This is a preprint of the paper: Nikos Gounakis, Michalis Mountantonakis and Yannis Tzitzikas, "Evaluating a Radius-based Pipeline for Question Answering over Cultural (CIDOC-CRM based) Knowledge Graphs". Accepted for publication in ACM Hypertext 2023, Research Track.

Evaluating a Radius-based Pipeline for Question Answering over Cultural (CIDOC-CRM based) Knowledge Graphs

NIKOS GOUNAKIS, MICHALIS MOUNTANTONAKIS, and YANNIS TZITZIKAS, Institute of Computer Science - FORTH-ICS, and Computer Science Department - University of Crete, , Greece

CIDOC-CRM is an event-based international standard for cultural documentation that has been widely used for offering semantic interoperability in the Cultural Heritage (CH) domain. Although there are several Knowledge Graphs (KGs) expressed by using CIDOC-CRM, the task of Question Answering (QA) has not been studied over such graphs. For this reason, in this paper we propose and evaluate a Radius-based QA pipeline over CIDOC-CRM KGs for single-entity factoid questions. In particular, we propose a generic QA pipeline that comprises several models and methods, including a keyword search model for recognizing the entity of the question (and linking it to the KG), methods that are based on path expansion for constructing subgraphs of different radius (or depths) starting from the recognized entity, i.e., for being used as a context, and pre-trained neural models (based on BERT) for answering the question using the mentioned context. Moreover, since there are no available benchmarks over CIDOC-CRM KGs, we construct (by using a real KG) an evaluation benchmark having 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. Concerning the results, the QA pipeline achieves satisfactory results both in the entity recognition step (78% accuracy) and in the QA process (51% F1 score).

Additional Key Words and Phrases: Knowledge Graph, Natural Language Processing, Resource Description Framework, Cultural Heritage, Answer Extraction, Event-Based Ontology, Path Expansion, Entity Recognition, Linked Data

ACM Reference Format:
Nikos Gounakis, Michalis Mountantonakis, and Yannis Tzitzikas. 2018. Evaluating a Radius-based Pipeline for Question Answering over Cultural (CIDOC-CRM based) Knowledge Graphs. In *ACM Hypertext 2023, September 2023.* ACM, New York, NY, USA, 14 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The digitization and scientific documentation of cultural heritage objects is a research field that has grown significantly in the last two decades, since it is of primary importance to curate, restore and preserve cultural artefacts [14]. For this reason, formal models have been created for modelling cultural objects, like the CIDOC Conceptual Reference Model (CIDOC-CRM); an ISO 21127 standard event-based ontology for the cultural domain [8] that has been widely used [1, 33] for offering interoperability between the Cultural Heritage (CH) domain metadata standards and ontologies. However, due to its complex (event-based) nature, it is difficult for non-experts to exploit the data expressed through the CIDOC-CRM model. A user-friendly interface to such Knowledge Graphs is to provide a Question Answering (QA)

Manuscript submitted to ACM

service, where any user can express a natural question (e.g., such as those in [4]). Indicatively, such QA pipelines can be used for enabling users to ask questions through text or voice (e.g., chatbots) [34] and to retrieve answers from a Knowledge Graph. For instance, 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 of the painting, etc.

However, there are no evaluated pipelines for QA over CIDOC-CRM [33], especially, due to the following difficulties: a) CIDOC-CRM model complexity, 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, i.e., 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 (e.g., [22]). Therefore, one has to exploit various deductions from the KG. Regarding c), there are no available benchmarks for evaluating such QA tasks that support CIDOC-CRM KGs [33]. For tackling these limitations, in this paper we focus on answering the following research questions:

- RQ1: How effective is an existing generic QA pipeline over non-event based models, such as for 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. In particular, i) we explain why existing generic QA pipelines, such as Elas4RDF-QA [22], 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 [30]. 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 questions for evaluating the effectiveness and efficiency of the proposed pipeline.

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 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, and 51% F1 score for the QA process (+28.4% comparing to the original Elas4RDF-QA pipeline). Finally, the average query time is approximately 1 second.

The rest of the paper is described as follows: Section 2 discusses the related work, and Sect. 3 presents the evaluation benchmark and the requirements. Sect. 4 introduces the proposed QA pipeline. Sect. 5 presents comparative results for the proposed methods. Finally, Sect. 6 concludes the paper and identifies directions for future research.

2 RELATED WORK

Here, we 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.

QA over RDF KGs. There is an increasing trend for QA approaches over KGs [7], which can be divided in 3 categories [31]: i) template based approaches [2, 16], i.e., matching questions to SPARQL templates, ii) semantic parsing methods [12, 18], i.e., translating questions into logic query forms, and iii) information retrieval-based methods [31], i.e., they

extract the entity and words of each question, and tries 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 [17] and Wikidata [35], e.g., see QAnswer [6], Platypus [32] and Elas4RDF-QA [22].

QA and Collections for Event-based KGs. Concerning event-based QA, [31] combines information retrieval methods and similarity functions for detecting the best path for answering a question. Also, [27] exploits KG embeddings for finding the best answer for multi-hop QA. Regarding event QA collections, there exists the EVENT-QA [28] and LC-QUAD [10] benchmarks, that use the EventKG [13], DBpedia [17] and Wikidata [35] 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 [36] is a large scale multi-hop collection (from one to three hops) with more than 400k questions in the movie domain.

Concerning CIDOC-CRM, [29] performs QA over genealogical graphs expressed in GEDCOM format. These graphs are converted into subgraphs represented using CIDOC-CRM, and then text passages and question-answer pairs are generated from the obtained subgraphs using a combination of Deep Neural Network models. In [5] the authors proposed a logic-based QA system over CIDOC-CRM, that transforms a question to a SPARQL query and returns the result in natural text. However, it does not provide an evaluation of the approach.

NLP tasks for CIDOC-CRM KGs. Apart from QA, there are few approaches that use NLP techniques over CIDOC-CRM [33]. Firstly, TEXTCROWD [11] offers part-of-speech tagging and Named Entity Recognition for Italian archaeological reports and produces the output using CIDOC-CRM, whereas [9] extracts entities and relations from Chinese cultural texts and uses CIDOC-CRM classes for classifying the extracted entities. Furthermore, in [20] text classification and extraction is performed over Portuguese National Archives records, for modelling the extracted information data by using CIDOC-CRM.

Comparison and 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 [29], whereas we create and convert subgraphs to texts instead of transforming the question into a SPARQL query [5]. 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 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.

3 EVALUATION BENCHMARK AND REQUIREMENTS

This section presents the evaluation benchmark, the context and the requirements.

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

Since there are no available benchmarks for QA over such KGs [33], we create a benchmark for evaluating CIDOC-CRM QA approaches. Specifically, we use the Smithsonian American Art Museum [30] (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

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

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 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. Below, we provide a small description for each question type.

A. Single Entity Factoid Questions (Q1-Q10): There are 5,000 questions from 10 templates (from radius 1 to 4), and they contain questions about a single artwork or artist.

B. Comparative Questions (Q11-Q15): There are 2,500 questions from 5 templates (from radius 1 to 4), and they contain comparative questions about either pairs of art works or pairs of artists.

C. Confirmation Questions (Q16-Q20): 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".

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. 1 shows the SPARQL query of Q9. Regarding the output of this process, an indicative benchmark entry of Q9 is shown in Fig. 2.

How will we use this benchmark in this paper: We decided to first investigate techniques for factoid single-entity questions since this is a fundamental step for more complex questions, thereby the evaluation is conducted on the question templates Q1-Q10 of Table 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 .
}
```

Fig. 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"
```

Fig. 2. An indicative JSON entry for the template Q9

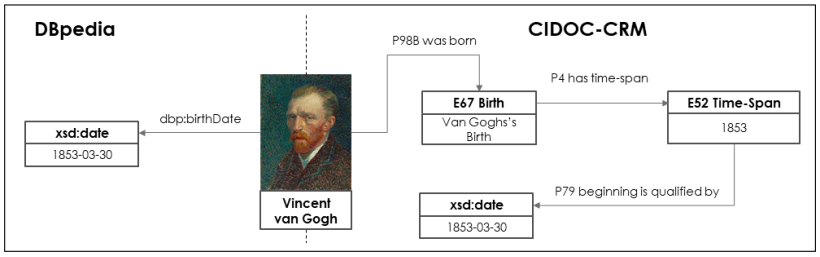

Fig. 3. Van Gogh birth date representation in DBpedia vs CIDOC-CRM

3.2 Context and Requirements

In this paper, we extend the generic Elas4RDF-QA pipeline [22], for being able to answer single-entity factoid questions over any CIDOC-CRM KG. Elas4RDF-QA uses a Keyword Search over RDF [15], SPARQL queries for text generation (to be used as a context), BERT for Answer Extraction, and Answer Type Prediction. Although Elas4RDF has been successfully used for DBpedia, it is not sufficient in its current form for CIDOC-CRM KGs for the following reasons:

• *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 3). 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 3).

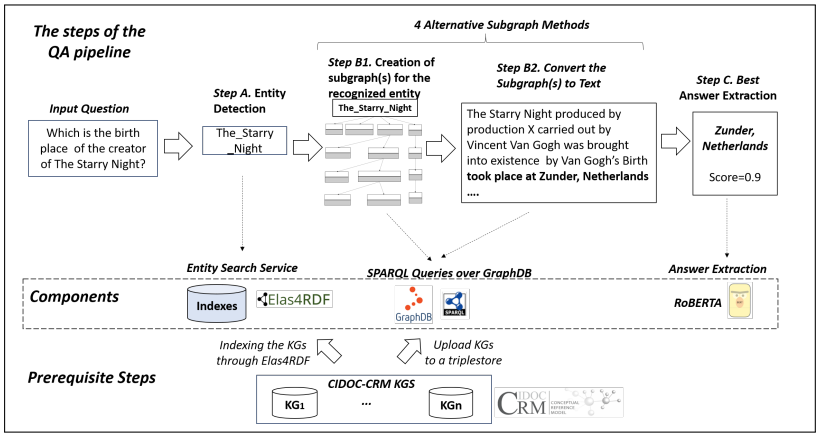

Fig. 4. The proposed QA pipeline over any CIDOC-CRM KG for single entity factoid questions and a running example

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 1.

4 THE PROPOSED QA 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 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 CIDOC-CRM KG

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).

Indexes for Enabling Entity Detection. The first step is to create an index from the desired KG(s) using the Elas4RDF index service [15]. 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.

Using a triplestore for storing and querying the KGs. 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).

4.2 Step A. Entity Detection

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, 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).

Output. The output of this stage is the top-K (K is configurable) entities in a ranked list, described by their URI. 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 Sect. 3.1), it is preferable to select $K = 1$.

The Process. Since there are no Named Entity Recognition and Linking tools for CIDOC-CRM based KGs [33] (e.g., in comparison with other KGs such as DBpedia, Wikidata, etc.), we use the Keyword Search System of [15], which returns the top-K entities (and their URI) according to the question q . The URI will be used as the starting point for creating one or more subgraphs that will be used as the context for a pretrained model.

4.3 Prerequisites for Step B - Creation of Radius Subgraphs

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.

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).

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.

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.

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 5 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 6, 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 can be found in https://github.com/NicolaiGoon/CIDOC-QA-BENCHMARK. 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).

4.4 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. However, a key problem is which subgraph(s) to create, since a) different questions

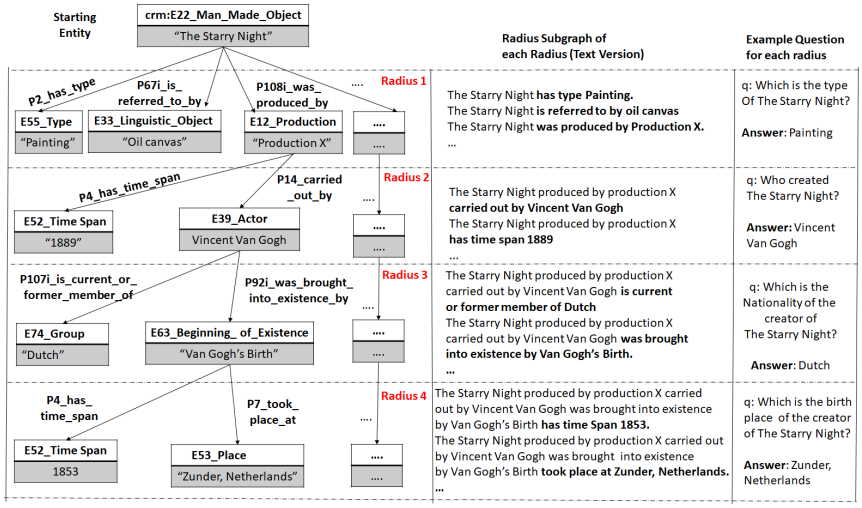

Fig. 5. The subgraph(s) for the painting The Starry Night of Vincent Van Gogh

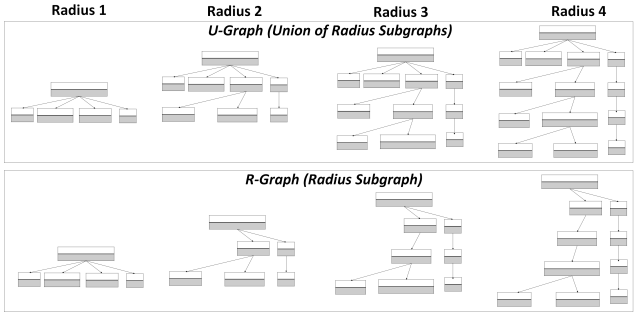

Fig. 6. U-Graphs vs R-graphs for the running example (radius 1 to 4)

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. 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.

Method 1. Known-Radius (KR) - Knowing the Radius a priori: 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.

Method 2. Fixed Subgraph 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 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 5 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).

Method 3. Best of subgraphs (*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 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).

Method 4. Threshold based - Best of subgraphs (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.

Models for Answer Extraction. We can use any BERT-based model that supports extractive QA [26], e.g., such as those listed in https://rajpurkar.github.io/SQuAD-explorer/. In our evaluation, we have selected the RoBERTa [19] model, which was fine-tuned on the SQuAD dataset [25]. We selected this model over BERT due to the increased difficulty of the extractive QA task.

5 EXPERIMENTAL EVALUATION

This section 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) of the evaluation benchmark of Section 3.1. 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.

5.1 Effectiveness Results - Single-Entity Factoid Questions

We provide results for the single entity factoid questions of the benchmark for evaluating *RQ1* and *RQ2*.

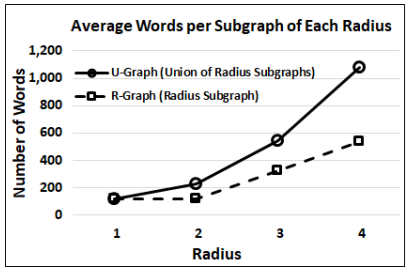

Fig. 7. Average words per subgraph of each radius

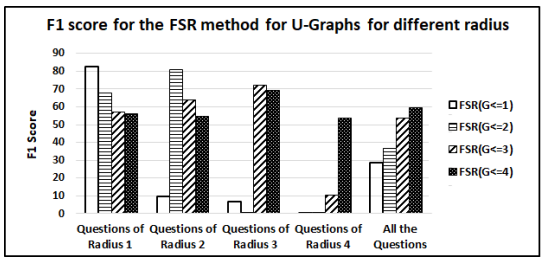

Fig. 8. F1score for the *FSR* method for U-Graphs (grouped by questions radius)

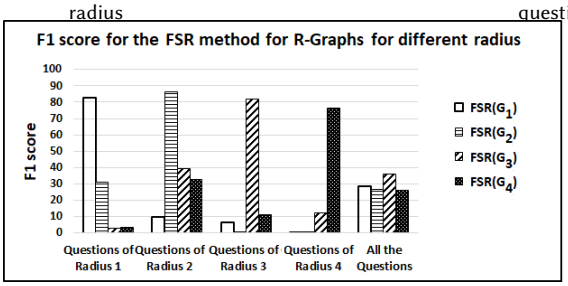

Fig. 9. F1score for the FSR method for R-Graphs (grouped by questions radius)

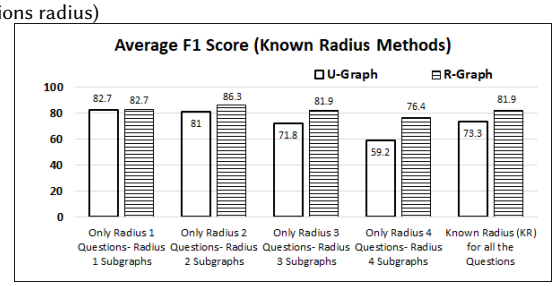

Fig. 10. Comparison of Known Radius (KR) Methods for U-graphs and R-graphs

Methods and Metrics used. We compare the methods of §4. 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. 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].

Finally, we compute the average percentage (%) of these metrics for all the questions.

Effectiveness of Step A. Entity Detection. From the 5,000 questions, we recognized and linked correctly the entity to its URI in 3,920 cases, i.e., 78.4%. 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).

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 2 we also provide the results of the whole process.

R-Graph vs U-Graph. Figure 7 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. 8 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

	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
$BoS_{G_{\leq r}}$ ($r \in [1,4]$)	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**
$t\text{-}BoS_{G_{\leq r}}$ ($r \in [1,4]$, $t = 0.7$)	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 2. Effectiveness Results for (automatic and known) methods for both i) perfect entity Detection and ii) for the full QA process

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. 9 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).

Known radius (RD) Methods. Figure 10 compares the known radius methods. 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).

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 2 presents the results of all the methods, 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 [22]. 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.

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. 5.3.

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 2, 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).

5.2 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 questions of radius $r > 1$ (i.e., its F1score equals 23.0). Regarding the *RQ2*, the extended pipeline can be effectively used for QA over CIDOC-CRM KGs. Concerning the most effective method, it is the BoS_{G_r} method with F1score=51.4 for the full QA process, and with F1score=64.5 for questions with perfect entity

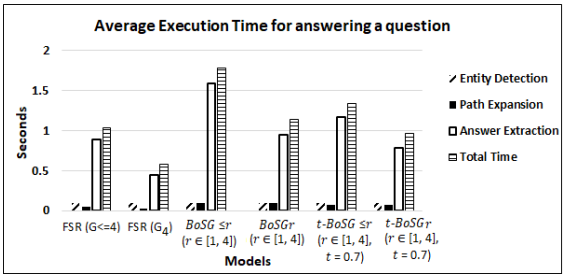

Fig. 11. Average Execution time (per question) for each model and each step

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.

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).

5.3 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 QA process, Fig. 11 shows the average execution time for answering a question for the automated models of Table 2.

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%.

Total Execution Time. The FSR models are quite fast, however, they are less effective compared to best-of methods (see Table 2). 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 $t\text{-}BoS_{G_r}$ (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.

6 CONCLUDING REMARKS

In this paper, 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). 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 [30], where most of these questions require to traverse subgraphs of a large radius for being answered. Regarding the results, we used the 5,000 single-entity factoid questions of the benchmark, and we achieved 78% accuracy for the Entity Recognition step and an F1score of 51.4% (on average) for the whole process, which highly outperforms the baseline (F1score for baseline was 23%). Finally we managed to answer each question approximately in 1 second (on average).

As a future work, we plan to a) extend the evaluation benchmark and provide techniques for answering more question types, b) propose ways for predicting the exact radius of the given question, c) create a web application for enabling the QA over any CIDOC-CRM KG as a service, d) evaluate other neural network models, including ChatGPT [23], and e) exploit machine translation techniques, such as those in [21, 24], for enabling multilingual QA.

Acknowledgments. This work has received funding from the European Union's Horizon 2020 coordination and support action 4CH (Grant agreement No 101004468).

REFERENCES

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

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

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

[4] Martin Doerr Christianna-Despina Pratikaki. 2020. *Analysis of Scientific Questions in Archaeology.* Technical Report. ICS-FORTH.

[5] Bernardo Cuteri, Kristian Reale, and Francesco Ricca. 2019. A Logic-Based Question Answering System for Cultural Heritage. In *Logics in Artificial Intelligence*, Francesco Calimeri, Nicola Leone, and Marco Manna (Eds.). Springer International Publishing, Cham, 526–541.

[6] Dennis Diefenbach, Pedro Henrique Migliatti, Omar Qawasmeh, Vincent Lully, Kamal Singh, and Pierre Maret. 2019. 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*. 3507–3510.

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

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

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

[10] Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. 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*. Springer, 69–78.

[11] Achille Felicetti, Daniel Williams, Ilenia Galluccio, Douglas Tudhope, and Franco Niccolucci. 2018. 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)*. IEEE, 1–8.

[12] Liu-jie GAO, Wen ZHAO, Jun-fu ZHANG, and Bo JIANG. 2021. G2S: Semantic segment based semantic parsing for question answering over knowledge graph. *ACTA ELECTONICA SINICA* 49, 6 (2021), 1132.

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

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

[15] Giorgos Kadilierakis, Pavlos Fafalios, Panagiotis Papadakos, and Yannis Tzitzikas. 2020. Keyword Search over RDF Using Document-Centric Information Retrieval Systems. In *The Semantic Web*, Andreas Harth, Sabrina Kirrane, Axel-Cyrille Ngonga Ngomo, Heiko Paulheim, Anisa Rula, Anna Lisa Gentile, Peter Haase, and Michael Cochez (Eds.). Springer International Publishing, Cham, 121–137.

[16] Masayu Leylia Khodra, Ary Setijadi Prihatmanto, Carmadi Machbub, et al. 2018. 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)*. IEEE, 536–541.

[17] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2014. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal* 6 (01 2014). https://doi.org/10.3233/SW-140134

[18] Zhicheng Liang, Zixuan Peng, Xuefeng Yang, Fubang Zhao, Yunfeng Liu, and Deborah L McGuinness. 2021. BERT-based Semantic Query Graph Extraction for Knowledge Graph Question Answering.. In *ISWC (Posters/Demos/Industry)*.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://doi.org/10.48550/ARXIV.1907.11692

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

[21] Michalis Mountantonakis, Michalis Bastakis, Loukas Mertzanis, and Yannis Tzitzikas. 2022. Tiresias: Bilingual Question Answering over DBpedia. (2022).

[22] Christos Nikas, Pavlos Fafalios, and Yannis Tzitzikas. 2021. 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*. Springer-Verlag, Berlin, Heidelberg, 235–251. https://doi.org/10.1007/978-3-030-88361-4_14

[23] Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. 2023. ChatGPT versus Traditional Question Answering for Knowledge Graphs: Current Status and Future Directions Towards Knowledge Graph Chatbots. *arXiv preprint arXiv:2302.06466* (2023).

[24] Aleksandr Perevalov, Andreas Both, Dennis Diefenbach, and Axel-Cyrille Ngonga Ngomo. 2022. Can Machine Translation be a Reasonable Alternative for Multilingual Question Answering Systems over Knowledge Graphs?. In *Proceedings of the ACM Web Conference 2022*. 977–986.

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

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

[27] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. 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*. 4498–4507.

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

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

[30] Pedro Szekely, Craig A Knoblock, Fengyu Yang, Xuming Zhu, Eleanor E Fink, Rachel Allen, and Georgina Goodlander. 2013. 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*. Springer, 593–607.

[31] Wei Tang, Qingchao Kong, Wenji Mao, and Xiaofei Wu. 2022. 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)*. IEEE, 360–364.

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

[33] Yannis Tzitzikas, Michalis Mountantonakis, Pavlos Fafalios, and Yannis Marketakis. 2022. CIDOC-CRM and Machine Learning: A Survey and Future Research. *Heritage* 5, 3 (2022), 1612–1636. https://doi.org/10.3390/heritage5030084

[34] Savvas Varitimiadis, Konstantinos Kotis, Dimitris Spiliotopoulos, Costas Vassilakis, and Dionisis Margaris. 2020. "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*. Springer, 281–299.

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

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