>?x. }
  UNION
    { res:John_F._Kennedy <http://dbpedia.org/ontology/successor>?x. }
}
```



## 4 The Empirical Evaluation and Results

---

### 4.1 Evaluation Measures

In our approach, we worked with NL questions written in English. The ScoQAS system has been implemented in Python. Most important parts of ScoQAS code<sup>1</sup> are accessible in Github. We need to evaluate the ScoQAS in the two scenarios: a) evaluation of the closed-domain (domain-restricted) approach, b) the evaluation of open domain (domain independent) approach. The results of the ScoQAS on the training and the test set are summarized in separate tables for both scenarios. Based on the nature of the thesis, we apply two different training set and test set questions. There is a set of dimensions in order to analyze the efficiency of the semantic-based QA system which shows a negative/positive impact on the runtime and the accuracy of the ScoQAS system. For the evaluation of the experiments, we use the standard and most commonly applied evaluation metrics.

We suppose 2-by-2 contingency table as shown in Table 4.1.

Table 4.1: The 2-by-2 contingency table

	Correct	Not Correct
Selected	True Positive (TP)	False Negative (FN)
Not Selected	False Positive (FP)	True Negative (TN)

The precision and recall formulas based on our categorization in Table 4.1 are as follows:

---

<sup>1</sup> <https://github.com/mlatifi/ScoQAS>

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

To evaluate the overall performance of the ScoQAS, we tune parameters to achieve the best possible results. So the result could be favorable answer or it is not expected answer or even inadequate answer. The results of the ScoQAS are compared to the gold standard results and evaluated in terms of precision and recall. For the set of questions (SoQs), precision, recall and F-measure are computed as follows:

$$\text{Recall}(\text{SoQs}) = \frac{\text{no.of correct system answers for a set of questions}}{\text{no.of gold standard answers for a set of questions}}$$

$$\text{Precision}(\text{SoQs}) = \frac{\text{no. of correct system answers for a set of questions}}{\text{no.of system answers for a set of questions}}$$

$$\text{F - Measure}(\text{SoQs}) = \frac{2 \times \text{Precision}(\text{SoQs}) \times \text{Recall}(\text{SoQs})}{\text{Precision}(\text{SoQs}) + \text{Recall}(\text{SoQs})}$$

The set of questions for evaluation of the closed-domain scenario can be found in Appendix A. Those for the open domain scenario, the QALD-2, QALD-3, QALD-4, and QALD-5 test sets are shown in Appendices D, E, F, and G respectively and these are briefly described in the following sections.

Evaluation framework has to be obviously different in both scenarios because of lacking of an external golden data set for the Enterprise domain. Therefore, our evaluation of the first scenario is mainly qualitative while the evaluation of the second scenario, disposing in this case of several golden sets in quantitative. We hope that the combination of both evaluations will produce a fair result.

## 4.2 The Evaluation of Closed-domain Approach

In the closed-domain approach, one of the major challenges of the evaluation of the ontology-based QA is a lack of predefined benchmark(s) as a starting point and also the lack of tools to facilitate the evaluation process of Semantic based QA systems, which has not been proposed in literature. Therefore, the main idea to determine our evaluation criteria to measure interpretation of the questions and inference answer was to define measures that demonstrate the actual hardness of the problem and the actual efficiency of our approaches. Hence, we have determined a baseline analysis model in order to evaluate the accuracy of the most important modules of ScoQAS architecture. We created a benchmark for closed domain QA system based on six steps in the process consisting of Stanford parsing failure, QC, EAT, building Constraint, building QGraph, and inferencing correct answer (see



Figure 4.1). Then we have evaluated the domain restricted ScoQAS based on precision and recall. In this phase of our experimental research, we have collected as said in Section 3.6 a set of 40 NL questions over Enterprise ontology which are supposed to be representative (see Appendix A). Before choosing the questions in Enterprise ontology, we studied the current problems in textual inferences in FraCas (Framework for Computational Semantics) Project [52], NIST TREC<sup>1</sup>, and challenging topics in QALD series.

So the questions were extracted from the Enterprise ontology by analyzing and considering the more challenging complex questions and unresolved issues in current QA systems. The provided questions are the template for various type of questions such as list questions, Yes/No questions, Wh-questions, time related questions, etc. adapted to Enterprise ontology.

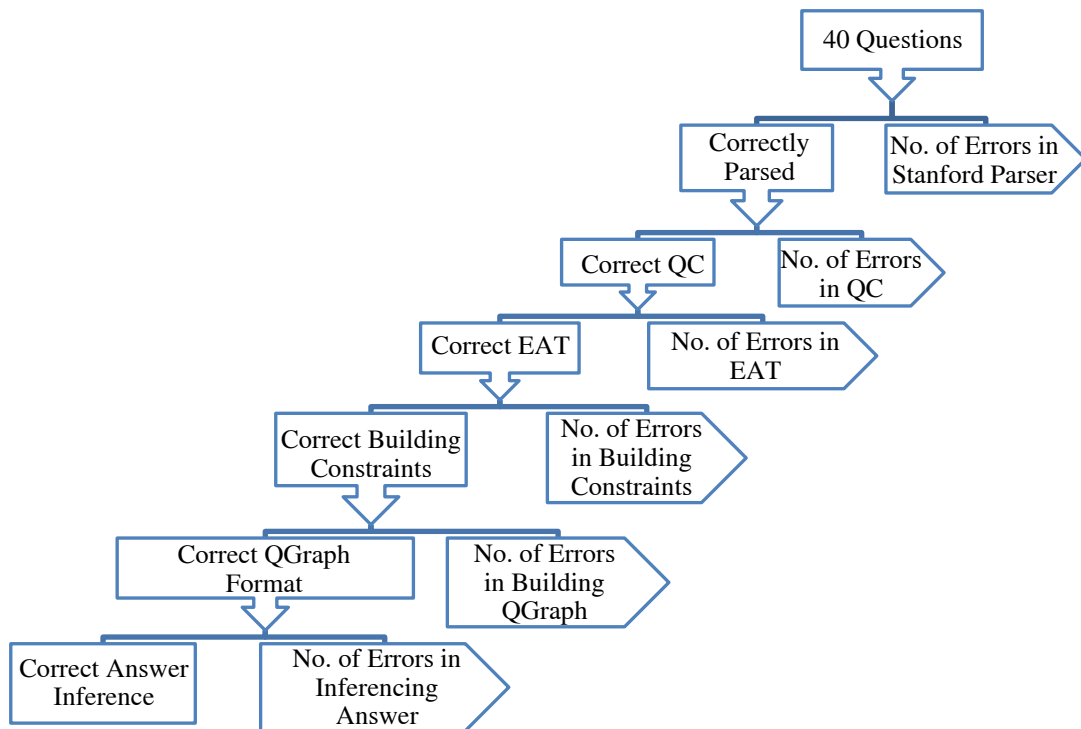


Figure 4.1: Evaluation and error analyzing steps for closed-domain (first scenario)

Regarding the first issue as shown in Figure 4.1, correctly parsed, we wanted to check the preprocessing results, the linear preprocessing information such as morphological analysis (e.g. lemma, pos, ne, etc.) and the dependency parsing results. In the case where the parser is failed to generate correct parsing information

<sup>1</sup> <http://trec.nist.gov/>

then it is not possible to justify the accuracy in next steps results in presented ScoQAS architecture. At the same way the final answer could not been achieved correctly while other steps have not been executed successfully. So we assume negative (No) for such conditions because the ScoQAS basically acts based on NLP preprocessing results.

In the second issue, QC, we analyze that the question classification algorithm is corresponded with the actual classification, i.e., the concept of question is correctly classified as a question type (i.e. QT: semantic-based structure-feature pattern) and the preliminary variable values have been assigned perfectly. Otherwise, it could be seen the technical errors at this stage for associated question. In the next step, EAT issue, we look at the first item of pattern which specifies the type of question (i.e. who, what, where, etc.) and then considering its corresponding tokens, 'tk\_Type', which are responsible to determine the expected answer. Moreover, this module looks for ontology items based on type of question and expands it to bind all of the related ontology concepts according to the QT.

The fourth issue, building constraints, is evaluated based on all of the related predicates, variables and ontology items, which are produced in QC, step according to the Ss-fP pattern. Here it should carefully be checked that all of the preliminary assigned tokens for corresponding pattern have been extracted correctly. In addition, relations between them were determined according to the variable-bind approach. In addition, making sure about the dependencies between pattern items and also the assignment of the ontology items and their associated variables. Each of the pattern items should have their associated ontology items. So let us consider the previous example Q1. The constraints and their predicates are shown in Table 4.2 for Person in QT: Where\_Person\_Action. In this table, we have categorized the bounded variables to associated tokens and the bounded ontology items such as Class, Slot, and Instance columns for Person, respectively. The constraint tk\_PER([3, X1]) has been generated in order to bind token with position 3 ("manager") to variable 'X1' in the QC step because after traversing ontology, class "Manager" has been matched as a subclass of "i\_en\_proper\_Person". The constraint ont\_PER\_0([X1, X5]) is produced after traversing the ontology classes in building constraint step. These dependencies have been exploited from Stanford dependency parsing. The other algorithm attempts to expand the EAT to achieve high accuracy based on the result of the EAT step.

As seen in the Slot column, 5 slots of class "Manager" have been expanded as slot\_PER\_*i* (*i* is a number, which has been produced for each slot as index). The expanding algorithm continues to find instances of the slot\_PER\_*i* and bind new variables to the new constraints as well. The new constraints in form of instance\_PER\_*i* are made, so for *i* = 1, instance\_PER\_1(['Organization manager', 'X12']). The relations between these instances and associated slots are determined by Inst\_*i*. For example, consider the Inst\_1 (['X9', 'X12']) which 'X9' indicates the slot 'manager\_title'. There are slots that their types (ranges) are instance of other class, e.g., "has\_authority". So Instances of this type of Slot have been listed like as "http://protege.stanford.edu/rdfQA-Enterprise\_Instance\_2". In this case, this label as

an Instance could be expanded while we do not address to deal with such labels in the current work.

Table 4.2: Binding ontology items for Person in pattern QT:Where\_Person\_Action

Bound Tokens for Person	Class	Slot	Instance
tk_PER ([ '3', 'X1' ])	class_PER_0 ([ 'Manager', 'X5' ])	slot_PER_0 ([ 'make', 'X6' ])	instance_PER_0 ([ 'Executive director', 'X11' ])
	ont_PER_0 ([ 'X1', 'X5' ])	Slot_0 ([ 'X5', 'X6' ])	Inst_0 ([ 'X9', 'X11' ])
		slot_PER_1 ([ 'control', 'X7' ])	instance_PER_1 ([ 'Organization manager', 'X12' ])
		Slot_1 ([ 'X5', 'X7' ])	Inst_1 ([ 'X9', 'X12' ])
		slot_PER_2 ([ 'has_authority', 'X8' ])	instance_PER_2 ([ 'Chief of security', 'X13' ])
		Slot_2 ([ 'X5', 'X8' ])	Inst_2 ([ 'X9', 'X13' ])
		slot_PER_3 ([ 'manager_title', 'X9' ])	instance_PER_3 ([ 'Chief of Administrative affair', 'X14' ])
		Slot_3 ([ 'X5', 'X9' ])	Inst_3 ([ 'X9', 'X14' ])
		slot_PER_4 ([ 'define', 'X10' ])	instance_PER_4 ([ 'Resources development and logistics vice president', 'X15' ])
		Slot_4 ([ 'X5', 'X10' ])	Inst_4 ([ 'X9', 'X15' ])
			instance_PER_5 ([ 'Coordinating vice president', 'X16' ])
			Inst_5 ([ 'X9', 'X16' ])
			instance_PER_6 ([ 'Financial manager', 'X17' ])
			Inst_6 ([ 'X9', 'X17' ])
			instance_PER_7 ([ 'http://protege.stanford.edu/rdfQA-Enterprise_Instance_2', 'X18' ])
			Inst_7 ([ 'X8', 'X18' ])
			instance_PER_8 ([ 'Education vice president', 'X19' ])
			Inst_8 ([ 'X9', 'X19' ])

			instance_PER_9(['Product manager', 'X20'])
			Inst_9(['X9', 'X20'])
			instance_PER_10(['ITC manager', 'X21'])
			Inst_10(['X9', 'X21'])
			instance_PER_11(['Library manager', 'X22'])
			Inst_11(['X9', 'X22'])
			instance_PER_12(['http://protege.stanford.edu/rdfenterprise_Class70010', 'X23'])
			Inst_12(['X7', 'X23'])
			instance_PER_13(['research and Planning vice president', 'X24'])
			Inst_13(['X9', 'X24'])
			instance_PER_14(['Theoretical education vice president', 'X25'])
			Inst_14(['X9', 'X25'])
			instance_PER_15(['Chief of planning department', 'X26'])
			Inst_15(['X9', 'X26'])
			instance_PER_16(['Sales manager', 'X27'])
			Inst_16(['X9', 'X27'])

The fifth issue, building the QGraph, shows that for QT and its Constraints corresponding to the question whether it is produced a complete graph, which include all of the predicates belongs to QC, EAT, and Constraints stages. Here the QGraph parts such as node index, node name, edge (source index, destination index), edge name, and the other node types such as EAT items should be analyzed. Finally, in the inferencing answer, we consider that after doing heuristic inference over produced QGraph; whether or not the appropriate answer is achieved. All of the mentioned issues have been summarized in Appendix A after running the questions over presented ScoQAS.

We assume that the whole set of questions has an answer in order to measure accuracy. Therefore:

$$\text{Accuracy} = \frac{\text{no. of questions answered correctly}}{\text{no. of processed questions}}$$

Table 4.3: Accuracy of closed-domain questions in ScoQAS

Training Questions	Processed	Correctly Parsed	Correct QC	Correct EAT	Correct Building Constraints	Correct Graph Format	Correct Answer Inference	Answered Correctly
ScoQAS – Closed Domain	40	39	37	31	30	30	29	29

We have processed 40 different types of questions for all of the steps. It is supposed that all the questions have answer in golden system (here ontology). In Table 4.4 is shown the evaluation and the accuracy result after making answer inference. We reach an average recall of 0.85 and precision of 0.85 leading to an F-measure of 0.85. The results obtained from the first training of the ScoQAS indicate that it can be developed to achieve high performance and reasonable results. According to the above accuracy formula, the accuracy will be 0.73 for first scenario.

Table 4.4: Experimental results over Enterprise ontology for closed domain

Closed Domain	Processed	Answer Produced	Correct	Not Correct	Recall	Precision	F-Measure
Enterprise Ontology	40	34	29	11	0.85	0.85	0.85

### 4.3 The Evaluation of Open Domain Approach

In open domain approach, with respect to other semantic-based QA system evaluation benchmarks, we use a series of evaluation campaigns on question answering over linked data (QALD<sup>1</sup>). The QALD challenges provide an up-to-date benchmark for assessing and comparing systems that mediate between users, expressing his or her information need in NL, and RDF data and the researchers which working on QA over Semantic Web data and querying Linked Open Data. The task is to extract correct answers for natural language questions or corresponding keywords from one of the given RDF repositories. Although, there are other tools that used as mapping language which maps RDB to RDF (e.g. R2RML) such as Stardog<sup>2</sup>, OnTop<sup>3</sup> but they are not appropriate for QA system evaluation. We started with 188 selected questions from QALD's training set (QALD-2, QALD-3, and QALD-4 training set) as our training data set in order to complete Rules and Constraints for different types of questions [103]. We could only translate NL questions written in English, while the organizers offer questions in the other languages. We filtered all of those questions, which have solutions and answers in the golden system. We analyzed the open domain ScoQAS based on four dimensions consisting of QC, question Constraints, EAT, and mapping question to formal query (SPARQL). We completed our training questions over 188 questions. Then we

<sup>1</sup> <http://qald.sebastianwalter.org/>

<sup>2</sup> <http://stardog.com>

<sup>3</sup> <http://ontop.inf.unibz.it>

continued to test the ScoQAS over other QALD versions. We started to convert questions in QALD-2, QALD-3, QALD-4, and QALD-5 test set into NIF format for analyzing the accuracy of the system. All training questions are annotated with keywords, corresponding to SPARQL queries and, if indicated, answers retrieved from the provided SPARQL endpoint. Annotations have been provided in the XML format. The overall document is enclosed by a tag that specifies an ID for the dataset indicating the domain and whether it is training or test. Here, rules associated with traversing domain ontology are ignored. Finally, SPARQL templates that correspond to the satisfied rules and the question type have been implemented. The SPARQL query that has been produced based on assigned related terms and question type is executed in OpenLink Virtuoso<sup>1</sup> SPARQL protocol endpoint.

The evaluation is proceeding based on two steps. In the first step, the verification of producing the SPARQL query is evaluated. In the case that the SPARQL query has been generated then we will look to the results of applying this query over KB (LOD). The results of the ScoQAS system on the QALD test sets are summarized in Table 4.5. It presents the questions answered correctly, as well as their individual scores. The column “Size” shows the number of provided questions by every QALD series. The column “Processed” states for how many of the questions have been performed in ScoQAS over the QALD series. The “Answer Produced” indicates that how many of the processed questions have been provided with an answer. The column “Correct” specifies how many of these produced answer questions were answered properly. The remain columns such as “Recall”, “Precision”, and “F-Measure” have been calculated based on the left side’s columns values.

These results obtained by the ScoQAS can be compared to those obtained recent years ago, when participating in the second edition of QALD<sup>2</sup> challenge. We executed ScoQAS system on the above mentioned questions of the QALD’s series (training and test set) benchmarks. After comparing the results of the constructed queries with the results given by the gold standard queries, we reach an average recall of 0.7025 and precision of 0.6075 leading to an F-measure of 0.6254 as illustrated in Table 4.5.

Table 4.5: Evaluation of ScoQAS over QALD benchmark

QALD Series	Size	Processed	Answer Produced	Correct	Recall	Precision	F-Measure
QALD-2 Test Set	99	99	81	35	0.82	0.43	0.5642
QALD-3 Test Set	99	80	37	26	0.46	0.70	0.5552
QALD-4 Test Set	50	50	36	28	0.72	0.78	0.7488
QALD-5 Test Set	60	59	48	25	0.81	0.52	0.6334
<b>ScoQAS Average</b>	<b>308</b>	<b>288</b>	<b>202</b>	<b>114</b>	<b>0.7025</b>	<b>0.6075</b>	<b>0.6254</b>

<sup>1</sup> <http://virtouso.openlinksw.com>

<sup>2</sup> <http://qald.sebastianwalter.org/index.php?x=home&q=2>

In Table 4.6, Table 4.7, Table 4.8, and Table 4.9 the results in QALD competitions are shown which includes winner of the competition, average, and median of each track. The evaluation method was defined by the challenge organizers every year. It consists in calculating, for each test query, the precision, the recall and the F-measure of the SPARQL translation returned by the systems, compared with handmade queries of a gold standard document. In Table 4.6, we summarize the results from the second of QALD evaluation campaigns [104]. For QALD-2, both of the training and test sets in QALD-1 (a set of 50 training and 50 test questions) have been combined to build a new training set, provided together with a newly created test set, leading to 100 training and 100 test questions for DBpedia, and 100 training and 50 test questions for MusicBrainz.

Table 4.6: QALD-2 competitions results

QALD-2	Participating systems	Recall	Precision	F-Measure
Winner	SemSeK	0.48	0.44	0.46
Median	QAKiS	0.37	0.39	0.38
	MHE	0.4	0.36	0.38
Average	-	0.43	0.40	0.42

Table 4.7 shows the results achieved in QALD-3 for DBpedia test set [105]. The QALD-3 was the first contest including multilingual tasks. Three datasets were provided:

- English DBpedia 3.8<sup>1</sup>
- Spanish DBpedia<sup>2</sup>
- MusicBrainz<sup>3</sup>

Although the main focus of the challenge in QALD-3 has been on multilingualism but all participating systems worked on English data only. This shows that the multilingual scenario was not broadly addressed in this track. In this competition, the participants have not been involved in many aspects of linked data such as biomedical or clinical QA systems.

Table 4.7: QALD-3 competitions results for DBpedia test set

QALD-3	Participating systems	Recall	Precision	F-Measure
Winner	Squall2sparql	0.88	0.93	0.90 <sup>4</sup>
Median	CASIA	0.36	0.35	0.36
Average	-	0.40	0.40	0.40

<sup>1</sup> <http://dbpedia.org>

<sup>2</sup> <http://es.dbpedia.org>

<sup>3</sup> <http://musicbrainz.org>

<sup>4</sup> This extremely high figure is due to the inclusion of a manual step in the process.

Table 4.8 report on the results obtained by the participating systems on Tasks 1, multilingual question answering over DBpedia, in QALD-4 [106].

Table 4.8: QALD-4 competitions results in multilingual QA over DBpedia

QALD-4	Participating systems	Recall	Precision	F-Measure
Winner	Xser	0.71	0.72	0.72
Median	CASIA	0.40	0.32	0.36
Average	-	0.35	0.32	0.34

QALD-5 questions were compiled from the QALD-4 training and test questions, slightly modified in order to account for changes in the DBpedia dataset [107]. In the case of hybrid questions they were also building on the data provided by the INEX Linked Data track<sup>1</sup>. Later, systems were evaluated on 60 different test questions, comprising 50 multilingual ones and 10 hybrid ones. The participating systems were seven teams while two of them submitted results only for the multilingual questions, two participants presented the results only for the hybrid questions. Of these, three applicants participated in both kinds of questions. In Table 4.9 we show the brief report of the participating systems on the 50 multilingual questions.

Table 4.9: QALD-5 competitions results for multilingual QA

QALD-5	Participating systems	Recall	Precision	F-Measure
Winner	Xser	0.72	0.74	0.73
Median	QAnswer	0.35	0.46	0.40
Average	-	0.42	0.44	0.43

As seen, our results on QALD-2 and QALD-3 are higher than median. Although our result on QALD-2 is over the winner. In the measuring of test set for QALD-4, obviously, accuracy is over the median. Compared to the recent competition, the QALD-5, the result of ScoQAS can be seen as above the average. It is worth mentioning that ScoQAS tries to care both closed-domain and LOD-based domain while participants system(s) in the QALD challenges are specially designed for the task. So we can conclude that our results on this task are considerable.

The ScoQAS is compared to the gold standard with respect to the precision and recall achieved for QALD series. It is worth noting that a comparison with systems participating in the challenges has to be considered with care because the QA systems participating in QALD challenges are specially designed to this task while the ScoQAS aims to face both restricted-domain and open-domain scenarios using mostly the same components. The winner F-measure and median F-measure results

<sup>1</sup> <http://inex.mmci.uni-saarland.de/tracks/dc/index.html>



obtained for each of the QALD series which is shown in Table 4.10 are used as a baseline for our experiments. Our results in ScoQAS on QALD series is shown in the column “ScoQAS F-Measure” and its average in the last row “ScoQAS Average” can easily be compared with others. As can be seen in Table 4.10, our results on QALD-2 are upper than median and also upper than top of the participants. In QALD-3, the ScoQAS F-Measure=0.5193 are clearly over the median and lower than winner with F-Measure = 0.90. In QALD-4 is seen a considerable results that the ScoQAS reaches to F-Measure=0.7488 upper that top participant. Finally, for QALD-5 the F-Measure value is decreased in comparison with QALD-4 and located among the median and top values.

With regard to this aspect, we can conclude that the results in this scenario are remarkable.

Table 4.10: The summary of the QALD competitions results in F-Measure

QALD	Median F-Measure	Top F-Measure	ScoQAS F-Measure
QALD-2	0.38	0.46	0.5642
QALD-3	0.36	0.90	0.5552
QALD-4	0.36	0.72	0.7488
QALD-5	0.40	0.73	0.6334
<b>ScoQAS Average</b>	-		<b>0.6254</b>

#### 4.4 Error Analysis

The cases in which the ScoQAS system fails to construct an appropriate query can be pinned down to reasons that can roughly be categorized as the *internal built-in error* and *external failure*.

By the *internal built-in error*, we mean questions that could in principle be analyzed and answered, but for one reason or the other, the system fails to do so. There are mainly three reasons for such failure:

- The first reason is lack of definition of patterns in QC module, e.g., “How many companies were founded in the same year as Google?” for this example, the appropriate pattern “Howmany\_Action\_TimeRelation” has not been defined as a QT to classify the question.
- The second reason for the failure in specifying EAT for combined temporal and comparative question. There are cases where the meaning of the whole question is related to the composition of the temporal and comparative of the question. This involves components that do not contribute anything to the overall meaning (e.g. Does the new Battlestar Galactica series have more episodes than the old one?).
- The third reason is due to the incomplete coverage of building the QSiS. Example of question that cannot be completed for this reason is “Give me

all actors starring in movies directed by and starring William Shatner?”, or also in some specific cases of counting with respect to a restriction (e.g. Who is the Formula 1 race driver with the most races?).

The *External failures* mean failures for which ScoQAS system has not responsibility. Some of the errors are due to incorrect parsing by Stanford CoreNLP, ill-formed questions, or natural language resources used in the ScoQAS, namely the access to WordNet, to assess their impact on the overall answer accuracy. Let us consider this example, “Through which countries does the Yenisei river flow?” As shown in Figure 4.2, the token “flow” has been recognized as a direct object (dobj) for the verb “does” in the Stanford dependency parsing. Here, the POS of the word “flow” is noun while, in fact, it is an auxiliary verb for “does”. Therefore, within this dependency parsing results, the system could not classify this question correctly as QT: “Where\_Properties\_GEO\_Action”.

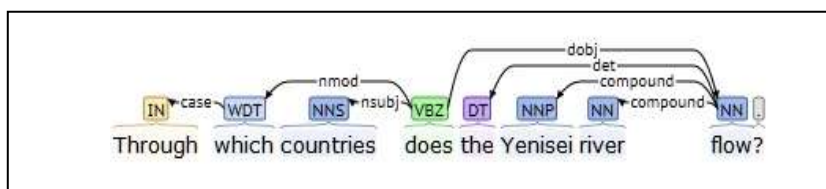


Figure 4.2: External error in Stanford parsing over question

The ill-formed questions comprise two types that are:

1. Syntactically ill-formed questions: questions that are incomplete or ungrammatical and therefore are not parsed (e.g., what are the capital city in Iran?)
2. Semantically ill-formed questions: questions that violate restrictions (e.g. which states border the Missouri river?)

Continuing on, we analyze the results achieved by ScoQAS running on QALD series test sets. The results of ScoQAS system are summarized in comparison with golden systems in Table 4.11, Table 4.12, and Table 4.13.

Let us consider Table 4.11 after applying ScoQAS system over QALD-2 test set. As can be seen, there are three True Negative (TN) questions and not False Positive (FP) question among 99 questions (see turquoise color rows in Appendix D). Here we can refer to the *internal built-in errors*, e.g., the question “What is the melting point of copper?” has been classified as “Where\_CompoundProperties”. The achieved EAT indicates to the location while the real EAT should be a quantifier (e.g., melting point). Another notable aspect is the high value of the False Negative (e.g., FN=61). The main reason for being the high FN was due to the initial system implementation with a limited number of the Ss-fp rules. For almost twenty types of questions, we did not manually define rules in the ScoQAS system. Hence, in the first stage, the system has been failed in the QC task.

Table 4.11: Results over QALD-2 Test Set

		ScoQAS System results	
		Positive	Negative
Golden System	True	TP = 35	FN = 61
	False	FP = 0	TN = 3

Table 4.12 shows the results for QALD-3 test set. As shown, there is only one False Positive (FP) result, which in golden system is not the answer but ScoQAS finds the answer, e.g. “Who invented the zipper?” In this example, all of the steps are done accurately, i.e., the QC, Constraints, EAT, and mapping to SPARQL return the correct values while in the QALD-3 was not defined an equivalent SPARQL query.

There are two True Negative (TN) questions, which ScoQAS system could not process correctly in order to produce corresponding SPARQL queries. These two TN questions are categorized in the *internal built-in error*, e.g., the question “What is the average temperature on Hawaii?” is not completed with its constraints while it has been categorized as “What\_GEO”. The remarkable high value of the False Negative (e.g., FN=51) is considered. This was the first test set that we used in ScoQAS and after analysing we found that the most errors stem from low eccuracy of Stanford dependency parser to deal with list questions such as “Give me, List all, etc.”. This issue has been solved in the recent version in this parser for this type of questions. Some other fails return to the lack of defining and implementing the enough Ss-fP rules to coverage the most similar question types.

Table 4.12: Results over QALD-3 Test Set

		ScoQAS System results	
		Positive	Negative
Golden System	True	TP = 26	FN = 51
	False	FP = 1	TN = 2

In QALD-4 test set, we analyzed 50 questions. Table 4.13 shows the results for this set of questions. There are two True Negative (TN) questions, which also ScoQAS system could not process correctly them. One of them is “How many gold medals did Michael Phelps win at the 2008 Olympics?” and the other is “In which studio did the Beatles record their first album?” which both of them are categorized in the *internal built-in error*. The former is not completed with its constraints, and the latter was not mapped to the appropriate SPARQL query.

Table 4.13: Results over QALD-4 Test Set

		ScoQAS System results	
		Positive	Negative
Golden System	True	TP = 28	FN = 20
	False	FP = 0	TN = 2

Table 4.14 shows the results of analyzing 59 test set questions in QALD-5 (see Appendix G). As shown, there is only one True Negative (TN) question. The question is “In which city were the parents of Che Guevara born?” which the *internal built-in error* caused. Two QTs have been produced for this question: “Where\_Person\_Action” and “Where\_CompoundProperties\_Action”. Although in both of the QTs, the EAT is the same and tk\_Type=2 (city) but it was not mapped to the proper SPARQL query to retrieve the correct answer.

Table 4.14: Results over QALD-5 Test Set

		ScoQAS System results	
		Positive	Negative
Golden System	True	TP = 25	FN = 33
	False	FP = 0	TN = 1

---

## **5 Conclusions and Future Work**

---

### **5.1 Conclusions**

The goal of this thesis research was to develop a semantic QA system by constraining the search space for the answer beyond simple factoid questions using NLP tools. The aim is obtaining a system able to perform both an open domain and closed domain scenarios, taking part of the late of the ontology defining the domain. For this purpose, in this dissertation we developed ScoQAS, which performs over ontologies and operates on two scenarios. The purpose of these scenarios was to sketching out the specific conduct in two types of domains with the aim of consolidating the pros and cons for designing and implementing the integrated approach. Thus, considering this purpose led to demonstrate the adaptabilities of our approach using common components to achieve efficiency and robustness. Four research questions were discussed throughout this thesis. In this chapter, they are revisited and briefly discussed to highlight the contributions made in this work. The research questions were as follows:

1. How accurately can we make the semantic QA system technology more coherent infrastructure and hopefully to be able to move it a step closer to a full-fledged QA system?
2. How to provide a set of semantically motivated question types able to be used without launchment to both restricted domain and open domain settings.
3. How can we formulate and generalize the constraints of the complex questions in order to find the syntactic and semantic relations in factoid and non-factoid questions?
4. What kind of structure can be modeled to facilitate the question interpretation in order to find inference mechanism to extract answer(s)?

We presented five technical contributions that above mentioned research questions were targeted by these contributions. In order to answer the research question No. 1, a brief description is presented in the first scientific contribution in the introduction chapter (i.e. Section 1.7). With respect to the lack of standardized architecture in semantic QA, the developed ScoQAS architecture has been described with more details in Chapter 3. It was empirically and theoretically enhanced to overwhelmed the weaknesses raised from different semantic QA models. Our initial model [36] was developed in order to complete and implement the real semantic QA system based on closed and open domains. The ScoQAS has been presented, discussed and evaluated in [90]. The result of empirical evaluation shows the effectiveness, robustness as well as scalability of ScoQAS architecture with applied approach.

Question No. 2 is exhaustively addressed in Section 3.4 where we presented an empirical approach to classifying questions by defining the semantic-based structure-feature patterns (Ss-fP). The Ss-fP consists of features that are extracted by hand-crafted rules and then assigned to the associated QT. We have used state-of-the-art tools in preprocessing step to exploit them in classifying question in both closed and open domains. A list of rules was built accordingly to the QT. The rule-based classification includes semantic aspects within the tag where each of these QTs can cover a large number of questions with the similar grammatical structure. It also can be instantiated with different expression types looking for the same answer(s). In this way, the same tag set can be used for different scenario.

In order to answer to question No. 3, it has been presented an innovative and practical way to solve the problem of generalizing the constraints of the complex questions in Section 3.6.2. In this section, we describe how constraints are generated based on information provided to this step by upstream process. This approach leads to be able to identify the appropriate answers. The constraints are simply relations that can hold between the related terms. We have developed the QSiS that is compatible to constructing such constraints which provides more information about the nature of question and leads to analyze the question semantically. The QSiS is the main block of knowledge to obtain the answer. Constraints are the principal part of the QSiS, which is used in downstream steps.

Our approach to answer to question No. 4 was presenting and implementing the QGraph that is generated after automatic processing and analysis of question in the first scenario. We have explained with details in Section 3.6. The QGraph is used both as a search space for locating the answer and as a resource to exploit it for enriching the constraints sets and EATs. We have introduced an algorithm that demonstrates how we generate the QGraph. Thus, as a consequence of this, another algorithm was developed to search nodes and edges over produced QGraph which consists of semantic information for question, and hence this algorithm shows how we can infer the final answer. In front of this, in second scenario, we presented mapping approach to interpret the question in Section 3.7. Here, the goal is to construct queries (i.e. SPARQL queries) for a given set of constraints in QSiS to crawl in LOD resources.

This dissertation focused on presenting an automatic empirical approach to solve some problems in semantic-based closed and open domains QA System. My research attempts to consolidate the foundations of the bridge between NLP and Semantic Web technologies so that each of these runs its own pace. We employed NLP techniques, specific pattern combined with rule-based (e.g. Ss-fP), developing syntactic-semantic structure of question (e.g. QSiS) and graph-based representation (e.g. QGraph) to interpret questions and make inference (or map to formal query in the second scenario). The applications of these state-of-the-art artificial intelligence techniques have several advantages, which fulfill the interpretation and conversion action. This action converts questions into a form of semantic and lexical structure or a formal semantic query automatically. On the one hand, in the first scenario, NLP techniques are used to preprocess the question by using Stanford CoreNLP parser and then building the QSiS. By creating QSiS, it is possible to generate a question graph that demonstrates the nature of the question and its relation to the expected answers in a well-formed model, which facilitates the process of inferring the final answer. On the other hand, in the second scenario, the presented approach provides a convenient method that map the corresponding QT (e.g. Ss-fP) with generated constraints into the formal query template (e.g. SPARQL). This approach provides the comprehensive information in order to crawl in the LOD resources to find the answers.

The result of the evaluation confirms the feasibility and remarkable accuracy of this model. In this thesis, we showed that although the ScoQAS is in its infancy, it has appropriate potential for development within the framework of scientific and technical standards.

## **5.2 Future Work**

Due to the complexity of semantic QA systems, in order to align and integrate the various approaches, in our future work, we intend to focus on ways to accelerate the development of ScoQAS that further facilitates adaptation for different domains. The preliminary evaluation results show that despite the significant development and technical basis of the ScoQAS architecture, it has to be enhanced in order to increase the accuracy. The future work would be applying the ScoQAS in other domains so that by improving the potential capacities of components to become realistic and applicable. This will improve the flexibility and stability of this system and become consistent in the generalization process to take the necessary steps in the progress of the standardization. It is important to continuously improve the components and tools by generalizing the algorithms. This is an advantage of the scalability, compatibility, and reusability requirements that we imposed on the tools and components.

In the following, we present some open problems and viewpoints for future works:

- The most frequent questions in our case study are the single-relation questions that can be formulated by classifying, providing QSiS, generating QGraph in order to do answer inference. The more complex

questions like as anaphora relation or multi-relation questions remain in ongoing investigation.

- The research findings show that, using statistical techniques in NLP is promising particularly in terms of recall. The future work is open to apply statistical features in some of the processes, e.g. question classification, which the QC module satisfies more than one QTs for some questions (when more than one rules matches) in order to increase the accuracy and efficiency of the ScoQAS.
- A possible future research direction is to make use of various languages (Multilanguage) other than English to use and evaluate the flexibility of the QSiS and QGraph format to infer the answer or even to map SPARQL query format. Hence, multilingualism remains to develop in the future effort.
- For those Slots in domain ontology that its type (its range) is instance of other domain concept. Labels, in the case of concepts, often catch the meaning of the semantic entity, while in relations its meaning is given by the type of its domain and its range rather than by its name. From a technical point of view, rationality behind this is that expanding relations based on its labels or names is more difficult to traverse such concepts based on Slot labels.
- The answer selection is a complex mechanism involving several layers of filtering and ranking functions in second scenario, which could be addressed, with more details.
- In order to get more insights on the robustness of our approach we want to define another scenario, within the domain-restricted framework, corresponding to a domain for which golden datasets exist. The medical domain can be an excellent choice for this task.
- Finally, in this research the temporal cost of obtaining the answer has not been considered and has not been evaluated. A nice way of including this issue could be to focus on a more real scenario. LiveQA<sup>1</sup> challenges provide such scenario where questions are extracted from a real QA site and answer has to be produced in less than one minute. Another advantage of this setting is that questions correspond to a large set of popular topics.

---

<sup>1</sup> <https://sites.google.com/site/trecliveqa2017/home>



## Bibliography

---

- [1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - A crystallization point for the Web of Data," *J. Web Semant.*, vol. 7, no. 3, pp. 154–165, 2009.
- [2] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *IJSWIS*, vol. 5, no. 3, pp. 1–22, 2009.
- [3] A. Ben Abacha and P. Zweigenbaum, "MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies," *Inf. Process. & Manag.*, vol. 51, no. 5, Sep. 2015.
- [4] C. Unger, C. Forascu, V. López, A.-C. N. Ngomo, E. Cabrio, P. Cimiano, S. Walter, C. Forescu, V. Lopez, A.-C. N. Ngomo, E. Cabrio, P. Cimiano, and S. Walter, "Question Answering over Linked Data (QALD-5)," 2015.
- [5] M. S. Fabian, K. Gjergji, and W. Gerhard, "YAGO: A core of semantic knowledge unifying wordnet and wikipedia," *16th Int. World Wide Web Conf. ....*, pp. 697–706, 2007.
- [6] L. Hirschman and R. Gaizauskas, "Natural language question answering: the view from here," *Nat. Lang. Eng.*, vol. 7, no. 4. pp. 275–300, 2001.
- [7] A. Bordes, N. Usunier, S. Chopra, and J. Weston, "Large-scale Simple Question Answering with Memory Networks," Jun. 2015.
- [8] A. Bordes, S. Chopra, and J. Weston, "Question Answering with Subgraph Embeddings," Jun. 2014.
- [9] A. Bordes, J. Weston, and N. Usunier, "Open Question Answering with Weakly Supervised Embedding Models," Springer, Berlin, Heidelberg, 2014, pp. 165–180.
- [10] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Learning to Compose Neural Networks for Question Answering," pp. 1545–1554, 2016.
- [11] M. T. Maybury and M. Pa, "New Directions in Question Answering," *Comput. Linguist.*, vol. 9, no. page 83, pp. 383–386, 2004.
- [12] V. Tablan, D. Damljanovic, and K. Bontcheva, "A natural language query interface to structured information," in *LNCS*, 2008, vol. 5021 LNCS, pp. 361–375.
- [13] C. Shah, S. Oh, and J. S. Oh, "Research agenda for social Q&A," *Library and Information Science Research*, vol. 31, no. 4. pp. 205–209, 2009.
- [14] J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C. Y. Lin, S. Maiorano, G. Miller, and Et Al., "Issues , Tasks and Program Structures to Roadmap Research in Question & Answering ( Q & A )," *New York*, pp. 1–35, 2001.
- [15] X. Li and D. Roth, "Learning question classifiers," in *Proc. 19th Int. Conf. Comput. Linguist. -*, 2002, vol. 1, pp. 1–7.
- [16] R. Perera, "IPedagogy: Question answering system based on web information clustering," in *Proc. - 2012 IEEE 4th Int. Conf. Technol. Educ. T4E 2012*, 2012, pp. 245–246.

- [17] Ó. Ferrández, R. Izquierdo, S. Ferrández, and J. L. Vicedo, “Addressing ontology-based question answering with collections of user queries,” *IPM*, vol. 45, no. 2, pp. 175–188, 2009.
- [18] Ó. Ferrández, “Textual entailment recognition and its applicability in NLP tasks,” [Sociedad Española para el Procesamiento del Lenguaje Natural], 2009.
- [19] A. Hallili, E. Cabrio, and C. F. Zucker, “QALM: a benchmark for question answering over linked merchant websites data,” in *Proceedings of the 2014 International Conference on Posters & Demonstrations Track - Volume 1272*, 2014, pp. 389–392.
- [20] E. Cabrio, C. F. Zucker, F. Gandon, A. Hallili, and A. Tettamanzi, “Answering N-Relation Natural Language Questions in the Commercial Domain,” in *2015 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, 2015, pp. 169–172.
- [21] K. Zhang, W. Wu, H. Wu, Z. Li, and M. Zhou, “Question Retrieval with High Quality Answers in Community Question Answering,” in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manag. - CIKM '14*, 2014, pp. 371–380.
- [22] Y. El Adlouni, I. Lahbari, H. Rodríguez, M. Meknassi, S. Ouatik, E. Alaoui, and N. Ennahahi, “UPC-USMBA at SemEval-2017 Task 3: Combining Multiple Approaches for CQA for Arabic,” in *Proc. 11th Int. Work. Semant. Eval.*, 2017, pp. 275–279.
- [23] P. Nakov, L. Arquez, A. Moschitti, W. Magdy, H. M. Abed, A. Freihat, J. Glass, B. Randeree, and Q. Living, “SemEval-2016 Task 3: Community Question Answering,” in *Proceedings of SemEval-2016*, 2016, pp. 525–545.
- [24] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun, “Finding question-answer pairs from online forums,” in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '08*, 2008, p. 467.
- [25] X. Xue, J. Jeon, and W. B. Croft, “Retrieval models for question and answer archives,” in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08*, 2008, p. 475.
- [26] C. Unger and P. Cimiano, “Pythia: Compositional meaning construction for ontology-based question answering on the semantic web,” in *NLDB*, 2011, pp. 153–160.
- [27] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, “Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks.” 2015.
- [28] V. López, V. Uren, M. Sabou, and E. Motta, “Is Question Answering fit for the Semantic Web?: A survey,” *Semant. Web*, vol. 2, no. 2, pp. 125–155, Jan. 2011.
- [29] I. Androustopoulos, G. D. Ritchie, P. Thanisch, B. W. Ballard, D. E. Stumberger, Unknown, B. W. Ballard, B. W. Ballard, J. C. Lusth, N. L. Tinkham, R. J. Bobrow, P. Resnik, R. M. Weischedel, R. A. Capindale, R. G. Crawford, J. G. Carbonell, S. Ceri, G. Gottlob, L. Tanca, S. Ceri, G. Gottlob, G. Wiederhold, J. Clifford, J. Clifford, D. S. Warren, E. F. Codd, F. J. Damerou, S. M. Dekleva, D. R. Dowty, R. E. Wall, S. Peters, D. R. Dowty, R. E. Wall, S. Peters, S. S. Epstein, J. M. Ginsparg, B. J. Grosz, B. J. Grosz, D. E. Appelt, P. A. Martin, F. C. N. Pereira, G. Guida, C. Tasso, Unknown, C. D. Hafner, C. D. Hafner, C. D. Hafner, K. Godden, L. R. Harris, G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, J. Slocum, G. Hirst, M. Jarke, J. A. Tuner,

- E. A. Stohr, Y. Vassiliou, N. H. White, K. Michielsen, S. J. Kaplan, N. Ott, R. J. H. Scha, M. Templeton, J. Burger, H. R. Tennant, K. M. Ross, R. M. Saenz, C. W. Thompson, J. R. Miller, H. R. Tennant, K. M. Ross, C. W. Thompson, B. H. Thompson, F. B. Thompson, B. H. Thompson, F. B. Thompson, F. B. Thompson, B. H. Thompson, D. L. Waltz, R. M. Weischedel, G. Whittemore, K. Ferrara, and H. Brunner, "Natural language interfaces to databases – an introduction," *Nat. Lang. Eng.*, vol. 1, no. 1, pp. 29–81, Mar. 1995.
- [30] W. Tunstall-Pedoe, "True Knowledge: Open-Domain Question Answering Using Structured Knowledge and Inference," *AI Mag.*, vol. 31, no. 3, pp. 80–92, Jul-2010.
- [31] P. Forner, D. Giampiccolo, and B. Magnini, "Evaluating multilingual question answering systems at CLEF," *Target*, pp. 2774–2781, 2010.
- [32] E. Kaufmann and A. Bernstein, "How useful are natural language interfaces to the semantic Web for casual end-users?," in *LNCS*, 2007, vol. 4825 LNCS, pp. 281–294.
- [33] D. Mollá and J. L. Vicedo, "Question Answering in Restricted Domains: An Overview," *Comput. Linguist.*, vol. 33, no. 1, pp. 41–61, 2007.
- [34] V. Uren, Y. Lei, V. Lopez, H. Liu, E. Motta, and M. Giordanino, "The usability of semantic search tools: a review," *Knowl. Eng. Rev.*, vol. 22, no. 4, pp. 361–377, Dec. 2007.
- [35] K. Höffner, S. Walter, E. Marx, R. Usbeck, J. Lehmann, and A.-C. Ngonga Ngomo, "Survey on Challenges of Question Answering in the Semantic Web," *Semant. Web*, vol. 0, pp. 1–26, 2016.
- [36] M. Latifi and M. Sánchez-Marrè, "The Use of NLP Interchange Format for Question Answering in Organizations," in *IOS Press, Frontiers in Artificial Intelligence and Applications*, 2013, pp. 235–244.
- [37] V. López, V. Uren, E. Motta, and M. Pasin, "AquaLog: An ontology-driven question answering system for organizational semantic intranets," *Web Semant.*, vol. 5, no. 2, pp. 72–105, 2007.
- [38] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, "Building Watson: An Overview of the DeepQA Project," *AI Mag.*, vol. 31, no. 3, pp. 59–79, 2010.
- [39] A. Kalyanpur, B. K. Boguraev, S. Patwardhan, J. W. Murdock, A. Lally, C. Welty, J. M. Prager, B. Coppola, A. Fokoue-Nkoutche, L. Zhang, Y. Pan, and Z. M. Qiu, "Structured data and inference in DeepQA," *IBM J. Res. Dev.*, vol. 56, no. 3.4, pp. 351–364, 2012.
- [40] E. Cabrio, J. Cojan, A. Palmero Aprosio, B. Magnini, A. Lavelli, and F. Gandon, "QAKiS: an open domain QA system based on relational patterns," in *Int. Conf. Posters Demonstr. Track-Volume 914*, 2012, pp. 9–12.
- [41] S. Shekarpour, E. Marx, A. C. Ngonga Ngomo, and S. Auer, "SINA: Semantic interpretation of user queries for question answering on interlinked data," *Journal of Web Semantics*, 2013.
- [42] S. Mithun, L. Kosseim, and V. Haarslev, "Resolving quantifier and number restriction to question OWL ontologies," in *3rd SKG 2007*, 2007, pp. 218–223.
- [43] C. Wang, M. Xiong, Q. Zhou, and Y. Yu, "PANTO: A Portable Natural Language

- Interface to Ontologies,” *Eswc*, vol. 4519, pp. 473–487, 2007.
- [44] R. Wilensky, D. N. Chin, M. Luria, J. Martin, J. Mayfield, and D. Wu, “The berkeley UNIX consultant project,” *Comput. Linguist.*, vol. 14, no. 4, pp. 35–84, Dec. 1988.
- [45] O. Herzog, J. H. Siekmann, and C. R. Rollinger, “Text Understanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence - Final Report on the LILOG-Project,” Sep. 1991.
- [46] D. Warren and F. Pereira, “An efficient easily adaptable system for interpreting natural language queries,” *Comput. Linguist.*, vol. 8, no. 3, 1982.
- [47] C. D. Hafner, “Interaction of knowledge sources in a portable natural language interface,” in *Proc. 22nd Annu. Meet. Assoc. Comput. Linguist.* -, 1984, pp. 57–60.
- [48] M. Templeton and J. Burger, “Problems in natural-language interface to DBMS with examples from EUFID,” in *Proc. first Conf. Appl. Nat. Lang. Process.* -, 1983, p. 3.
- [49] B. W. Ballard, “The syntax and semantics of user-defined modifiers in a transportable natural language processor,” in *Proceedings of the 10th international conference on Computational linguistics* -, 1984, pp. 52–56.
- [50] F. J. Damerau, “Operating statistics for the transformational question answering system,” *Comput. Linguist.*, vol. 7, no. 1, pp. 30–42, Jan. 1981.
- [51] B. W. Ballard and D. E. Stumberger, “Semantic acquisition in TELI,” in *Proc. 24th Annu. Meet. Assoc. Comput. Linguist.* -, 1986, pp. 20–29.
- [52] R. Cooper, D. Crouch, and J. Van Eijck, “Using the framework,” *FraCaS ...*, 1996.
- [53] J. Allan, B. Croft, A. Moffat, and M. Sanderson, “Frontiers, Challenges, and Opportunities for Information Retrieval: Report from SWIRL 2012 the Second Strategic Workshop on Information Retrieval in Lorne,” *SIGIR Forum*, vol. 46, no. 1, pp. 2–32, 2012.
- [54] P. R. Comas i Umbert, “Factoid question answering for spoken documents,” Universitat Politècnica de Catalunya (UPC), 2012.
- [55] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, “YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia,” *Artif. Intell.*, vol. 194, pp. 28–61, Jan. 2013.
- [56] S. Hakimov, H. Tunc, M. Akimaliev, and E. Dogdu, “Semantic question answering system over linked data using relational patterns,” in *Proc. Jt. EDBT/ICDT 2013 Work. - EDBT '13*, 2013, pp. 83–88.
- [57] K. Shearer and K. (COAR) Müller, “COAR » 7 things you should know about...Linked Data.” 2014.
- [58] C. ( U. of S. Gutteridge, “Linked Data Basics for Techies - OpenOrg.” 2015.
- [59] T. Berners-Lee, “Linked Data - Design Issues,” *W3C*. 2006.
- [60] M. C. Daconta, L. J. O. Smith, and K. T., “The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management,” *John Wiley Sons*. 2003.
- [61] R. F. Simmons, “Natural language question-answering systems: 1969,” *Commun. ACM*, vol. 13, no. 1, pp. 15–30, Jan. 1970.
- [62] R. F. Simmons, “Answering English questions by computer: a survey,” *Commun.*




- ACM*, vol. 8, no. 1, pp. 53–70, Jan. 1965.
- [63] M. W. Volker Haarslev, Ralf Möller, “Querying the semantic web with Racer + nRQL,” 2004.
- [64] Y. Li, H. Yang, and H. Jagadish, “NaLIX: an interactive natural language interface for querying XML,” *Proc. 2005 ACM SIGMOD ...*, pp. 900–902, 2005.
- [65] N. Vitucci, M. A. Neri, R. Tedesco, and G. Gini, “Semanticizing syntactic patterns in NLP processing using SPARQL-DL queries,” in *OWLED: CEUR-WS.org*, 2012.
- [66] V. López, M. Fernández, E. Motta, and N. Stieler, “PowerAqua: Supporting users in querying and exploring the Semantic Web,” *Semant. Web*, vol. 3, no. 3, pp. 249–265, 2012.
- [67] T. Finin, P. Reddivari, R. S. Cost, and J. Sachs, “Swoogle : A Search and Metadata Engine for the Semantic Web,” in *ACM conference on Information and knowledge management*, 2004, pp. 652–659.
- [68] M. d’Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, and D. Guidi, “Toward a New Generation of Semantic Web Applications,” *IEEE Intell. Syst.*, vol. 23, no. 3, 2008.
- [69] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker, “Searching and browsing Linked Data with SWSE: The Semantic Web Search Engine,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 9, no. 4, pp. 365–401, 2011.
- [70] A. Funk, V. Tablan, K. Bontcheva, H. Cunningham, B. Davis, and S. Handschuh, “CLOnE: Controlled language for ontology editing,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007, vol. 4825 LNCS, pp. 142–155.
- [71] D. Damljjanovic, M. Agatonovic, and H. Cunningham, “Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction,” in *LNCS*, 2010, vol. 6088 LNCS, no. PART 1, pp. 106–120.
- [72] A. M. Moussa and R. F. Abdel-kader, “QASYO: A Question Answering System for YAGO Ontology,” *Int. J. Database Theory Appl.*, vol. 4, no. 2, pp. 99–112, 2011.
- [73] Y. Schabes, “Mathematical and Computational Aspects of Lexicalized Grammars,” 1991.
- [74] P. Cimiano, “Flexible semantic composition with DUDES,” pp. 272–276, Jan. 2009.
- [75] P. Cimiano, P. Buitelaar, J. McCrae, and M. Sintek, “LexInfo: A declarative model for the lexicon-ontology interface,” *J. Web Semant.*, vol. 9, no. 1, pp. 29–51, 2011.
- [76] J. Lehmann, T. Furche, G. Grasso, A.-C. N. Ngomo, C. Schallhart, A. Sellers, C. Unger, L. Bühmann, D. Gerber, K. Höffner, D. Liu, and S. Auer, “DEQA: Deep web extraction for question answering,” in *ISWC 2012*, 2012, vol. 7650, pp. 131–147.
- [77] T. Furche, G. Gottlob, G. Grasso, C. Schallhart, and A. J. Sellers, “OXPath: A Language for Scalable, Memory-efficient Data Extraction from Web Applications,” *Proceedings of the VLDB Endowment 4.11*, 2011. .
- [78] T. Furche, G. Gottlob, G. Grasso, C. Schallhart, and A. Sellers, “OXPath: A language for scalable data extraction, automation, and crawling on the deep web,” *VLDB J.*, vol. 22, no. 1, pp. 47–72, 2013.

- [79] A.-C. Ngonga Ngomo and S. Auer, "LIMES - A time-efficient approach for large-scale link discovery on the web of data," in *IJCAI Int. Jt. Conf. Artif. Intell.*, 2011, pp. 2312–2317.
- [80] C. Unger and L. Bühmann, "Template-based question answering over RDF data," *Proc. 21st Int. Conf. World Wide Web*, pp. 639–648, 2012.
- [81] S. Kalaivani and K. Duraiswamy, "Comparison of Question Answering Systems Based on Ontology and Semantic Web in Different Environment," *J. Comput. Sci.*, vol. 8, no. 9, pp. 1407–1413, Sep. 2012.
- [82] J. Suchal, "Caching Spreading Activation Search," *IIT. SRC*. pp. 151–155, 2007.
- [83] R. Mahendra, L. Wanzare, R. Bernardi, A. Lavelli, and B. Magnini, "Acquiring relational patterns from wikipedia: A case study," in *Proc. 5th Lang. Technol. Conf.*, 2011.
- [84] A. Fader, "Open Question Answering," 2014.
- [85] A. Fader, L. S. Zettlemoyer, and O. Etzioni, "Paraphrase-Driven Learning for Open Question Answering.," *ACL*, pp. 1608–1618, 2013.
- [86] S. Shekarpour, E. Marx, A. C. Ngonga Ngomo, and S. Auer, "SINA: Semantic interpretation of user queries for question answering on interlinked data," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 30, pp. 39–51, Jan. 2015.
- [87] S. Shekarpour, S. Auer, A.-C. Ngonga Ngomo, D. Gerber, S. Hellmann, and C. Stadler, "Generating SPARQL queries using templates," *Web Intell. Agent Syst. An Int. J.*, vol. 11, no. 3, pp. 283–295, Jan. 2013.
- [88] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum, "Natural language questions for the web of data," in *EMNLP-CoNLL '12 Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 379–390.
- [89] M. Yahya, K. Berberich, S. Elbassuoni, G. Weikum, M. Yahya, K. Berberich, S. Elbassuoni, and G. Weikum, "Robust question answering over the web of linked data," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manag. - CIKM '13*, 2013, pp. 1107–1116.
- [90] M. Latifi, H. Rodríguez, and M. Sánchez-Marrè, "ScoQAS: A semantic-based closed and open domain question answering system," *Proces. del Leng. Nat.*, vol. 59, pp. 73–80, 2017.
- [91] H. Sundblad, "Question Classification in Question Answering Systems," Linkopings University, 2007.
- [92] S. Xu, G. Cheng, and F. Kong, "Research on question classification for Automatic Question Answering," in *2016 Int. Conf. Asian Lang. Process.*, 2016, pp. 218–221.
- [93] X. Li and D. Roth, "Learning question classifiers: the role of semantic information," *Nat. Lang. Eng.*, vol. 12, no. 3, p. 229, Sep. 2006.
- [94] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Dev. Informaion Retr. - SIGIR '03*, 2003, p. 26.
- [95] R. Srihari, C. Niu, and W. Li, "A hybrid approach for named entity and sub-type tagging," in *Proc. sixth Conf. Appl. Nat. Lang. Process. -*, 2000, pp. 247–254.

- 
- [96] U. Hermjakob, "Parsing and question classification for question answering," in *Proc. Work. Arab. Lang. Process. status Prospect.*, 2001, vol. 12, pp. 1–6.
- [97] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: An On-line Lexical Database \*," *Int. J. Lexicogr.*, vol. 3, no. 4, pp. 235–244, Jan. 1990.
- [98] M. Latifi, H. Khotanlou, and H. Latifi, "An efficient approach based on ontology to optimize the organizational knowledge base management for advanced queries service," in *2011 IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011*, 2011, pp. 269–273.
- [99] L. Yujian and L. Bo, "A normalized Levenshtein distance metric.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1091–1095, Jun. 2007.
- [100] C. Biemann, "Ontology Learning from Text: A Survey of Methods," *LDV-Forum*, vol. 20, no. 2, pp. 75–93, 2005.
- [101] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," Springer Berlin Heidelberg, 2007, pp. 722–735.
- [102] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee, "Linked data on the web (LDOW2008)," in *Proceeding 17th Int. Conf. World Wide Web - WWW '08*, 2008, p. 1265.
- [103] P. Cimiano, V. López, C. Unger, E. Cabrio, A.-C. Ngonga Ngomo, and S. Walter, "Multilingual Question Answering over Linked Data (QALD-3): Lab Overview," Springer Berlin Heidelberg, 2013, pp. 321–332.
- [104] V. López, C. Unger, P. Cimiano, and E. Motta, "Evaluating question answering over linked data," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 21, pp. 3–13, 2013.
- [105] P. Cimiano, V. López, C. Unger, E. Cabrio, A.-C. Ngonga Ngomo, and S. Walter, "Multilingual Question Answering over Linked Data (QALD-3): Lab Overview," in *Lect. Notes Comput. Sci.*, Springer Berlin Heidelberg, 2013, pp. 321–332.
- [106] C. Unger, C. Forascu, V. López, A.-C. N. Ngomo, E. Cabrio, P. Cimiano, and S. Walter, "Question Answering over Linked Data (QALD-4)," 2014.
- [107] C. Unger, C. Forescu, V. López, A.-C. N. Ngomo, E. Cabrio, P. Cimiano, and S. Walter, "Question Answering over Linked Data (QALD-5)," 2015.

# Appendix: A

## Appendix A: Set of Questions for Closed Domain Empirical Evaluation (1<sup>st</sup> Scenario)

	Correct Answer (Y)
	Not Answer Produced (N)
	Not Answer (N)

ID	Question	Stanford Parsing Without Failure	Q Classification (Ss-fp)	EAT (Class-Instance)	Building Constraints	Building QGraph	Inference Final Answer
1	How many employees have two responsibilities?	Y	Y	Y	Y	Y	Y
2	Are there any activities that are in a suspended status?	Y	Y	Y	Y	Y	Y
3	Is there any manager who is controlled by the executive director?	Y	Y	N	N	N	N
4	Is the financial manager a member of the financial council committee?	Y	Y	Y	Y	Y	Y
5	What is the Administrative department number?	Y	Y	Y	Y	Y	Y
6	Who are the invited members in the documentation committee?	Y	Y	N	N	N	N
7	What number of members is financial council committee?	Y	N	N	N	N	N
8	Which activities are performed by public relations expert?	Y	Y	Y	Y	Y	Y
9	Give me a list of departments in the organization?	Y	Y	Y	Y	Y	Y
10	In which city of Iran is the University of Bu-Ali Sina?	Y	Y	Y	Y	Y	Y
11	Does the organization pay rent rebate for each employee?	Y	Y	Y	Y	Y	N
12	Does Mehdi work as a teacher?	Y	Y	Y	Y	Y	Y
13	Who are the experts in the research and planning department?	Y	Y	Y	Y	Y	Y
14	Which section does Reza Kiani work?	Y	Y	Y	Y	Y	Y
15	List all manager of organisation?	Y	Y	Y	Y	Y	Y
16	Did the organization employ Mehdi in 2002?	Y	Y	Y	Y	Y	Y
17	Was Mehdi employed before Ali?	Y	Y	N	N	N	N
18	Was Majid's employment after Reza?	N	N	N	N	N	N
19	What is the Ramin Seraj's birthday?	Y	N	N	N	N	N
20	What is the surname of Reza?	Y	Y	Y	Y	Y	Y
21	List all chief of the departments of the organisation?	Y	Y	N	N	N	N



22	Which division is aimed at attracting funds?	Y	Y	Y	Y	Y	Y
23	Where is the manager of ITC working in the organization?	Y	Y	Y	Y	Y	Y
24	Who is executive director?	Y	Y	Y	Y	Y	Y
25	Who are controlled by executive chief of organization?	Y	Y	Y	Y	Y	Y
26	Who are the members of committee Frontier of IT?	Y	Y	N	N	N	N
27	What are the activities that are listed in running condition?	Y	Y	Y	Y	Y	Y
28	How much is the premium deductions for Ali?	Y	Y	Y	Y	Y	Y
29	What are the responsibilities of committee Frontier of IT?	Y	Y	Y	Y	Y	Y
30	Which division or sub division has economic goals?	Y	Y	Y	Y	Y	Y
31	What are the responsibilities of IT manager?	Y	Y	Y	Y	Y	Y
32	When is the working schedule for managers?	Y	Y	Y	Y	Y	Y
33	When is the start date for employee Mehdi?	Y	Y	Y	Y	Y	Y
34	How much is the value of laptop?	Y	Y	Y	Y	Y	Y
35	What are the authorities of IT manager?	Y	Y	N	N	N	N
36	Which regulations have been defined by IT manager?	Y	Y	Y	Y	Y	Y
37	Who are the main members of committee Frontier of IT?	Y	Y	Y	Y	Y	Y
38	What are committees that combine the team central council?	Y	Y	Y	N	N	N
39	Which company made the video projector product?	Y	Y	Y	Y	Y	Y
40	What is the manufacturer of equipment video projector?	Y	Y	Y	Y	Y	Y



# Appendix: B

## Appendix B: QALD's Training Data Set for Open Domain (2<sup>nd</sup> Scenario)

We have used as training questions set by selecting a set of 188 questions from QALD's (QALD-2, QALD-3, and QALD-4) training set series altogether.

ID	Question	QT
0	Give me all female Russian astronauts.	Who_Properties
1	In which country does the Nile start?	Where_GEO_Action
2	Which German cities have more than 250000 inhabitants?	Where_CompoundProperties
3	Who was the successor of John F. Kennedy?	Who_CompoundProperties_Person
4	Who is the mayor of Berlin?	Who_CompoundProperties_GEO
5	How many students does the Free University in Amsterdam have?	Howmany_CompoundProperties_GEO
6	What is the second highest mountain on Earth?	Where_CompoundProperties
7	Give me all professional skateboarders from Sweden.	Who_Properties_GEO
8	When was Alberta admitted as province?	When_GEO_Action
9	To which countries does the Himalayan mountain system extend?	Where_Properties_GEO_Action
10	Give me a list of all trumpet players that were bandleaders.	Who_Properties
11	Which countries have places with more than two caves?	Where_CompoundProperties
12	What is the total amount of men and women serving in the FDNY?	Howmany_CompoundProperties_Action_Entity
13	Who produces Orangina?	Who_Action_Entity
14	Who is the Formula 1 race driver with the most races?	Who_CompoundProperties
15	Give me all world heritage sites designated within the past five years.	Where_Properties_Action
16	Who is the youngest player in the Premier League?	Who_CompoundProperties
17	Give me all members of Prodigy.	Who_Member
18	What is the longest river?	Where_Properties
19	Does the new Battlestar Galactica series have more episodes than the old one?	None
20	Give me all cars that are produced in Germany.	What_Action_GEO
21	Give me all people that were born in Vienna and died in Berlin.	Who_Action_GEO
22	Is proinsulin a protein?	YNo_SubType
23	How tall is Michael Jordan?	Quantifier_Person
24	What is the capital of Canada?	Where_GEO
25	Who is the governor of Wyoming?	Who_CompoundProperties_GEO
26	Do Prince Harry and Prince William have the same mother?	None
27	Who was the father of Queen Elizabeth II?	Who_CompoundProperties_Person
28	Which U.S. state has been admitted latest?	Where_Properties_Action
29	How many official languages are spoken on the	Howmany_Action_GEO_Properties

Appendix: B

	Seychelles?	
30	Sean Parnell is the governor of which U.S. state?	Where_CompoundProperties_Person
31	Give me all movies directed by Francis Ford Coppola.	What_Person_Action
32	Give me all actors starring in movies directed by and starring William Shatner.	None
33	Which classis do tree frogs belong to?	What_SubType
34	What is the birth name of Angela Merkel?	What_CompoundProperties
35	Give me all current Methodist national leaders.	None
36	How often did Nicole Kidman marry?	Quantifier_Person_Action
37	Give me all Australian nonprofit organizations.	None
38	In which military conflicts did Lawrence of Arabia participate?	What_Person_Action
39	Who developed Minecraft?	Who_Action
40	How many inhabitants does Maribor have?	Howmany_GEO
41	Give me all companies in Munich.	None
42	How tall is Claudia Schiffer?	Quantifier_Person
43	List all games by GMT.	None
44	Who founded Intel?	Who_Action_Entity
45	Who is the husband of Amanda Palmer?	Who_CompoundProperties_Person
46	Give me all breeds of the German Shepherd dog.	None
47	Which cities does the Weser flow through?	Where_GEO_Action
48	Which countries are connected by the Rhine?	Where_GEO_Action
49	Which professional surfers were born on the Philippines?	Who_Action_GEO
50	In which UK city are the headquarters of the MI6?	Where_CompoundProperties_GEO
51	Which other weapons did the designer of the Uzi develop?	What_CompoundProperties_Action
52	Who created Goofy?	Who_Action_Entity
53	Was the Cuban Missile Crisis earlier than the Bay of Pigs Invasion?	YNo_CompoundProperties_TimeRelation
54	Give me all Frisian islands that belong to the Netherlands.	Where_GEO_Member
55	What is the ruling party in Lisbon?	None
56	What are the nicknames of San Francisco?	What_Synonym
57	Which Greek goddesses dwelt on Mount Olympus?	None
58	When were the Hells Angels founded?	When_Action_Entity
59	Give me the Apollo 14 astronauts.	None
60	What is the time zone of Salt Lake City?	What_GEO
61	Which U.S. states are in the same time zone as Utah?	Where_Properties_GEO
62	Give me the capitals of all countries in Africa.	None
63	Give me a list of all lakes in Denmark.	None
64	How many space missions have there been?	Howmany_Properties
65	Did Socrates influence Aristotle?	YNo_Person_Action
66	Give me all Argentine films.	None
67	Give me all launch pads operated by NASA.	What_Properties_Action_Entity
68	Which instruments did John Lennon play?	What_Person_Action

69	Which ships were called after Benjamin Franklin?	What_Person_Action
70	Who are the parents of the wife of Juan Carlos I?	Who_CompoundProperties
71	How many employees does Google have?	Howmany_Entity
72	Did Tesla win a nobel prize in physics?	YNo_Person_Action
73	Give me all cities in New Jersey with more than 100000 inhabitants.	Where_CompoundProperties_GEO
74	Is Michelle Obama the wife of Barack Obama?	YNo_CompoundProperties_Person
75	When was the Statue of Liberty built?	When_Action_CompoundProperties
76	In which U.S. state is Fort Knox located?	Where_Properties_GEO
77	How many children did Benjamin Franklin have?	Howmany_Person
78	When did Michael Jackson die?	When_Person_Action
79	Which daughters of British earls died in the same place they were born in?	Who_Properties_Action_GEO
80	List the children of Margaret Thatcher.	None
81	Who was called Scarface?	Who_Synonym
82	Was Margaret Thatcher a chemist?	YNo_SubType
83	Was Dutch Schultz a jew?	YNo_SubType
84	Which museum exhibits The Scream by Munch?	Which_Person_Action_Entity
85	Give me all books by William Goldman with more than 300 pages.	
86	Which books by Kerouac were published by Viking Press?	What_Properties_Person_Action
87	Give me a list of all American inventions.	None
88	How high is the Mount Everest?	Quantifier_GEO
89	Who created the comic Captain America?	Who_Action_Entity_Properties
90	How many people live in the capital of Australia?	None
91	What is the largest city in Australia?	Where_Properties_GEO
92	Who composed the music for Harold and Maude?	Who_Action_Ent_Person
93	Which films starring Clint Eastwood did he direct himself?	What_Person_Action
94	In which city was the former Dutch queen Juliana buried?	Where_Properties_Person_Action
95	Is Egypts largest city also its capital?	YNo_Equal
96	Where is the residence of the prime minister of Spain?	Where_Properties_GEO
97	Which U.S. state has the abbreviation MN?	Where_Properties_Synonym
97	Which U.S. state has the abbreviation MN?	Where_CompoundProperties_GEO
98	Show me all songs from Bruce Springsteen released between 1980 and 1990.	What_Person_Action_TimeRelation
99	Which movies did Kurosawa direct after Rashomon?	Which_Person_Action
99	Which movies did Kurosawa direct after Rashomon?	Which_Action_TimeRelation
100	What is the founding year of the brewery that produces Pilsner Urquell?	When_Action_CompoundProperties_Entity
101	Who wrote the lyrics for the Polish national anthem?	None
102	Give me all B-sides of the Ramones.	None
103	Who painted The Storm on the Sea of Galilee?	Who_CompoundProperties_Action_GEO

Appendix: B

104	Which country does the creator of Miffy come from?	Where_CompoundProperties_Action
105	For which label did Elvis record his first album?	None
106	Give me the birthdays of all actors of the television show Charmed.	What_CompoundProperties
107	How many employees does IBM have?	Howmany_Entity
108	Which states border Illinois?	Where_GEO_Action
108	Which states border Illinois?	Where_GEO
109	In which country is the Limerick Lake?	Where_Properties_GEO
110	Which television shows were created by Walt Disney?	Which_Person_Action_Properties
111	Which mountain is the highest after the Annapurna?	Where_GEO_TimeRelation
112	In which films directed by Garry Marshall was Julia Roberts starring?	What_Person_Action
113	Which bridges are of the same type as the Manhattan Bridge?	Where_Properties_GEO
114	Was U.S. president Jackson involved in a war?	YNo_Person_Action_GEO
115	Which European countries have a constitutional monarchy?	Where_Properties_Entity
116	Which awards did WikiLeaks win?	Which_Person_Action
117	Who is the daughter of Ingrid Bergman married to?	Who_CompoundProperties_Action
117	Who is the daughter of Ingrid Bergman married to?	Who_Person_Action
118	Which state of the USA has the highest population density?	Where_CompoundProperties_GEO
119	What is the currency of the Czech Republic?	What_CompoundProperties
119	What is the currency of the Czech Republic?	What_GEO
120	Which countries in the European Union adopted the Euro?	Where_CompoundProperties_GEO
121	What is the area code of Berlin?	What_CompoundProperties
121	What is the area code of Berlin?	What_GEO
122	Which countries have more than two official languages?	Where_CompoundProperties
122	Which countries have more than two official languages?	Where_Properties_Entity
123	Who is the owner of Universal Studios?	Who_CompoundProperties_Person
123	Who is the owner of Universal Studios?	Who_CompoundProperties_ORG
124	Through which countries does the Yenisei river flow?	None
125	When did Latvia join the EU?	When_GEO_Action
126	Which monarchs of the United Kingdom were married to a German?	Who_CompoundProperties_Action
126	Which monarchs of the United Kingdom were married to a German?	Who_Action_GEO
127	When was the Battle of Gettysburg?	When_CompoundProperties
128	Which river does the Brooklyn Bridge cross?	Where_GEO_Action
129	What is the highest mountain in Australia?	Where_Properties_GEO
130	Give me all soccer clubs in Spain.	None

131	What are the official languages of the Philippines?	Howmany_CompoundProperties_GEO
132	Who is the mayor of New York City?	Who_CompoundProperties_GEO
133	Who designed the Brooklyn Bridge?	Who_Action_GEO
134	Which telecommunications organizations are located in Belgium?	What_CompoundProperties_GEO
135	Is Frank Herbert still alive?	YNo_Person_Status
136	What is the highest place of Karakoram?	Where_GEO
137	Give me the homepage of Forbes.	None
138	Give me all companies in the advertising industry.	None
139	How many monarchical countries are there in Europe?	Howmany_Properties_GEO
140	What did Bruce Carver die from?	None
141	Give me all school types.	What_Properties
142	Which presidents were born in 1945?	What_Properties_Action
143	Give me all presidents of the United States.	None
144	Who was the wife of U.S. president Lincoln?	Who_CompoundProperties_Person
145	Who developed the video game World of Warcraft?	Who_CompoundProperties
146	What is the official website of Tom Cruise?	What_CompoundProperties
147	List all episodes of the first season of the HBO television series The Sopranos!	None
148	Who produced the most films?	Who_Properties_Person
149	Give me all people with first name Jimmy.	Who_CompoundProperties_Person
150	In which city did John F. Kennedy die?	Where_Person_Action
151	Is there a video game called Battle Chess?	
152	Which mountains are higher than the Nanga Parbat?	Where_Properties
153	Who created Wikipedia?	None
154	Give me all actors starring in Last Action Hero.	None
155	Which software has been developed by organizations founded in California?	What_Action_GEO
156	Which companies work in the aerospace industry as well as on nuclear reactor technology?	What_Properties_GEO
157	Is Christian Bale starring in Batman Begins?	YNo_Person_Action
158	Give me the websites of companies with more than 500000 employees.	None
159	Which actors were born in Germany?	Who_Action_GEO
160	Which caves have more than 3 entrances?	Where_CompoundProperties
160	Which caves have more than 3 entrances?	Where_Properties_Entity
161	Is the wife of president Obama called Michelle?	YNo_CompoundProperties_Synonym
162	Give me all films produced by Hal Roach.	What_Person_Action
163	Give me all video games published by Mean Hamster Software.	What_Properties_Action
164	Which languages are spoken in Estonia?	What_Action_GEO
165	Who owns Aldi?	Who_Action_Entity
166	Which capitals in Europe were host cities of the summer olympic games?	Where_Properties_GEO
166	Which capitals in Europe were host cities of the summer olympic games?	Where_Properties_Entity
167	Who has been the 5th president of the United	Who_CompoundProperties_GEO

Appendix: B

	States of America?	
167	Who has been the 5th president of the United States of America?	Who_Properties_Person
168	How many films did Hal Roach produce?	Howmany_Person_Action
169	Which music albums contain the song Last Christmas?	What_Properties_Action
170	Give me all books written by Danielle Steel.	What_Person_Action
171	Which airports are located in California, USA?	Where_CompoundProperties_GEO
172	Which states of Germany are governed by the Social Democratic Party?	Where_Properties_GEO_Action
173	Give me all Canadian Grunge record labels.	What_Properties
174	Which country has the most official languages?	Where_Properties_Entity
175	In which programming language is GIMP written?	What_Properties_Action
176	Who produced films starring Natalie Portman?	Who_Action_Ent_Person
177	Give me all movies with Tom Cruise.	None
178	In which films did Julia Roberts as well as Richard Gere play?	What_Properties_Person_Action
179	Give me all female German chancellors.	Who_Person
180	Who wrote the book Les Piliers de la terre?	Who_Action_Entity_Properties
181	How many films did Leonardo DiCaprio star in?	Howmany_Person
182	Give me all soccer clubs in the Premier League.	None
183	Which U.S. states possess gold minerals?	Where_Properties_Action
183	Which U.S. states possess gold minerals?	Where_Entity_Action
184	When was Capcom founded?	When_Action_Entity
185	Which organizations were founded in 1950?	What_Properties_Action
186	What is the highest mountain?	None
187	Was Natalie Portman born in the United States?	YNo_Person_Action_GEO
187	Was Natalie Portman born in the United States?	YNo_Person_Action



# Appendix: C

---

## Appendix C: Bounded Variables and Constraints for Q1 in Closed Domain (1<sup>st</sup> Scenario)

Q1: Where is the manager of ITC working in the organization?

QT: Where\_Person\_Action

Variables: { X1, X2, ... X101 }

Constraints (MC):

```
tk_Type(['0', 'X1'])
tk_ACT(['6', 'X2'])
tk_PER(['3', 'X3'])
tk(['5', 'X4'])
advmod(['X1', 'X2'])
tk(['1', 'X5'])
aux(['X5', 'X2'])
nsubj(['X3', 'X2'])
tk(['9', 'X6'])
nmod(['X6', 'X2'])
det(['X3'])
nmod(['X4', 'X3'])
tk(['4', 'X7'])
case(['X7', 'X4'])
tk(['7', 'X8'])
case(['X8', 'X6'])
det(['X6'])
```

----- Predicate with bounded variables for PERSON involved ontology items -----

```
class_PER_0(['i_en_proper_company', 'X7'])
ont_PER_0(['X4', 'X7'])
class_PER_1(['Manager', 'X8'])
ont_PER_1(['X3', 'X8'])
slot_PER_0(['has_authority', 'X9'])
Slot_0(['X8', 'X9'])
class_PER_2(['Authority', 'X10'])
slotType_PER_2(['X9', 'X10'])
slot_PER_1(['control', 'X11'])
Slot_1(['X8', 'X11'])
class_PER_3(['Expert', 'X12'])
slotType_PER_3(['X11', 'X12'])
slot_PER_2(['manager_title', 'X13'])
```

Slot\_2(['X8', 'X13'])  
slot\_PER\_3(['Authority\_title', 'X14'])  
Slot\_3(['X10', 'X14'])  
slot\_PER\_4(['define', 'X15'])  
Slot\_4(['X8', 'X15'])  
slot\_PER\_5(['Expert\_title', 'X16'])  
Slot\_5(['X12', 'X16'])  
slot\_PER\_6(['make', 'X17'])  
Slot\_6(['X8', 'X17'])  
class\_PER\_4(['Detailed\_Schedule', 'X18'])  
slotType\_PER\_4(['X17', 'X18'])  
instance\_PER\_0(['Executive director', 'X19'])  
Inst\_0(['X13', 'X19'])  
instance\_PER\_1(['Authority to decide task assignments', 'X20'])  
Inst\_1(['X14', 'X20'])  
instance\_PER\_2(['http://protege.stanford.edu/rdfenterprise\_Class100049', 'X21'])  
Inst\_2(['X11', 'X21'])  
instance\_PER\_3(['expert technical and professional education and knowledge',  
'X22'])  
Inst\_3(['X16', 'X22'])  
instance\_PER\_4(['Organization manager', 'X23'])  
Inst\_4(['X13', 'X23'])  
instance\_PER\_5(['Product manager', 'X24'])  
Inst\_5(['X13', 'X24'])  
instance\_PER\_6(['Expert of councils and student activities', 'X25'])  
Inst\_6(['X16', 'X25'])  
instance\_PER\_7(['http://protege.stanford.edu/rdfQA-Enterprise\_Instance\_5', 'X26'])  
Inst\_7(['X9', 'X26'])  
instance\_PER\_8(['Payroll Expert', 'X27'])  
Inst\_8(['X16', 'X27'])  
instance\_PER\_9(['ITC manager', 'X28'])  
Inst\_9(['X13', 'X28'])  
instance\_PER\_10(['Development of private schools and public participation expert',  
'X29'])  
Inst\_10(['X16', 'X29'])  
instance\_PER\_11(['Chief of planning department', 'X30'])  
Inst\_11(['X13', 'X30'])  
instance\_PER\_12(['Authority to decide personal effectiveness appraisal and merit  
recognition', 'X31'])  
Inst\_12(['X14', 'X31'])

instance\_PER\_13(['Network Expert', 'X32'])  
 Inst\_13(['X16', 'X32'])  
 instance\_PER\_14(['Sales manager', 'X33'])  
 Inst\_14(['X13', 'X33'])  
 instance\_PER\_15(['Expert for Physical Education', 'X34'])  
 Inst\_15(['X16', 'X34']) involving variables ['X16', 'X34']  
 instance\_PER\_16(['http://protege.stanford.edu/rdfQA-Enterprise\_Instance\_2',  
 'X35'])  
 Inst\_16(['X9', 'X35'])  
 instance\_PER\_17(['expert of theoretical education and skills', 'X36'])  
 Inst\_17(['X16', 'X36'])  
 instance\_PER\_18(['research and Planning vice president', 'X37'])  
 Inst\_18(['X13', 'X37'])  
 instance\_PER\_19(['Financial manager', 'X38'])  
 Inst\_19(['X13', 'X38'])  
 instance\_PER\_20(['Supervise the design, development and implementation of  
 critical ICT Projects across the Public Service', 'X39'])  
 Inst\_20(['X14', 'X39'])  
 instance\_PER\_21(['Security Expert', 'X40'])  
 Inst\_21(['X16', 'X40'])  
 instance\_PER\_22(['Expert of assessment and evaluation of education', 'X41'])  
 Inst\_22(['X16', 'X41'])  
 instance\_PER\_23(['Coordinating vice president', 'X42'])  
 Inst\_23(['X13', 'X42'])  
 instance\_PER\_24(['Education vice president', 'X43'])  
 Inst\_24(['X13', 'X43'])  
 instance\_PER\_25(['Approve major projects, IT budgets, priorities, standards,  
 procedures, and overall IT performances', 'X44'])  
 Inst\_25(['X14', 'X44'])  
 instance\_PER\_26(['student health and nutrition expert', 'X45'])  
 Inst\_26(['X16', 'X45'])  
 instance\_PER\_27(['Expert opinion and pre-college and undergraduate education for  
 adults', 'X46'])  
 Inst\_27(['X16', 'X46'])  
 instance\_PER\_28(['Resources development and logistics vice president', 'X47'])  
 Inst\_28(['X13', 'X47'])  
 instance\_PER\_29(['Expert for parents and teachers association', 'X48'])  
 Inst\_29(['X16', 'X48'])  
 instance\_PER\_30(['Public relations expert', 'X49'])

Inst\_30(['X16', 'X49'])  
instance\_PER\_31(['Theoretical education vice president', 'X50'])  
Inst\_31(['X13', 'X50'])  
instance\_PER\_32(['Chief of security', 'X51'])  
Inst\_32(['X13', 'X51'])  
instance\_PER\_33(['Facilitate and regulate the design, implementation and use of  
ICTs in the public service', 'X52'])  
Inst\_33(['X14', 'X52'])  
instance\_PER\_34(['http://protege.stanford.edu/rdfQA-Enterprise\_Instance\_3',  
'X53'])  
Inst\_34(['X9', 'X53'])  
instance\_PER\_35(['http://protege.stanford.edu/rdfenterprise\_Class70010', 'X54'])  
Inst\_35(['X11', 'X54'])  
instance\_PER\_36(['Sales Expert', 'X55'])  
Inst\_36(['X16', 'X55'])  
instance\_PER\_37(['Expert of Educational Technology Group', 'X56'])  
Inst\_37(['X16', 'X56'])  
instance\_PER\_38(['Library manager', 'X57'])  
Inst\_38(['X13', 'X57'])  
instance\_PER\_39(['Cultural expert', 'X58'])  
Inst\_39(['X16', 'X58'])  
instance\_PER\_40(['Expert for estate rights and legal support of employees', 'X59'])  
Inst\_40(['X16', 'X59'])  
instance\_PER\_41(['Legal Expert', 'X60'])  
Inst\_41(['X16', 'X60'])  
instance\_PER\_42(['Educational evaluation expert', 'X61'])  
Inst\_42(['X16', 'X61'])  
instance\_PER\_43(['Chief of Administrative affair', 'X62'])  
Inst\_43(['X13', 'X62'])

--Predicate with bounded variables for EAT (Location Organization) involved  
ontology items --







EAT\_class\_0(['Employee', 'X63'])  
ont\_Where\_0(['X1', 'X63'])  
EAT\_class\_1(['i\_en\_proper\_person', 'X64']) 64  
ont\_Where\_1(['X1', 'X64'])  
EAT\_class\_2(['Record', 'X65'])  
ont\_Where\_2(['X1', 'X65'])  
EAT\_class\_3(['Department', 'X66']) 66  
ont\_Where\_3(['X1', 'X66'])  
EAT\_slot\_0(['DEP\_Name', 'X67'])  
Slot\_0(['X66', 'X67'])

EAT\_slot\_1(['start\_date', 'X68'])  
Slot\_1(['X65', 'X68'])  
EAT\_slot\_7(['DEP\_chief', 'X69'])  
Slot\_7(['X66', 'X69'])  
EAT\_slot\_8(['DEP\_NO', 'X70'])  
Slot\_8(['X66', 'X70'])  
EAT\_inst\_0(['Planning Dep.', 'X71'])  
Inst\_0(['X67', 'X71'])  
EAT\_inst\_1(['1372/10/10', 'X72'])  
Inst\_1(['X68', 'X72'])  
EAT\_inst\_2(['Engineering Dep.', 'X73'])  
Inst\_2(['X67', 'X73'])  
EAT\_inst\_3(['Telecommunication Dep.', 'X74'])  
Inst\_3(['X67', 'X74'])  
EAT\_inst\_4(['1380/10/02', 'X75'])  
Inst\_4(['X68', 'X75'])  
EAT\_inst\_5(['1380/1/1', 'X76'])  
Inst\_5(['X68', 'X76'])  
EAT\_inst\_6(['Marketing Dep.', 'X77'])  
Inst\_6(['X67', 'X77'])  
EAT\_inst\_7(['http://protege.stanford.edu/rdfenterprise\_Class130068', 'X78'])  
Inst\_7(['X69', 'X78'])  
EAT\_inst\_8(['103', 'X79'])  
Inst\_8(['X70', 'X79'])  
EAT\_inst\_9(['http://protege.stanford.edu/rdfenterprise\_Class50001', 'X80'])  
Inst\_9(['X69', 'X80'])  
EAT\_inst\_10(['Research and Development Dep.', 'X81'])  
Inst\_10(['X67', 'X81'])  
EAT\_inst\_11(['1375/02/02', 'X82'])  
Inst\_11(['X68', 'X82'])  
EAT\_inst\_12(['1378//01/01', 'X83'])  
Inst\_12(['X68', 'X83'])  
EAT\_inst\_13(['104', 'X84'])  
Inst\_13(['X70', 'X84'])  
EAT\_inst\_14(['1381/12/1', 'X85'])  
Inst\_14(['X68', 'X85'])  
EAT\_inst\_15(['http://protege.stanford.edu/rdfenterprise\_Class30008', 'X86'])  
Inst\_15(['X69', 'X86'])  
EAT\_inst\_16(['100', 'X87'])  
Inst\_16(['X70', 'X87'])  
EAT\_inst\_17(['105', 'X88'])  
Inst\_17(['X70', 'X88'])  
EAT\_inst\_18(['1382/03/05', 'X89'])  
Inst\_18(['X68', 'X89'])  
EAT\_inst\_19(['1378/10/13', 'X90'])  
Inst\_19(['X68', 'X90'])  
EAT\_inst\_20(['Administrative Dep.', 'X91'])  
Inst\_20(['X67', 'X91'])  
EAT\_inst\_21(['1375/11/13', 'X92'])  
Inst\_21(['X68', 'X92'])

EAT\_inst\_22(['Educational Dep.', 'X93'])  
Inst\_22(['X67', 'X93'])  
EAT\_inst\_23(['Cultured Dep.', 'X94'])  
Inst\_23(['X67', 'X94'])  
EAT\_inst\_24(['101', 'X95'])  
Inst\_24(['X70', 'X95'])  
EAT\_inst\_25(['1385/11/10', 'X96'])  
Inst\_25(['X68', 'X96'])  
EAT\_inst\_26(['107', 'X97'])  
Inst\_26(['X70', 'X97'])  
EAT\_inst\_27(['1379/01/01', 'X98'])  
Inst\_27(['X68', 'X98'])  
EAT\_inst\_28(['106', 'X99'])  
Inst\_28(['X70', 'X99'])  
EAT\_inst\_29(['1372/12/1', 'X100'])  
Inst\_29(['X68', 'X100'])  
EAT\_inst\_30(['102', 'X101'])  
Inst\_30(['X70', 'X101'])

# Appendix: D

## Appendix D: Results of ScoQAS over QALD-2 Test Set for Open Domain (2<sup>nd</sup> Scenario)

		Correct Answer (Y)
		Not Answer in Golden System (N)
		Not Answer (N)

No.	Question	Stanford Parsing Without Failure	Q Classification	Q Constraints	Q EAT	Q Mapping SPARQL	Solution in Golden system	Final Answer
1	Which German cities have more than 250000 inhabitants?	Y	N	N	Y	N	Y	N
2	Who was the successor of John F. Kennedy?	Y	Y	Y	Y	N	Y	N
3	Who is the mayor of Berlin?	Y	Y	Y	Y	Y	Y	Y
4	How many students does the Free University in Amsterdam have?	Y	Y	Y	Y	N	Y	N
5	What is the second highest mountain on Earth?	Y	Y	Y	Y	Y	Y	Y
6	Give me all professional skateboarders from Sweden.	Y	Y	N	Y	N	Y	N
7	When was Alberta admitted as province?	Y	Y	Y	Y	Y	Y	Y
8	To which countries does the Himalayan mountain system extend?	Y	Y	Y	Y	N	Y	N
9	Give me a list of all trumpet players that were bandleaders.	Y	Y	Y	Y	N	Y	N
10	What is the total amount of men and women serving in the FDNY?	Y	Y	Y	Y	N	Y	N
11	Who is the Formula 1 race driver with the most races?	Y	Y	N	Y	N	Y	N
12	Give me all world heritage sites designated within the past five years.	Y	Y	Y	Y	N	Y	N
13	Who is the youngest player in the Premier League?	Y	Y	N	Y	N	Y	N
14	Give me all members of Prodigy.	Y	Y	Y	Y	Y	Y	Y
15	What is the longest river?	Y	N	N	N	N	Y	N
16	Does the new Battlestar Galactica series have more episodes than the old one?	Y	Y	N	Y	N	Y	N
17	Give me all cars that are produced in Germany.	Y	Y	Y	Y	Y	Y	Y
18	Give me all people that were born in Vienna and died in Berlin.	Y	Y	Y	Y	N	Y	N
19	How tall is Michael Jordan?	Y	Y	Y	Y	Y	Y	Y
20	What is the capital of Canada?	Y	Y	Y	Y	Y	Y	Y
21	Who is the governor of Texas?	Y	Y	Y	Y	Y	Y	Y
22	Do Harry and William, Princes of Wales, have the same mother?	Y	N	N	Y	N	Y	N
23	Who was the father of Queen Elizabeth II?	Y	Y	N	Y	N	Y	N
24	Which U.S. state has been admitted latest?	Y	N	N	N	N	Y	N
25	How many official languages are spoken on the Seychelles?	Y	Y	Y	Y	Y	Y	Y
26	Sean Parnell is the governor of which U.S. state?	Y	N	N	N	N	Y	N
27	Give me all movies directed by Francis Ford Coppola.	Y	Y	Y	Y	Y	Y	Y
28	Give me all actors starring in movies directed by and starring William Shatner.	Y	N	N	N	N	Y	N
29	What is the birth name of Angela Merkel?	Y	Y	Y	Y	Y	Y	Y

Appendix: D

30	Give me all current Methodist national leaders.	Y	N	N	N	N	Y	N
31	How often did Nicole Kidman marry?	Y	Y	Y	Y	N	Y	N
32	Give me all Australian nonprofit organizations.	Y	N	N	N	N	Y	N
33	In which military conflicts did Lawrence of Arabia participate?	Y	Y	Y	Y	N	Y	N
34	Who developed Skype?	Y	Y	Y	Y	Y	Y	Y
35	What is the melting point of copper?	Y	N	N	Y	N	N	N
36	Give me all sister cities of Brno.	N	N	N	N	N	Y	N
37	How many inhabitants does Maribor have?	Y	Y	Y	Y	Y	Y	Y
38	Give me all companies in Munich.	Y	N	N	N	N	Y	N
39	List all boardgames by GMT.	Y	N	N	N	N	Y	N
40	Who founded Intel?	Y	Y	Y	Y	Y	Y	Y
41	Who is the husband of Amanda Palmer?	Y	Y	Y	Y	N	Y	N
42	Give me all breeds of the German Shepherd dog.	Y	N	N	N	N	Y	N
43	Which cities does the Weser flow through?	N	N	Y	Y	N	Y	N
44	Which countries are connected by the Rhine?	Y	Y	Y	Y	Y	Y	Y
45	Which professional surfers were born on the Philippines?	Y	Y	Y	Y	Y	Y	Y
46	What is the average temperature on Hawaii?	Y	Y	Y	Y	N	N	N
47	In which UK city are the headquarters of the MI6?	Y	Y	Y	Y	N	Y	N
48	Which other weapons did the designer of the Uzi develop?	Y	Y	Y	Y	Y	Y	Y
49	Was the Cuban Missile Crisis earlier than the Bay of Pigs Invasion?	Y	Y	N	Y	N	Y	N
50	Give me all Frisian islands that belong to the Netherlands.	Y	Y	Y	Y	N	Y	N
51	Who invented the zipper?	Y	Y	Y	Y	Y	N	N
52	What is the ruling party in Lisbon?	Y	N	N	N	N	Y	N
53	What are the nicknames of San Francisco?	Y	Y	Y	Y	Y	Y	Y
54	Which Greek goddesses dwelt on Mount Olympus?	Y	N	Y	N	N	Y	N
55	When were the Hells Angels founded?	Y	Y	Y	Y	Y	Y	Y
56	Give me the Apollo 14 astronauts.	N	N	N	N	N	Y	N
57	What is the time zone of Salt Lake City?	Y	Y	Y	Y	N	Y	N
58	Which U.S. states are in the same timezone as Utah?	Y	Y	Y	Y	Y	Y	Y
59	Give me a list of all lakes in Denmark.	N	N	N	N	N	Y	N
60	How many space missions have there been?	Y	Y	Y	Y	Y	Y	Y
61	Did Socrates influence Aristotle?	Y	Y	Y	Y	Y	Y	Y
62	Give me all Argentine films.	N	N	N	N	N	Y	N
63	Give me all launch pads operated by NASA.	Y	Y	N	Y	N	Y	N
64	Which instruments did John Lennon play?	Y	Y	Y	Y	Y	Y	Y
65	Which ships were called after Benjamin Franklin?	Y	Y	N	N	N	Y	N
66	Who are the parents of the wife of Juan Carlos I?	Y	Y	N	Y	N	Y	N
67	How many employees does Google have?	Y	Y	Y	Y	Y	Y	Y
68	Did Tesla win a nobel prize in physics?	Y	Y	Y	Y	N	Y	N
69	Is Michelle Obama the wife of Barack Obama?	Y	Y	N	Y	N	Y	N
70	When was the Statue of Liberty built?	Y	Y	Y	Y	Y	Y	Y
71	In which U.S. state is Area 51 located?	Y	Y	Y	Y	Y	Y	Y
72	How many children did Benjamin Franklin have?	Y	Y	Y	Y	N	Y	N
73	When did Michael Jackson die?	Y	Y	Y	Y	Y	Y	Y
74	Which daughters of British earls died in the same place they were born in?	Y	Y	N	Y	Y	Y	N
75	List the children of Margaret Thatcher.	Y	Y	N	N	N	Y	N
76	Who was called Scarface?	Y	Y	Y	Y	Y	Y	Y
77	Was Margaret Thatcher a chemist?	Y	Y	Y	Y	Y	Y	Y









78	Was Dutch Schultz a Jew?	Y	Y	Y	Y	Y	Y	Y
79	Give me all books by William Goldman with more than 300 pages.	Y	N	N	N	N	Y	N
80	Which books by Kerouac were published by Viking Press?	Y	Y	N	Y	N	Y	N
81	Give me a list of all American inventions.	N	N	N	N	N	Y	N
82	How high is the Mount Everest?	Y	Y	Y	Y	Y	Y	Y
83	Who created the comic Captain America?	Y	Y	N	Y	N	Y	N
84	How many people live in the capital of Australia?	Y	Y	Y	Y	Y	Y	Y
85	What is the largest city in Australia?	Y	Y	Y	Y	Y	Y	Y
86	Who composed the music for Harold and Maude?	Y	Y	Y	Y	N	Y	N
87	Which films starring Clint Eastwood did he direct himself?	Y	Y	N	Y	N	Y	N
88	In which city was the former Dutch queen Juliana buried?	Y	Y	Y	Y	N	Y	N
89	Where is the residence of the prime minister of Spain?	Y	Y	Y	Y	N	Y	N
90	Which U.S. State has the abbreviation MN?	Y	Y	Y	Y	N	Y	N
91	Show me all songs from Bruce Springsteen released between 1980 and 1990.	Y	Y	Y	Y	N	Y	N
92	Which movies did Sam Raimi direct after Army of Darkness?	Y	N	N	Y	N	Y	N
93	What is the founding year of the brewery that produces Pilsner Urquell?	Y	Y	Y	Y	N	Y	N
94	Who wrote the lyrics for the Polish national anthem?	Y	Y	Y	Y	Y	Y	Y
95	Give me all B-sides of the Ramones.	Y	N	N	N	N	Y	N
96	Who painted The Storm on the Sea of Galilee?	Y	Y	Y	Y	N	Y	N
97	Which country does the creator of Miffy come from?	Y	Y	Y	Y	N	Y	N
98	For which label did Elvis record his first album?	Y	Y	Y	Y	N	Y	N
99	Who produces Orangina?	Y	Y	Y	Y	Y	Y	Y



# Appendix: E

## Appendix E: Results of ScoQAS over QALD-3 Test Set for Open Domain (2<sup>nd</sup> Scenario)

		Correct Answer (Y)
		Not Answer in Golden System (N)
		Not Answer (N)

ID	Question	Stanford Parsing Without Failure	Q Classification	Q Cconstraints	Q EAT	Q Mapping SPARQL	Solution in Golden system	Final Answer
1	Which books by Kerouac were published by Viking Press?	Y	Y	Y	Y	N	Y	N
2	Which U.S. states are in the same timezone as Utah?	Y	Y	Y	Y	N	Y	N
3	Which daughters of British earls died in the same place they were born in?	Y	Y	N	Y	N	Y	N
4	Which instruments did John Lennon play?	Y	Y	Y	Y	Y	Y	Y
5	When was the Statue of Liberty built?	Y	Y	Y	Y	Y	Y	Y
6	Give me all people that were born in Vienna and died in Berlin.	Y	Y	Y	Y	N	Y	N
7	How tall is Michael Jordan?	Y	Y	Y	Y	Y	Y	Y
8	What is the total amount of men and women serving in the FDNY?	Y	Y	N	Y	N	Y	N
9	Who composed the music for Harold and Maude?	Y	Y	Y	Y	N	Y	N
10	In which city was the former Dutch queen Juliana buried?	Y	Y	Y	Y	Y	Y	N
11	What are the nicknames of San Francisco?	Y	Y	Y	Y	Y	Y	Y
12	Where is the residence of the prime minister of Spain?	Y	Y	Y	Y	N	Y	N
13	Which other weapons did the designer of the Uzi develop?	Y	Y	Y	Y	N	Y	N
14	To which countries does the Himalayan mountain system extend?	Y	Y	N	Y	N	Y	N
15	What is the founding year of the brewery that produces Pilsner Urquell?	Y	Y	Y	Y	N	Y	N
16	Which country does the creator of Miffy come from?	Y	Y	Y	Y	N	Y	N
17	Was Margaret Thatcher a chemist?	Y	Y	Y	Y	Y	Y	N
18	Which German cities have more than 250000 inhabitants?	Y	Y	Y	Y	N	Y	N
19	For which label did Elvis record his first album?	Y	Y	Y	Y	N	Y	N
20	What is the capital of Canada?	Y	Y	Y	Y	N	Y	N
21	What is the average temperature on Hawaii?	Y	Y	N	Y	N	N	N
22	In which U.S. state is Fort Knox located?	Y	Y	Y	Y	N	Y	N
23	Give me a list of all trumpet players that were bandleaders.	Y	Y	N	Y	N	Y	N
24	Do Prince Harry and Prince William have the same mother?	Y	Y	Y	Y	N	Y	N

Appendix: E

25	In which military conflicts did Lawrence of Arabia participate?	Y	Y	Y	Y	Y	Y	Y
26	Who invented the zipper?	Y	Y	Y	Y	Y	N	Y
27	Who developed Minecraft?	Y	Y	Y	Y	Y	Y	Y
28	How many space missions have there been?	Y	Y	Y	Y	Y	Y	Y
29	Give me all cars that are produced in Germany.	Y	Y	Y	Y	Y	Y	Y
30	How many children did Benjamin Franklin have?	Y	Y	Y	Y	N	Y	N
31	Who was the successor of John F. Kennedy?	Y	Y	N	Y	N	Y	N
32	Is Michelle Obama the wife of Barack Obama?	Y	Y	Y	Y	Y	Y	N
33	Who is the youngest player in the Premier League?	Y	Y	N	Y	N	Y	N
34	Give me all world heritage sites designated within the past five years.	Y	Y	Y	Y	N	Y	N
35	Was Dutch Schultz a jew?	Y	Y	Y	Y	Y	Y	Y
36	What is the second highest mountain on Earth?	Y	Y	Y	Y	N	Y	N
37	How often did Nicole Kidman marry?	Y	Y	Y	Y	N	Y	N
38	What is the largest city in Australia?	Y	Y	N	Y	N	Y	N
39	Who painted The Storm on the Sea of Galilee?	Y	Y	N	Y	Y	Y	N
40	Give me all launch pads operated by NASA.	Y	Y	N	Y	N	Y	N
41	Who wrote the lyrics for the Polish national anthem?	Y	Y	Y	Y	N	Y	N
42	Who created the comic Captain America?	Y	Y	N	Y	N	Y	N
43	Who was the father of Queen Elizabeth II ?	Y	Y	Y	Y	N	Y	N
44	Which U.S. state has the abbreviation MN?	Y	Y	Y	Y	Y	Y	Y
45	Which movies did Kurosawa direct after Rashomon?	Y	Y	Y	Y	Y	Y	N
46	Which professional surfers were born on the Philippines?	Y	Y	Y	Y	Y	Y	Y
47	Which films starring Clint Eastwood did he direct himself?	Y	Y	Y	Y	Y	Y	N
48	Who are the parents of the wife of Juan Carlos I?	Y	Y	N	Y	N	Y	N
49	What is the birth name of Angela Merkel?	Y	Y	N	Y	N	Y	N
50	Who was called Scarface?	Y	Y	Y	Y	N	Y	N
51	Did Socrates influence Aristotle?	Y	Y	Y	Y	Y	Y	Y
52	In which UK city are the headquarters of the MI6 ?	Y	N	N	Y	N	Y	N
53	What is the time zone of Salt Lake City ?	Y	Y	Y	Y	N	Y	N
54	Does the new Battlestar Galactica series have more episodes than the old one?	Y	N	N	N	N	Y	N
55	When was Alberta admitted as province?	Y	Y	Y	Y	Y	Y	N
56	Which countries are connected by the Rhine?	Y	Y	Y	Y	Y	Y	Y
57	Give me all Frisian islands that belong to the Netherlands.	Y	Y	Y	Y	N	Y	N
58	Which ships were called after Benjamin Franklin?	Y	Y	Y	Y	Y	Y	N
59	Who is the husband of Amanda Palmer?	Y	Y	Y	Y	N	Y	N
60	How many employees does Google have?	Y	Y	Y	Y	Y	Y	Y
61	When did Michael Jackson die?	Y	Y	Y	Y	Y	Y	Y
62	How many inhabitants does Maribor have?	Y	Y	Y	Y	Y	Y	Y
63	Which Greek goddesses dwelt on Mount Olympus?	N	N	N	N	Y	Y	N
64	How high is the Mount Everest?	Y	Y	Y	Y	Y	Y	Y
65	Did Tesla win a nobel prize in physics?	Y	Y	Y	Y	N	Y	N
66	Who is the governor of Wyoming?	Y	Y	Y	Y	Y	Y	Y
67	When were the Hells Angels founded?	Y	Y	Y	Y	Y	Y	Y
68	Who is the mayor of Berlin?	Y	Y	Y	Y	Y	Y	Y
69	How many people live in the capital of Australia?	Y	Y	Y	Y	Y	Y	Y
70	Who founded Intel?	Y	Y	Y	Y	Y	Y	Y

71	Which cities does the Weser flow through?	N	N	Y	Y	N	Y	N
72	Give me all movies directed by Francis Ford Coppola.	Y	Y	Y	Y	Y	Y	N
73	Who produces Orangina?	Y	Y	Y	Y	Y	Y	Y
74	What is the melting point of copper?	Y	N	N	N	N	N	N
75	Was the Cuban Missile Crisis earlier than the Bay of Pigs Invasion?	Y	Y	Y	Y	N	Y	N
76	Who is the Formula 1 race driver with the most races?	Y	Y	N	Y	N	Y	N
77	How many official languages are spoken on the Seychelles?	Y	Y	Y	Y	Y	Y	Y
78	Give me all professional skateboarders from Sweden.	Y	Y	N	Y	N	Y	N
79	Give me all members of Prodigy.	Y	Y	Y	Y	N	Y	N
80	How many students does the Free University in Amsterdam have?	Y	Y	Y	Y	N	Y	N



# Appendix: F

## Appendix F: Results of ScoQAS over QALD-4 Test Set for Open Domain (2<sup>nd</sup> Scenario)

<span style="background-color: #00FF00; border: 1px solid black; display: inline-block; width: 15px; height: 10px;"></span> <span style="color: green;">→</span> Correct Answer (Y)
<span style="background-color: #00FFFF; border: 1px solid black; display: inline-block; width: 15px; height: 10px;"></span> <span style="color: blue;">→</span> Not Answer in Golden System (N)
<span style="background-color: #FF0000; border: 1px solid black; display: inline-block; width: 15px; height: 10px;"></span> <span style="color: red;">→</span> Not Answer (N)

No.	Question	Stanford Parsing Without Failure	Q Classification	Q Constraints	Q EAT	Q Mapping SPARQL	Solution in Golden system	Final Answer
1	Give me all taikonauts.	Y	Y	Y	Y	Y	Y	Y
2	How many languages are spoken in Colombia?	Y	Y	Y	Y	Y	Y	Y
3	Which poet wrote the most books?	Y	Y	N	Y	Y	Y	Y
4	How many programming languages are there?	N	N	N	N	N	Y	N
5	Give me all Dutch parties.	Y	Y	Y	Y	Y	Y	Y
6	When was Carlo Giuliani shot?	N	N	N	N	N	Y	N
7	Does the Isar flow into a lake?	Y	Y	N	Y	N	Y	N
8	Which rivers flow into a German lake?	Y	Y	Y	Y	Y	Y	Y
9	How heavy is Jupiter's lightest moon?	Y	Y	Y	Y	Y	Y	Y
10	Who is the youngest Darts player?	Y	Y	Y	Y	Y	Y	Y
11	Give me all animals that are extinct.	Y	Y	Y	Y	Y	Y	Y
12	How many pages does War and Peace have?	Y	N	N	Y	N	Y	N
13	Which ingredients do I need for carrot cake?	Y	Y	Y	Y	Y	Y	Y
14	What is the most frequent death cause?	N	N	N	N	N	Y	N
15	Who has Tom Cruise been married to?	Y	Y	Y	Y	Y	Y	Y
16	Who is the tallest player of the Atlanta Falcons?	Y	Y	Y	Y	Y	Y	Y
17	What is the bridge with the longest span?	Y	Y	Y	Y	Y	Y	Y
18	Give me all films produced by Steven Spielberg with a budget of at least \$80 million.	Y	N	N	Y	N	Y	N
19	Is Cola a beverage?	Y	Y	Y	Y	Y	Y	Y
20	Which actor was casted in the most movies?	Y	Y	Y	Y	Y	Y	Y
21	Where was Bach born?	Y	Y	Y	Y	Y	Y	Y
22	Which of Tim Burton's films had the highest budget?	N	N	N	N	N	Y	N
23	Does Abraham Lincoln's death place have a website?	Y	Y	N	Y	N	Y	N
24	Who are the four youngest MVP basketball players?	Y	Y	N	Y	N	Y	N
25	What are the top-10 action role-playing video games according to IGN?	N	N	N	N	N	Y	N
26	Give me all actors who were born in Berlin.	Y	Y	Y	Y	Y	Y	Y
27	Give me all actors who were born in Paris after 1950.	Y	N	N	Y	N	Y	N
28	What was Brazil's lowest rank in the FIFA World Ranking?	Y	N	N	Y	N	Y	N
29	Give me all Australian metalcore bands.	Y	N	N	N	N	Y	N
30	When is Halloween?	Y	Y	Y	Y	Y	Y	Y
31	How many inhabitants does the largest city in Canada have?	Y	Y	Y	Y	Y	Y	Y
32	In which countries can you pay using the West	Y	Y	Y	Y	Y	Y	Y







Appendix: F

	African CFA franc?							
33	Give me the capitals of all countries that the Himalayas run through.	Y	Y	N	Y	N	Y	N
34	Who was the first to climb Mount Everest?	Y	Y	Y	Y	Y	Y	Y
35	To which artistic movement did the painter of The Three Dancers belong?	Y	Y	N	Y	N	Y	N
36	Which pope succeeded John Paul II?	Y	Y	Y	Y	Y	Y	Y
37	What was the last movie with Alec Guinness?	Y	Y	Y	Y	Y	Y	Y
38	How many James Bond movies are there?	Y	N	N	N	N	Y	N
39	Which actor played Chewbacca?	Y	Y	Y	Y	Y	Y	Y
40	Give me the grandchildren of Bruce Lee.	Y	N	N	N	N	Y	N
41	Give me all writers that won the Nobel Prize in literature.	Y	Y	Y	Y	Y	Y	Y
42	What is the official color of the University of Oxford?	Y	Y	Y	Y	Y	Y	Y
43	Give me all Swedish oceanographers.	Y	Y	Y	Y	Y	Y	Y
44	How deep is Lake Placid?	Y	Y	N	N	Y	Y	N
45	Is James Bond married?	Y	Y	Y	Y	Y	Y	Y
46	Which spaceflights were launched from Baikonur?	Y	N	N	N	N	Y	N
47	Give me all actors called Baldwin.	Y	Y	Y	Y	Y	Y	Y
48	What does CPU stand for?	Y	N	N	N	N	Y	N
49	In which studio did the Beatles record their first album?	Y	Y	Y	Y	N	N	N
50	How many gold medals did Michael Phelps win at the 2008 Olympics?	Y	Y	N	Y	N	N	N



# Appendix: G

## Appendix G: Results of ScoQAS over QALD-5 Test Set for Open Domain (2<sup>nd</sup> Scenario)

		Correct Answer (Y)
		Not Answer in Golden System (N)
		Not Answer (N)

No.	Question	Stanford Parsing Without Failure	Q Classification	Q Constraints	Q EAT	Q Mapping SPARQL	Solution in Golden system	Final Answer
1	Give me all ESA astronauts.	N	N	N	N	N	Y	N
2	Give me all Swedish holidays.	N	N	N	N	N	Y	N
3	Who is the youngest Pulitzer Prize winner?	Y	Y	N	Y	N	Y	N
4	Which animals are critically endangered?	Y	Y	Y	Y	Y	Y	Y
5	Which soccer players were born on Malta?	Y	Y	Y	Y	Y	Y	Y
6	Did Arnold Schwarzenegger attend a university?	Y	Y	Y	Y	Y	Y	Y
7	Which programming languages were influenced by Perl?	Y	Y	Y	Y	N	Y	N
8	Is Barack Obama a democrat?	Y	Y	Y	Y	Y	Y	Y
9	How many children does Eddie Murphy have?	Y	Y	Y	Y	Y	Y	Y
10	Who is the oldest child of Meryl Streep?	Y	Y	Y	Y	Y	Y	Y
11	Who killed John Lennon?	Y	Y	N	Y	Y	Y	Y
12	Which frequent flyer program has the most airlines?	Y	N	N	Y	N	Y	N
13	In which city is Air China headquartered?	Y	Y	Y	Y	Y	Y	Y
14	Which artists were born on the same date as Rachel Stevens?	Y	Y	Y	Y	N	Y	N
15	How many scientists graduated from an Ivy League university?	Y	Y	Y	Y	Y	Y	Y
16	Which types of grapes grow in Oregon?	Y	N	N	Y	N	Y	N
17	Who is starring in Spanish movies produced by Benicio del Toro?	Y	Y	N	Y	N	Y	N
18	Who is the manager of Real Madrid?	Y	Y	Y	Y	Y	Y	Y
19	Give me the currency of China.	N	N	N	N	N	Y	N
20	Which movies starring Brad Pitt were directed by Guy Ritchie?	Y	Y	N	Y	N	Y	N
21	How many companies were founded by the founder of Facebook?	Y	Y	Y	Y	Y	Y	Y
22	How many companies were founded in the same year as Google?	Y	N	N	Y	N	Y	N
23	Which subsidiary of Lufthansa serves both Dortmund and Berlin Tegel?	Y	Y	N	Y	N	Y	N
24	How many airlines are members of the Star Alliance?	Y	N	N	N	N	Y	N
25	Give me all spacecrafts that flew to Mars.	Y	N	N	Y	N	Y	N
26	Which musician wrote the most books?	Y	Y	Y	Y	Y	Y	Y
27	Show me everyone who was born on Halloween.	Y	N	N	N	N	Y	N
28	Give me all Swiss non-profit organizations.	Y	N	N	N	N	Y	N
29	In which country is Mecca located?	Y	Y	Y	Y	Y	Y	Y

Appendix: G

30	What is the net income of Apple?	Y	Y	Y	Y	Y	Y	Y
31	What does the abbreviation FIFA stand for?	Y	Y	Y	Y	Y	Y	Y
32	When did the Ming dynasty dissolve?	Y	Y	Y	Y	Y	Y	Y
33	Which museum in New York has the most visitors?	Y	Y	N	Y	N	Y	N
34	Is Lake Baikal bigger than the Great Bear Lake?	Y	N	N	Y	N	Y	N
35	Desserts from which country contain fish?	N	N	N	N	N	Y	N
36	What is the highest mountain in Italy?	Y	Y	Y	Y	Y	Y	Y
37	Where did the architect of the Eiffel Tower study?	Y	Y	Y	Y	Y	Y	Y
38	Which Greek parties are pro-European?	Y	N	N	N	N	Y	N
39	What is the height difference between Mount Everest and K2?	Y	N	N	Y	N	Y	N
40	Who is the mayor of Rotterdam?	Y	Y	Y	Y	Y	Y	Y
41	In which city were the parents of Che Guevara born?	Y	Y	Y	Y	Y	N	N
42	How high is the Yokohama Marine Tower?	Y	Y	Y	Y	Y	Y	Y
43	Are Taiko a kind of Japanese musical instruments?	Y	Y	N	Y	N	Y	N
44	How many ethnic groups live in Slovenia?	Y	Y	Y	Y	Y	Y	Y
45	List the seven kings of Rome.	Y	N	N	N	N	Y	N
46	Who were the parents of Queen Victoria?	Y	Y	Y	Y	Y	Y	Y
47	Who is the heaviest player of the Chicago Bulls?	Y	N	N	Y	N	Y	N
48	Which volcanos in Japan erupted since 2000?	Y	Y	N	Y	N	Y	N
49	Who is the tallest basketball player?	Y	Y	Y	Y	Y	Y	Y
50	Where was the "Father of Singapore" born?	Y	N	N	N	N	Y	N
51	Which Secretary of State was significantly involved in the United States' dominance of the Caribbean?	Y	N	N	Y	N	Y	N
52	Who is the architect of the tallest building in Japan?	Y	Y	Y	Y	Y	Y	Y
53	What is the name of the Viennese newspaper founded by the creator of the croissant?	Y	N	N	Y	N	Y	N
54	In which city where Charlie Chaplin's half brothers born?	Y	Y	N	Y	N	Y	N
55	Which German mathematicians were members of the von Braun rocket group?	Y	Y	N	Y	N	Y	N
56	Which writers converted to Islam?	Y	Y	Y	Y	Y	Y	Y
57	Are there man-made lakes in Australia that are deeper than 100 meters?	Y	N	N	N	N	Y	N
58	Which movie by the Coen brothers stars John Turturro in the role of a New York City playwright?	Y	N	N	Y	N	Y	N
59	Which of the volcanoes that erupted in 1550 is still active?	N	N	N	N	N	Y	N

# Appendix: H

---

## Appendix H: Pseudocode of Generating QGraph

---

**Algorithm** Building Provided Graph to Use in Prolog

---

**procedure** BUILD\_CONSTRAINTGRAPH\_QT(currentRule R<sub>0</sub>, constraintGraphFile)

input = constraintGraphFile

dotPr=currentPrologDGraph(R<sub>0</sub>)

**for** textLine **in** readline(input) **do**

    newtxtTab = split(textLine, "tab space")

**if** newtxtTab[0] == "Node" **then**

        i = 1

        argNode1 = newtxtTab[1]

**while** newtxtTab[i+1] != ":" **do**

            argNode1 = argNode1 + " " + newtxtTab[i+1]

            i = i + 1

**end while**

        nodeLabel = str(newtxtTab[i+3])

        argLabel = newtxtTab[i+5]

**if** nodeLabel.startswith("Class") **then**

            add\_Node\_Class\_Prolog(dotPr, argNode1, argLabel)

**else if** nodeLabel.startswith("Instance") **then**

            add\_Node\_Instance\_Prolog(dotPr, argNode1, argLabel)

**else**

            add\_Node\_Instance\_Prolog(dotPr, argNode1)

**end if**

**else if** newtxtTab[0] == "Edge" **then**

        i=2

        edge = findVertex(newtxtTab[i])

**if** edgeLabel.startswith("EAT\_class\_") **or** edge.startswith("class\_") **then**

            add\_Edge\_Var2Class\_Prolog(dotPr, edge)

**else if** edgeLabel.startswith("EAT\_slot\_") **or** edge.startswith("slot\_") **then**

            add\_Edge\_Var2Slot\_Prolog(dotPr, edge)

**else if** edgeLabel.startswith("EAT\_inst\_") **or** edge.startswith("inst\_") **then**

            add\_Edge\_Var2InstanceEAT\_Prolog(dotPr, edge)

```
    else if edgeLabel.startswith("instance_") then
        add_Edge_Var2Instance_Prolog(dotPr, edge)
    else if edgeLabel == "Class" then
        add_Edge_Class_Prolog(dotPr, edge)
    else if edgeLabel == "Instance"
        add_Edge_Instance_Prolog(dotPr, edge)
    else
        add_Edge_Var2tk_Prolog(dotPr,e,edge)
    end if
end if
end for
end procedure
```

---

# Appendix: I

---

## Appendix I: List of the Publications

In this appendix, the list of our peer-reviewed publications is attached towards on the development of thesis work which have been presented the basic and advanced framework of the ScoQAS.

1. M. Latifi, H. Rodríguez, and M. Sànchez-Marrè, “Facing Closed and Open Domain QA over Knowledge Bases,” (under submitting to Natural Language Engineering (NLE) journal), 2018.
2. M. Latifi, H. Rodríguez, and M. Sànchez-Marrè, “ScoQAS: A semantic-based closed and open domain question answering system,” *Proces. Leng. Nat.*, vol. 59, pp. 73–80, 2017.
3. M. Latifi and M. Sànchez-Marrè, “The Use of NLP Interchange Format for Question Answering in Organizations.,” in *IOS Press, Frontiers in Artificial Intelligence and Applications*, pp. 235–244, 2013.
4. M. Latifi, “Proposal for Using NLP Interchange Format for Question Answering in Organizations”, 7th International Web Rule Symposium, CEUR Workshop Proceedings, Vol. 1004., University of Washington, Seattle, U.S.A, pp. 1-10, 2013.