Question Answering over CIDOC-CRM based Knowledge Graphs

Nikos Gounakis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Prof. *Yannis Tzitzikas*

Thesis Supervisor: *Michalis Mountantonakis*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

University of Crete
Computer Science Department

Question Answering over CIDOC-CRM based Knowledge Graphs

Thesis submitted by
Student Nikos Gounakis
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Nikos Gounakis

Committee approvals: _____
Yannis Tzitzikas
Professor, Thesis Advisor

_____
Dimitris Plexousakis
Professor, Committee Member

_____
Kostas Magoutis
Associate Professor, Committee Member

Departmental approval: _____
Polyvios Pratikakis
Associate Professor, Director of Graduate Studies

Heraklion, November 2023

Question Answering over CIDOC-CRM based Knowledge Graphs

Abstract

CIDOC-CRM is a standard for documenting cultural heritage based on events that enables semantic interoperability among different sources of data in the Cultural Heritage (CH) domain. Despite the existence of several Knowledge Graphs (KGs) that use CIDOC-CRM, the problem of Question Answering (QA) over such graphs has not been explored extensively. Therefore, in this thesis we propose and evaluate a Radius-based QA pipeline over CIDOC-CRM KGs for mostly answering single-entity factoid questions ,while also covering confirmation questions. Specifically, we present a generic QA pipeline that consists of various models and methods, such as a keyword search model for identifying the entity of the question (and linking it to the KG), methods that rely on path expansion for building subgraphs of different radius (or depths) starting from the identified entity, i.e., for providing a context, and pre-trained neural models (based on BERT) for answering the question using the given context. Furthermore, since there are no available benchmarks over CIDOC-CRM KGs, we create (by using a real KG) an evaluation benchmark with 10,000 questions, i.e., 5,000 single-entity factoid, 2,500 comparative and 2,500 confirmation questions. For evaluating the QA pipeline, we use the 5,000 single-entity factoid questions and the 2,500 confirmation. Finally, we create a simple web application that enables the QA task to the users by utilizing the pipeline. Regarding the evaluation results, the QA pipeline achieves satisfactory results for factoid questions both in the entity recognition step (78% accuracy) and in the QA process (51% F1 score), while in confirmation 54% accuracy for entity detection and 76% accuracy for biased methods in the QA process, indicating the need for radius prediction.

Απάντηση Ερωτήσεων επί Γνωσιακών Γράφων που βασίζονται στο CIDOC -CRM

Περίληψη

Το CIDOC-CRM είναι ένα διεθνές πρότυπο για την τεκμηρίωση πολιτιστικών αγαθών το οποίο βασίζεται σε γεγονότα (event-based model) το οποίο επιτρέπει τη μοντελοποίηση, ανταλλαγή και συνάθροιση ετερογενών πληροφοριών πολιτισμικής κληρονομιάς και την επίτευξη σημασιολογικής διαλειτουργικότητας. Παρά την ύπαρξη πολλαπλών Γνωσιακών Γράφων (Knowledge Graphs) που χρησιμοποιούν το CIDOC-CRM , το πρόβλημα της απάντησης ερωτήσεων (QA) πάνω σε τέτοιους γράφους δεν έχει διερευνηθεί εκτενώς, εξαιτίας α) της πολυπλοκότητας του μοντελου CIDOC-CRM , β) της έλλειψης ροών εργασιών για την απάντηση ερωτήσεων για event-based μοντέλα, και γ) της απουσίας συλλογών για την αξιολόγηση μηχανισμών απάντησης ερωτήσεων που αφορούν Γνωσιακούς Γράφους βασισμένους σε CIDOC-CRM. Για την αντιμετώπιση αυτών των προβλημάτων, στην παρούσα εργασία προτείνουμε και αξιολογούμε μια ροή εργασιών για απάντηση ερωτήσεων πάνω σε γνωσιακούς γράφους που έχουν μοντελοποιηθεί με τη χρήση του CIDOC-CRM, η οποία βασίζεται στην ακτίνα (βάθος) του γράφου. Η μέθοδος έχει σχεδιαστεί κυρίως για ερωτήματα που αφορούν ένα συγκεκριμένο γεγονός για μία οντότητα (single factoid questions), και δευτερευόντως για ερωτήματα που αφορούν την απάντηση ερωτήσεων επιβεβαίωσης (confirmation questions). Συγκεκριμένα, παρουσιάζουμε μία γενική ροή εργασιών που αποτελείται από διάφορα μοντέλα και μεθόδους, όπως ένα μοντέλο αναζήτησης λέξεων-κλειδιών για τον εντοπισμό της οντότητας της ερώτησης (και τη σύνδεσή της με τον γνωσιακό γράφο), μεθόδους που βασίζονται στην επέκταση του μονοπατιού της οντότητας για τη δημιουργία υπογράφων διαφορετικής ακτίνας (ή βάθους) ξεκινώντας από την αρχική οντότητα με σκοπό την δημιουργία ενός κειμένου σε φυσική γλώσσα, και προ-εκπαιδευμένα νευρωνικά μοντέλα (με βάση το BERT) για την απάντηση της ερώτησης χρησιμοποιώντας το προαναφερθέν κείμενο.

Επιπλέον, δεδομένης της έλλειψης συλλογών αξιολόγησης για την αξιολόγηση ερωτήσεων/απαντήσεων που αφορούν CIDOC-CRM γνωσιακούς γράφους, παρουσιάζουμε τη δημιουργία μιας συλλογής αξιολόγησης (χρησιμοποιώντας έναν πραγματικό γράφο με δεδομένα από μουσεία) που περιλαμβάνει 10.000 ερωτήσεις (και απαντήσεις). Συγκεκριμένα 5.000 ερωτήσεις που αφορούν ένα συγκεκριμένο γεγονός για μία οντότητα, 2.500 συγκριτικές ερωτήσεις και 2.500 ερωτήσεις επιβεβαίωσης. Για την αξιολόγηση της ροής εργασιών, χρησιμοποιούμε τις 5.000 ερωτήσεις που αφορούν γεγονότα για μία οντότητα και τις 2.500 ερωτήσεις επιβεβαίωσης. Όσον αφορά τα αποτελέσματα της αξιολόγησης, η ροή εργασιών επιτυγχάνει ικανοποιητικά αποτελέσματα για τις ερωτήσεις γεγονότων για μία οντότητα, τόσο στο στάδιο της αναγνώρισης οντοτήτων (78% ακρίβεια) όσο και στη διαδικασία απάντησης ερωτήσεων (51% F1 score), ενώ για τις ερωτήσεις επιβεβαίωσης τα αντίστοιχα αποτελέσματα είναι 54% ακρίβεια για τον εντοπισμό οντότητας και 76% ακρίβεια για μια προκατειλημμένη (biased) μέθοδο που γνωρίζει εκ των προτέρων το βάθος, συμπεραίνοντας την ανάγκη ενός μηχανισμού για την πρόβλεψη του βάθους της ακτίνας του υπο-γράφου για κάθε

απάντηση. Τέλος, δημιουργούμε μια απλή διαδικτυακή εφαρμογή που επιτρέπει στους χρήστες να κάνουν ερωτήσεις σε γνωσιακούς γράφους CIDOC-CRM χρησιμοποιώντας την ροή εργασιών που αναφέραμε.

Ευχαριστίες

Ευχαριστώ τον επόπτη καθηγητή μου Γιάννη Τζίτζικα και τον Μιχάλη Μουντακη για την πολύτιμη υποστήριξη και καθοδήγηση στα πλαίσια της ολοκλήρωσης αυτής της εργασίας. Επίσης ευχαριστώ και το Ινστιτούτο Πληροφορικής του ΙΤΕ για την υποτροφία που μου προσέφερε κατα τη διάρκεια της μεταπτυχιακής μου εργασίας. Τέλος ευχαριστώ τους φίλους μου και την οικογένεια μου για την συνεχή στήριξη όλα αυτα τα χρόνια.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, the digitization and scientific documentation of cultural heritage objects is a research field that has grown significantly, since it is crucial to curate, restore and preserve cultural artefacts [17]. To model these cultural objects, formal models have been created, such as the CIDOC Conceptual Reference Model (CIDOC-CRM); an ISO 21127 standard event-based ontology for the cultural domain [10] that has been widely used [1, 40] for offering interoperability between the Cultural Heritage (CH) domain metadata standards and ontologies. However, due to its complex (event-based) nature, it is not an easy task for non-experts to exploit the data expressed through the CIDOC-CRM model, as they have to query a KG directly. There are other ways of easy accessing like applying a faceted search, but in this study we will experiment with a more user-friendly approach, and that is to provide a Question Answering (QA) service, where any user can express a natural question (e.g., such as those in [4]) were the answer will be provided through a dedicated QA pipeline. Indicatively, such QA pipelines can be used for enabling users to ask questions through text or voice (e.g., chatbots) [41] and to retrieve answers from a Knowledge Graph. For example, suppose a scenario where a museum visitor stands in front of a painting [3] and desires to ask more questions about the painting, such as about its creator, the history, and any other question relevant to the painting.

However, there are no evaluated pipelines for QA over CIDOC-CRM [40], especially, due to the following difficulties: a) CIDOC-CRM is a complex model, b) lack of QA pipelines for (complex) event-based ontologies, and c) absence of QA benchmarks for CIDOC-CRM based KGs. In particular, regarding a) and b), CIDOC-CRM has a complex structure as it is an event-centric ontology with a plethora of classes and associations structured in specialization hierarchies, which makes it difficult to apply successful QA techniques that are applicable for simpler ontologies/models like DBpedia [21] (e.g., [28]). Therefore, one has to exploit various deductions from the KG which is not supported by the current approaches. Regarding c), there are no available benchmarks for evaluating such QA tasks that support CIDOC-CRM KGs [40]. For tackling these limitations, in this thesis we

focus on answering the following research questions:

• RQ1: How effective is an existing generic QA pipeline that works on non-event based models, on CIDOC-CRM?

• RQ2: How to traverse the CIDOC-CRM KG for creating the subgraph that contains the desired answer, given that: a) subgraphs of a small radius may not contain the desired answer and b) subgraphs of a large radius may contain redundant data?

Concerning our contribution, since there is a high need for facilitating access to cultural knowledge through interactive pipelines (and applications), we provide a radius-based QA pipeline for answering single-entity factoid questions mainly, while also it can answer confirmation questions. In particular, i) we explain why existing generic QA pipelines, such as Elas4RDF-QA [28], are not (in their current form) sufficient for CIDOC-CRM KGs, ii) we propose an extension of Elas4RDF-QA, for being compatible with event-based models (by focusing on CIDOC-CRM), by supporting different entity path expansion methods for the creation of subgraphs (for text construction), and iii) we construct an evaluation benchmark with 10,000 question-answer pairs, called CIDOC-QA, by using the real Smithsonian American Art Museum (SAAM) KG [36]. It includes 5,000 single-entity factoid questions, 2,500 comparative and 2,500 confirmation questions. Finally, iv) we use the mentioned 5,000 single-entity factoid and 2,500 confirmation questions for evaluating the effectiveness and efficiency of the proposed pipeline while keeping most of the focus on the factoid questions.

As regards the novelty, to the best of our knowledge it is the first work that offers a) a QA pipeline for answering natural questions over any CIDOC-CRM KG (or event based ontology) and b) an evaluation benchmark of QA over CIDOC-CRM KGs.

The results of our evaluation show that through the path expansion methods, it is feasible to answer questions that require a certain radius from a starting resource. Indicatively, we achieved 78% accuracy for the entity recognition step for factoid questions and 54% for confirmation while 51% F1 score for the full QA process in factoid questions (+28.4% comparing to the original Elas4RDF-QA pipeline) and 76% for a known depth method in confirmation . Finally, the average query time is approximately 1 second for factoid and 9 seconds for the confirmation.

The rest of the thesis is described as follows: Chapter 2 discusses the related work. Chapter 3 presents the evaluation benchmark, while Chapter 4 introduces the proposed QA pipeline including methods for subgraph creation and discuss the requirements. Chapter 5 introduces the functionality of the web application that utilizes the pipeline in order to bring the qa task with ease to the users. Chapter 6 presents comparative results for the proposed methods. Finally, Chapter 7 concludes the thesis and Chapter 8 identifies directions for future research.

It is also important to mention that this thesis is an extension of our following publication [16]:

Nikos Gounakis, Michalis Mountantonakis, and Yannis Tzitzikas. Evaluating a radius-based pipeline for question answering over cultural

(cidoc-crm based) knowledge graphs. In Proceedings of the 34th ACM Conference on Hypertext and Social Media, HT '23, New York, NY, USA, 2023. Association for Computing Machinery

Chapter 2

Background & Related Work

In this Chapter, we present the background and then describe approaches for a) QA over RDF KGs, b) for QA over event-based KGs (including CIDOC-CRM KGs), and c) NLP tasks over CIDOC-CRM.

2.1 Knowledge Graphs & RDF

Knowledge graphs play a crucial role in the organization and representation of information in various domains. With the exponential growth of data in today's digitally connected world, the need for efficient and effective methods of knowledge representation has become increasingly important. The term "Knowledge Graph" is widely used to refer to a large-scale semantic network that consists of entities and concepts, as well as the semantic relationships among them. These knowledge graphs are structured using representation languages such as Resource Description Framework and RDF Schema. The Resource Description Framework has become a standard format for many publicly available knowledge graphs, including DBpedia [21] and Wikidata [43]. The knowledge graphs consist of triples, which are composed of subject-predicate-object statements. These triples allow for the expression of complex relationships between entities, providing a rich and interconnected representation of knowledge. Knowledge graphs are not just limited to a specific domain but encompass a wide range of topics, making them versatile and applicable in various fields. A simple example can be seen in figure 2.1

2.2 CIDOC-CRM

CIDOC-CRM is an ISO standard ontology that serves as a common vocabulary for representing cultural heritage information. It is widely used in domains related to cultural heritage, such as museums, archives, and libraries, as a means of integrating data from various sources. This ontology provides definitions and a formal structure for representing both implicit and explicit knowledge included in cultural heritage documentation. The CIDOC CRM has proven to be particularly

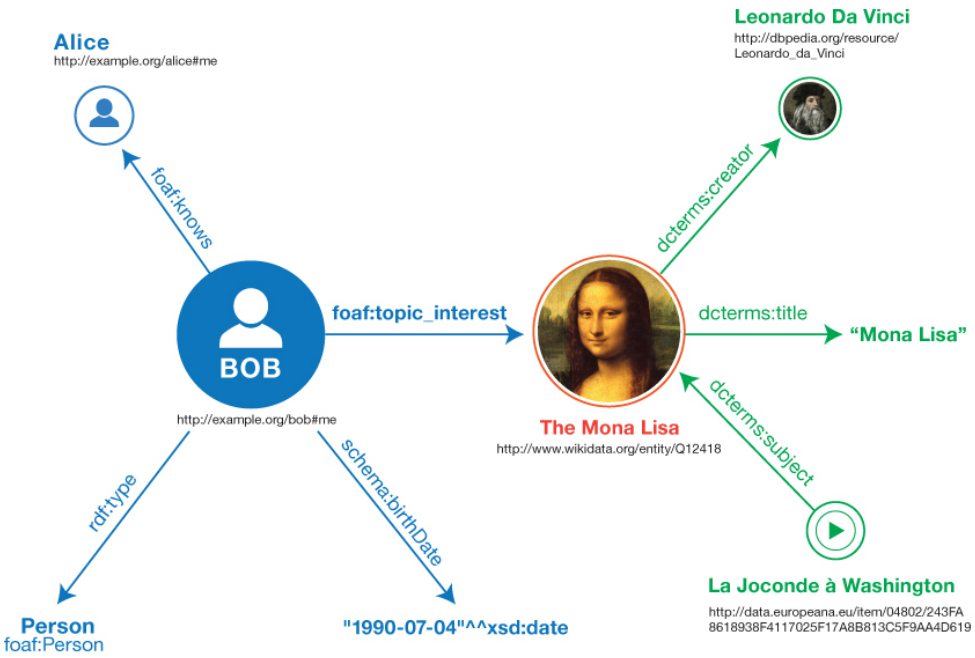

Figure 2.1: A graph of RDF triples (W3C)

useful in the archaeological sector, as it allows for the structured representation of data that goes beyond simple keywords [42]. However, while CIDOC-CRM offers immense potential for organizing and exchanging cultural heritage information, it comes with its own set of challenges when it comes to retrieving information from CIDOC CRM knowledge graphs. The event based structure of the CIDOC CRM, which enables n-ary relationships between entities, adds complexity to the retrieval process. The properties and relationships defined within the CIDOC CRM can be difficult to navigate and query efficiently, especially for users who are not familiar with the ontology. You can find more information in the following link `https://www.cidoc-crm.org`. A CIDOC-CRM graph example can be seen in figure 2.2.

2.3 Challenges for accessing RDF data

In the ever-evolving landscape of RDF data retrieval and presentation, addressing challenges related to user-friendly access and versatile information requirements is paramount. Some of the challenges that need to be addressed in displaying/presenting RDF data are as follows: 1) **Lack of a Distinct Retrieval and Presentation Unit.** In the realm of RDF data, the concept of a document or web page, as commonly encountered in web searches, is absent. RDF data are interconnected and the information is not in one place. [7]. 2) **Absence of a Clearly Defined Information Requirement** The user query represents

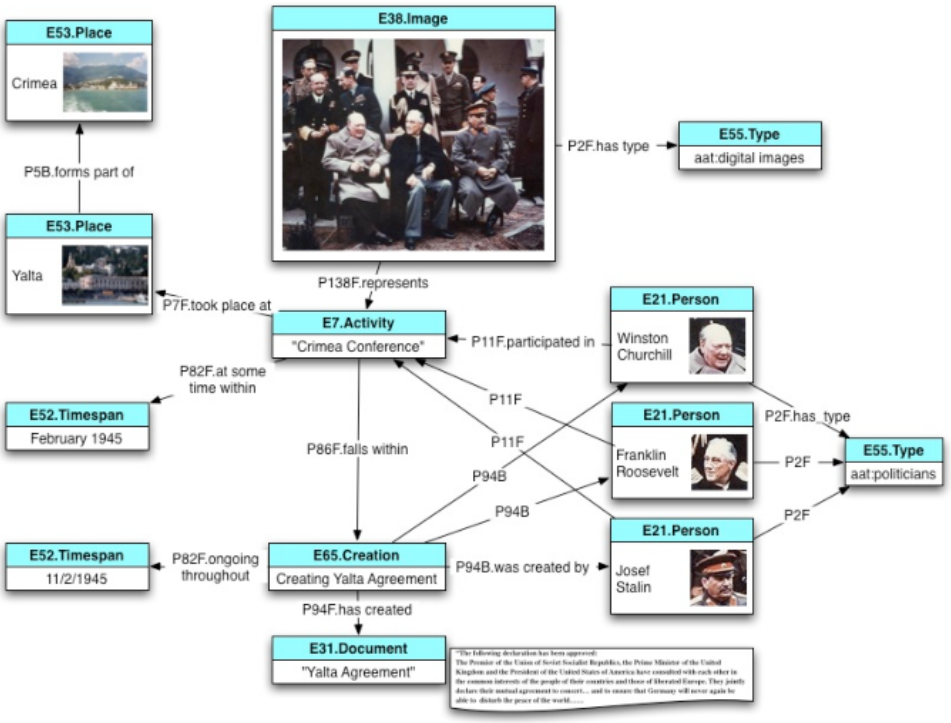

Figure 2.2: A CIDOC-CRM graph

an initial effort to articulate their information requirements. These needs can vary widely, ranging from a single piece of information or a list of entities to understanding the relationships between entities or pursuing exploratory inquiries, among other possibilities. 3) **Incomplete Dataset.** The dataset is typically incomplete, and retrieved triples may lack completeness. 4) **A universal presentation approach does not suit all types of information requirements.** A standardized method for displaying RDF results across various query types has yet to be established, indicating that a one-size-fits-all approach may not be applicable to all potential demands. Diverse information requirements necessitate distinct result presentation approaches.

The following study [19] suggests that by employing multiple methods (Triple view, Entity view, Graph view, QA view) that focus on different segments of top results and their context increases the chances of finding valuable information that meets the user's needs. Considering the above and the domain research needs and also the fact that Elas4RDF QA approach [28] in its current form is not enough for CIDOC-CRM, it is worth to investigate a QA approach over CIDOC-CRM, as the rest of the approaches (Triple view, Entity view etc.) are covered through Elas4RDF [19].

2.4 QA over RDF KGs

There is an increasing trend for QA approaches over KGs [9], which can be divided in 3 categories [37]: i) template based approaches [20, 2], i.e., matching questions to SPARQL templates, ii) semantic parsing methods [22, 14], i.e., translating questions into logic query forms, and iii) information retrieval-based methods [37], i.e., extracting the entity and words of each question, and trying to find the best candidate answer (e.g., by ranking the different triples/paths). The proposed approach, which extends Elas4RDF-QA, is hybrid, i.e., it combines Information Retrieval, SPARQL and Neural Networks techniques. Concerning the KGs that are used from QA systems, there are usually popular KGs, such as DBpedia [21] and Wikidata [43], e.g., see QAnswer [8], Platypus [38] and Elas4RDF-QA [28].

Regarding Elas4RDF-QA [28] Nikas C. et al. have developed a Question Answering pipeline that utilizes the DBpedia Knowledge Graph. The pipeline employs a keyword search system over RDF [18] to retrieve the top 10 entities relevant to the question along with their accompanying textual descriptions. Subsequently, the pipeline employs a two-stage answer type prediction using Deep Neural Networks to determine the type of question. Based on the determined question type, additional text is generated from triples matching the question type. Finally, the generated text is appended to each entity's textual description, and the final answer is extracted using the RoBERTa model [23].

2.5 QA over event-based KGs

Concerning event-based QA, [37] combines information retrieval methods and similarity functions for detecting the best path for answering a question. Also, [33] exploits KG embeddings for finding the best answer for multi-hop QA. Regarding event QA collections, there exists the EVENT-QA [34] and LC-QUAD [12] benchmarks, that use the EventKG [15], DBpedia [21] and Wikidata [43] KGs, accordingly. These benchmarks contain thousands of complex and diverse questions, however, the complexity of the queries in the dataset is restricted in a maximum of two relations (two hops). Moreover, the MetaQA [44] is a large scale multi-hop collection (from one to three hops) with more than 400k questions in the movie domain.

Concerning CIDOC-CRM, [35] performs QA over genealogical graphs expressed in GEDCOM format. These graphs are converted into CIDOC-CRM subgraphs using the classes Person (E21) for individuals, and Group (E74) for families and also events and properties such as "birth" and "brought into life." Then text passages are generated from each subgraphs using a knowledge-graph-to-text DNN model along with knowledge-graph-to-text template-based methodology. From the generated passages they generate question-answer pairs using a rule based approach. The process culminates in the creation of a SQuAD format dataset in order to fine

tune BERT [22] and create Uncle BERT, a Deep Neural Network for Question Answering over genealogical data in GEDCOM format that have been processed and converted in CIDOC-CRM triples and then to a text passage in which the system extract answers for a given question. In [6] the authors proposed a logic-based QA system over CIDOC-CRM. The input questions are converted into SPARQL queries, which are then run on a knowledge base adhering to the CIDOC ontology. To achieve this, the questions go through a rule-based syntactic classification module that operates using an Answer Set Programming system. The initial stages of processing involve natural language processing steps such as identifying named entities, tokenization and part-of-speech tagging, and dependency parsing. Following these steps, a template matching mechanism takes over to categorize questions from a syntactic perspective and extract the question terms necessary for forming the query to retrieve the answer and form a natural text response. However, they only provide a performance evaluation of the approach.

2.6 NLP Tasks over CIDOC-CRM KGs

The approach involves the transformation of the input question into a three-level syntactic representation. The representation is then categorized by a logical template system, which maps the template to an intent that precisely identifies the purpose of the question. The intent is transformed into a query for the KG, and after its execution, the result is transformed into a natural language answer. Apart from QA, there are few approaches that use NLP techniques over CIDOC-CRM [40]. Firstly, TEXTCROWD [13] offers part-of-speech tagging and Named Entity Recognition for Italian archaeological reports and produces the output using CIDOC-CRM, whereas [11] extracts entities and relations from Chinese cultural texts and uses CIDOC-CRM classes for classifying the extracted entities. Furthermore, in [24] text classification and extraction is performed over Portuguese National Archives records, for modelling the extracted information data by using CIDOC-CRM.

2.7 Comparison & Novelty

Comparing to QA approaches over CIDOC-CRM, we provide a general QA pipeline that can be adjusted for any CIDOC-CRM KG, and not for a specific domain, e.g., genealogical data [35], whereas we create and convert subgraphs to texts instead of transforming the question into a SPARQL query [6]. As regards event-based evaluation collections, the existing ones are not applicable for CIDOC-CRM KGs, i.e., they include Knowledge Graphs that have not been modelled through CIDOC-CRM. Moreover, they contain questions that need paths of length 2 to be answered, whereas we cover also questions for larger paths (i.e. of a large radius). On the contrary, they offer a larger diversity (i.e., questions are dissimilar to others), whereas we mainly use similar (the same template) questions for different

entities/events. Regarding the novelty, to the best of our knowledge it is the first work that offers a) a generic QA pipeline for answering natural questions over any CIDOC-CRM KG (by also supporting entity recognition and linking), and b) an evaluation benchmark of QA over CIDOC-CRM KGs, including thousands of questions over a real KG, which can be useful for researchers that desire to study the same problem in the future.

Chapter 3

Evaluation Benchmark

In this chapter we present and discuss about the evaluation benchmark.

3.1 CIDOC-QA: Evaluation Benchmark over CIDOC-CRM KGs

Since there are no available benchmarks for QA over such KGs [40], we create a benchmark for evaluating CIDOC-CRM QA approaches. Specifically, we use the Smithsonian American Art Museum [36] (SAAM) KG, which contains 2,792,865 triples and 720,767 entities, including thousands of artworks and artists (e.g., paintings, sculptures, photographies). The objective is to focus on the radius complexity, i.e., for including questions that need subgraphs of different radius for being answered. For automating the process of creating the questions, we created 20 question templates (each one having 500 questions), for three question types. Specifically, Table 3.1 shows each template, grouped by their question type and radius (from radius 1 to 4), the number of questions of each template, and the average words for each question and answer.

The benchmark is rule-based generated, by sending SPARQL queries to the endpoint of the SAAM KG (`https://triplydb.com/smithsonian/american-art-museum/`). The evaluation benchmark, the code for creating the questions and more details are available in `https://github.com/NicolaiGoon/CIDOC-QA-BENCHMARK/`.

Fig. 3.1 shows the SPARQL query of Q9. Regarding the output of this process, an indicative benchmark entry of Q9 (Factoid) is shown in Fig. 3.2 while also for Q19 (Confirmation) in Fig 3.3 and Q14 (Comparative) in Fig. 3.4.

How will we use this benchmark in this thesis: We decided to focus most in investigating techniques for factoid single-entity questions since this is a fundamental step for more complex questions, therefore the evaluation is conducted in both confirmation and factoid questions with a slight different methods while we use the question templates Q1-Q10 (factoid) and Q16-Q20 (confirmation) of Table 3.1.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cidoc: <http://www.cidoc-crm.org/cidoc-crm/>

SELECT ?artwork ?label ?place WHERE {
    ?artwork rdfs:label ?label .
    ?artwork cidoc:P108i_was_produced_by ?production .
    ?production cidoc:P14_carried_out_by ?actor .
    ?actor cidoc:P92i_was_brought_into_existence_by ?existence .
    ?existence cidoc:P7_took_place_at ?placeLabel .
    ?place rdfs:label ?placeLabel .
}
```

Figure 3.1: The SPARQL Query of template Q9

```
    "id": 3501,
    "question": "Which is the birth place of the creator of Head of a Woman in Jerusalem?",
    "entity": "<http://data.americanart.si.edu/object/id/1983.95.194>",
    "answer": ["Pittsburgh, Pennsylvania, United States"],
    "type": "single-entity factoid",
    "radius": "4"
```

Figure 3.2: An indicative JSON entry for the template Q9 (Factoid)

```
    "id":9002,
    "question": "Was the production of Rockbound Coast, Cape Ann ended before 1900?",
    "entity":  "<http://data.americanart.si.edu/object/id/1910.9.14>",
    "answers": ["No"],
    "type":"confirmation",
        "property":"production_year",
    "radius": 3
```

Figure 3.3: An indicative JSON entry for the template Q19 (Confirmation)

```
"id": 7003,
 "question": "Which Artwork produced first, Autumn Fields or Moonlight?",
 "entity": [
     "<http://data.americanart.si.edu/object/id/1983.90.210>",
     "<http://data.americanart.si.edu/object/id/1909.10.2>"
 ],
 "answers": [
     "Moonlight"
 ],
 "type": "comparative",
 "property": "artCreatedFirst",
 "radius": 3
```


Figure 3.4: An indicative JSON entry for the template Q14 (Comparative)

3.1.1 Single Entity Factoid Questions Q1-Q10)

A single entity factoid question is a type of question that seeks a specific piece of information about a particular entity, such as a person, place, or thing. These types of questions typically require a concise, factual answer that provides a clear and accurate piece of information about the entity in question. For example, a single entity factoid question might ask, "What is the capital of France?" The answer to this question is a simple fact, "Paris." In our benchmark, there are 5,000 questions from 10 templates (from radius 1 to 4), and they contain questions about a single artwork or artist.

3.1.2 Comparative Questions (Q11-Q15)

Comparative questions are a type of question that seek to establish a relationship between two or more entities, by comparing and contrasting their similarities and differences. These types of questions often use comparative adjectives, such as "better," "worse," "more," or "less," to compare the attributes of the entities being compared. For example, a comparative question might ask, "Is running more effective than cycling for weight loss?" This type of question allows for a more nuanced and detailed response, as it requires an analysis of the pros and cons of each entity being compared. Our benchmark contains 2,500 questions from 5 templates (from radius 1 to 4), including comparative questions about either pairs of art works or pairs of artists.

3.1.3 Confirmation Questions (Q16-Q20)

Confirmation questions are a type of question that seeks to verify or confirm the understanding of the speaker or listener regarding a specific topic or idea. These questions are designed to elicit a response that clarifies or affirms the accuracy and

completeness of the information being conveyed, thereby reducing the likelihood of misunderstanding or miscommunication. A simple paradigm (of radius 2) from our benchmark is as follows: **Is Cullen Yates the creator of Rockbound Coast, Cape Ann?** where the answer could be a "yes" or "no" (in this case the correct answer is "yes"). There are 2,500 confirmation questions from 5 templates (from radius 1 to 4), about artworks and artists. Each template includes 250 questions with answer "Yes" and 250 with answer "No".

ID	Question Template	Radius	Number of Questions	Question words length	Answer Words length
Single Entity Factoid Questions (5000 Questions)					
Q1	Which is the type of {Art Work}?	1	500	8.66	1.45
Q2	What material was used for creating the {Art Work}?	1	500	7.65	3.59
Q3	Who gave the {Art Work} to the museum?	1	500	7.71	7.00
Q4	Who is the creator of {Art Work}?	2	500	7.65	2.37
Q5	Which is the birth place of {Artist}?	2	500	8.32	3.57
Q6	When the production of {Art Work} started?	3	500	4.76	1.00 (date)
Q7	When the production of {Art Work} ended?	3	500	7.69	1.00 (date)
Q8	Which is the nationality of the creator of {Art Work}?	3	500	10.67	1.00
Q9	Which is the birth place of the creator of {Art Work}?	4	500	11.59	4.11
Q10	Which year died the creator of {Art Work}?	4	500	8.70	1.00
Comparative Questions (2500 Questions)					
Q11	Which painting is taller {Painting 1} or {Painting 2}?	1	500	13.04	4.05
Q12	Who has more art works in the museum, {Artist 1} or {Artist 2}?	1	500	12.64	2.34
Q13	Who was born first, {Artist 1} or {Artist 2}?	2	500	9.66	2.45
Q14	Which Artwork produced first, {Art Work 1} or {Art Work 2}?	3	500	15.39	4.75
Q15	Who was born first, the creator of {Art Work 1} or {Art Work 2}?	4	500	21.56	2.46
Confirmation Questions (2500 Questions)					
Q16	Was {Art Work} given as a gift to the museum?	1	500	10.57	1.00 (Yes/No)
Q17	Had the {Material} used for the production of {Art Work}?	1	500	14.70	1.00 (Yes/No)
Q18	Is {Artist} the creator of {Art Work}?	2	500	8.98	1.00 (Yes/No)
Q19	Was the production of {Art Work} ended before 1900?	3	500	9.70	1.00 (Yes/No)
Q20	Is {Place} the birth place of the creator of {Art Work}?	4	500	14.72	1.00 (Yes/No)

Table 3.1: Evaluation Benchmark: Question templates (in total 10000 questions) and statistics of the benchmark

Chapter 4

The Proposed QA Pipeline

In this chapter we present the proposed pipeline. First we analyse the prerequisite steps and then we discuss about each step of the pipeline. We describe a QA pipeline that can be used over any CIDOC-CRM based KG. The steps of the QA pipeline are illustrated in Figure 4.2 through the use of a running example, i.e., for the question "Which is the birth place of the creator of The Starry Night" (a painting of Vincent Van Gogh) that requires to traverse a subgraph of radius 4 to be answered.

4.1 Prerequisite Steps

For any given CIDOC-CRM KG, we need to perform two prerequisite steps for creating the required components of the QA pipeline (lower part of Figure 4.2).

4.1.1 Context and Requirements

In this thesis, we extend the generic Elas4RDF-QA pipeline [28], for being able to answer single-entity factoid questions mainly, while also confirmation questions, over any CIDOC-CRM KG. Elas4RDF-QA uses a Keyword Search over RDF [18], SPARQL queries for text generation (to be used as a context), BERT for Answer Extraction (AE), and Answer Type Prediction (ATP). In the Elas4RDF pipeline the ATP component [27] needs to be fine tuned in the current ontology in order to predict the question's answer type (answer type matches an ontology class). Regarding its methodology, Elas4RDF [28] has configured the keyword search to include the value of the property **rdfs:comment** for each entity. This property is utilized to create a textual description along with additional text generated from the Entity Expansion module, which is derived from the direct triples of the entity that match the answer type.

Although Elas4RDF has been successfully used for DBpedia, it is not sufficient in its current form for CIDOC-CRM KGs for the following reasons:

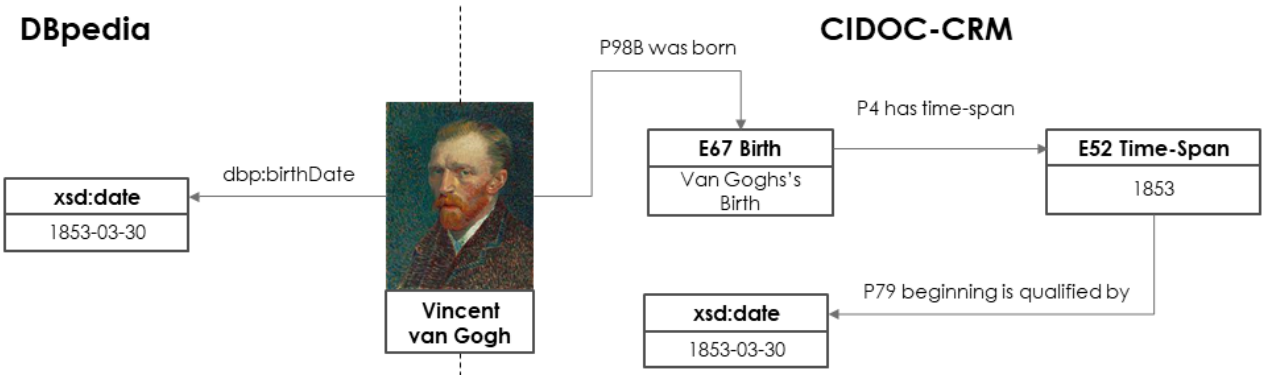

Figure 4.1: Van Gogh birth date representation in DBpedia vs CIDOC-CRM

• *Named Entity Recognition and Linking.* The Elas4RDF-QA pipeline recognizes DBpedia entities, by using indexing mechanisms over DBpedia, that use the suffix of the URI of each entity. In DBpedia, the suffix of the URIs is informative, however, this is not the case for several KGs (including Wikidata and CIDOC-CRM KGs like SAAM), since they use identifiers in their URIs. Therefore, indexes should be constructed by using the labels of each URI.

• *Direct Triples versus Large Paths.* The Elas4RDF-QA pipeline can answer questions that are described in the direct triples of an entity (i.e., direct neighbors). However, in event-based models usually larger paths (i.e., multi-hops) need to be traversed for answering most of the questions. For instance, it is simple to find the birth date of Vincent Van Gogh by using DBpedia, as the property "dbo:birthDate" is directly connected to that entity (see left part of Figure 4.1). On the other hand, for finding the birth date by using CIDOC-CRM, it requires to follow a larger path, since it is modelled as an event (see right part of Figure 4.1).

The requirements for enabling QA over any CIDOC-CRM KG follow: a) offer Entity Recognition for any CIDOC-CRM KG, by focusing on indexing the labels of the URIs and not only of their suffix, and b) support methods for constructing the context from subgraphs even of a large radius, starting from an entity/event, i.e., since we desire to answer questions requiring to follow paths of a large radius, such as those in Table 3.1. Concerning the possible effectiveness problems, they are mainly related to *RQ2*, i.e., the event-oriented nature of CIDOC-CRM requires the creation of small or large subgraphs (see Steps B1 and B2) to be converted to text and serve as context for answering a query q. Regarding the efficiency, we expect more time will be needed for constructing the subgraph(s) in comparison with the Elas4RDF QA pipeline utilized in DBpedia.

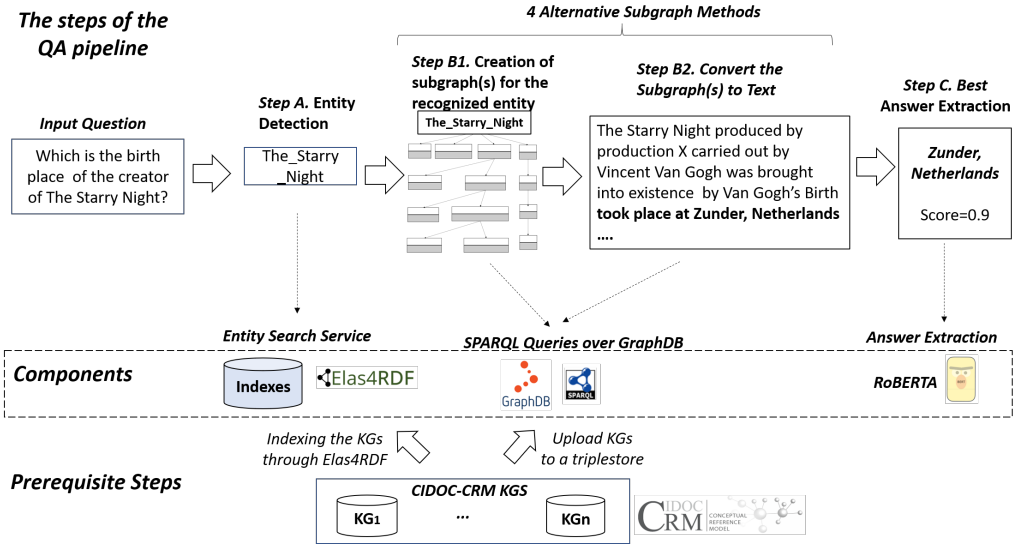

Figure 4.2: The proposed QA pipeline over any CIDOC-CRM KG for single entity factoid questions and a running example

4.1.2 Indexes for Enabling Entity Detection

The proposed process is based on [18] which is a system for Elastic keyword search over RDF. The first step is to create an index from the desired KG(s) using the Elas4RDF index service [18]. The objective is to load the index in an elastic search instance and use the Elas4RDF search upon it, for enabling the retrieval of the top-K entities (and of their URI) for a given question q, i.e., for enabling entity recognition and linking (or entity detection) for any CIDOC-CRM KG.

It is important to utilize the extended index to incorporate supplementary triple information. This feature can be tailored to encompass specific attributes such as **rdfs:comment** and **rdfs:label**. This is beneficial in cases where the resource URI is not descriptive (e.g. comprises of an identifier) and reduces the necessity for additional queries to the knowledge graph.

4.1.3 KG storage

Apart from the indexes, the KGs should also be stored in a triplestore, e.g., in GraphDB (`https://www.ontotext.com/products/graphdb/`), for enabling the execution of SPARQL queries (i.e., for creating at real time the context from the subgraphs). We chose Onto-graph GraphDB, because it offers path search algorithms and it easy to create graphs from a resource using a different radius (depth) at a time.

4.2 Step A. Entity Recognition

The objective is to detect the main entity (or entities) of the question q, and to retrieve its URI in the KG. For instance, see Step A in Figure 4.2, where we retrieved the main entity of the question.

Input. A question q in natural language for a CIDOC-CRM KG which has been previously indexed (i.e., see §4.1.2).

Output. The output of this stage is the top-K (K is configurable) entities in a ranked list, described by their URI and a short textual description of the entity extracted by a descriptive property e.g., skos:label or rdfs:label. The value of K depends on the needs of each application, i.e., for questions containing a single entity/event (such as in our evaluation benchmark, which is presented in Chapter 3). For this task it is preferable to select $K = 1$.

4.3 Prerequisites for Step B - Sub Graph Creation

The objective for the recognized entity e (or the top-K entities) is to create one or more subgraphs through path expansion of CIDOC-CRM properties starting from the detected entity, and then to transform each path to text, for being used as a context for the given question q.

4.3.1 Step B1. Creation of subgraph(s)

First, we define a CIDOC-CRM directed path of radius r (or depth) for an entity e, any path of the form: $e \xrightarrow{p_1} u_1 \xrightarrow{p_2} ... \xrightarrow{p_r} u_r$, where e is starting entity (URI), $p_1, ..., p_r$ are CIDOC-CRM forward properties, $u_1, ..., u_r$ are URIs, and r is the radius (path length) between e and and u_r (directly connected through CIDOC-CRM properties).

4.3.1.1 R-Graph

Radius Subgraph (R-Graph) of e given a radius r. We define as $G_r(e)$ the radius subgraph of e, i.e., it includes all the URI sequences starting from e, that contains CIDOC-CRM paths exactly of radius r.

4.3.1.2 U-Graph

Union of Radius-Subgraphs (U-Graph) of e until a radius r. The union of all radius subgraphs of e until r is defined as: $G_{\leq r}(e) = \bigcup_{i=1}^{r} G_i(e)$, i.e., the union of all the (CIDOC-CRM) paths having radius from 1 to r (i.e., the union of all the R-Graphs from 1 to d).

4.3.2 Step B2. From URIs to text

Since we will use the subgraph(s) as a context, we need to transform them to text. In particular, for any CIDOC-CRM path of the constructed subgraph(s), each URI is replaced by its string representation (e.g., through rdfs:label, rdf:value, etc.) , i.e., $label(e) \xrightarrow{label(p_1)} label(u_1) \xrightarrow{label(p_2)} ... \xrightarrow{label(p_r)} label(u_r)$.

Running Example. Figure 4.3 shows all the radius subgraphs for the painting "The Starry Night".In particular, the left part shows the subgraph of each radius, the middle part its textual version, and the right side indicative questions that can be answered (from the subgraph of each radius). Certainly, the U-Graph of $r = 4$ contains all the sentences shown in the middle part, i.e., is the union of the radius subgraphs for each r from 1 to 4. On the contrary, the radius graph of a specific r contains only the texts of that radius. The difference can be also seen in Figure 4.4, i.e., it compares the R-graph and U-Graph of the running example for each radius (from $r = 1$ to $r = 4$).

The process for creating subgraphs from SPARQL queries. For creating either the R-Graph or the U-Graph for a given entity e and radius r, we send a SPARQL query in a GraphDB triplestore, which enables the creation of paths starting from e. The query that we send is shown in Figure 4.5 (Also can be found in `https://github.com/NicolaiGoon/CIDOC-QA-BENCHMARK`). The terms {entity.uri} and {DEPTH} are the variable part of the query. Each time we want to expand the path of a desired entity in a desirable depth, we replace these values with the actual ones. As regards the order of paths that are generated from the query, it depends on the triplestore that one is using for storing and querying the KG. In our case, the SPARQL query, that is sent in GraphDB, first returns the paths (i.e., their textual representation) of the selected radius r (the largest paths), then of radius $r - 1$ and finally of radius 1 (the smallest paths).

4.3.3 Steps B-C. Methods based on Radius Subgraphs for Answer Extraction

The objective is to provide an answer to the question q, by exploiting one or more subgraphs of e and a textual QA model, e.g., a BERT-based model, or any model that can answer a question q given a text t. However, a key problem is which subgraph(s) to create, since a) different questions can require to follow paths of different radius to be answered, and b) large subgraphs can add redundant data that can affect the effectiveness and efficiency (mainly for questions that can be answered by a subgraph of a smaller radius).

Here, we present four alternative methods that can support an R-Graph (i.e., $G_r(e)$) or a U-Graph (i.e., $G_{\leq r}(e)$) given a radius r, that are evaluated in chapter 6 (Evaluation). First, we present a method where we suppose that we know a priori the required radius for answering each question, and then three automatic methods, i.e., the required radius for answering each question is not given.

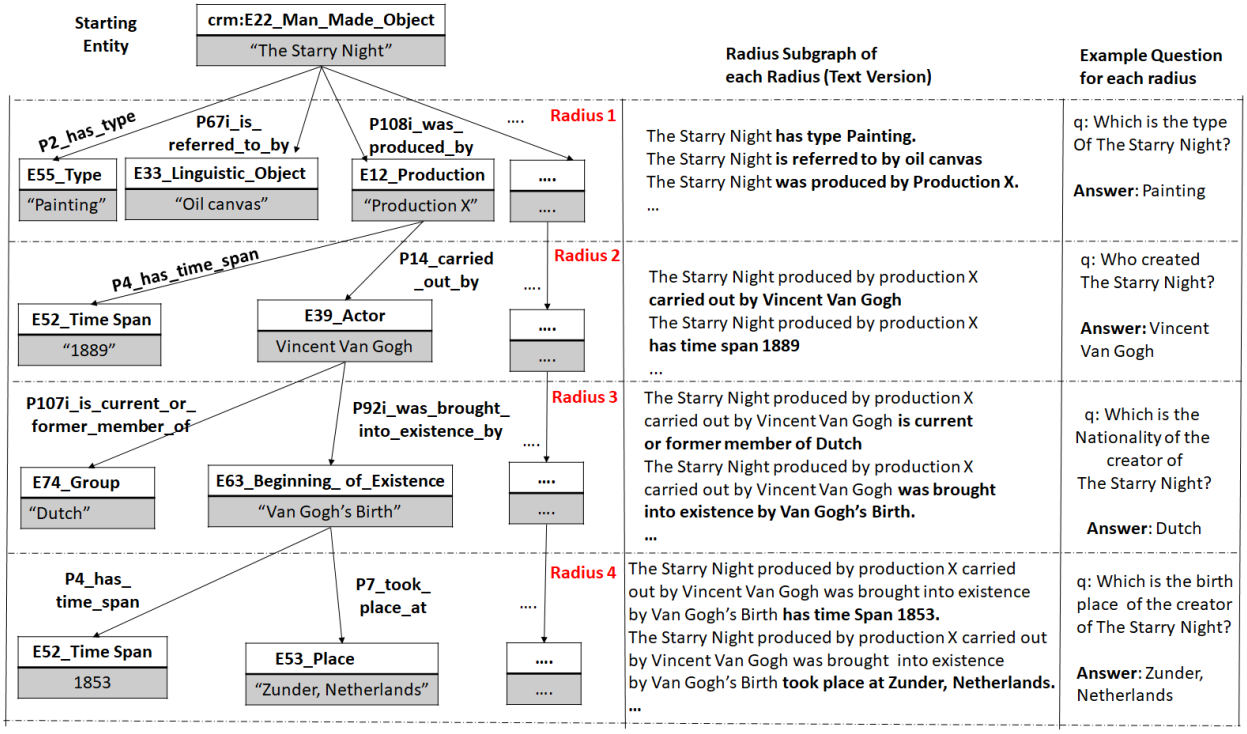

Figure 4.3: The subgraph(s) for the painting The Starry Night of Vincent Van Gogh

4.3.3.1 Known-Radius (KR)

We suppose that we know a priori the required radial r_q of answering a question q. Thereby, we create a single subgraph $G'(e, r_q)$, and the final answer is the following: $KR(G'(e, r_q)) = ans(G'(e, r_q), q)$ having a confidence score of the following range: $0 \leq score(ans(G'(e, r_q), q)) \leq 1$. In the R-Graph case, $G'(e, r_q)$ equals $G_{r_q}(e)$, whereas in the U-Graph case it equals $G_{\leq r_q}(e)$.

- **Advantages and Drawbacks**: The ideal case is to know a priori the radius of each question for avoiding to include noisy information from other radius. However, this is not trivial since it requires to implement mechanisms for answer radius (and type) prediction, which is one of our future directions.

4.3.3.2 Fixed Sub Graph of Radius r (FSR)

The notion is similar to KR method, however, the radius of the question (r_q) is neither given nor predicted. Thereby, we use a fixed radius r for any question q, i.e., it returns $FSR(G'(e, r), q) = ans(G'(e, r), q)$ (r is probably different than r_q)

- **Advantages and Drawbacks**: Concerning the U-Graph, i.e., $G_{\leq r}(e)$, by

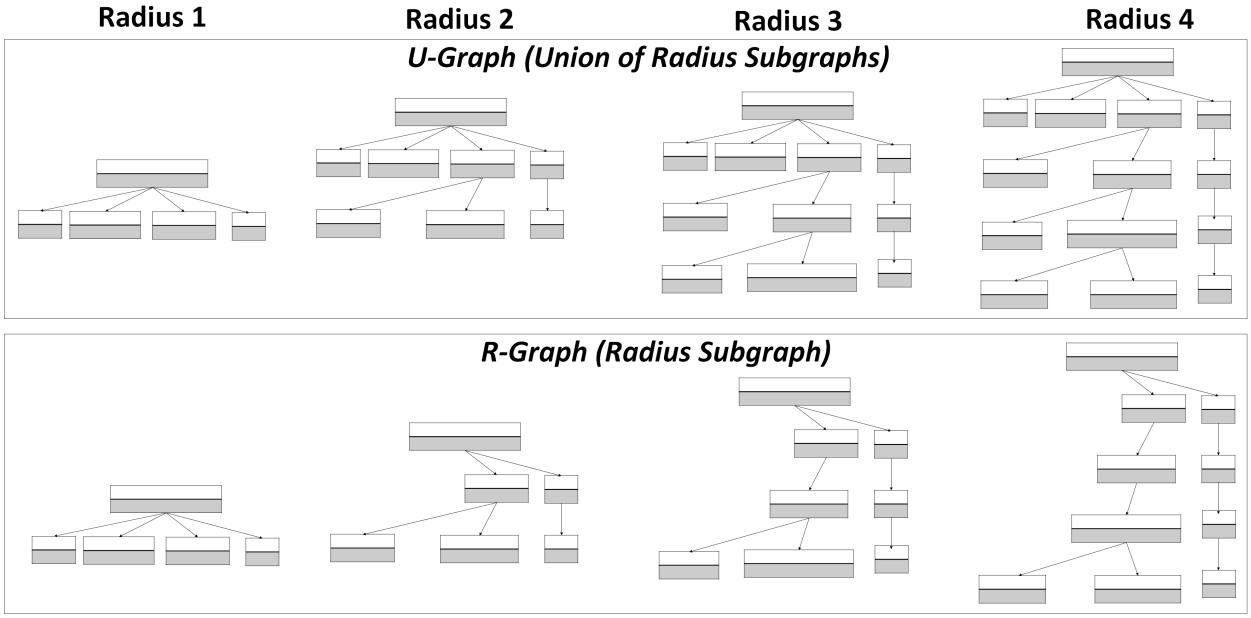

Figure 4.4: U-Graphs vs R-graphs for the running example (radius 1 to 4)

creating the union of radius subgraphs of a fixed radius r, the answer will be included in the context, even for questions requiring a radius $r_q < r$. In Figure 4.3 the question "Which is the type of The Starry Night?", can be answered from the $G_{\leq 4}(e)$, however, a lot of redundant data are included. Concerning the R-Graph, i.e., $G_r(e)$, it can be more effective for questions of radius r, but it would be infeasible in most cases to answer questions of a radius $< r$, e.g., by selecting $r = 2$, we can answer the question "Who created the Starry Night". However, we cannot answer the question about "the type of The Starry Night" (i.e., it is covered only in G_1).

4.3.3.3 Best of Sub Graphs (BoS)

It creates all the subgraphs $G'(e, i)$ for each different radius, i.e., $i \in [1, r]$ (r should be pre-configured). Afterwards, each $G'(e, i)$ is used as context (its text version), and it provides a separate answer for each radius, i.e., r answers are provided (each one having a unique confidence score). Finally, it returns the answer that maximizes the confidence score, i.e., $BoS(e, r, q) = ans(G'(e, i), q)$, s.t., $arg_i max\ score(ans(G'(e, i), q)), i \in [1, r]$. It is applicable for both R-Graph and U-Graph,

• **Advantages and Drawbacks**: Concerning the $BoS_{G_{\leq r}}$ (i.e., U-graph), we expect a positive impact for questions of a small radius, however, again redundant data (from a smaller radius) are included. Regarding BoS_{G_r} (i.e., R-graph), we expect a positive impact for questions of any radius, mainly for questions of a

```
SELECT DISTINCT (?start as ?s) (?property as ?p)
(?end as ?o) (?startLabel as ?sLabel)
(?endLabel as ?oLabel) (?startValue as ?sValue)
(?endValue as ?oValue) (?index as ?depth)
WHERE {
    {
        SELECT ?start ?property ?end ?index
        WHERE {
            VALUES (?src) {
            (<{entity.uri}>)
        }
        SERVICE path:search {
            <urn:path> path:findPath path:allPaths ;
            path:sourceNode ?src ;
            path:destinationNode ?dst ;
            path:minPathLength 1 ;
            path:maxPathLength {DEPTH} ;
            path:startNode ?start;
            path:propertyBinding ?property ;
            path:endNode ?end;
            path:resultBindingIndex ?index ;
            path:pathIndex ?path .
        }
    }
}
OPTIONAL {
    ?start rdfs:label ?startLabel .
    FILTER(lang(?startLabel) = "en" || lang(?startLabel) = "")
}
OPTIONAL {
    ?start rdf:value ?startValue
}
 OPTIONAL {
     ?end rdfs:label ?endLabel .
     FILTER(lang(?endLabel) = "en" || lang(?endLabel) = "")
}
OPTIONAL {
    ?end rdf:value ?endValue
}
FILTER(REGEX(STR(?property), "http://www.cidoc-crm.org/cidoc-crm")) .
} ORDERBY DESC(?index)
```

Figure 4.5: SPARQL query for expanding the path of an entity

large radius. Finally, for both cases (mainly for $BoS_{G_{\leq r}}$) the execution time will be increased (since the answer extraction step is performed r times).

4.3.3.4 Threshold based - Best of Sub Graphs (t-BoS)

For avoiding to perform the answer extraction step r times, we can create the subgraphs incrementally, by using a threshold t. Starting from $r = 1$, we check if $score(ans(G'(e, 1), q)) \geq t$. If it holds, we return the answer, otherwise we continue with the subgraph of the next radius (until finding a $score \geq t$). In case of failing to reach the threshold, i.e., if $\forall i \in [1, r], score(ans(G'(e, i), q)) < t$, we select the answer with the maximum score (i.e., $arg_i max\ score(ans(G'(e, i), q))$. It is applicable for both R-graph and U-graph.

• **Advantages and Drawbacks**: The major advantage is that we can avoid to perform r times the answer extraction phase, however, by selecting a low threshold t, it will possibly not return the answer with the highest score.

4.4 Answer Extraction

In this section we describe the methods we used in order to answer factoid and confirmation questions as they need different approaches.

4.4.1 Answering Factoid Questions

This step receives a list of entities along with their textual descriptions and a question. Using RoBERTa it generates answers for each entity, sorted by their score. We can use any BERT-based model that supports extractive QA [32], e.g., such as those listed in `https://rajpurkar.github.io/SQuAD-explorer/`. In our evaluation, we have selected the RoBERTa [23] model, which was fine-tuned on the SQuAD dataset [31]. We selected this model over BERT due to the increased difficulty of the extractive QA task and the better performance it provides.

4.4.2 Answering Confirmation Questions

To answer confirmation questions at first we tried a fine-tuned version of RoBERTa in the BoolQ [5] dataset which contains Yes/No question answer pairs with a given passage. This method seemed not to be working well as the model would answer all the questions with "No". That was due to the complex text produced by the CIDOC-CRM sub graphs that could not be easily understood by the model, thus it provided wrong answers. Considering the previous statements we decided to use a chat model in order to answer confirmation questions. We decided to use the Llama2 [39] which is a family of pre-tarined and fine-tuned large language models ranging from 7B to 70B parameters from the AI group at Meta, similar to chat-gpt

[29] by OpenAI. We used the 70B parameter model which is the most accurate, through Hugging face API which can be found here: `https://huggingface.co/spaces/ysharma/Explore_llamav2_with_TGI`. We first set up a prompt to help the model undesrtand the concept and how it must respond. The prompt we used is the following: **"Answer a given question based on the context. Keep the answer as short as possible. You must always provide an answer."** At first we tried to feed the model with a single text containing the question and the context like the following string representation: **"Context: ... Question: ..."**. Again, the results were similar with the previous method , but the model was also explaining that the passage did not contain any useful information to answer the question which is a false claim. The method that finally worked required to literally chat with the model. At first we sent only the context **"Context: ..."**, while ignoring the response of the model. Next, in another request we send the question **"Question: ..."** and considering as an answer this response of the model. Following this tactic the model was able to understand better the context and provide more accurate answers than the previous approaches, even if the text was noisy and difficult to understand. Finally in table 4.1 you can see the parameters of the model that we used.

Parameter	**Value**
Temperature	0.9
Max new tokens	128
Top-p (nucleus sampling)	0.6
Repetition penalty	1.2

Table 4.1: Parameters for the Llama2 model

Chapter 5

CIDOC-QA Web

In order to enable to the user the QA over CIDOC-CRM graphs we construct a web app called CIDOC-QA Web. The purpose of the web app is to utilize the QA service and the other resources and provide to the user the ability to access the CIDOC-CRM graph through questions without requiring special knowledge.

5.1 User Interface and Interaction

The interface is composed from the following components:

- Search Bar
- Examples
- Configuration
- Answer

In the **Search Bar** the user can type the desired question and click the search icon or press the key 'enter' in order to send the question to the QA system.

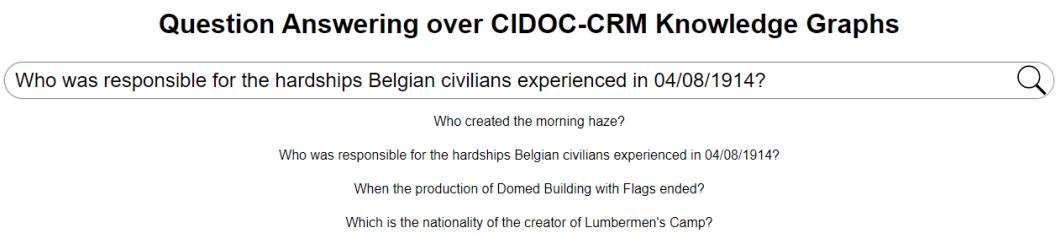

Figure 5.1: CIDOC-QA Web Search bar & Examples

Following there are the **Examples** component. This component has a list of example questions that each one can be clicked resulting in auto-filling the selected question into the search bar.

In figure 5.1 we can see the Search bar when the question "Who was responsible for the hardships Belgian civilians experienced in 04/08/1914?" has been clicked from the Examples component.

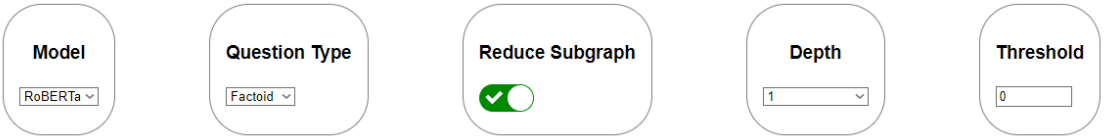

Figure 5.2: CIDOC-QA Web Configuration

Following, the **Configuration** component is responsible for giving the user the ability to choose the model that will be used for answer extraction, the question type and the method that the system will use to create the subgraph text from which the answer will be extracted. The options span between: RoBERTA or Llama2 for the models, Confirmation or Factoid for the question type, choosing between R-Graph or U-Graph (Reduce Subgraph) , selecting depth for the fixed subgraph method or selecting best of subgraphs method and finally using a threshold based approach by setting a threshold. The Configuration component can be seen in figure 5.2.

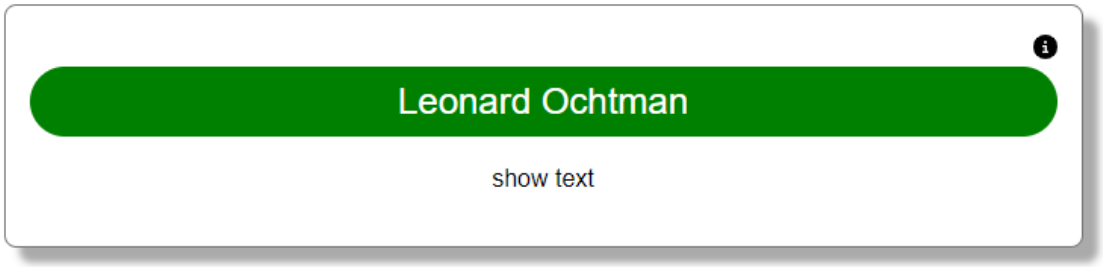

Figure 5.3: CIDOC-QA Web Answer block closed

Finally the **Answer** Component (in figure 5.3) displays the answer along with the passage and the entity that was used in order to extract the answer.

In order to see the text the user must click the "show text" button as you can see in figure 5.4.

Note that the text is the text version of the subgraph that was produced using the configuration the user has selected. The user can also see the score of the question and the time it took for the hole process of the pipeline (see Fig.5.5) while the color of the answer indicates the confidence of the answer according to the score.

Finally in figure 5.6 we can see the whole user interface.

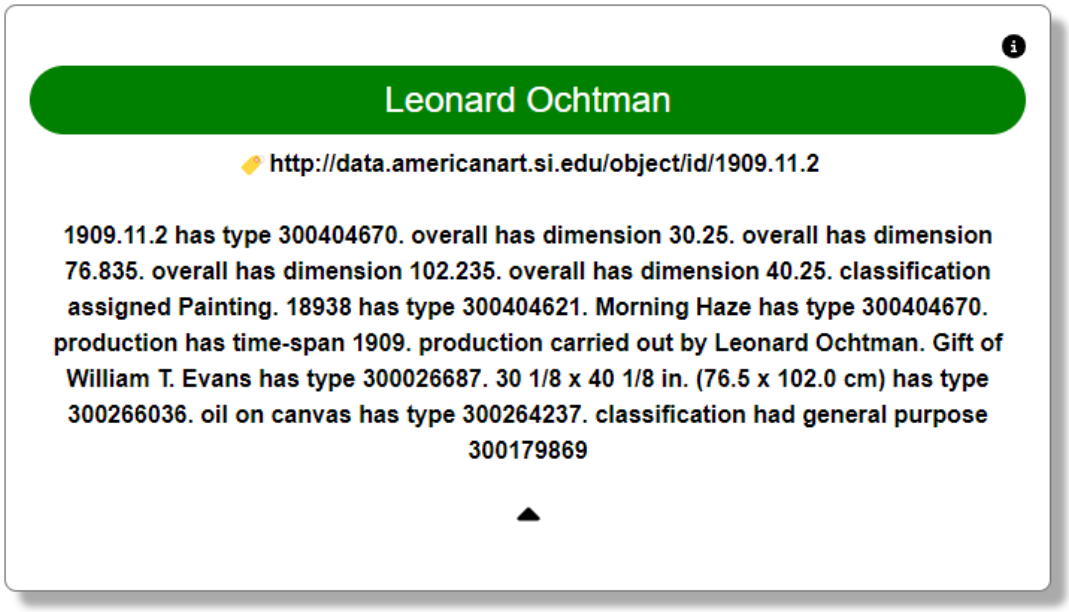

Figure 5.4: CIDOC-QA Web Answer block open

Figure 5.5: CIDOC-QA Web Answer with info

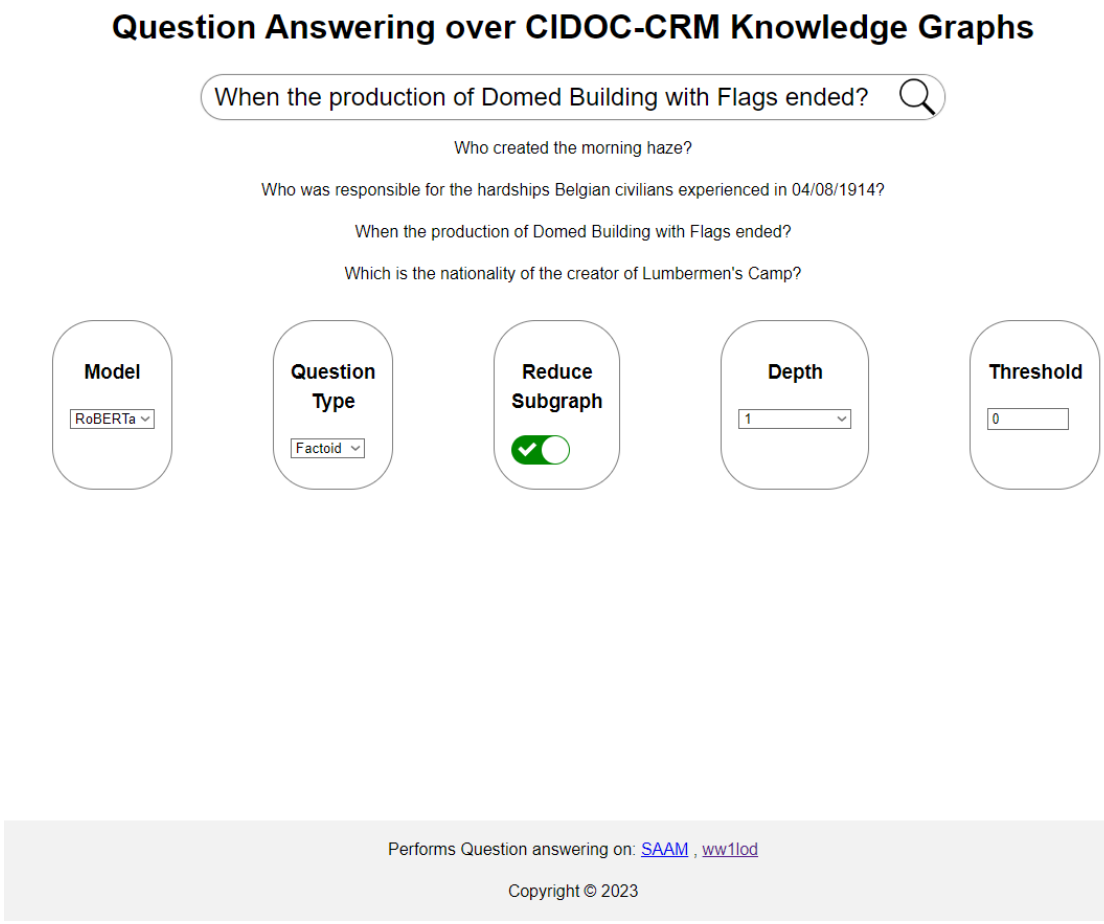

Figure 5.6: CIDOC-QA Web User Interface

Chapter 6

Evaluation

This chapter presents the experimental results for the proposed methods of the QA pipeline by using the 5,000 single- entity factoid questions (question templates Q1-Q10) and the 2,500 Confirmation questions (question tempaltes Q16-Q20) of the evaluation benchmark of Chapter 3. All the experiments have been conducted in a single machine with 16 GB RAM, 8 cores, GTX 1050 Ti GPU and 256 GB disk space.

6.1 Effectiveness

We provide results for the single entity factoid questions and the confirmation questions of the benchmark , while focusing on factoid questions for evaluating *RQ1* and *RQ2*.

6.1.1 Methods and Metrics

We compare the methods of chapter 4 for the Factoid, and only the FSR and KR for the confirmation questions as the chat model does not apply a score to the answers. Since our evaluation benchmark contains questions of radius $r \in [1, 4]$, we use $r = 4$ as the max radius for the best of methods. The baseline method is the one that uses the subgraph of radius=1 (the direct neighbor of each entity). Concerning the metrics, for each question there is a single golden answer. We define for a question q, as $tokens_{gold}(q)$ the set of tokens of the golden answer, and as $tokens_{pred}(q)$ the tokens of the predicted answer. For confirmation questions there is only one token for the correct answer (Yes/No). Afterwards, we compute the metrics below for each question:

- **Precision**: $Prec(q) = \frac{|tokens_{gold}(q) \cap tokens_{pred}(q)|}{|tokens_{pred}(q)|}$, with range [0,1].
- **Recall**: $Recall(q) = \frac{|tokens_{gold}(q) \cap tokens_{pred}(q)|}{|tokens_{gold}(q)|}$, with range [0,1].
- **F1score**: $F1(q) = \frac{|2*Prec(q)*Recall(q)|}{|Prec(q)+Recall(q)|}$, with range [0,1].

- **Accuracy**:

$$Acc(q) = \begin{cases} 1, & \text{if q is correct} \\ 0, & \text{otherwise} \end{cases}$$

The first three metrics are used for the Factoid questions and the Accuracy is only used for the Confirmation ones. Finally, we compute the average percentage (%) of these metrics for all the questions

6.1.2 Effectiveness of Step A. Entity Detection

In this section we evaluated the first stage of the pipeline as shown in figure 4.2. From the 5,000 Factoid questions, we recognized and linked correctly the entity to its URI in 3,920 cases, i.e., 78.4%. Regarding the Confirmation questions we managed to recognise and link 1,362 out of 2,500 entities of the questions or 54.4%. Concerning the most errors, there were ambiguous entities (e.g., paintings having as a title the name of an artist), and entities with popular words that occur in many artworks (e.g., landscape, money).

6.1.3 Effectiveness of Steps B and C. Comparison of methods

The target is to evaluate the effectiveness of the models based on subgraphs. For this reason, we first provide results by ignoring the questions where we did not manage to recognize and link correctly the entity. Afterwards, in Table 6.1 we also provide the results of the whole process.

R-Graph vs U-Graph. Figure 6.1 shows the average size of the words for U-graph (i.e., $G_{\leq r}$) and the R-graph (i.e., G_r) for each different radius (for the entities of the evaluation collection). The size of $G_{\leq r}$ increases exponentially, as the radius grows, whereas the size of the G_r is quite smaller.

Fixed Subgraph Radius (FSR) methods. Fig. 6.2 shows the F1score of the $FSR(G_{\leq r})$, for the questions grouped by their radius r. For each question group we achieved the highest score by using the $(G_{\leq r})$ of the same r. An advantage of $FSR(G_{\leq r})$ is that it can answer questions requiring a smaller r even by using subgraphs of a large r. However, its F1 score is decreased as r increases, whereas even for the questions of the same radius (mainly for large r) it can have a negative impact due to the noisy data of the previous radius. Indeed, Fig. 6.3 shows that the $FSR(G_r)$ (R-graph) is more effective for the questions of the same r. However, it has low scores for questions of different r (and for the overall case). Regarding confirmation questions Fig. 6.4 shows the accuracy for the $FSR(G_{\leq r})$ and Fig. 6.5 shows the accuracy for the $FSR(G_r)$, again for questions grouped by their radius r. For each question group we can see that we achieved similar results, but using $FSR(G_r)$ we see a stability as almost every combination is exceeding the baseline of 50% accuracy. The previous advantage of $FSR(G_{\leq r})$, that can answer questions of smaller radius is not visible in this evaluation due to the nature of the

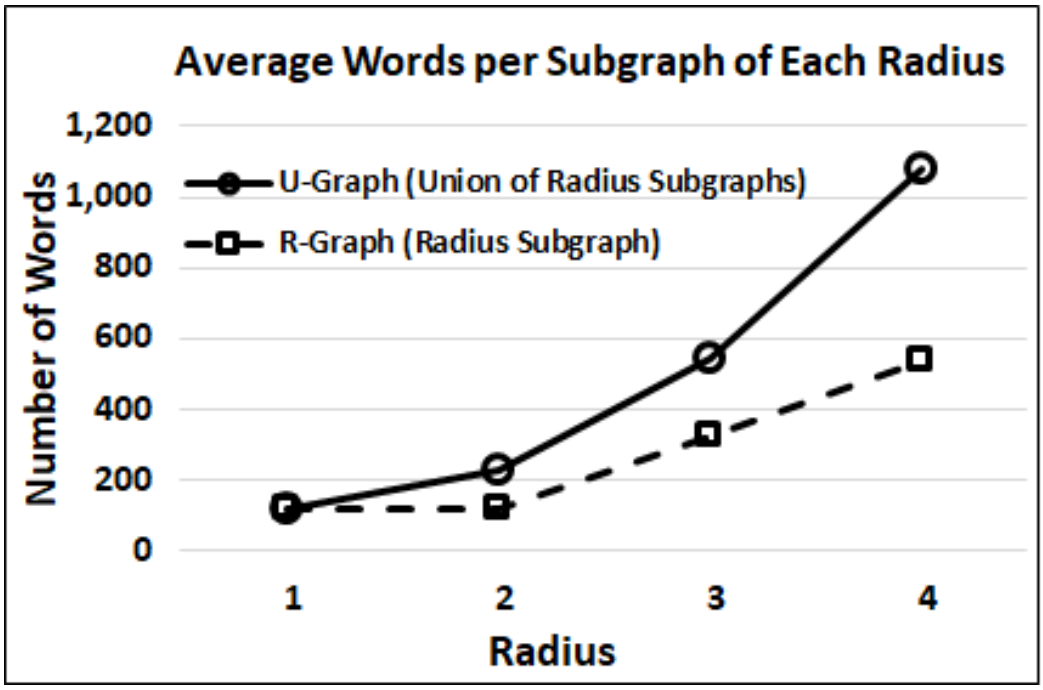

Figure 6.1: Average words per subgraph of each radius

confirmation questions that can be either yes or no. It is easy to answer a question correctly or not while not intend to. The impact of noise in large radius is visible as in Fig. 6.2 $FSR(G \leq 4)$ gets 61.4% accuracy while $FSR(G_4)$ in Fig. 6.5 gets 70.6% accuracy for questions of depth 4.

Known radius (RD) Methods. Figure 6.6 compares the known radius methods for Factoid questions. By knowing the correct radius a priori, the R-graphs are more effective (i.e., they contain less redundant data in the context), especially as r increases, e.g., for the questions of $r = 4$ the $KR(G_4)$ has a difference of +17 compared to the $KR(G_{\leq 4})$. Moreover, concerning the overall case, by using the R-graphs we reached an F1score of 81.9 (i.e., +8.6 compared to the case of using the U-graphs) Regarding Confirmation questions, we achieve similar results as you can see in Fig. 6.7 that shows the accuracy per graph type, per question radius. R-Graphs again, are more effective while containing less redundant information in the text version of the graph. The biggest difference can be seen for depth = 2, R-graphs achieve 74.4% accuracy a +13.2 difference compared to the U-Graph.

The remaining of the evaluation focuses on Factoid questions while mentioning accuracy for the full QA process regarding confirmation questions.

Effectiveness of Best of Methods. Since we do not perform answer radius (and type) prediction, we would like to evaluate the performance of the automatic methods (i.e., the required radius is not given a priori), and to compare their effectiveness with the KR methods. Table 6.1 presents the results of all the methods,

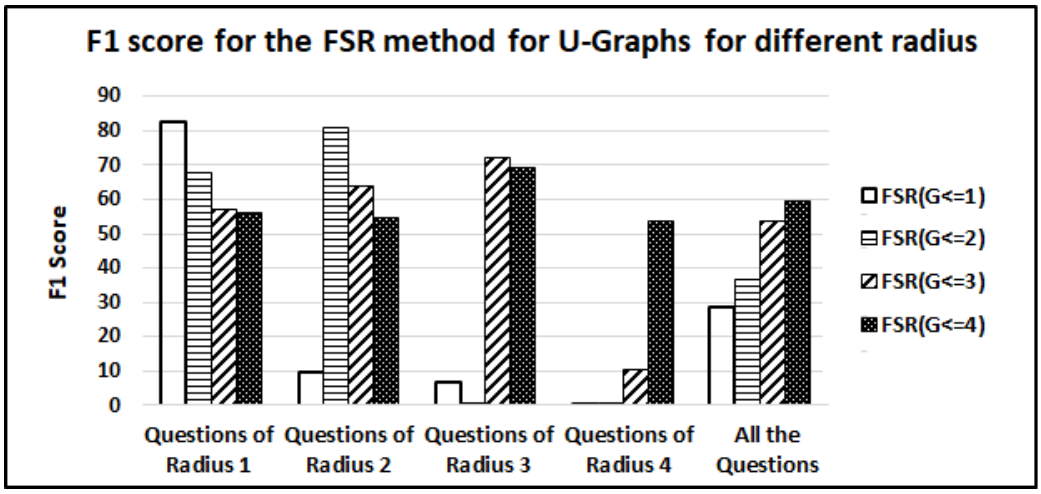

Figure 6.2: F1score for the FSR method for U-Graphs in Factoid Questions (grouped by questions radius)

i.e., on the left side for the questions that we recognized correctly the entity (perfect entity detection) and on the right side for the full QA process. We denote as the baseline method, the one that includes only paths of radius 1, i.e., the direct neighbour of each entity, as in [28]. Since most questions require larger paths (radius) to be answered, it has very low scores. Concerning the best-of methods, they are more effective than the FSR ones, indeed, the BoS_{G_r} achieved the highest F1score, i.e., 64.5. Regarding the full QA process for Confirmation questions, we achieved 37% accuracy for $FSR(G \leq 4)$ method and 30.4% for the $FSR(G)$ which is below the baseline for Confirmation questions (50%).

Threshold-based Methods. By checking several values for the threshold (from 0.1 to 0.9), we decided to use $t = 0.7$. As we can see, they offer similar results to the best-of-methods and they are faster (on average), i.e., see Sect. 6.3.

Best-of vs Known Radius Methods. Although the best-of methods are the most effective automatic methods, they are far from reaching the scores of the KR methods. This means that in many cases, although the correct answer is provided in the r possible answers, it does not have the highest confidence score.

Precision vs Recall. For all the methods of Table 6.1, the precision is higher compared to the recall, which means that the predicted answer contains usually a part of the desired answer (but not the whole one), especially for the questions whose answer has a high average word length (i.e., Q2, Q3, Q5, Q9).

6.2 Discussion & Possible Improvements

Here, we provide conclusions with respect to the research questions. Concerning the *RQ1*, the baseline method is not effective, since it cannot answer effectively Factoid questions of radius $r > 1$ (i.e., its F1score equals 23.0). Regarding the

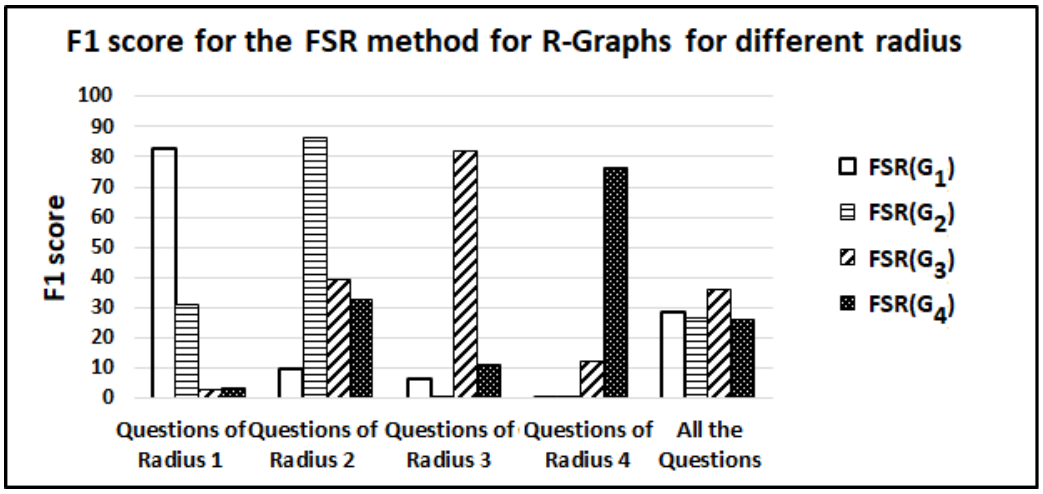

Figure 6.3: F1score for the FSR method for R-Graphs in Factoid Questions (grouped by questions radius)

RQ2, the extended pipeline can be effectively used for QA over CIDOC-CRM KGs (Factoid Questions). Concerning the most effective method, it is the BoS_{G_r} with F1score=51.4 for the full QA process, and with F1score=64.5 for questions with perfect entity detection. However, the difference in the results of the KR methods reveals that there is space for improvements, since in many cases the answer with the highest confidence score is not the correct one.

Regarding Confirmation questions, KR methods show the potential of the pipeline while achieving 76.16% accuracy for all questions using R-Graph. From that we deduct the need for radius prediction as the other methods (BoS, threshold) do not work for Confirmation due to lack of scoring mechanism.

Since this is the first attempt for providing a generic QA pipeline for CIDOC-CRM KGs, there is a plenty of space for improvements, as they follow: i) investigating methods for predicting the required radius, ii) proposing methods for further minimizing the context, by trying to predict the exact path for answering a given question, iii) evaluating the methods by using more BERT models except for RoBERTA, and by adding more KGs to the evaluation benchmark and iv) by adding even more question types and templates (for increasing question diversity), v) investigating Confirmation questions more by seeing the problem as binary classification and computing F1, AUC metrics vi) improving indexing performance as it takes too long even for half a GB.

6.3 Efficiency

First, we needed 9 hours for constructing the index, which is used for the Entity Detection step. However the indexing process needs to be done once for each KG. The KG size is 450 MB and the resulting index is 1.17 GB on disk. Concerning the

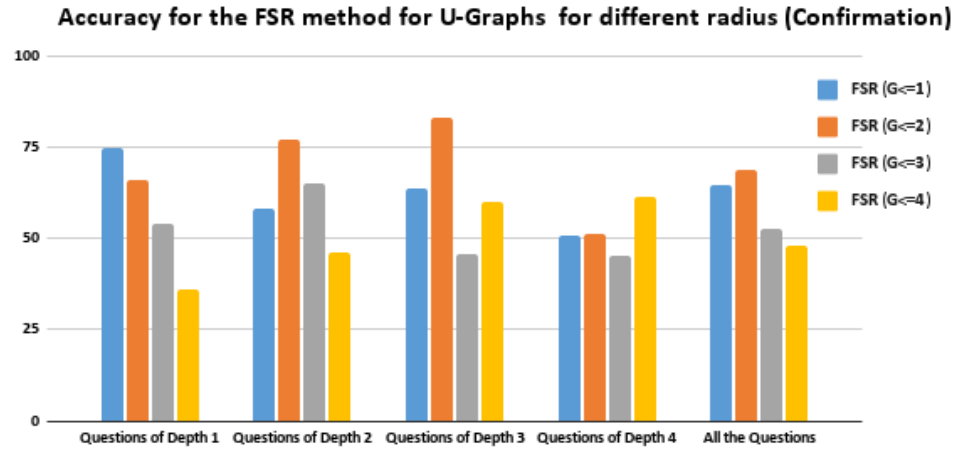

Figure 6.4: Accuracy for the FSR method for U-Graphs in Confirmation Questions (grouped by questions radius)

QA process, Fig. 6.8 shows the average execution time for answering a question for the automated models of Table 6.1.

Execution time of each step. For all the models, the most-time consuming step is the answer extraction, especially for the best of methods. Indicatively, for the BoS_{G_r} case, for the entity detection step we needed the 8.8% of the total time, for the path expansion the 8%, and for the answer extraction the 83.2% of the time. For Confirmation questions the average answer extraction step took 9.61s.

Total Execution Time. The FSR models are quite fast, however, they are less effective compared to best-of methods (see Table 6.1). Concerning the best-of methods, the fastest ones are those using the R-Graph, i.e., for the BoS_{G_r} the average time per question was 1.14 seconds, whereas for the t-BoS_{G_r} (using $t = 0.7$) the average time was 0.96 seconds. In the latter case we achieved a 1.18× speedup, by having similar precision, recall and F1score.

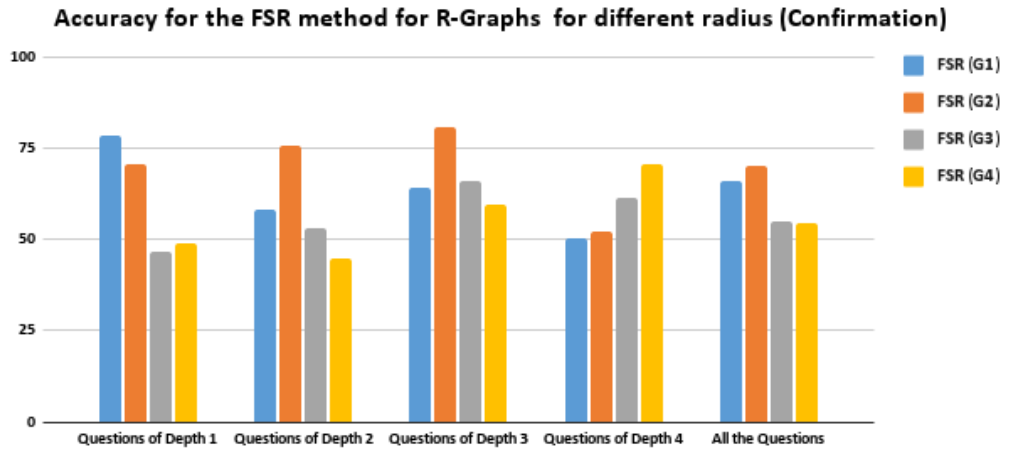

Figure 6.5: Accuracy for the FSR method for R-Graphs in Confirmation Questions (grouped by questions radius)

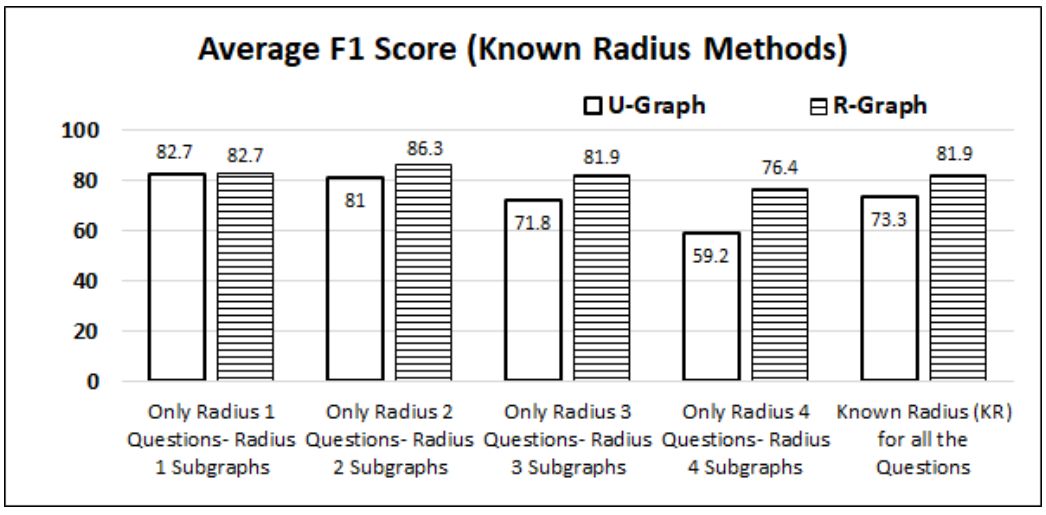

Figure 6.6: Comparison of Known Radius (KR) Methods for U-graphs and R-graphs in Factoid Questions

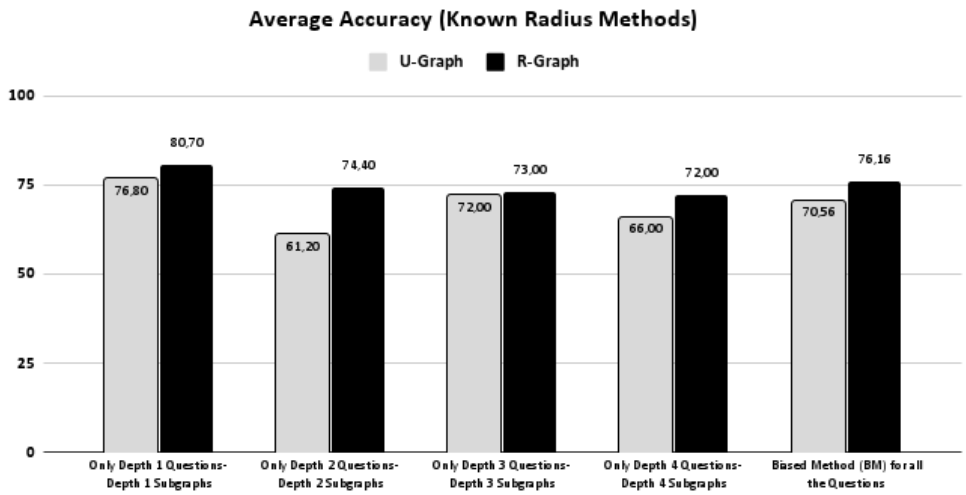

Figure 6.7: Comparison of Known Radius (KR) Methods for U-graphs and R-graphs in Confirmation Questions

	Perfect Entity Detection			**Full QA Process**		
Automatic Methods	Prec. (%)	Recall (%)	F1score (%)	Prec. (%)	Recall (%)	F1score (%)
$FSR(G_{\leq 1})$ (radius 1) **(Baseline)**	31.4	28.0	28.8	25.2	22.4	23.0
$FSR(G_{\leq 4})$ (max radius 4)	63.6	57.7	59.2	52.2	47.7	48.8
$FSR(G_4)$ (only radius 4)	28.8	25.6	26.1	21.3	18.8	19.3
$BoS_{G_{<r}}$ $(r \in [1,4])$	66.4	59.7	61.7	54.2	49.0	50.5
BoS_{G_r} $(r \in [1,4])$	**70.5**	**61.9**	**64.5**	**56.0**	**49.4**	**51.4**
$t\text{-}BoS_{G_{<r}}$ $(r \in [1,4], t = 0.7)$	66.5	59.8	61.8	54.2	49.0	50.5
$t\text{-}BoS_{G_r}$ $(r \in [1,4], t = 0.7)$	70.4	61.8	64.4	55.9	49.3	51.3
Known Radius Methods	Prec. (%)	Recall (%)	F1score (%)	Prec. (%)	Recall (%)	F1score (%)
KR $(G_{\leq r})$ $(r = r_q$ for each question $q)$	79.2	71.1	73.3	64.4	58.0	59.7
KR (G_r),$(r = r_q$ for each question $q)$	**88.9**	**79.1**	**81.9**	**76.7**	**68.6**	**70.9**

Table 6.1: Effectiveness Results for (automatic and known) methods for both i) perfect entity Detection and ii) for the full QA process in Factoid Questions

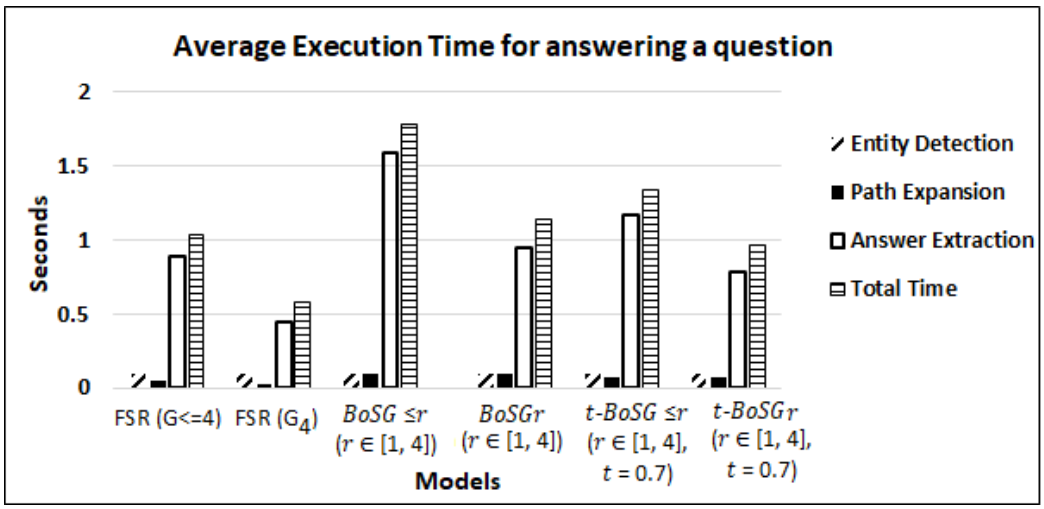

Figure 6.8: Average Execution time (per question) for each model and each step

Chapter 7

Conclusion

In this thesis, we proposed and evaluated a radius-based QA pipeline for answering single-entity factoid questions over CIDOC-CRM (event-based) KGs, since there are not available such QA approaches for the mentioned standard (which is highly used from cultural institutions). We also evaluated our approach in answering Confirmation questions. Since CIDOC-CRM KGs require traversing small or even large subgraphs to answer questions, the pipeline uses methods based on a) elastic search, for recognizing the main entity of the question, b) subgraph creation through path expansion, which transforms subgraphs (even for large radius) of the detected entity to text, for being used as a context, and c) neural network models, for extracting the desired answer from the context. Moreover, we created a benchmark for evaluating the approach having 10,000 questions from the SAAM KG [36], where most of these questions require traversing subgraphs of a large radius for being answered. Regarding the results, we used the 5,000 single-entity factoid questions and 2,500 Confirmation questions of the benchmark, and we achieved 78% accuracy for Factoid and 54% for Confirmation questions for the Entity Recognition step. We also achieved an F1score of 51.4% (on average) for the whole process, which highly outperforms the baseline (F1score for baseline was 23%) for Factoid questions while for Confirmation questions it seems that the KR methods works best with R-Graphs (76.12% accuracy) indicating the need for radius prediction. Finally we managed to answer each question approximately in 1 second (on average) for Factoid questions and ≈ 9 seconds for Confirmation.

Chapter 8

Future Work

As a future work, we plan to a) extend the evaluation benchmark and provide techniques for answering the Comparative questions, b) propose ways for predicting the exact radius of the given question, c) evaluate other models, including large language models like ChatGPT [29, 26] or Llama [39] for Factoid and Comparative questions, d) exploit machine translation techniques, such as those in [25, 30], for enabling multilingual QA. Further more e) Confirmation questions needs a more in depth analysis of the results by approaching the problem as a binary classification one. f) Regarding the indexing mechanism [18] we plan to improve the time required to produce the elastic search indexes for large size KGs. Another solution would be to find a triplestore that enables keyword search in order to avoid the extra indexing and storing the KG in one place.

Bibliography

[1] Vladimir Alexiev et al. Museum linked open data: Ontologies, datasets, projects. *Digital Presentation and Preservation of Cultural and Scientific Heritage*, (VIII):19–50, 2018.

[2] Ram G Athreya, Srividya K Bansal, Axel-Cyrille Ngonga Ngomo, and Ricardo Usbeck. Template-based question answering using recursive neural networks. In *2021 IEEE 15th international conference on semantic computing (ICSC)*, pages 195–198. IEEE, 2021.

[3] Pietro Bongini, Federico Becattini, Andrew D Bagdanov, and Alberto Del Bimbo. Visual question answering for cultural heritage. In *IOP Conference Series: Materials Science and Engineering*, volume 949, page 012074. IOP Publishing, 2020.

[4] Martin Doerr Christianna-Despina Pratikaki. Analysis of scientific questions in archaeology. Technical report, ICS-FORTH, March 2020.

[5] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Bernardo Cuteri, Kristian Reale, and Francesco Ricca. A logic-based question answering system for cultural heritage. In Francesco Calimeri, Nicola Leone, and Marco Manna, editors, *Logics in Artificial Intelligence*, pages 526–541, Cham, 2019. Springer International Publishing.

[7] Aba-Sah Dadzie, Emmanuel Pietriga, Aba-Sah Dadzie, and Emmanuel Pietriga. Visualisation of linked data – reprise. *Semant. Web*, 8(1):1–21, jan 2017.

[8] Dennis Diefenbach, Pedro Henrique Migliatti, Omar Qawasmeh, Vincent Lully, Kamal Singh, and Pierre Maret. Qanswer: A question answering prototype bridging the gap between a considerable part of the lod cloud and end-users. In *The World Wide Web Conference*, pages 3507–3510, 2019.

[9] Eleftherios Dimitrakis, Konstantinos Sgontzos, and Yannis Tzitzikas. A survey on question answering systems over linked data and documents. *Journal of intelligent information systems*, 55:233–259, 2020.

[10] Martin Doerr. The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. *AI magazine*, 24(3):75–75, 2003.

[11] Jinhua Dou, Jingyan Qin, Zanxia Jin, and Zhuang Li. Knowledge graph based on domain ontology and natural language processing technology for chinese intangible cultural heritage. *Journal of Visual Languages & Computing*, 48:19–28, 2018.

[12] Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*, pages 69–78. Springer, 2019.

[13] Achille Felicetti, Daniel Williams, Ilenia Galluccio, Douglas Tudhope, and Franco Niccolucci. Nlp tools for knowledge extraction from italian archaeological free text. In *2018 3rd Digital Heritage International Congress (DigitalHERITAGE) held jointly with 2018 24th International Conference on Virtual Systems & Multimedia (VSMM 2018)*, pages 1–8. IEEE, 2018.

[14] Liu-jie GAO, Wen ZHAO, Jun-fu ZHANG, and Bo JIANG. G2s: Semantic segment based semantic parsing for question answering over knowledge graph. *ACTA ELECTONICA SINICA*, 49(6):1132, 2021.

[15] Simon Gottschalk and Elena Demidova. Eventkg–the hub of event knowledge on the web–and biographical timeline generation. *Semantic Web*, 10(6):1039–1070, 2019.

[16] Nikos Gounakis, Michalis Mountantonakis, and Yannis Tzitzikas. Evaluating a radius-based pipeline for question answering over cultural (cidoc-crm based) knowledge graphs. In *Proceedings of the 34th ACM Conference on Hypertext and Social Media*, HT '23, New York, NY, USA, 2023. Association for Computing Machinery.

[17] Yumeng Hou, Sarah Kenderdine, Davide Picca, Mattia Egloff, and Alessandro Adamou. Digitizing intangible cultural heritage embodied: State of the art. *Journal on Computing and Cultural Heritage (JOCCH)*, 15(3):1–20, 2022.

[18] Giorgos Kadilierakis, Pavlos Fafalios, Panagiotis Papadakos, and Yannis Tzitzikas. Keyword search over RDF using document-centric information retrieval systems. In Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez, editors, *The Semantic Web*, pages 121–137, Cham, 2020. Springer International Publishing.

[19] Giorgos Kadilierakis, Christos Nikas, Pavlos Fafalios, Panagiotis Papadakos, and Yannis Tzitzikas. Elas4rdf: Multi-perspective triple-centered keyword search over rdf using elasticsearch. In Andreas Harth, Valentina Presutti, Raphaël Troncy, Maribel Acosta, Axel Polleres, Javier D. Fernández, Josiane Xavier Parreira, Olaf Hartig, Katja Hose, and Michael Cochez, editors, *The Semantic Web: ESWC 2020 Satellite Events*, pages 122–128, Cham, 2020. Springer International Publishing.

[20] Masayu Leylia Khodra, Ary Setijadi Prihatmanto, Carmadi Machbub, et al. A question answering system using graph-pattern association rules (qagpar) on yago knowledge base. In *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 536–541. IEEE, 2018.

[21] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6, 01 2014.

[22] Zhicheng Liang, Zixuan Peng, Xuefeng Yang, Fubang Zhao, Yunfeng Liu, and Deborah L McGuinness. Bert-based semantic query graph extraction for knowledge graph question answering. In *ISWC (Posters/Demos/Industry)*, 2021.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2019.

[24] Dora Melo, Irene Pimenta Rodrigues, and Davide Varagnolo. A strategy for archives metadata representation on cidoc-crm and knowledge discovery. *Semantic Web*, (Preprint):1–32, 2021.

[25] Michalis Mountantonakis, Michalis Bastakis, Loukas Mertzanis, and Yannis Tzitzikas. Tiresias: Bilingual question answering over dbpedia. 2022.

[26] Michalis Mountantonakis and Yannis Tzitzikas. Using multiple rdf knowledge graphs for enriching chatgpt responses. *arXiv preprint arXiv:2304.05774*, 2023.

[27] Christos Nikas, Pavlos Fafalios, and Yannis Tzitzikas. Two-stage semantic answer type prediction for QA using BERT and class-specificity rewarding. 2020.

[28] Christos Nikas, Pavlos Fafalios, and Yannis Tzitzikas. Open domain question answering over knowledge graphs using keyword search, answer type prediction, SPARQL and pre-trained neural models. In *The Semantic Web – ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings*, page 235–251, Berlin, Heidelberg, 2021. Springer-Verlag.

[29] Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *arXiv preprint arXiv:2302.06466*, 2023.

[30] Aleksandr Perevalov, Andreas Both, Dennis Diefenbach, and Axel-Cyrille Ngonga Ngomo. Can machine translation be a reasonable alternative for multilingual question answering systems over knowledge graphs? In *Proceedings of the ACM Web Conference 2022*, pages 977–986, 2022.

[31] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[32] Anna Rogers, Matt Gardner, and Isabelle Augenstein. QA dataset explosion: A taxonomy of NLP resources for question answering and reading comprehension. *ACM Computing Surveys (CSUR)*, 2022.

[33] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4498–4507, 2020.

[34] Tarcísio Souza Costa, Simon Gottschalk, and Elena Demidova. Event-qa: A dataset for event-centric question answering over knowledge graphs. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 3157–3164, 2020.

[35] Omri Suissa, Maayan Zhitomirsky-Geffet, and Avshalom Elmalech. Question answering with deep neural networks for semi-structured heterogeneous genealogical knowledge graphs. *Semantic Web*, 14(2):209–237, 2023.

[36] Pedro Szekely, Craig A Knoblock, Fengyu Yang, Xuming Zhu, Eleanor E Fink, Rachel Allen, and Georgina Goodlander. Connecting the smithsonian american art museum to the linked data cloud. In *The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings 10*, pages 593–607. Springer, 2013.

[37] Wei Tang, Qingchao Kong, Wenji Mao, and Xiaofei Wu. Contrastive semantic similarity learning for multi-hop question answering over event-centric knowledge graphs. In *2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, pages 360–364. IEEE, 2022.

[38] Thomas Pellissier Tanon, Marcos Dias de Assunçao, Eddy Caron, and Fabian Suchanek. *Platypus–A Multilingual Question Answering Platform for Wikidata*. PhD thesis, LIP-ENS Lyon, 2018.

[39] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[40] Yannis Tzitzikas, Michalis Mountantonakis, Pavlos Fafalios, and Yannis Marketakis. CIDOC-CRM and machine learning: A survey and future research. *Heritage*, 5(3):1612–1636, 2022.

[41] Savvas Varitimiadis, Konstantinos Kotis, Dimitris Spiliotopoulos, Costas Vassilakis, and Dionisis Margaris. "talking" triples to museum chatbots. In *Culture and Computing: 8th International Conference, C&C 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings 22*, pages 281–299. Springer, 2020.

[42] Geert Verhoeven and Christopher Sevara. Trying to break new ground in aerial archaeology, 11 2016.

[43] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[44] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.