# An Ontology-Based Question-Answering, from Natural Language to SPARQL Query

Davide Varagnolo[1,3], Dora Melo[2,3][a] and Irene Pimenta Rodrigues[1,3][b]
[1]Department of Informatics, University of Évora, Évora, Portugal
[2]Polytechnic of Coimbra, Coimbra Business School - ISCAC, Coimbra, Portugal
[3]NOVA Laboratory for Computer Science and Informatics, NOVA LINCS, Caparica, Portugal

Keywords: Natural Language Processing, Knowledge Representation, Knowledge Discovery, Semantic Web, Archives Linked Data Semantic Representation.

Abstract: In this paper an Ontology-based Question-Answering system for exploring the information on CIDOC-CRM ontology representing the Portuguese Archives metadata text descriptions is presented. The proposed approach transforms the natural language input question into a SPARQL query over the target knowledge base, the Portuguese Archives CIDO-CRM Population. To interpret the user´s natural language questions, a pipeline with a natural language grammar, Stanza, a Discourse Representation Structure builder and the final question interpretation on a Query ontology is used. After obtaining the best representation of the user question on the Query ontology, the query constraints classes and properties are translated to CIDOC-CRM ontology and a SPARQL query is generated. The matching of the questions DRS on the query ontology is done as a constraint satisfaction problem and the choice of the best interpretation (matching) is obtained by solving a multi-objective optimizer.



## 1 INTRODUCTION

The representation of information on an Ontology, such as CIDOC-CRM (Conceptual Reference Model) which was developed for museums by the International Committee for Documentation (CIDOC) of the International Council of Museums (ICOM) (Meghini and Doerr, 2018; ICOM/CIDOC, 2020), enables new searches using dedicated interfaces or using a specific query language, such as SPARQL. In the research project EPISA (Entity and Property Inference for Semantic Archives), one of the achieved results is the migration of the Portuguese Archival Information into CIDOC-CRM Ontology (Melo et al., 2023). This OWL knowledge base also includes the representation of the information extracted from the semi-structured text fields of the Portuguese Archives metadata, such as passport requisitions or baptism registers (Melo et al., 2023; Melo et al., 2020; Varagnolo. et al., 2022).

Information on Birth events could be extracted from the metadata of the archival materials of the Portuguese Catholic Church Archives referring to baptisms and marriages, or from the Portuguese Civil Administration Archives metadata, referring to passport requisitions, testaments, marriages or divorces (Varagnolo. et al., 2022). To explore this information represented in CIDOC-CRM, SPARQL queries or DL-Queries are adequate tools to be used. Portuguese Archives users who are interested in exploring the content of the archive find difficult the use of a formal query language such as SPARQL. A Question-Answering system for exploring the information on CIDOC-CRM ontology was developed to enable these Portuguese Archives users to explore the information migrated to CIDOC-CRM. This system translates natural language questions into SPARQL queries on CIDOC-CRM.

SPARQL[1] is the standard query language and protocol for Linked Open Data on the web or for RDF triplestores that enables users to query information from knowledge bases mapped to RDF, such as OWL knowledge bases. Querying such knowledge bases using SPARQL queries is difficult and complex, even for experts. In addition to the syntax and semantics

[a] https://orcid.org/0000-0003-3744-2980
[b] https://orcid.org/0000-0003-2370-3019

[1]https://www.w3.org/TR/sparql11-overview/



Varagnolo, D., Melo, D. and Rodrigues, I.
An Ontology-Based Question-Answering, from Natural Language to SPARQL Query.
DOI: 10.5220/0012180000003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 2: KEOD*, pages 174-181
ISBN: 978-989-758-671-2; ISSN: 2184-3228

of the SPARQL language, it is also necessary to know the information representation model of the knowledge base.

The natural language processing module uses a state-of-the-art statistical English parser, Universal dependencies parser - Stanza [2] and the semantic representation of the question is a simplified form of a discourse representation structure (Kamp and Reyle, 1993; Geurts et al., 2020).

The remainder of this paper is organized as follows. Section 2 presents the proposed approach on how a natural language question is transformed into a SPARQL query representation. The process of transforming the natural language input question into its discourse representation structures variants, based on syntactic analysis and dependencies tree, is explained in Section 3. The Query Ontology, which serves as a middle layer to adequately interpret the vocabulary used in the input question and the knowledge base, is detailed in Section 4. Afterwards, Section 5 explains the methodology used to choose the best semantic interpretation solution. In Section 6, the transformation process of the semantic interpretation solution, as a Query Ontology representation, into the corresponding SPARQL query is presented. The evaluation of the proposed Question-Answering System is detailed in Section 7. Finally, in Section 8, conclusions and future work are drawn.

# 2 QUESTION ANSWERING ARCHITECTURE

The strategy followed in transforming a natural language question into a SPARQL query representation, see Figure 1, includes a pipeline with three modules: Partial Semantic Representation, Pragmatic interpretation, and SPARQL generator.

The Partial Semantic Representation module has two steps. First, a dependency parser, Stanza[3], is applied to the question and, then, the resulting parser tree is transformed into a set of partial Discourse Representation Structures (DRSs), performed by the DRS process and detailed in Subsection 3.1. This module is language-independent and domain-independent, the Stanza is defined for many languages and the DRS process uses the Universal Dependencies Tags defined equally for all languages. As an illustrative example, consider the question *"Which children were born in 1900?"*. The dependency tree is shown in Figure 2 and the corresponding DRS obtained by the Partial Semantic Representation module is illustrated in Figure 3.

The next module, Pragmatic interpretation, rewrites the partial semantic representations of a question into a set of variant meanings of the question in the domain-specific context, see Subsection 3.3 for further details. This process uses an ontology-based domain representation detailed in Section 4, and a multi-objective optimization approach to choose the best interpretation of the question in the domain ontology context, further details are presented in Section 5. Regarding the illustrative question example, Figure 4 presents the solution obtained by this module.

Finally, a SPARQL Query Builder is applied to the Semantic Query Representation solution and generates the corresponding SPARQL query representation, more details are presented in see Section 6.

# 3 DRS BUILDER AND MAPPING DRS'S ON THE ONTOLOGY

Each partial question representation results from: first, applying a Universal Dependencies Parser[4] to obtain the syntactic dependency representation of the question, which contains the question terms properly classified and the existing syntactic relations between them; and, then, an algorithm is applied to rewrite the dependency parser into a simplified discourse representation structure. This algorithm is based on the Discourse Representation Theory (Kamp and Reyle, 1993) restricted to some discourse phenomena, such as establishing that variables are always existentially quantified and conditionals are not considered (Silva et al., 2023). These options make the question representation consistent with the domain-specific ontology structure. In the future, this algorithm can be extended or replaced by a new tool such as a semantic parser (Žabokrtský et al., 2020; Reddy et al., 2016).

To interpret all the DRS variants, each DRS is mapped on the Query ontology by representing this process as a constraint satisfaction problem.

## 3.1 DRS Builder from Dependency Parser

The partial semantic representation of the question, a simplified discourse representation structure (DRS)

[2]Stanza library https://stanfordnlp.github.io/stanza/

[3]https://stanfordnlp.github.io/stanza/

[4]Stanza library(https://stanfordnlp.github.io/stanza/) is used to perform the syntactic analysis of the question. Stanza is a Python natural language analysis package, containing a collection of accurate and efficient tools for the linguistic analysis of many human languages.

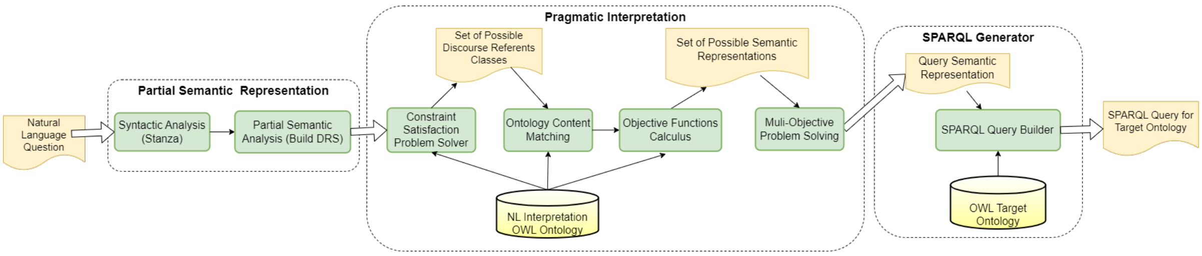


Figure 1: Natural Language Questions Representation as SPARQL Queries Architecture.

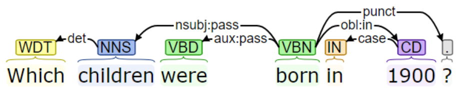


Figure 2: Dependencies Tree of the question *"Which children were born in 1900?"*.

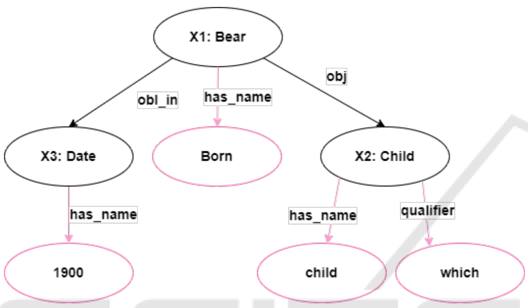


Figure 3: DRS of the question *"Which children were born in 1900?"*.

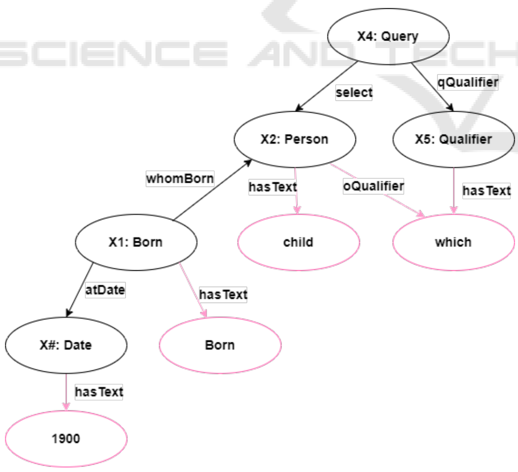


Figure 4: Query Ontology Solution of the question *"Which children were born in 1900?"*.

(Kamp and Reyle, 1993), is defined by using the dependency tags, where:

- Each noun phrase gives rise to a corresponding discourse referent.
- Each verb also gives rise to a corresponding discourse referent representing the event or action.
- Each subject, object and indirect object of a verb, and modifiers, such as propositional phrases, adjectives and adverbs, define the relations between discourse referents, namely the conditions.

A discourse entity is represented by a referent that is always existentially quantified and the information on the determinant and the lemma will be kept. A condition is defined as having a name composed of the lemma and the syntactic role, preposition or subject, and one or two discourse referents.

## 3.2 Constraint Satisfaction Problem

To map a DRS into an Ontology as a constraint satisfaction problem (CSP) the problem is defined by:

1. the set of variables $X = \{X_1, X_2, \ldots, X_n\}$, $X_i$, the discourse referents;
2. the set of variable domain values $D = \{D_1, D_2, \ldots, D_n\}$, $D_i$ the ontology classes;
3. the set of constraints $C = (C_1 \vee C_2 \vee \ldots) \wedge \ldots \wedge (C_m \vee C_n \vee \ldots)$, established by the conditions in the partial representation and the object properties in the ontology.

The set of variables $X$ is defined by the set of discourse referents and the corresponding domain for each variable is defined by the set of ontology classes, i.e., each variable $X_i$ can take any value from the nonempty domain $D$. Therefore, each variable can take any value from the ontology classes.

The set of constraints is defined by applying restrictions to the variables according to the conditions in the DRS and the object properties in the ontology, i.e., for each condition from discourse referent $X_i$ to discourse referent $X_j$ in the semantic representation, the conjunction of the following restrictions is added: for each object property in the ontology with domain $D_k$ and range $D_l$, the disjunction of the constraint $(X_i = D_k \wedge X_j = D_l)$ is added.

An evaluation of the variables is a function from a subset of variables to a particular set of values in the corresponding subset of domains. An evaluation $v$ satisfies a constraint $C_j$ if the values assigned to the variables satisfy the constraint $C_j$. A solution is an evaluation that is consistent, i.e. does not violate any of the constraints, and that is complete, i.e. includes all variables. Such an evaluation is said to solve the constraint satisfaction problem.

The Python tool `CP-SAT Solver`[5] is used to solve the constraint satisfaction problem.

### 3.3 Ontology Content-Matching

Ontology Content-Matching consists of obtaining the instances of ontology classes and properties that complete each CSP solution, i.e., consists of the matching process between the DRS's relations and the ontology properties for each CSP solution.

For each DRS condition, $C_k = (X_i, P_m, X_j)$ in a CSP solution, it is determined the ontology property $OP_n$ that links the ontology classes values of the variables $X_i$ and $X_j$, obtaining the ontology representation of the solution. If there exists more than one ontology property in these conditions, it means various ontology representations exist for the question.

Completing the ontology content-matching process results in a set of ontology solutions for each DRS.

## 4 QUERY ONTOLOGY

Query Ontology is a domain-specific knowledge base structure where concepts and properties reflect the vocabulary used in the input question and the knowledge base, the expressed information needs and the representation of the knowledge base, and the syntax and semantics of the SPARQL queries. In the following subsections, the Query Ontology is explained in more detail.

### 4.1 Ontology for Representing SPARQL Queries

The user input question concept is defined as a 'Query', in the Query Ontology. A `'Query'` has an `'Object'` or an `'Action'`, and a `'Qualifier'` that corresponds to the SPARQL structure:

```
SELECT (Qualifier(?Object) as ?result)
WHERE { (...) }
```

For instance, consider the question

*"Which are the children born in 1900?"*

[5] https://developers.google.com/optimization/cp, https://developers.google.com/optimization/reference/python/sat/python/cp_model

The representation of this question is defined as a 'Query' composed of a 'qualifier', with the value 'which', and an 'Object' corresponding to the child born in 1900.

If it is considered syntactic variants of the question, such as *"Which children were born in 1900?"*, then their representation in the Query Ontology is the same as the previous one.

Now, consider the following question:

*"How many children were born between 1900 and 1910, per woman?"*

This question has a modifier, the prepositional phrase 'per woman', reflecting a grouping to perform the answer. The intention of the user is to be informed of the total of children for each woman during a period of time.

To enable this type of query, the ontology must be extended to allow aggregations:

```
SELECT ?Aggregate (Qualifier(?Object) as ?result)
WHERE { (...) } GROUP BY ?Aggregate
```

The modifier 'per woman' is associated with this type of SPARQL query, where `'Aggregate'` corresponds to 'woman' and in the clause 'Where' there must exist a property in the ontology linking the `'Object'` to the `'Aggregation'`. The question 'Qualifier' is 'how many'.

The ontology extension to represent queries with aggregations is:

- Classes = {`Query; Object; Qualifier`}
- Object properties = {`Query select Object; Query aggregate Object; Query qQualifier Qualifier`}

This ontology will be populated from the content of the user input questions, and then from the ontology instances, a SPARQL query can be generated.

### 4.2 Ontology for Matching the Questions DRS

A user input question DRS is the semantic representation of the user question and is defined as: a set of discourse referents and a set of conditions on the discourse referents and constants.

The ontology matching is done by assigning an Ontology Class to each discourse referent and an Object Property or Data Property to each condition. Consider now the question:

*"Which are the children born?"*

Its DRS is composed of:

- Discourse referents = {`X1 - to be; X2 - Which; X3 - Child; X4 - Born`}

- Conditions = {`X1 subj X2; X1 obj X3; X4 obj X3`}

The corresponding Ontology classes to each entity referent are {`X1 - Query; X2 - Qualifier; X3 - Person; X4 - Born`}.

Therefore, the Query Ontology is extended to include the classes corresponding to the concepts 'Person' and 'Activity', as follows.

- Classes = {`Query; Object - Person; Qualifier; Activity - Born`}
- Object properties = {`Query select Object; Query aggregate Object; Query qQualifier Qualifier; Born whomBorn Person; Born fromBorn Person`}

The corresponding Ontology Object Properties to the DRS relations between referents should be { `X1 subj X2 - qQualifier; X1 obj X3 - select; X4 obj X3 - whomBorn` }

Note that the matching of `'X4 obj X3'` could be `'whomBorn'` or `'fromBorn'`, since both properties have the same domain and range classes.

#### 4.2.1 Ontology Properties Annotations

To enable preferences taking into account the question syntactic features, ontology properties are annotated with their syntactic preferences. For instance, the property `'whomBorn'` is annotated with the term `'obj'`, and the property `'fromBorn'` is annotated with the term `'obl_from'` since, for instance, in a question, the parents are indicated in a propositional phrase with the preposition `'from'`, as in *'John was born from a young mother'*.

#### 4.2.2 Ontology New Instances and Properties Inference Rules

Another problem that must be dealt with is when the question's syntactic structure is not as expected, such as when a question modifier is attached to another constituent as in the question:

*"How many children were born per woman?"*

This question DRS is composed of:

- Discourse referents = {`X1 - Born; X2 - Child; X3 - Women`}
- Conditions = {`X1 subj X2; X1 obl_per X3; X2 oQualifier 'how many'`}

The ontology matching of this DRS results in {`X1 - Born; X2 - Person; X3 - Person; X1 subj X2 - whomBorn; X1 obl_per X3 - fromBorn; X2 nQualifier 'how many'`}.

In this case, the question interpretation is not a `'Query'` but it is easy to guess what the query should be and fix the representation: an `'Object'` can have a data property `'oQualifier'` and a rule that should be triggered when a question does not have a `Query` in its representation:

```
oQualifier(A,Nql) -> new(Q,Query), select(Q,A),
                     new(Ql,Qualifier), has_name(Ql,Nql),
                     qQualifier(Q,Ql).
```

This rule creates a new instance of `'Query'` and a new instance of `'Qualifier'`, the query object is the instance that has a `'oQualifier'`. The question DRS is then updated to {`X1 - Born; X2 - Person; X3 - Person; X4 - Query; X5 - Qualifier; X1 subj X2 - whomBorn; X1 obl_per X3 - fromBorn; X2 nQualifier 'how many' - oQualifier; X4 select X2; X4 qQualifier X5`}. However, the interpretation of the question is not yet complete, since its representation does not have an aggregation.

The creation of an object property over the `'Activity'` will allow defining the aggregation relation between an activity and an object.

```
Activity activity_agregation Object
```

And adding the following inference rule will guarantee the existence of an aggregation over a `'Query'`:

```
activity_aggregation(A,O1), query(Q) -> aggregation(A,O1)
```

At this point, it is necessary to assure that there is a chain of properties linking the object selected to the aggregation. The following properties, with the corresponding annotation, are included in the Query Ontology.

```
Person child_of Person
    child_of:annot{"n per", "n for"}
Person ichild_of Person
    ichild_of:annot{"obl_per", "obl_for"}
```

The matching of `X2 obl_per X3` can be the intended relation `'ichild_of'`, since the annotation `"obl_per"` may be used when preferring the solution.

The representation of the question DRS is then {`X1 - Born; X2 - Person; X3 - Person; X4 - Query; X5 - Qualifier; X1 subj X2 - whomBorn; X1 obl_per X3 - activity_aggregate; X2 nQualifier 'how many' - oQualifier; X4 select X2; X4 qQualifier X5; X4 aggregate X3; X2 child_of X3`}.

#### 4.2.3 Ontology Class Annotations

The question vocabulary should enable preferences on the Ontology classes assigned to the question discourse referents. For instance, the term 'child' associated with the ontology class `'Person'` is a better choice than associating it with the Class `'Place'`. To take into account the vocabulary preferences, the ontology classes are annotated with 'lemmas' that better express their sense. For instance, the Ontology class 'Person' can be annotated with the lemmas: 'child', 'mother', 'father', 'couple', etc. This information can be imported from domain-controlled vocabularies or from general-purpose taxonomies, such as WordNet.

#### 4.2.4 Inference Rules to Model Preferences on Ontology Representations

The ontology can model representation preferences by defining SWRL (Semantic Web Rule Language) rules[6] such as:

```
select(?q,?o) -> entity_ok(?q, "1"^^xsd:int)
```

The data property 'entity_ok' contains the evaluation of a question discourse referent interpretation. The rule above evaluates a query to '1' if there is an object associated with the query.

The next rule, states that a query with an aggregation is better than a query with a select.

```
select(?q,?o), aggregate(?q,?o1)
                    -> entity_ok(?q, "2"^^xsd:int)
```

These preference rules are defined to give preference weights to the interpretation of domain-specific knowledge. For instance, the following rule states that, in question interpretations, where there is a person that has been born from a person on a date, is better than the interpretation where the person is only associated with a parent.

```
whomBorn(?a,?o1), fromBorn(?a,?o2)
            -> entity_ok(?a, "1"^^xsd:int)
```

The data property 'entity_nok' will also contain the evaluation of questions discourse referents interpretation, to express that some interpretation is not correct.

```
oQualifier(?o, ?nq), Qualifier(?q),
sentence(?q,?nn)   ->  entity_nok(?q, -1)
```

The above rule states that when a question interpretation has a qualifier attached to a noun, and there is another 'Qualifier' in the interpretation which results from a question discourse referent, then the question interpretation is evaluated as '-1', i.e., the interpretation should be passed over by another possible interpretation of the question.

#### 4.2.5 Language Model Classes and Properties

In the ontology for representing SPARQL queries, there are some classes and properties that are used for meta-reasoning on the question representations. These classes and properties should not be used to match the discourse referents to obtain the corresponding question interpretation. To formalize the classes and properties that can be used to model the questions, a special ontology class, named 'Language Model' is defined as a superclass of those classes and the same procedure is done for object and data properties.

[6]https://www.w3.org/Submission/SWRL/

## 5 CHOOSING THE BEST QUESTION INTERPRETATION

Given a DRS, a semantic interpretation of the question in the domain-specific ontology is defined as: the assigning of an ontology class to each discourse referent; the assigning of an ontology object property to each condition; and the assigning of an ontology data property to each one argument condition, such as 'has_text', 'has_name', 'has_value' or 'qualifier'.

Some of these assignments are more adequate than others, corresponding to a more adequate semantic interpretation of the question. A set of weighting rules are applied to choose the best interpretation of the question in the domain-specific ontology.

The time complexity of assigning an ontology class to each discourse referent can be as high as $\#C^{\#V}$, where $\#C$ is the number of ontology classes and $\#V$ is the number of discourse referents.

The question interpretation in the domain-specific ontology can be seen as a multi-objective optimization problem, where the weighting semantic, syntactic, and lexical rules define the set of objective functions.

### 5.1 Objective Functions Calculus

The ontology content-matching process produces a set of possible solutions. To reduce this set of solutions, it is applied a set of measures calculated from each solution that evaluates the vocabulary interpretation, the adequacy of the object properties to the original syntactic structure, the number of proper names, and the semantic adequacy of the entities in the solution. In the best case, this reduction can lead to only one ontology solution, corresponding to the best interpretation of the question in the domain-specific ontology.

The evaluation measures, i.e., the objective functions, considered are:

- Lexical

  $F_c$ = the number of discourse referents that have a class assigned matching the referent lemma.

$F_w =$ the number of discourse referents that have a lemma belonging to the annotations of the assigned class.

- Syntactic

  $F_{pp} =$ the number of object properties annotated with the lemma of the DRS condition, translated by the object property.

  $F_{pn} =$ the number of object properties annotated with the 'n'+lemma of the DRS condition, translated by the object property.

  $F_p = F_{pp} - F_{pn}$

- Semantic

  $F_{s_1}$ the sum of all `V`, for all `I` that `entity_ok(I,V)`

  $F_{s_2}$ the sum of all `V`, for all `I` that `entity_nok(I,V)`

  $F_n$ - the number of proper names

For each ontology solution, the calculus of the corresponding objective function values is performed, and the maximum value of each objective function, for all solutions, is calculated.

The set of best solutions is obtained by selecting those that verify the greatest number from all the maximal values of the six objective functions.

# 6 GENERATING A SPARQL QUERY

The SPARQL Query Generation consists of transforming the specific-domain ontology solution to a SPARQL Query format, in the context of the Portuguese Archival CIDOC-CRM population (Melo et al., 2023; Melo et al., 2020; Varagnolo. et al., 2022; Varagnolo et al., 2021).

The translation of the specific-domain solution into SQARQL Query is accomplished by applying a set of mapping description rules that for each individual in the Query Ontology defines: its representation in the Portuguese Archival CIDOC-CRM representation; or the corresponding part of the SPARQL scheme, as explained in Subsection 4.1. Table 1 presents some of the mapping rules regarding the examples illustrated throughout this work. For instance, given the Query Ontology property `whomBorn`, with domain `Born` and range `Person`, two variables are defined, one to address the domain and the other the range, it is intended to find in the Portuguese Archival CIDOC-CRM Population the persons that were born. For this purpose the CIDOC-CRM property `cidoc:P98_brought_into_life` is used. Therefore, the CIDOC-CRM representation of `?Born1 whomBorn ?Person1` is `?Born1 cidoc:P98_brought_into_life ?Person1`.

# 7 QUERY-ANSWER SYSTEM EVALUATION

The evaluation of the proposed system can be done on a dataset with natural language questions and the corresponding SPARQL representation. However, at the current stage of this project development, this dataset is still under construction.

The evaluation of the proposed system should be done by assessing the performance of the different modules: the Partial Semantic Representation correction, the correction of the Pragmatic Interpretation, the SPARQL Generator correction, and finally the question answers.

The Partial Semantic Representation has good results on the correction of the question representation, with time and space efficiency. The question analyses are done in milliseconds with no memory problems. The Pragmatic interpretation has also good results on the correction, but time complexity is a problem since the number of possible solutions grows exponentially with the number of classes, `nc`, and the number of discourse referents, `nr` ($nr^{nc}$). The SPARQL Generator is correct and is very efficient, regarding time and space. Finally, the evaluation of the question answers helps in the task of evaluating the SPARQL generator. For the time being, the evaluation is done manually by annotating a dataset with questions and the corresponding SPARQL answers, which is then used to evaluate the system's precision (number of correct answers).

# 8 CONCLUSIONS

A Query-Answer System to translate natural language questions to SPARQL queries was proposed. This system uses a dependency parser to analyse the natural language question. The parser analysis builds a simplified Discourse Representation Structure that is interpreted in an Ontology, which was also built with this purpose. The Query Ontology uses concepts close to the ones used by users in their input questions, ontology annotation on the classes to add specific vocabulary information, ontology annotations on the properties to provide syntactic role preferences, and semantic web rules (SWLR) to evaluate the adequacy of the question representation. The question interpretation is obtained by matching the DRS initial

Table 1: Mapping rules.

| #Rule | Property | Relation | CIDOC-CRM Triple |
|---|---|---|---|
| 1 | whomBorn | ?Born1 whomBorn ?Person1 | ?Born1 cidoc:P98_brought_into_life ?Person1 . |
| 2 | bornFrom | ?Born1 bornFrom ?Person1 | ({?Born1 cidoc:P97_from_father ?Person1} UNION {?Born1 cidoc:P96_by_mother ?Person1}) |
| 3 | child_Of | ?Person1 child_of ?Person2 | {?newBirth1 cidoc:P97_from_father ?Person2 .} UNION {?newBirth1 cidoc:P96_by_mother ?Person2 .} ?newBirth1 cidoc:P98_brought_into_life ?Person1 . |

representation on the Query ontology. With this strategy, each question can have many interpretations, and the choice of the best solution is resolved as a multi-objective problem, where the objective values are obtained for each solution using lexical, syntactic and semantic information. The evaluation of the proposed approach is still ongoing and includes the extension of the Query Ontology with more classes and properties to cover the DBpedia information and with new migration rules to a new target ontology, DBpedia, with the purpose of using the publicly available datasets in the evaluation of the system. This question-answering system is language-independent, except for the annotations on the query ontology that are language-dependent. To adapt this system to a new domain, the Query Ontology must be designed to represent the new domain questions concepts. A new set of migration rules must be written to transform the classes and properties of the solution into classes and properties of the target ontology of the new domain.

## ACKNOWLEDGEMENTS

This work is financed by National Funds through FCT - Foundation for Science and Technology I.P., within the scope of the project UIDP/04516/2020 (NOVA Laboratory for Computer Science and Informatics).

## REFERENCES


Geurts, B., Beaver, D. I., and Maier, E. (2020). Discourse Representation Theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2020 edition.

ICOM/CIDOC (2020). *Definition of the CIDOC Conceptual Reference Model*. ICOM/CRM Special Interest Group, 7.0.1 edition.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer Academic Publishers.

Meghini, C. and Doerr, M. (2018). A first-order logic expression of the cidoc conceptual reference model. *International Journal of Metadata, Semantics and Ontologies*, 13(2):131–149.

Melo, D., Rodrigues, I. P., and Koch, I. (2020). Knowledge discovery from isad, digital archive data, into archonto, a cidoc-crm based linked model. In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, KEOD - Volume 2*, pages 197–204. INSTICC, SciTePress.

Melo, D., Rodrigues, I. P., and Varagnolo, D. (2023). A strategy for archives metadata representation on cidoc-crm and knowledge discovery. *Semantic Web*, 14(3):553–584.

Reddy, S., Täckström, O., Collins, M., Kwiatkowski, T., Das, D., Steedman, M., and Lapata, M. (2016). Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Silva, J. Q., Melo, D., Rodrigues, I. P., Seco, J. C., Ferreira, C., and Parreira, J. (2023). An ontology-based task-oriented dialogue to create outsystems applications. *SN Computer Science*, 4(12):1–17.

Varagnolo., D., Antas., G., Ramos., M., Amaral., S., Melo., D., and Rodrigues., I. P. (2022). Evaluating and exploring text fields information extraction into cidoc-crm. In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KEOD,*, pages 177–184. INSTICC, SciTePress.

Varagnolo, D., Melo, D., and Rodrigues, I. P. (2021). A tool to explore the population of a cidoc-crm ontology. *Procedia Computer Science*, 192:158–167. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021.

Žabokrtský, Z., Zeman, D., and Ševčíková, M. (2020). Sentence Meaning Representations Across Languages: What Can We Learn from Existing Frameworks? *Computational Linguistics*, 46(3):605–665.