Move Cultural Heritage Knowledge Graphs in Everyone's Pocket

Maria Angela Pellegrino [a,*], Vittorio Scarano [a] and Carmine Spagnuolo [a]

[a] *Dipartimento di Informatica, Università degli Studi di Salerno, Italy*
E-mails: mapellegrino@unisa.it, vitsca@unisa.it, cspagnuolo@unisa.it

Abstract. Last years witnessed a shift from the potential utility in digitization to a crucial need to enjoy activities virtually by noting that while in the past (before 2019) data curators recognised the utility in performing data digitization, during the COVID-19, due to the lockdown, no one could enjoy Cultural Heritage in person and it required a great investment in remotely offering activities to make culture survive. The Cultural Heritage community heavily invested in digitization campaigns, mainly modeling data as Knowledge Graphs by becoming one of the most successful application domains of the Semantic Web technologies.

Despite the vast investment in Cultural Heritage Knowledge Graphs, the syntactic complexity of RDF query languages, e.g., SPARQL, negatively affects and threatens data exploitation, risking leaving this enormous potential untapped. Thus, we aim to support the Cultural Heritage community (and everyone interested in Cultural Heritage) in querying Knowledge Graphs without requiring technical competencies in Semantic Web technologies.

We propose an engaging exploitation tool accessible to all without losing sight of developers' technological challenges. Engagement is achieved by letting the Cultural Heritage community leave the passive position of visitor and actively create their Virtual Assistant extensions to exploit proprietary or public Knowledge Graphs in question-answering. Accessible to all underlines that we propose a software framework freely available on GitHub and Zenodo with an open-source license. We do not lose sight of developers' technical challenges, carefully considered both in the design and in the evaluation phases.

This article, first, analyzes the effort invested in publishing Cultural Heritage Knowledge Graphs to quantify data on which developers can rely in designing and implementing data exploitation tools in this domain. Moreover, we point out data aspects and challenges that developers may face in exploiting them in automatic approaches. Second, it presents a domain-agnostic Knowledge Graph exploitation approach based on virtual assistants as they naturally enable question-answering features where users formulate questions in natural language directly by their smartphones. Then, we discuss the design and implementation of this approach within an automatic community-shared software framework (a.k.a. generator) of virtual assistant extensions and its evaluation on a standard benchmark of question-answering systems. Finally, according to a taxonomy of the Cultural Heritage field, we present a use case for each category to show the applicability of the proposed approach in the Cultural Heritage domain. In overviewing our analysis and the proposed approach, we point out challenges that a developer may face in designing virtual assistant extensions to query Knowledge Graphs, and we show the effect of these challenges in practice.

Keywords: Community-shared software framework, Question-answering, Virtual assistant, Knowledge graph, SPARQL

1. Introduction

In the last decade, public institutions and private organizations have invested in massive digitization campaigns to create vast digital collections, repositories, and portals that allow online and direct access to billions of resources [1]. Digitization causes an extraordinary acceleration in digital transformation processes [2] that affected any field, from education to business models [3], from health care [4] to Cultural Heritage (CH) [2]. Focusing on the CH field, public and private organizations have invested in digitizing any form of data to ensure its long-term preservation and support the knowledge economy [1].

The United Nations Educational, Scientific and Cultural Organization (UNESCO) defines CH as "*the legacy of physical artifacts and intangible attributes of a group or society inherited from past generations,*

[*] Corresponding author. E-mail: mapellegrino@unisa.it.

maintained in the present and bestowed for the benefit of future generations" [5]. CH includes *tangible* culture (such as buildings, monuments, landscapes, books, works of art, and artifacts); *intangible* culture (such as folklore, traditions, language, and knowledge), and *natural* heritage (including culturally significant landscapes, and biodiversity) [5].

Nowadays, CH has become one of the most successful application domains of the Semantic Web technologies [6]. Both public institutions (e.g., galleries, libraries, archives, and museums, a.k.a. GLAM institutions) and private providers modelled and published CH as Knowledge Graphs (KGs), i.e., a combination of ontologies to model the domain of interest and data published in the linked open data (LOD) format [7], both as independent datasets or by enriching aggregators (such as Europeana [8]) [6].

CH as LOD improves data reusability and allows easier integration with other data sources [6]. It behaves as a promising approach in facing CH challenges, such as syntactically and semantically heterogeneity, multilingualism, semantic richness, and interlinking nature [9]. The availability of CH data in digital machine-processable form has enabled a new research paradigm called Digital Humanities [6] and aims to facilitate researchers, practitioners, and generic users to consume cultural objects [9].

However, KG exploitation is mainly affected by i) required technical competences in generic query languages, such as SPARQL, and in understanding the semantics of the supported operators [10], too challenging for lay users [10–14], and ii) conceptualization issues to understand how data are modelled [10, 11].

Natural Language (NL) interfaces mitigate these issues, enabling more intuitive data access and unlock the potentialities of KGs to the majority of end-users [15]. NL interfaces provide lay users with question answering (QA) functionalities where users can adopt their terminology and receive a concise answer. Researchers argue that multi-modal communication with virtual characters using NL is a promising direction in accessing KGs [16]. Consequently, virtual assistants (VAs) have witnessed an extraordinary and increasing interest as they naturally behave as QA systems. Many companies and researchers have combined (CH) KGs and VAs [1, 17, 18], but no one has provided end-users with a generic methodology to generate extensions to querying KGs automatically.

To fill this gap, our *goal* is the definition of a general-purpose approach that makes KGs accessible to all by requiring minimum-no technical knowledge in Semantic Web technologies. VAs usually give the possibility to extend their capabilities by programming new features, also referred to as VA extensions. It implies that (potentially) everyone can implement custom extensions and personalize the VA behaviour. However, playing the VA extension creator's role requires programming competences to design and implement the application logic. Moreover, users must be aware that VA extensions are provider-dependent meaning that an extension implemented for Alexa will not be directly reusable for other providers.

We desire to empower lay-users by letting them leave VA users' passive position and play the role of VA extensions creator by requiring little/no-technical competences. We reformulate the goal of this work as i) enabling QA over KGs (KGQA) by VAs and ii) allowing (lay) users to automatically create ready-to-use VA extensions to query KGs by popular VAs, e.g., Amazon Alexa and Google Assistant. Thus, we propose a community-shared software framework (a.k.a. generator) that enables lay users to create custom extensions for performing KGQA for any cloud provider, unlocking the potentialities of the Semantic Web technologies by bringing KGs in everyone's "*pocket*", accessible from smartphones or smart speakers.

To determine the quantity of CH data modeled as KGs on which developers can rely in designing data exploitation tools in this domain, we overview the CH community effort to create, publish, and maintain KGs belonging to any category determined by the CH taxonomy. During the performed analysis, we point out which KG aspects and challenges developers may face in designing an automatic approach to exploit CH KGs. This analysis behaves as a starting point to design the proposed domain-agnostic approach to query (CH) KGs via VAs. To materialize this approach, we implement it in an automatic generator of VA extensions provided with KGQA functionalities. We summarize the configuration of the generator and the process to create a VA extension in Fig. 1. The generator architecture in Fig. 1 represents the community shared software framework that will be detailed in Fig. 6. The process starts with a user-defined URL of the link to a working SPARQL endpoint of interest. The returned VA extension is ready-to-be-use, and it can be used to perform QA, as simulated in Fig. 1 which will be detailed in Fig. 5 to understand the VA extension behaviour fully. We overview VA extensions in the CH field as use cases. In particular, we present a VA extension for each CH data category to demonstrate the generator in action and show that the proposed approach

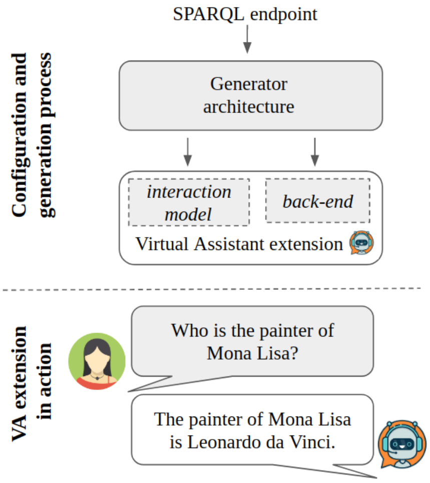

SPARQL endpoint
Configuration and generation process
Generator architecture
interaction model
back-end
Virtual Assistant extension
VA extension in action
Who is the painter of Mona Lisa?
The painter of Mona Lisa is Leonardo da Vinci.

Fig. 1. Overview of the process to configure the generator and create the Virtual Assistant extensions (detailed in Fig. 6) and the extension in action (interaction which will be discussed in Fig. 5).

is general enough to work with any CH data. To assess the quality of the produced VA extensions and draw out differences in generator configuration options, we design VA extensions for well-known general-purpose KGs, i.e., DBpedia and Wikidata, and we evaluate them on a standard evaluation benchmark for KGQA systems, i.e., QALD. Finally, we perform i) a preliminary user experience to estimate the usability according CH experts in using an auto-generated VA extension for the UNESCO Thesaurus and ii) we collect the perceived impact and utility of the proposed approach according to end-users and data curators.

The major *contributions* of this paper follows.
- A design methodology to enable lay-users without technical competences in programming and query languages to author VA extensions (Section 4).
- An approach to make KGs compliant with VAs for the KGQA task (Section 4).
- A software tool architecture to automatically generate personalized, configurable, and ready-to-use VA extensions where ready-to-use means that they can be uploaded on VA service providers as manually generated ones (Section 5).
- The open-source release of the software framework *v*1.0 that supports Amazon Alexa, publicly available on the project GitHub repository[1].
- A detailed review and analysis of the CH community effort in publishing KGs and registering them in standard dataset repositories (Section 3).
- The open-source release of a pool of Alexa skills resulting from the generator exploitation to query CH KGs (Section 6 and GitHub repository[1]). We present a use case for each CH category. In particular, for the tangible category, we propose the Mapping Manuscript Migrations (MMM) use case for the movable subcategory, and the Hungarian museum use case for the immovable one; DBTune for the intangible category; and NaturalFeatures for the natural heritage category. During the performed analysis, we noticed a particular interest in taking care of CH terminology and modeling approaches by thesaurus and models. Therefore, we also present the UNESCO thesaurus use case for the Terminology category.

The rest of this article is structured as follows: Section 2 overviews related work in (CH) KGQA by traditional approaches and by VAs; Section 3 quantifies the CH community effort in publishing KGs by analyzing the status of the provided services and the amount of published data. This analysis aims to justify the advantages of investing in designing and developing technological solutions to engage lay-users interested in CH and exploit the vast amount of available data in this domain. Section 4 details the proposed domain-agnostic approach to query KGs by VAs by pointing out technological challenges in interfacing KGs and VAs, providing design principles and the implementation methodology, and discussing its strengths and limitations. Section 5 overviews the VA extension generator that embeds the proposed general-purpose approach in querying KGs, while Section 6 presents a pool of VA extensions to query CH KGs by showing the general approach in a domain-specific application and by focusing on the impact of the design challenges in the CH context. Section 7, first, assesses the performance of the generated VA extensions by evaluating its accuracy on general-purpose use cases (DBpedia and Wikidata) by using standard evaluation benchmarks, the QALD dataset, second, reports the user experience of the HETOR group in using the UNESCO VA extension to simulate the support in class in clarifying terminology and term hierarchies concerning CH, and, finally, discusses the impact and the potentialities of the proposed approach according to end-users and CH experts. Finally, it concludes with some final remarks and future directions.

[1] `mapellegrino/virtual_assistant_generator`. Permanent URI: https://zenodo.org/record/4605951

2. Related work

QA systems can be classified as domain-specific (a.k.a. closed domain) or domain-independent (a.k.a. open domain). While in domain-independent QA, there is no restriction on the question domain, and systems are usually based on a combination of Information Retrieval and Natural Language Processing techniques [19]; in domain-specific QA, questions are bound to a specific context [20] and developers can rely on techniques that are tailored to the domain of interest [21]. Besides the scope, they can be classified by the type of questions it can accept (e.g., facts or dialogs) and queried sources (structured vs. unstructured data) [22]. While systems querying text collections are classified as tools working on unstructured data (e.g., WEBCOOP [23]), systems querying KGs are classified as tools working on structured data. According to this classification, we propose an approach to pose factoids questions (wh-queries, e.g., who, what, and how many, and affirmation/negation questions) over semantically structured data where questions aim to be as general as possible to classify our proposal as a domain-independent approach.

KGQA is a widely explored research field [24–26]. While it is rare to observe keyword-based questions, most of the KGQA systems address full NL questions. Usually, questions can be posed in English, while some tools deal with European and non-European languages [24]. There is a consistent effort in proposing domain-independent QA systems to query DBpedia and Wikidata [24, 25] by exploiting heterogeneous solutions ranging from combinatorial approaches [24] to neural networks [25], from graph-based solutions [26] to NL request mapping to SPARQL queries [27].

By focusing on **CH KGQA**, i.e., domain-specific systems in the CH domain, they can benefit from many standard data sources. CIDOC Conceptual Reference Model (CRM) is an example in this direction, and it is widely adopted as a base interchange format by GLAM institutions all over the world [28]. CIDOC-CRM has been identified as the knowledge reference model for PIUCULTURA project, funded by the Italian Ministry for Economic Development, which aims to devise a multi-paradigm platform that facilitates the fruition of Italian CH sites. Within the PIUCULTURA project, Cuteri et al. [21] proposed a QA system tailored to the CH domain to query both general (e.g., online data collections) and specific (e.g., museums databases) CIDOC-compliant knowledge sources by exploiting logic-based transformation. As an alternative approach, PowerAqua [29] maps input questions to SPARQL templates under the hypothesis that the SPARQL query's overall structure is almost determined by the syntactic structure of the NL question.

KGQA via VAs is natively offered in well-known VAs, such as Google Assistant and Alexa, that provide users with content from generic KGs (Google Search and Microsoft Bing, respectively). Thus, available commercial VA providers offer inner KGQA to reply, among others, to questions concerning widely known and well established museums, monuments and artworks of interest for the general public. However, end-users miss the opportunity to customize VA extensions behaviour to query data of interest, less established data sources, custom available and working SPARQL endpoints. In fact, the main limitations of commercial VA providers are that these tools query proprietary and general-purpose KGs without exploring domain-specific QA, and the proposed mechanisms can not be extended by end-users and ported on other KGs. Therefore, the Semantic Web community invested in increasing VA capabilities by providing QA over open KGs. Among others, Haase et al. [30] proposed an Alexa skill to query Wikidata by a generic approach, while Krishnan et al. [31] made the NASA System Engineering domain interoperable with VAs.

By considering **CH KGQA via VAs**, CulturalERICA (Cultural hERItage Conversational Agent) [1] is an intelligent conversational agent to assist users in querying Europeana [8] via NL interactions and Google Assistant technology. Authors state that CulturalERICA is database independent and can be configured to serve information from different sources. Besides technological differences (we opt for Alexa while they opt for Google Assistant), while they enable iterative refinement of the queries, at the moment, we only provide one-step iterations. However, they only enable path traversal, while we also support more complex queries, such as sort pattern, numeric filters, class refinement. Anelli et al. [17] developed a VA to enable the exploitation of the Puglia Digital Library by delegating the speech recognition to Google Assistant. Through subsequent interactions, the VA creates and keeps the context of the request. While they enable keyword-based search, we opt for complete NL questions. Cuomo et al. [18] proposed an answering system and adapted it to implement a VA able to reply to questions about artworks exposed in Castel Nuovo's museum in Naples. Their proposal aims to reply to questions about artworks, their author, and related information posed by visitors during the touristic tour. Even if it represents an interesting work in the direction of CH KGQA via VAs, it is bound to hardware devices within the museum, and it is not a solution that users can exploit everywhere, by their smartphones.

About the integration of CH KGs and chatbots, we can cite the chatbot proposed by Lombardi et al. [32] able to support users during an archaeological park visit (i.e., in Pompeii) by simulating the interaction between visitors and a real guide to improve the touristic experience by exploiting NL processing techniques. In the same direction, Pilato et al. [33] propose a community of chatbots (with specialized or generic competencies) developed by combining the Latent Semantic Analysis methodology and the ALICE technology.

These works behave as evidence of the interest in developing KGQA via VA by promoting interesting applications to make CH KGs interoperable with VAs to accomplish the QA task, but they do not empower end-users by providing them with the opportunity to create their VA extensions. In fact, the main difference between our proposal and the ones reported so far is that the literature proposes ready-to-use VA extensions, while we are proposing a generator of VA extensions that is bounded to neither any KG nor any specific VA provider. To the best of our knowledge, the proposed community-shared software framework is the first attempt to provide users without technical competences in the Semantic Web technologies to create KGQA systems via VAs. Consequently, it represents the main *novelty* of our proposal.

3. Cultural Heritage Knowledge Graph Analysis

This section analyzes the CH community effort in publishing CH data as KGs, in making them accessible by either SPARQL endpoints or APIs, in maintaining working SPARQL endpoints in most of the cases, in attaching human-readable labels to resources to making them accessible by NL interfaces. The performed analysis aims to make evident the potentialities of proposing exploitation tools in this application domain due to the vast amount of available data. In particular, this survey quantifies the amount of available CH KGs behaving as a source for the proposed generator and it estimates some of the aspects that are crucial for making data accessible by any data exploitation tool, such as, accessibility by a working SPARQL endpoint, and by NL interfaces, such as VA providers, that require the use of labels attached to resources.

First, it overviews the used sources to retrieve the analyzed KGs; second, it provides KG details and quantitative analysis of available data, and, finally, it points out considerations to take into account in proposing an exploitation tool for (CH) KGs.

Selection approach It is worth clarifying that we do not aim to provide a complete overview of all published KGs in the CH context, but the described selection process seeks to point out the absence of bias in the selected KGs and, consequently, the impartiality of the considerations reported in the performed analysis.

We perform the KG selection as a non technical user, by looking at available aggregators of published KGs and querying their user interfaces. We exploit LOD cloud [34] (updated in May 2020) as it is one of the biggest aggregators of published KGs, and a combination of datasets and articles search engines. In particular, we explor datasets aggregators not specifically related to the Semantic Web, such as datahub [35]. Finally, we consider recent publications available in Scopus to identify also KGs published recently. The variety of queried sources aim to demonstrate the lack of bias in the performed analysis. We collect more than `60` KGs covering more than `20` countries.

1 - We exploit the LOD cloud [34] search interface to retrieve KGs containing *museum*, *library*, *archive*, *cultur**, *heritage*, *bibliotec**, *natural*, *biodiversity*, *geodiversity*. It is worth noting that the search engine requires that the dataset title includes English terms, but it does not pose any constraint on the provider country.

2 - We retrieve datasets registered in the datahub with format equals to `api/sparql`. We manually inspect the `710` returned datasets by looking for *museum*, *library*, *archive*, *culture*, *heritage*, *bibliography*, *natural*, *biodiversity*, *geodiversity*, and similar terms in dataset title and description. Datahub also returns the SPARQL endpoint attached to retrieved datasets. When the specified endpoint is not more available, we search the dataset name attached to “SPARQL endpoint” on the Google search engine to determine if any URL migration took place.

3 - We inspect articles indexed by Scopus and matching the article title, abstract, and keyword filter *("cultural heritage" and ("semantic web" OR "linked data" OR "knowledge graph"))* from 2020 to 2018 (i.e., last two years). It results in `150` articles. We manually check them to verify if they present a KG publication and if so, we further check if authors expose API or a SPARQL endpoint.

KG details. According to the taxonomy of the CH term, we classify CH KGs according to its content by distinguish *tangible* (further classified as *movable* and *immovable*) (see Tab. 1), *intangible* (see Tab. 2) and *natural heritage* (see Tab. 3). Moreover, we notice an interesting amount of KG dedicated to clarify and model CH terminology interpreted as the effort invested in defining thesaurus and data models. Therefore, we also consider the *terminology* class as reported in Tab. 4). If a KG contains elements belonging to multiple classes, we repeat it. For each KG, we report the *original name*, the *country of the provider*, the *service* that enables data exploitation (SPARQL endpoint or API), and the *SPARQL endpoint status* (working or unavailable). It represents the assessment of data accessibility, that is required by any data exploitation tool. For each KG, we also generate a *short name* (mainly combining country and some name keywords clarifying KG content) to refer them in the following analysis quickly. Main observations follow.

World-wide investment. We overview country distribution and CH KG categories of the retrieved collection (see Fig. 2). It is interesting to notice that there is a consistent contribution from European countries, probably due to the vast amount of available raw data and the interest posed in Semantic Web technologies. While Australia and United States gave an interesting contribution to tangible goods, Asian countries also invested in natural heritage. By zooming on Europe (Fig. 2), it is evident that almost every country contributes to CH KGs, mainly in tangible CH. Spain, Netherlands, and Germany can be recognized as main contributors, followed by Italy, England, and Finland. France mainly invested in terminology.

Investment in all the CH KG categories. There is a substantial interest not only in materializing data but also in defining models (mainly tailored to libraries, archives, and museums [36]) and precise terminology by thesaurus (10/61 = 17%). For instance, the CIDOC-CRM is a theoretical model for information integration in the field of CH. It can help researchers and interested people in modeling CH collections and documents. Data exploitation tools should verify the proposed approach effectiveness by querying KG belonging to all the categories to confirm if it is interoperable with any data format and content.

SPARQL endpoints VS APIs. Few KGs only provide APIs (8%), while most opt for SPARQL endpoints. Some providers, e.g., Europeana [8], invest in both the access points. Therefore, developers should be aware of available services in designing data exploitation tools to define the best approach to query (CH) KGs. We opt for querying them by SPARQL endpoints as represents a more general and standard approach to query KGs and most of the CH KGs configure them.

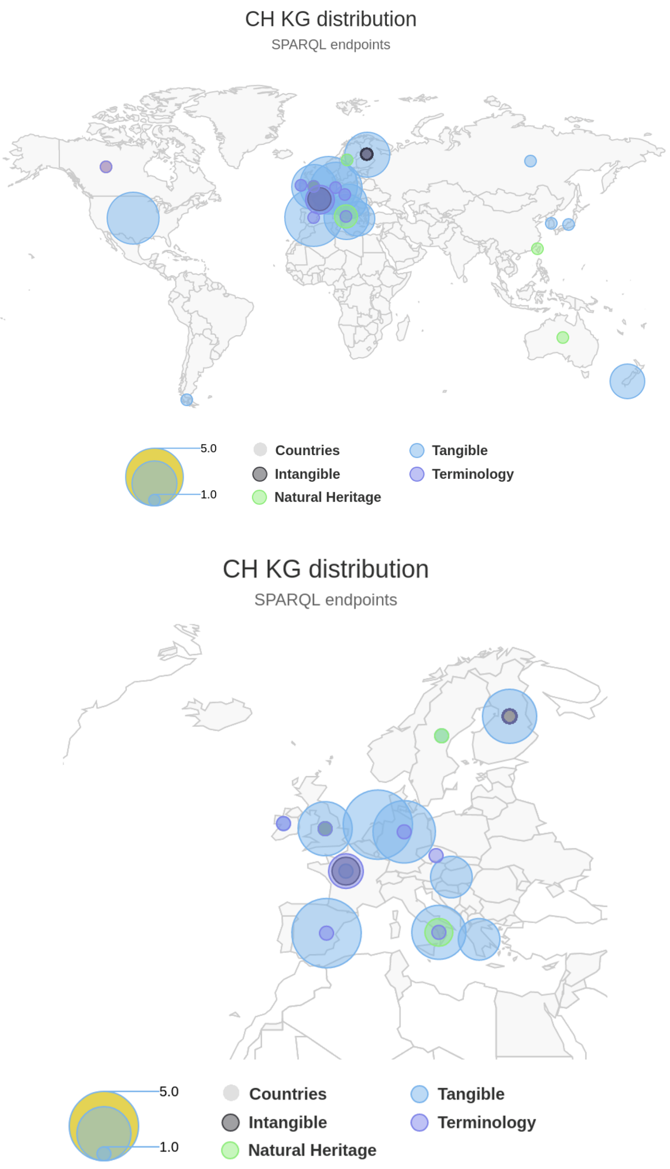


Fig. 2. Geographical distribution of CH KGs. The bubble size represents the number of available CH KGs.

Discontinuous effort. By looking at the ratio between working and discontinued SPARQL endpoints (see Fig. 3), in all the categories, there are SPARQL endpoints that are no more available. In some categories, such as tangible heritage, discontinued SPARQL endpoints reach almost half of the total available end-

Table 1

Overview of KGs related to *tangible CH*. It contains the *sub-category* interpreted as *movable* and *immovable*, a *short name* of KG to make shorter the following references, the complete *name*, the *country* of the provider, the *Service* that enables the LOD exploitation (SPARQL endpoint or API), and SPARQL endpoint *status* (✓ means that it works, while empty cells mean it does not; hyphen means not applicable).

Sub-category	Short name	Name	Country	Service	Status
Movable	ARCO	ARCO	IT	SPARQL	✓
	DigitalNZ	DigitalNZ	NZ	API	-
	Bibliopolis	Bibliopolis	USA	SPARQL	
	Europeana	Europeana	NL	SPARQL	✓
	FondazioneZeri	Fondazione Zeri	IT	SPARQL	✓
	MMM	Mapping Manuscript Migrations	FI	SPARQL	✓
	NL_maritime	Dutch Ships and Sailors	NL	SPARQL	✓
	Nomisma	Nomisma	DE	SPARQL	✓
	Yale	Yale center of British Art	GB	SPARQL	✓
Immovable	DPLA	Digital Public Library of America	USA	API	-
	NZ_museum	Auckland Museum	NZ	API	-
	ADL	Alexandria Digital Library Gazetteer	USA	SPARQL	
	Arc.	Architectural Data	IE	SPARQL	
	ARTIUM	Library and Museum of ARTIUM	ES	SPARQL	
	B3Kat	Libraries of Bavaria, Berlin and Brandenburg	DE	SPARQL	✓
	GB_museum	British museum	GB	SPARQL	
	Cervantes_lib	Biblioteca Virtual Miguel de Cervantes	ES	SPARQL	✓
	CL_library	Biblioteca del Congreso de Chile	CL	SPARQL	✓
	DE_library	Mannheim University Library	DE	SPARQL	
	ES_cultura	Spanish National Library	ES	SPARQL	✓
	ES_library	National Library of Spain	ES	SPARQL	✓
	FI_library	Finnish Public Libraries	FI	SPARQL	✓
	FI_museum	Finish museum	FI	SPARQL	✓
	FR_library	French National Library	FR	SPARQL	✓
	GB_library	British National Bibliography	GB	SPARQL	✓
	GR_library	National Library of Greece Authority Records	GR	SPARQL	
	GR_Veroia_lib	Public Library of Veroia	GR	SPARQL	
	HEBIS	HEBIS – service for libraries	DE	SPARQL	
	Hedatuz	Basque culture and science digital library	ES	SPARQL	
	HU_archive	National Digital Data Archive of Hungary	HU	SPARQL	✓
	HU_museum	Museum of Fine Arts Budapest	HU	SPARQL	✓
	IT_museum	Italian museums	IT	SPARQL	
	JP_library	Japan's National Library	JP	SPARQL	✓
	KR_library	National Library of Korea	KR	SPARQL	✓
	LIBRIS	LIBRIS: Swedish National Bibliography	SE	SPARQL	
	NL_library	Dutch National Bibliography	NL	SPARQL	✓
	NL_archeology	Linked Data Cultural Heritage Agency of the Netherlands	NL	SPARQL	
	Rijksmuseum	Rijksmuseum	NL	SPARQL	
	RU_museum	Russian Museum	RU	SPARQL	
	USA_museum	Smithsonian Art Museum	USA	SPARQL	

Table 2

Overview of KGs related to *intangible CH*. It contains a *short name* of KG to make shorter the following references, the complete *name*, the *country* of the provider, the *Service* that enables the LOD exploitation (SPARQL endpoint or API), and SPARQL endpoint *status* (✓means that it works, while empty cells mean it does not; hyphen means not applicable).

Short name	Name	Country	Service	Status
DBTune	DBTune Western Classical Music	GB	SPARQL	✓
EventMedia	EventMedia	FR	SPARQL	✓
FI_folklore	Semantic Kalevala and Folklore	FI	SPARQL	✓
Munnin	First World War (Muninn project)	CA	SPARQL	
MusicKG	MusicKG	FR	SPARQL	
WarSampo	WarSampo	FI	SPARQL	✓

Table 3

Overview of KGs related to *natural heritage*. It contains a *short name* of KG to make shorter the following references, the complete *name*, the *country* of the provider, the *Service* that enables the LOD exploitation (SPARQL endpoint or API), and SPARQL endpoint *status* (✓means that it works, while empty cells mean it does not; hyphen means not applicable).

Short name	Name	Country	Service	Status
ARCO	ARCO	IT	SPARQL	✓
EcoPortal	EcoPortal	IT	API	-
Ecology	Linked Open Data of Ecology	TW	SPARQL	
CarbonPortal	Carbon Portal	SWE	SPARQL	✓
NaturalFeatures	Natural Features	GB	SPARQL & API	✓
Ozymandias	Ozymandias	AUS	SPARQL	✓

Table 4

Overview of KGs related to *terminology*. It contains the *sub-category* interpreted as *thesaurus* and *model*, a *short name* of KG to make shorter the following references, the complete *name*, the *country* of the provider, the *Service* that enables the LOD exploitation (SPARQL endpoint or API), and SPARQL endpoint *status* (✓means that it works, while empty cells mean it does not; hyphen means not applicable).

Sub-category	Short name	Name	Country	Service	Status
Thesaurus	AAT	The Art & Architecture Thesaurus	CA	SPARQL	✓
	ES_thesaurus	Encabezamientos para las Bibliotecas Públicas	ES	SPARQL	✓
	FR_archive	Thesaurus for Local Archives	FR	SPARQL	
	GB_thesaurus	English Heritage Periods List	GB	SPARQL	✓
	Loanword	World Loanword Database	DE	SPARQL	
	Logainm	Placenames Database	IE	SPARQL	✓
	BNCF	Thesaurus National Central Library of Florence	IT	SPARQL	✓
	UNESCO	UNESCO thesaurus	FR	SPARQL	✓
Model	CIDOC-CRM	CIDOC-Conceptual Reference Model	FR	SPARQL	✓
	MONDIS	Monument Damage Ontology	CZ	API	-

points. Since many endpoints do not work anymore, it shows a discontinuous investment in CH KGs or the lack of attention in updating the dataset search engines when a SPARQL endpoint URL migration occurs.

Quantitative overview of available data. Concerning data quantity, we consider the amount of collected datasets and the number of classes, predicates and triples accessible by a working SPARQL endpoint. It quantifies CH KGs data to provide a perception of available sources that can be exploited by automatic data exploitation tools behaving as SPARQL query builders. From a quality point of view, we report the percentage of classes and predicates provided with a human-readable label, which is a crucial aspect for NL interfaces, such as VA extensions. For each working SPARQL endpoint listed in Tables 1 – 4, we retrieve:

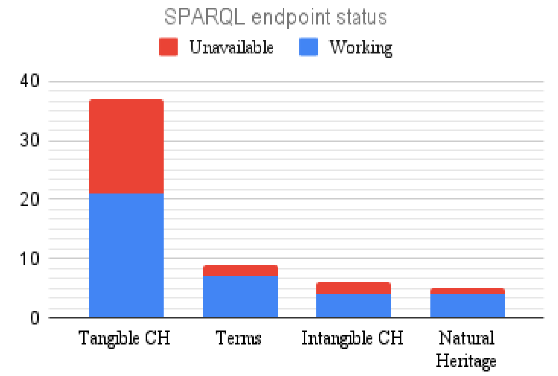


Fig. 3. CH KG SPARQL endpoints status. While blue represents working SPARQL endpoints, red represents unavailable ones.

- *classes*, both used classes returned by the `select count(distinct ?c) where {[] a ?c}` query, and the ones declared as rdfs:Class, skos:Concept and owl:Class. Moreover, we also ask for their labels (referred to by rdfs:label in all the cases but skos:Concept, where we asked for skos:prefLabel) (see Tab. 5).
- *properties*, both used properties returned by the `select count(distinct ?p) where {?s ?p ?o}` query, and the ones declared as owl:DatatypeProperty, owl:ObjectProperty, and rdf:property. Moreover, we also ask for their rdfs:label in all the cases (see Tab. 6).
- *triples* returned by the `select * where {?s ?p ?o}` query (see column Triples in Tab. 6).

Main observations follow, and they should guide developers in designing automatic data exploitation tools by considering technical constraints posed by available data access points and data properties.

Label provision. Table 5 and 6 detail the percentage of classes and properties provided with labels. If developers aim to rely on human-readable labels, they should carefully check them to avoid losing too much data if they only retrieve classes or properties already attached to labels. Some endpoints fail in retrieving labels, such as HU_archieve, KR_library, Nomisma, ARCO, B3Kat, NL_library, NL_maritime, and Yale (grey lines in Tab. 5 and 6). It evidences a lack of care in attaching human-readable labels to resources by standard approaches, such as rdfs:label. While there is a consistent interest in attaching human-readable labels to classes, properties are rarely provided with labels. Developers can complete missing labels by generating them from URI local names. However, this practice can be performed only if KGs adopt human-readable URIs. Lack of label provision is an obstacle to refer and understand resources.

Language support. Multilingualism is a desirable property in the CH community. However, in many cases, labels are defined in just one language (such as in Japanese for JP_library, Spanish for ES_Thesaurus). In some cases, KG providers expose at least labels in the national language and English (such as ARCO CL_library, FI_museum, KR_library). Rare are broader language support; e.g., Nomisma enumerates 177 languages. Moreover, sometimes the language tag is omitted. For instance, GB_thesaurus and Yale are provided with English labels, but if someone explicitly asks for `en` as language tag, it returns no results.

SPARQL support. If developers choose to query a SPARQL endpoint or exploit dedicated API directly, they must verify the SPARQL operator support and coverage. For instance, AAT, B3Kat, Cervantes_lib, and Ozymandias do not support the `COUNT` operator; JP_library, ES_library, ES_cultura, GB_thesaurus and CIDOC-CRM do not support the `BIND` operator; GB_thesaurus do not support the `DISTINCT` operator. This analysis affects the supported SPARQL patterns in QA applications (e.g., VA extension back-end).

Query failures. Even if some SPARQL endpoints work apparently, some of them partially or entirely fail in returning results. For example, ES_library and Europeana fail in returning properties by a SPARQL query. If developers require retrieving available data, they have to check the way to query them carefully.

Result limit. Some KGs pose a result limit that forces running multiple queries to retrieve all the results. It spans from 100 of KR_library, 500 for HU_archieve to 10000 Europeana. It should be taken into account in verifying the completeness of a single query result.

Running time. We tested SPARQL endpoint execution time by posing 10 times the query to retrieve a used class (by posing the `SELECT ?c WHERE{[] a ?c} LIMIT 1` query) and the one to retrieve a single triple (by posing the `SELECT * WHERE{?s ?p ?o} LIMIT 1` query). While 24/35 return a class in less than 19s, 3/35 require a half minute, KR_library requires 2m, ES_thesaurus requires 10m, and 4/35 fails in returning any reply. The triple query execution time returns comparable results to class retrieval run time. The running time may affect the performance of any interactive data exploitation tool. It is crucial to minimize it as much as possible.

Table 5

Overview of classes in CH KGs (by only considering SPARQL endpoints). It contains the *used classes* and the classes declares as *skos:Concept*, *rdfs:Class* and *owl:Class*. Moreover, it contains the percentage of classes provided with a label (besides its language). Grey lines are endpoints which fail at least a SPARQL query.

Short Name	Used Class		skos:Concept		rdfs:Class		owl:Class	
	TOT	% with label	TOT	% with label	TOT	% with label	TOT	% with label
AAT	75	24	2871894	100	93	25	27	85
ARCO	488	63	30000	100	56	100	615	77
B3Kat	31	0	270	0	18	0	0	0
Cervantes-lib	22	0	64	100	30	0	0	0
CIDOC-CRM	5	40	0	0	102	10	5	80
CL_library	502	39	8334	100	38	100	352	93
DBTune	14	0	0	0	0	0	1	0
ES_cultura	31	52	3	100	15	100	103	99
ES_library	28	14	10000	100	15	100	3	67
ES_thesaurus	2	0	30000	100	0	0	0	0
Europeana	30	13	10000	100	15	100	3	67
EventMedia	50	10	1471	100	56	100	3	67
FI_folklore	17	47	26122	100	0	0	23	87
FI_library	61	0	0	0	0	0	62	0
FI_museum	18	0	0	0	0	0	0	0
FondazioneZeri	105	0	0	0	0	0	0	0
FR_library	38	11	1000000	100	15	100	4	50
GB_library	46	0	1048576	0	0	0	0	0
GB_thesaurus	13	23	500	100	0	0	5	80
HU_archive	469	fail	500	100	384	100	500	100
HU_museum	89	43	10000	100	80	100	129	99
JP_library	5	0	100	0	0	0	0	0
KR_library	53	fail	100	100	0	0	85	33
Logainm	114	4	0	0	57	98	5	40
MMM	58	50	0	0	22	36	124	100
NaturalFeatures	322	90	2519	93	355	100	365	93
NL_library	34	3	113527	100	27	100	0	0
Nl_maritime	92	80	52282	59	131	100	86	90
Nomisma	64	33	107091	fail	4	50	47	70
Ozymandias	30	0	0	0	0	0	0	0
UNESCO	10	20	4427	100	0	0	4	50
WarSampo	90	9	7090	96	84	10	86	3
Yale	43	0	19020	99	52	0	0	0

4. Question-Answering over Knowledge Graph via Virtual Assistants

This section introduces the design methodology to make KGs compliant with VA to address the QA task. We focus on Amazon Alexa and its terminology without losing generality, as the same considerations can also be adapted for other customizable providers. Alexa VA extensions are named `skills`, and they include both the interaction model and the back-end logic. The interaction model defines the supported features referred to as `intents`, and each intent can be modelled by a set of `utterances`, i.e., phrases to invoke it. Utterances may specify a set of `slot` keywords, i.e., variables that will be instantiated according to the users' requests.

The KGQA task can be defined as follows: given an NL question Q and a KG K, the QA system produces the answer A, which is either a subset of entities in K or the result of a computation performed on this subset,

Table 6

Overview of properties in CH KGs and triples (by only considering SPARQL endpoints). It contains the *used properties* and the properties declared as *owl:ObjectProperty*, *owl:DatatypeProperty* and *rdf:Property*. Moreover, it contains the percentage of properties provided with a label (besides its language). Grey lines are endpoints which fail at least a SPARQL query.

Short Name	Used Property		owl:ObjectProperty		owl:DatatypeProperty		rdf:Property		Triples
	TOT	% with label	TOT	% with label	TOT	% with label	TOT	% with label	
AAT	43	2	350	100	11	100	490	71	32.094.409
ARCO	945	fail	838	91	244	85	0	0	372.182.177
B3Kat	265	fail	0	0	0	0	0	0	1.022.898.443
Cervantes-lib	128	0	0	0	0	0	0	0	fail
CIDOC-CRM	31	35	0	0	0	0	0	0	5.238
CL_library	357	34	0	0	69	84	0	0	45.413.189
DBTune	38	0	2	0	0	0	0	0	419.519
ES_cultura	354	69	0	0	0	0	0	0	867.535
ES_library	fail	fail	0	0	0	0	0	0	368.989.196
ES_thesaurus	14	0	0	0	0	0	0	0	90.056
Europeana	fail	6	0	0	0	0	0	0	10.000
EventMedia	199	4	0	0	0	0	0	0	11.916.783
FI_folklore	43	19	17	94	7	100	0	0	306.549
FI_library	114	0	0	0	28	0	0	0	4.363.198
FI_museum	60	0	0	0	0	0	0	0	210.986
FondazioneZeri	124	0	0	0	0	0	0	0	fail
FR_library	862	0	0	0	0	0	64	100	fail
GB_library	173	0	0	0	0	0	0	0	204.664.490
GB_thesaurus	51	35	17	100	1	100	28	100	500
HU_archive	fail	fail	2137	23	500	100	0	0	48.378.455
HU_museum	225	19	383	99	20	40	0	0	644.276
JP_library	43	0	0	0	0	0	0	0	21.884.879
KR_library	100	fail	33	100	100	100	0	0	100
Logainm	170	5	0	0	0	0	0	0	1.344.903
MMM	152	24	379	98	9	89	0	0	24.009.834
NaturalFeatures	456	87	116	93	52	96	0	0	918.664.981
NL_library	135	fail	0	0	0	0	0	0	182.580.001
NL_maritime	431	fail	128	100	49	98	590	84	fail
Nomisma	126	23	0	0	61	98	44	0	8.602.910
Ozymandias	196	0	0	0	0	0	0	0	fail
UNESCO	46	0	0	0	3	100	0	0	97.027
WarSampo	310	5	62	5	88	0	0	0	14.322.426
Yale	86	fail	1	0	0	0	93	1	fail

such as counting or assertion replies [37]. We draw a parallel between a general process for KGQA and a VA-based process (see Fig. 4).

A general KGQA workflow is composed of the question analysis phase, followed by the query construction to retrieve results [38]. We extend this workflow by adding a final step to formulate an NL reply to verbalize the retrieved results and return it to the user. Consequently, the high-level KGQA workflow is an adaptation of the methodological approach proposed in the literature by Diefenbach et al. [38]. How this general approach has been narrowed down as a VA-based process is a proper original contribution of the paper. While the general process reports a high-level approach detailing terminology commonly used in the context of KGQA, the VA-based process narrows it down to terms related to VA extensions (such as intent, slots) and reports low-levels detailed considered in implementing a KGQA via VAs. For instance, while the general phase to retrieve the entity or predicate URI

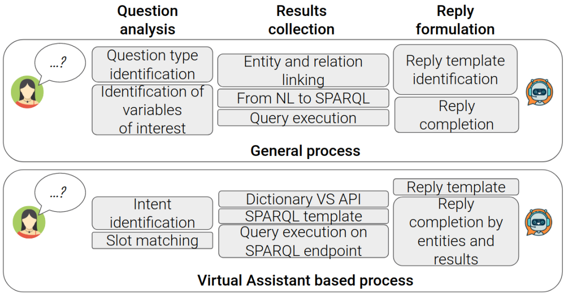

Question analysis
Results collection
Reply formulation
...?
Question type identification
Identification of variables of interest
Entity and relation linking
From NL to SPARQL
Query execution
Reply template identification
Reply completion
General process
...?
Intent identification
Slot matching
Dictionary VS API
SPARQL template
Query execution on SPARQL endpoint
Reply template
Reply completion by entities and results
Virtual Assistant based process

Fig. 4. Parallel of a general and a VA-based KGQA process.

attached to a NL label is usually named linking, in the VA-based process, it might be implemented by using dictionaries or calling APIs. While the general process focuses on the high-level role of each component, the VA-based process considers VA peculiarities and low-level implementation alternatives.

The *question analysis* step performs the question type identification and the linking phase. The *query construction* phase formulates the SPARQL query corresponding to the NL question and runs it on a SPARQL endpoint to retrieve raw results. During the *reply formulation* step, retrieved results are organized as an NL reply. In a VA-based process, users pose a question in NL by pronouncing or typing it via a VA app or dedicated device (e.g., Alexa app/device). During the *question analysis* phase, VAs interpret the request and identify the intent that matches the user query by an NL processing component. During the intent identification, VAs also solve intent slots. For instance, suppose that we implement a VA extension representing a thesaurus to recognize questions related to term definition. It might expect requests matching the template `Can you define the term <WORD>?`, where <WORD> is the slot that needs to be completed by the user. Therefore, when the user poses the question `Can you define the term <CULTURAL HERITAGE>?`, where CULTURAL HERITAGE behaves as a slot value. Once retrieved slot values, the VA extension performs the linking step to retrieve the URI(s), which may correspond to the label pronounced by users. The linking phase may be performed by consulting a lookup dictionary or by calling an API service. Completed the question analysis step, we can move to the *query formulation* step. If the KGQA system behaves as a query builder, the VA extension has to recognize the SPARQL pattern that fulfils the user request and formulate the SPARQL query. The SPARQL query can be run on the SPARQL endpoint. Finally, the VA extension performs the *reply formulation* step by identifying the reply template corresponding to the activated intent, completing it with actual results, and returning it to the user.

4.1. Design Challenges

Based on the analysis described in Section 3 and the overviewed KG aspects and issues, we identified the following challenges that must be faced in designing VA extensions to enable KGQA.

Label retrieval. According to LD principles [39], every resource must be referred to by a URI. Moreover, KG curators are encouraged to specify human-readable labels to make these URIs understandable by humans. It is crucial for making them callable by VA-based data exploitation tools. To easily configure systems able to automatically querying KGs, it is required to exploit a uniform (and standard) property to attach human-readable labels to resources. Most of the KGs attach labels to resources by standards properties, such as `rdfs:label`, `skos:prefLabel` or `foaf:name`. However, some KGs use domain-specific and custom label-properties (e.g., EventMedia uses `rnews:headline`) that makes the label retrieval step even more challenging.

Label coverage. Developers have to carefully check the percentage of URLs provided with labels (a.k.a. coverage) to avoid losing a high rate of resources by retrieving only URIs attached to human-readable labels.

Label readability. If labels contain codes (e.g., in HU_museum) or are wrongly formatted (e.g., labels are in camel notation, such as hasDate, hasUnit, shipType in NL_maritime), it is hard to recognize the desired resources when pronounced by humans.

Multilingualism. Language support is a desirable property. However, in many cases, labels are defined in just one language. It limits the use and exploitation of available sources.

Label ambiguity. If the same label is attached to several resources, it implies an ambiguous reference to a source of interest. For instance, if Apple is both used for the company and the fruit, it will be up to the VA back-end to solve the pronounced label. While it simplifies the question formulation by the user, it undermines the determinism of the question interpretation. A good trade-off must be detected to maintain the interaction as simple as possible without limiting the desired resources' user control.

Linking approach. To determine the URI corresponding to the pronounced label, developers can rely on i) APIs implemented by the KGs (such as Europeana provides search mechanism), ii) named entity resolution (NER) tools to solve entities (and properties), iii) define a dictionary to maintain a list of URIs for each label of interest, or iv) a combination of them. It affects the complexity, reliability, and size of the back-end. While the dictionary guarantees complete control of the entity and property resolution, it requires developer effort and highly affects the back-end size. As pointed out in the analysis described in Section 3, few CH KGs are provided with APIs. Concerning NER, it is a general solution to solve entity label, but i) it rarely works on properties, ii) it is hard to configure NER tools to work on KGs different from the one they are developed for, and iii) it strongly affects the reliability of the VA extension under the definition.

SPARQL support. If developers choose to query a SPARQL endpoint or exploit dedicated APIs directly, they have to check SPARQL operators' support and coverage in defining the mapping between NL and SPARQL queries in the QA tools.

Running time. Requests execution time strongly affects data exploitation tool performance.

Results limit. Results limit posed by KG services must be carefully checked since a low limit can compromise the completeness of the queries and require several questions provided by the `OFFSET` operator to have a complete reply.

4.2. Principles and Methodology

This section describes the proposed approach to design and implement a VA extension to enable KGQA by focusing on Amazon Alexa as a VA provider. It details the introduced concepts related to Alexa skills and the proposed implementation of a KGQA VA extension. It is not a loss of generality since it can be easily adapted to any other VA that enables custom VA extension definition, such as Google Assistant, or in bot implemented by Microsoft Azure Bot Service or Googlebot. We opt for Alexa instead of plausible alternatives as Amazon Alexa holds the record of the provider with the greatest number of sold devices. However, the architecture of the generator leads to an easy integration of novel VA providers, such as Google Assistant, that is actually under integration.

Amazon Alexa skills. As stated before, functionalities in Alexa are named skills. Among the supported types of Alexa skills, we are interested in *custom* Alexa skills where we can define the requests the Alexa skill can handle (intents), and the words users say to invoke those requests (utterances) [40]. An Alexa skill developer has to define a set of intents that represent actions that users can do with the resulting VA extension; a set of sample utterances that specify the words and phrases users can use to invoke the supported intents; an invocation name that identifies and wake-ups the resulting Alexa skill; a cloud-based service that accepts and fulfills these intents. Mapping utterances to intents defines the Alexa skill interaction model. Utterances can contain slots, i.e., variables bound by users when formulating their requests, that can be validated by attaching to each slot a list of valid options during the interaction model definition. The back-end code can be either an AWS Lambda function or a web service. An AWS Lambda (an Amazon Web Services offering) is a service that lets run code in the cloud without managing servers. When the user poses a question, Alexa recognizes the activated intent and communicates to back-end code both the recognized and slot(s) values. Then, the back-end can perform any necessary actions to collect results and elaborate a reply [40].

Virtual Assistants for Question-answering. We model each supported SPARQL query template as an intent. The implemented intents (listed in Table 7) are tailored towards SPARQL constructs, and they mainly cover questions related to a single triple enhanced by the refinement of the subject or object class. More in detail, we cover `SELECT` and `ASK` queries, `class specification`, numeric `filters`, `order by` to get the superlative and path traversal. Table 7 reports, for each intent, an exemplary NL query that activates the intent, the intent name, an utterance by specifying slots among braces, and the related SPARQL triples. In defining utterances, we separate the supported SPARQL patterns clearly to enable users to assess the query correctness generated out of their input. We also avoid utterance overlapping to ensure, as much as possible, a deterministic intent activation.

When the end-user poses a question, Alexa identifies the activated intent and notifies the back-end by communicating both the activated intent and the slot(s) values. For instance, in the CH use case reported in Fig. 5, users ask for Mona Lisa's painter. The VA recognises that it corresponds to the `getPropertyObject` intent with utterance *what/who is the {property} of {entity}*, *painter* as property slot, and *Mona Lisa* as entity slot.

Consequently, the entity and relation linking phase must be performed. It is worth noting that the performed task is a simplified version of the more gen-

Table 7

List of implemented intents by detailing an example that activates the intent, the intent name, an exemplary utterance where slots are represented among braces, and the SPARQL triple used in the SPARQL query formulation step.

Intent name	Utterance	SPARQL Triple
What is the {author} of {Mona Lisa}?		
getPropertyObject	What is the {p} of {e}?	<e><p>?
What is {cultural heritage}? Can you define {cultural heritage}?		
getDescription	What/Who is {e}?	<e><definition>?
Where is {Rome}? Where is the {Mona Lisa}?		
getLocation	Where is {e}?	<e><location>?
Show me {Paris}. Show me {Mona Lisa}.		
getImg	Show me {e}	<e><img>?
What has {Beethoven} as {author}?		
getPropertySubject	What has {e} as {p}?	? <p><e>
How many {paintings} are there?		
getClassInstances	How many {e} are there?	? <instanceof><e>
Which {pianist} were {influenced} by {Beethoven}?		
getPropertySubjectByClass	Which {c} were {p} by {e}?	? <instanceof><c>. ? <p><e>.
What has been {modifies} {in} {2020}?		
getNumericFilter	What has {p} {symbol} {val}?	? <p>?o. FILTER(?o <symbol><val>)
Which {source} has been {modified} {in} {2020}?		
getNumeriFilterByClass	Which {c} has {p} {symbol} {val}?	? <instanceof><c>. ? <p>?o. FILTER(?o <symbol><val>)
Which is the {creation} with the {maximum} {number of collaborators}?		
getSuperlative	What is the {c} with {sup} {p}?	? <p>?o. ORDER BY (?o). LIMIT 1
Can you verify if {intangible cultural heritage} as {folklore} as {narrower}?		
getTripleVerification	Can you verify if {s} has {o} as {p}?	ASK <s><p><o>
Give me all the results		
getAllResultsPreviousQuery	Give me all the results	-

eral entity and relation linking problem. Entity linking is generally referred to as identifying in a text snippet entities and matching these to the corresponding KG entity. For instance, mapping in the question *Who is the wife of the mayor of Rome?* the textual evidence of *Rome* has to be isolated first, and then it can be mapped to the corresponding KG entity. In our case, named entity textual evidence is already detected by VAs, and we have only to map the named entity textual evidence to a KG node (like *Rome* to the node in the graph representing the city of Rome). To perform this (simplified) linking phase an alternative is performing a dictionary lookup. In such a case, we store the mapping label-URIs in a dictionary by querying KG classes, predicates, and resources URIs and the corresponding labels. The VA extension back-end exploits the dictionary to retrieve the URI(s) corresponding to NL labels. Resolved entities and predicates are used to complete the SPARQL template. We attach to each intent a different SPARQL query template. Consequently, any NL query posed by end-users is matched to the corresponding intent (according to the VA interaction model), and each intent corresponds to a SPARQL query template (according to our approach). Readers can reconstruct the complete SPARQL query corresponding to each intent by proceeding as follows: introducing the SPARQL triple(s) reported in Table 7 with the SELECT operator and appending the

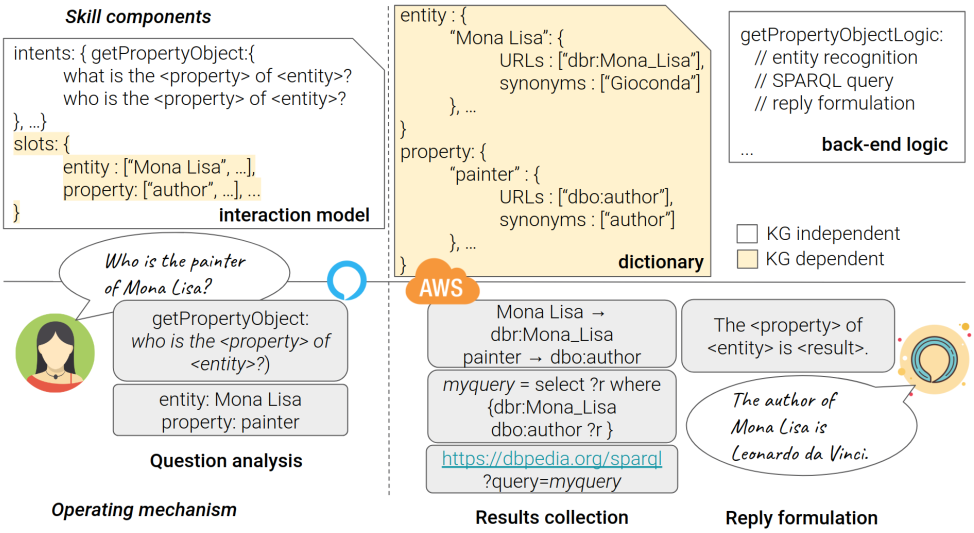

Skill components
intents: { getPropertyObject:{
what is the <property> of <entity>?
who is the <property> of <entity>?
}, ...}
slots: {
entity : ["Mona Lisa", ...],
property: ["author", ...], ...
}
interaction model
entity : {
"Mona Lisa": {
URLs : ["dbr:Mona_Lisa"],
synonyms : ["Gioconda"]
}, ...
}
property: {
"painter" : {
URLs : ["dbo:author"],
synonyms : ["author"]
}, ...
}
dictionary
getPropertyObjectLogic:
// entity recognition
// SPARQL query
// reply formulation
...
back-end logic
□ KG independent
▨ KG dependent
Who is the painter of Mona Lisa?
AWS
getPropertyObject: who is the <property> of <entity>?)
entity: Mona Lisa
property: painter
Question analysis
Operating mechanism
Mona Lisa → dbr:Mona_Lisa
painter → dbo:author
myquery = select ?r where {dbr:Mona_Lisa dbo:author ?r }
https://dbpedia.org/sparql
?query=myquery
Results collection
The <property> of <entity> is <result>.
The author of Mona Lisa is Leonardo da Vinci.
Reply formulation

Fig. 5. It is a graphical representation of the VA extension components where the yellow components are KG dependent and the VA extension in action in a CH use case by querying DBpedia.

optional request of the label attached to the variable of interest. For instance, the triple `<e><p>?` corresponds to the SPARQL query *SELECT DISTINCT ? ?label WHERE{* `SPARQL triple` *} OPTIONAL { ? <label> ?label. FILTER(LANG(?label)="en")}* (supposing that the VA extension language is English). The notation `<e>` means that the triple is completed by URIs attached to the label `e` in the dictionary. Once the query has been formulated, it can be posed to the SPARQL endpoint. We opt for running a GET query on the SPARQL endpoint and by asking for results in the JSON format. Once results are returned, the back-end formulates them as an NL reply. We attach to each intent a reply template. The back-end completes it with the resolved entities and retrieved results. The complete reply, i.e., the reply that includes the resolved entities, enables the end-users to inspect how the system interpreted the performed question implicitly. For instance, in the CH use case in Fig. 5, the end-user acknowledges that the *painter* word has been interpreted as *author*. It behaves as a step forward in the direction of the explicability of the application back-end logic.

4.3. Discussion of Strengths and Limitations

The proposed approach queries KGs in *real-time* by exploiting up-to-date data and it is entirely *KG-independent*. Fig. 5 makes evident components that must be reconfigured based on the KG of interest and which components can be left unchanged. It is also a *general-purpose* approach and it can be easily adapted to domain-specific applications (see Section 6). Although, the performance of the implemented approach highly depends on the queried KG. More in detail, the quality of the replies is up to the label coverage; the execution time is up to the endpoint settings; the completeness of the reply depends on the endpoint results limit (if any); the lack of control in accessed URI is due to the label ambiguity.

As a general process, utterances make no assumption on question interpretation and the application context. The covered SPARQL patterns contain at most three triples. We aim to extend the supported SPARQL patterns by implementing more complex queries. In particular, we are reasoning on iterative queries by consecutive query refinements conversation-based. It enables end-users to iteratively refine their questions, for instance, by applying filters consecutively.

The proposed approach is general enough to be exploited both in querying a single KG and in querying multiple KGs by aggregating query results in the reply formulation step, which means improving the back-end implementation without modifying the general approach. Although at the moment the generator can be configured to query a single KG a time, we aim to further investigate how to query multiple KGs.

5. Automatic Virtual Assistant Extensions Generator

This section overviews the architecture and implementation of the proposed software framework to automatically generate VA extensions implementing KGQA by requiring little/no-technical competences in programming and query languages. The proposed community shared software framework is implemented in Python by guaranteeing modularity and extendibility. Our framework provides users with the opportunity to customize VA extension capabilities and generate ready-to-use VA extensions. Each phase is kept separate by satisfying the modularity requirement, and it is implemented as an abstract module. The proposed generator architecture is reported in Fig. 6.

The generator takes as input a configuration file containing the VA extension customization process, as detailed in the following. The configuration file is parsed to verify the syntactical correctness, the semantic validity and, if all the checks pass, both the interaction model and the back-end implementation can be generated. The syntactical correctness is verified in terms of JSON valid format in the actual implementation, but it can be substituted according to the configuration file format. The semantic validation is in charge of spotting any configuration conflict and verifying consistency. Both the validations are performed by parsing the configuration file. Once passed these validations, the interaction model is created by extrapolating from a separated mapping file (stored in the back-end implementation as a JSON file) from each intent required by the configuration file, the corresponding set of utterances in the language configured by the end-user. It guarantees the ease in extending new supported languages, the possibility to revise utterances for each intent, and model new intents. The back-end is implemented in Node.js and maps to each intent the corresponding behavior. It is configured according to the required user language and the SPARQL endpoint of interest. The back-end is returned as a ZIP file containing both the Node.js webhook and the implementation of the linking approach. Further details follow.

VA generator input: the configuration file. The `VA Generator` module takes as input a configuration file containing the VA extension customization options: the `invocation name`, i.e., the VA extension wake-up word; the list of desired `intents`, according to supported intents listed in Table 7; the `SPARQL endpoint` the user aims to query; the `lang`, by choosing among *en* and *it* at the moment, even though further languages can be easily introduced. Moreover, users can specify a (incomplete) dictionary of entities and properties mapping URIs to labels.

Users can manually create the configuration file. Otherwise, they can exploit the `Configuration Generator` module that takes as input the URL of the SPARQL endpoint of interest and automatically retrieves both classes and properties labels and URIs. It looks for used classes/properties and the ones defined according to standard approaches, such as classes defined as `owl:Class` or `rdfs:Classes`, properties defined as `rdf:Property`. Moreover, it also expands labels with synonyms and variations by exploiting Wordnet, e.g., nouns used as properties are expanded by their verbal or adjective forms. The configuration file is returned as output, and it can be directly used to start the VA extension generation process. Users can manually check the auto-generated configuration file before generating the VA extension to revise supported resources.

Workflow & Output. Once provided the VA Generator module with the configuration file, it can start the generation workflow, i.e., i) it checks the syntactical correctness of the configuration file by the `Syntax checker`; ii) validates the semantic correctness of the configuration by the `Validator`; iii) creates the `interaction_model.json` by the `Interaction Model Generator` containing configured intents, its utterances and the slot values according to the configuration file; iv) generates the back-end code by the `Back-end generator` and it produces the `back-end` (as a ZIP file) containing the back-end logic implementation. While the syntax checker and the validator strictly depend on the configuration file, the interaction model and the back-end generator depend on the VA provider API. As we require a JSON configuration file, the `JSON Syntax Checker` has to verify that the file is a valid JSON file, while the `Validator` checks if all the mandatory fields are defined, and the configuration is consistent. If any error occurs, the generator immediately stops and returns a message reporting the occurred error. If the configuration is properly defined, the generator returns a folder entitled as the VA extension wake-up word containing the `interaction model` as JSON file and the `back-end` Node.js code as a ZIP file. It is worth noting that the generated VA extension is ready to be used, i.e., it can automatically be

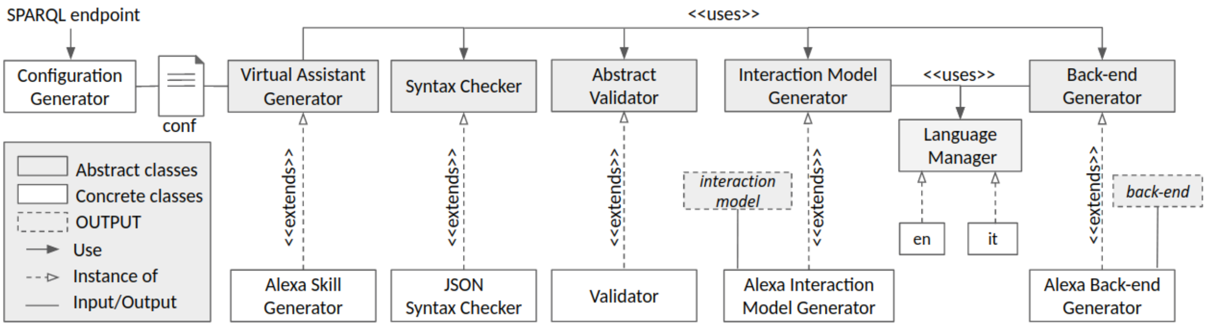


Fig. 6. Architecture of the proposed generator of Question Answering over Knowledge Graphs by Virtual Assistants.

uploaded on Amazon developer[2] and Amazon AWS[2], respectively. The generated code corresponds to manually created VA extensions but may reduce required technical competencies and development time.

Extension points. The version of the generator presented in this article ($v1.0$) supports the Amazon Alexa provider. Thus, once validated the configuration file, the Alexa skills components (the JSON interaction model and the ZIP file implementing the VA extension back-end that can be upload on Amazon AWS) can be created. Thanks to the architecture modularity, it is easy to develop new VA providers' support by focusing on the `Back-end generator` implementation. As an example, Google provides a vocal assistant named Google Assistant, and it can be enriched by programming functionalities named actions. As in Alexa, the interaction model is a JSON file containing intents, its example phrases (corresponding to utterances in Alexa) and parameters (corresponding to slots in Alexa). The intent back-end is named fulfillment and it is implemented by cloud-based webhooks, mainly in Java or Node.js. The integration of Google Assistant only requires the definition of the interaction model that is compliant with Google requirements, while the back-end used in the Alexa skill (already implemented in Node.js) can be almost reused also for Google actions. The extension point is guaranteed by the exploitation of abstract classes and the modular implementation that keeps general behaviour detached from the actual implementation. It implies that there is the possibility to integrate any VA provider properly modelling interaction model and back-end, without modifying the remaining functionalities, such as the linking mechanism, the configuration initialization mechanisms, the language manager.

Concerning the linking phase, it is performed in a dedicated function (as reported in the documentation) to enable end-users (with competences in programming and KG querying) to customize it, e.g., it by calling APIs (as we point out in Section 6). By default, the back-end exploits the (partial) dictionary to perform the linking step. If the slot value is resolved as a list of URIs by the dictionary lookup, it will exploit them during the SPARQL query formulation. Otherwise, the user value is used as-is in the SPARQL query formulation by comparing it with resource labels.

Moreover, developers may add new supported languages by translating utterances in the target language and extend the reply formulation mechanism to return replies in the desired language. At the moment, English and Italian are supported.

To add a new pattern, developers have to model the new intent as a set of utterances (by solving any arising conflict) and extend the back-end logic to formulate the related SPARQL query and the reply.

6. Use cases

This section overviews benefits and challenges in querying KGs by VAs by presenting a pool of Alexa skills for CH KGs. It is worth noting that it proposes use cases of the generator to demonstrate how a data curator might configure and use the generator to obtain a ready-to-use Alexa skill. Thus, we overview the generator configuration options, and we show the VA extension in action to make evident how the generator might be either used or configured to obtain VA-extensions and to simulate all the supported patterns in practice. The VA extensions back-end and its interaction model are freely available on GitHub[1]. More-

[2] Links for Alexa skill deployment: developer.amazon.com and aws.amazon.com

over, the reported use cases underline the impact of data sources on the generated VA extensions. As an example, the consequences of missing labels attached to resources. While this section provides data curators with guided examples to use the proposed generator, Section 7 reports scenarios foreseen by CH experts and lovers in adopting VA extensions in CH tasks.

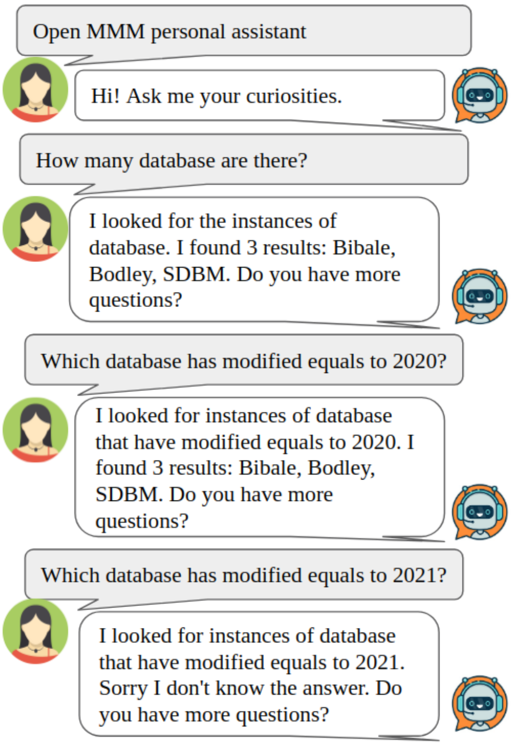
We propose a use case for each category of the CH taxonomy. In particular, for the tangible category, we propose the MMM use case for the movable sub-category, and the Hungarian museum use case for the immovable one; DBTune for the intangible category; NaturalFeatures for the natural heritage category; the UNESCO thesaurus for the terminology category.

6.1. Tangible Movable category: MMM

Mapping Manuscript Migrations (MMM) [41] is a semantic portal for finding and studying pre-modern manuscripts and their movements, based on linked collections of the Schoenberg Institute for Manuscript Studies, the Bodleian Libraries, and the Institute for Research and History of Texts. In particular, it models physical manuscript objects, the intellectual content of manuscripts, events, places, and people and institutions (referred to as actors) related to manuscripts.

Configuration. We automatically configured the MMM Alexa skill by exploiting the generator configuration component. The returned configuration file is directly used to initialize the generator.

VA extension in action. Fig. 7 reports a simulation of the interaction between humans and the MMM personal assistant. We ask for databases aggregated by the MMM portal by posing the *How many databases are there?* question. Used resources are i) Bibale (which stands for Bib[liothèque médiév]ale), a long-term project of the Codicological Section of the IRHT (The Institute for Research and History of Texts) in Paris; ii) Bodley, i.e., Medieval Manuscripts in Oxford Libraries, and iii) SDBM, i.e., Schoenberg Database of Manuscripts. The user request *How many databases are there?* match an utterance attached to the `getClassInstances` intent, which returns the instances of a given class (database in this case). To verify the timeliness of retrieved information, we ask *Which database has modified equals to 2020?* which corresponds to an utterance matching the `getNumericFilterByClass` intent that verifies which instance of a given class (database in our use case) has a property (modified in our case) matching a given numerical value (2020 in our case). It replies to the CH community need to verify the queried sources and the timeliness of the retrieved information.



Open MMM personal assistant

Hi! Ask me your curiosities.

How many database are there?

I looked for the instances of database. I found 3 results: Bibale, Bodley, SDBM. Do you have more questions?

Which database has modified equals to 2020?

I looked for instances of database that have modified equals to 2020. I found 3 results: Bibale, Bodley, SDBM. Do you have more questions?

Which database has modified equals to 2021?

I looked for instances of database that have modified equals to 2021. Sorry I don't know the answer. Do you have more questions?

Fig. 7. MMM use case for the tangible category related to the movable sub-category.

6.2. Tangible Immovable category: Hungarian museum

The Hungarian Museum [42] provides access to the Museum of Fine Arts Budapest data.

Configuration. We manually configured the Hungarian museum Alexa skill by retrieving `owl:class`, used classes and triples subjects, and the used properties. Labels are rare and are mainly provided in Hungarian, without English translation.

VA extension in action. Fig. 8 reports a simulation of the interaction between humans and the Hungarian museum personal assistant. By querying *what is the creation with the maximum value of had participant* we activated the `getSuperlative` pattern which returns the class instance (the creation in our case) corresponding to the maximum (or minimum) value of a given property (had participant in our use case). This scenario simulates the interest of CH lovers in retrieving information about artworks, paintings, sculptures.

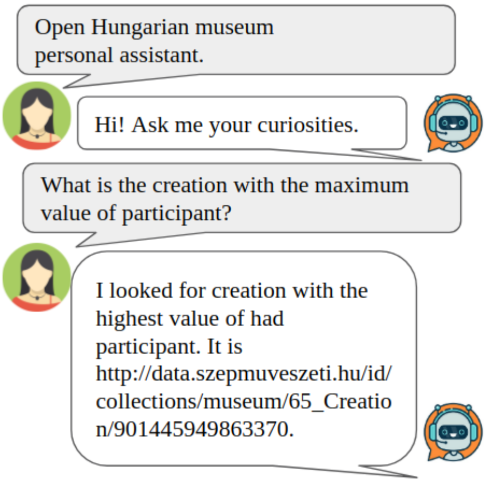

Open Hungarian museum personal assistant.
Hi! Ask me your curiosities.
What is the creation with the maximum value of participant?
I looked for creation with the highest value of had participant. It is http://data.szepmuveszeti.hu/id/collections/museum/65_Creation/901445949863370.

Fig. 8. Hungarian museum use case for the tangible category related to the immovable sub-category.

The VA extension usually refers to resources by labels. In this case, it returns the creation URL (see the reply in fig. 8). It makes evident the consequences of lack of labels attached to resources and the difficulties in exploiting them in VA-based applications.

6.3. Intangible category: DBTune classical

DBTune classical [43] describes concepts and individuals related to the Western Classical Music canon. It includes information about composers, compositions, performers, and influence relationships.

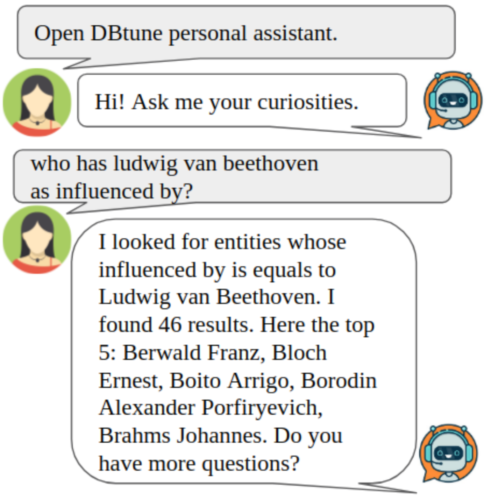

Open DBtune personal assistant.
Hi! Ask me your curiosities.
who has ludwig van beethoven as influenced by?
I looked for entities whose influenced by is equals to Ludwig van Beethoven. I found 46 results. Here the top 5: Berwald Franz, Bloch Ernest, Boito Arrigo, Borodin Alexander Porfiryevich, Brahms Johannes. Do you have more questions?

Fig. 9. DBTune classical use case for the intangible category.

Configuration. We automatically configured the DBTune classical Alexa skill by exploiting the generator configuration component. The returned configuration file is used to initialize the generator after applying basic configuration manipulation, such as identifying which relation can play the role of label predicate (alias [3] is exploited). This use case demonstrates developers' challenges when the KG adopts a non-standard way to attach human-readable labels to resources.

VA extension in action. Fig. 9 reports a simulation of interaction between humans and the DBTune classical personal assistant. *Who has Beethoven as influenced by?* activates the `getPropertySubject` intent which retrieves the subject of triples where *influenced by* is the property and *Beethoven* is the object. This use case addresses the CH community interest in retrieving curiosities about musicians and artists.

6.4. Natural Heritage category: Natural Features

It is part of Scotland's official statistics [44] that provides a range of statistical and geographic data about Scotland from various organisations. In particular, we are interested in aspects concerning geodiversity, ecology, and biodiversity.

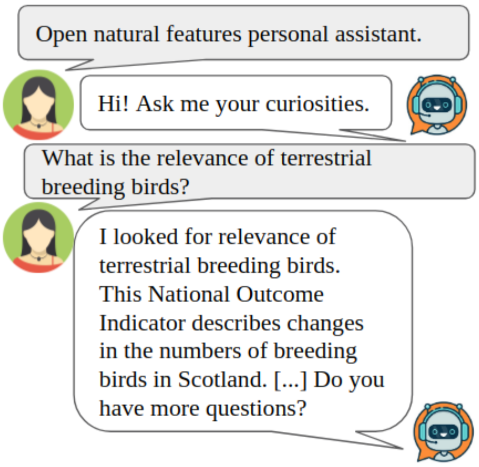

Open natural features personal assistant.
Hi! Ask me your curiosities.
What is the relevance of terrestrial breeding birds?
I looked for relevance of terrestrial breeding birds. This National Outcome Indicator describes changes in the numbers of breeding birds in Scotland. [...] Do you have more questions?

Fig. 10. Natural feature use case for the natural heritage category.

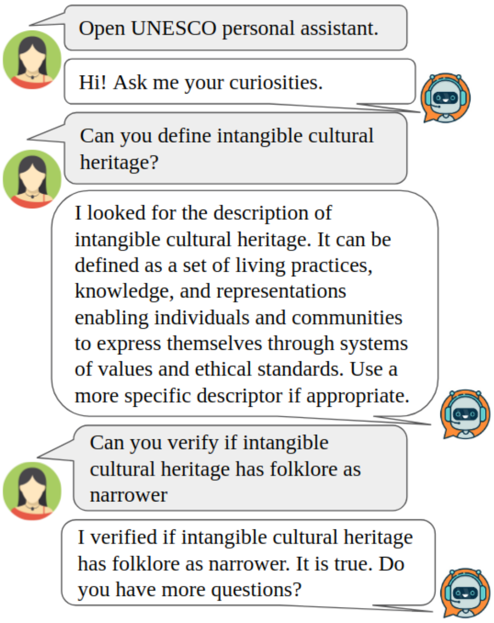
Configuration. We automatically configured the Natural feature Alexa skill by exploiting the generator configuration component. The returned configuration file is directly used to initialize the generator.

[3] http://dbtune.org/musicbrainz/resource/vocab/alias

VA extension in action. Fig. 10 reports a simulation of interaction between humans and the Natural feature personal assistant. *What is the relevance of terrestrial breeding birds?* activates `getPropertyObject` intent which returns the value playing the object role in triples related to *terrestrial breeding birds* as subject and *relevance* as predicate. CH lovers and experts joining the user survey on the impact and potentialities of the proposed approach in the CH domain stress that VA extensions might be useful for educational scenarios. In fact, this use case simulates the possibility to deeper domain-specific information for familiarizing with terminology or conduct researches.

6.5. Terminology category: The UNESCO Thesaurus

The UNESCO Thesaurus [45] is a controlled and structured list of terms used in subject analysis and retrieval of documents and publications in education, culture, natural sciences, social and human sciences, communication, and information. Continuously enriched and updated, its multidisciplinary terminology reflects the evolution of UNESCO programs and activities. Like a thesaurus, it mainly provides access to synonyms and related concepts. It also partially behaves like a dictionary by providing term definitions.



Open UNESCO personal assistant.

Hi! Ask me your curiosities.

Can you define intangible cultural heritage?

I looked for the description of intangible cultural heritage. It can be defined as a set of living practices, knowledge, and representations enabling individuals and communities to express themselves through systems of values and ethical standards. Use a more specific descriptor if appropriate.

Can you verify if intangible cultural heritage has folklore as narrower

I verified if intangible cultural heritage has folklore as narrower. It is true. Do you have more questions?

Fig. 11. UNESCO use case for the thesaurus category.

Configuration. We manually configured the UNESCO Alexa skill by retrieving $(4,421)$ `skos:Concept` that defines all the thesaurus terms and the used properties. All the concepts are attached to a human-readable label (by `skos:prefLabel`), while we generate property labels by local names of URIs.

VA extension in action. Fig. 11 reports a simulation of the interaction between humans and the UNESCO personal assistant. We can ask for the term definitions, e.g., *what is intangible cultural heritage?* (see Fig. 11). It activates the `getDescription` intent, i.e., a special case of `getPropertyObject` where the property is bound to a relation modelling term description. The VA extension retrieves the description (configured as skos:scopeNote) attached to intangible CH, and it returns the term definition. We can also pose ask queries. By querying *Can you verify if intangible cultural heritage as folklore as narrower* we activate the `getTripleVerification` pattern, which model ask queries that verify if the stated triple is modelled in the KG. It replies to the interest of the CH community to clarify and use domain-specific terms properly.

6.6. Discussion

By the overviewed use cases, we demonstrate most of the intents reported in tab. 7 in real settings. We verify that the proposed approach is general enough to query data concerning different categories of CH, from museums to manuscripts, from music to term definition. Moreover, we also experienced some issues related to some aspects already pointed out in the CH KG analysis (Section 3) and challenges described in Section 4. In the following, we summarize KG properties that affect VA-based KG exploitaiton.

Label coverage. To cope with the scarce provision of human-readable labels, they can be generated by local names of URIs, as we performed in UNESCO Thesaurus. This practice can be performed if resources have human-readable URLs. As evidenced in the Hungarian museum use case, the lack of label provision is an obstacle to resource understanding.

Multilingualism. Some KGs, such as Finland datasets, Hungarian museum, Cultura, only provide access to labels in the data provider native language without enriching resources with English translations. Lack of multilingualism prevents a wider data exploitation.

SPARQL support. A technical detail must be remarked. Before implementing the intents to SPARQL queries mapping, developers have to carefully check if the queried endpoint fully supports SPARQL or omits some patterns. For instance, to use alternative predicates, we exploited the `VALUES` pattern. It is not supported by some of the queried KGs, such as Munnin and CULTURA. It affects the back-end implementation or limits the endpoints that can be exploited by any KG exploitation mean. Moreover, there are endpoints, such as CIDOC-CRM and AAT, that do not support the `COUNT` aggregator. It affects queries as simple as *How many artifacts are hosted in the Uffizi museum*.

7. Evaluation

This section assesses the quality of the generated VA extensions and test to what extent configuration options affect the returned VA extensions. All the presented VA extensions and the discussed results are online available on the project GitHub repository[1]. It also tests the user experience of a group of CH experts in using an auto-generated Alexa skill and collects the impact and utility according to the CH community in making CH KGs interoperable with VAs.

7.1. Performance of the proposed mechanism

This evaluation tests the accuracy and the precision of the auto-generated VA extension as an approach to verify to what extent the configuration affects the proposed evaluation. It demonstrates that the generation of a VA extension in a single click already returns VA extensions that are as accurate as systems proposed in the literature evaluated on the same benchmark. Moreover, it also demonstrates that by tuning the generator configuration, end-users can significantly improve the accuracy and precision of the auto-generated VA extension. It is relevant to assess the performance of the auto-generated VA extensions as a special case of KGQA over VA compared with systems categorized as traditional KGQA.

7.1.1. Evaluation design

Methodology. The following questions (Qs) guides our evaluation process:
Q1 - Are the results achieved by the auto-configured VA extensions comparable with other KGQA systems in terms of precision, recall and F-score?
Q2 - To what extent the manual configuration refinement affect results?
Q3 - Which linking approach between the dictionary lookup and API-based approach achieve the best results?

While Q1 compares the proposed approach with alternative KGQA approaches, Q2 and Q3 have been formulated and evaluated to overcome any skepticism by end-users in terms of the impact the generator configuration on the generated VA extensions has in terms of performance. Thus, they analyze to what extent the linking approach and the lookup mechanism affect the performance of auto-generated VA extensions.

Dataset & Baselines. We rely on a standard benchmark on KGQA systems, QALD[4], as it contains benchmarks for multiple well-established KGs (i.e., DBpedia and Wikidata), and it tests both simple and complex questions. We prefer to evaluate the VA extensions created by the proposed generator on a standard benchmark for KGQA, first, to avoid overfitting in a specific context. Second, it easily enables comparison with systems that joined the same challenge. Finally, it implicitly behaves as a comparison between a VA-based and traditional KGQA approaches. We consider the QA system joining the challenge as baselines. We refer to official results published in the QALD report. While for DBpedia, we rely on QALD-9 [46], for Wikidata, we have to consider QALD-7. As systems joining the QALD-7 challenge relied on a different version of Wikidata, we report results achieved by the Wikidata Alexa skill generated by the proposed software framework and the updated version of the QALD-7 dataset to enable further comparisons.

Settings. We generate the DBpedia and Wikidata Alexa skills by the proposed software framework. The generated VA extensions are different in configuration options (manual VS auto) and linking approach (dictionary VS APIs). Further details follow.

Manual Configured DBpedia Alexa skill. The manual configuration option requires end-users to perform standard queries on the SPARQL endpoint of interest to retrieve all the classes, properties, and resources and to organize them in the JSON format, as described in Section 5. As Alexa requires the specification of custom slot values in the interaction model and poses a constraint on the interaction model size (1.5MB), developers have to query a sub-graph of the KG of interest. In the sub-graph retrieval, we focus on heterogeneous macro-areas. In particular, the entities dictionary contains all the declared classes (750) and 28.5K resources, distributed as follows: 5K people; 5K cities countries, and continents, 2K rivers and mountains related to the geography field; 3K films, 2.5K musical

[4] QALD challenge: http://qald.aksw.org/

works, and 3K books belonging to the entertainment category; 4K museums and monuments and 1.5K artworks belonging to the art field; 2.5K animals and celestial bodies, related to the scientific field. The property dictionary contains all the declared properties (5K). We take the first results returned by the DBpedia SPARQL endpoint without either applying any sorting option or checking the returned results' relevance. Then, we perform basic cleansing operations, such as lower-casing labels and removing codes as labels to avoid readability issues. Finally, we automatically generate the resulting Alexa skill.

Auto-configured DBpedia Alexa skill. Users can opt for the auto-configuration by specifying the URL of the endpoint of interest and the `Configuration generator` automatically creates the configuration file as described in Section 5. The configuration file contains DBpedia classes and property, while it lets end-users freely refer to resources, and the queried labels will be compared against KG resource labels during the query formulation step. Users can either accept the generated file or manually clean the configuration file before generating the VA extension. It behaves as a check-point to reduce the human effort and enable end-users to control the VA extension generation process. The configuration file initializes the generator.

Dictionary-based WikiSkill. We query a sub-graph of Wikidata. It results in a dictionary composed of 2K classes and 28.5K resources, obtained following the same topic distribution described for the manual configured DBpedia Alexa skill. The property dictionary contains all used properties (6.5K). We lowercase all the labels, and remove the ones containing unreadable codes. We add synonyms to entities and properties by retrieving the Wikidata `also known as` property. We generate a VA extension, and we use as-is without applying any further modification.

API-driven WikiSkill. The generator back-end provides the opportunity to modify the linking approach by affecting a dedicated script to customize back-end logic functions. We rely on wikibase-sdk[5], a library to make read queries to a Wikibase instance (e.g., Wikidata). `searchEntities` enables the opportunity to perform entity (and property) linking by resolving labels given as input. We create the `API-driven WikiSkill` by i) modifying the linking method from the dictionary lookup to the invocation of the `searchEntities` function and ii) the SPARQL query execution with `sparqlQuery` function in the `Dictionary-based WikiSkill` back-end.

Procedure. We perform the evaluation by retracing the following steps. Given the QALD (QALD-7 for Wikidata and QALD-9 for DBpedia) question set,
1 - for each question, we manually check if the pose question matches one of the supported intents (according to table 7), or it can be transformed into a chain of supported intents. For instance, the question "What is the time zone of Salt Lake City?" in QALD-9 on DBpedia matches the getPropertyObject intent ("What is the p of e?") where <time zone> plays the role of p and <Salt Lake City> plays the role of e. The question "What is the name of the university where Obama's wife studied?" in QALD-9 on DBpedia can be transformed into a chain of supported intents where first users can ask for "Who is the wife of Obama?" (corresponding to the getPropertyObject intent where wife is the predicate and Obama is the entity) and, then, "What is the school of Michelle Obama?" (corresponding to the getPropertyObject intent where school is the predicate and Michelle Obama is the entity). If not, we skip the question. Otherwise, we continue the procedure.
2 - we check the activated intent, and we formulate the query according to one of the supported utterances.
3 - we query the VA extension by the adapted question;
4 - all the replies returned by our VA extension (including empty results) are stored in a JSON file.
5 - We exploit the official system used to evaluate the QA systems joining the QALD challenge, GERBIL [47], to perform the result assessment.

For Wikidata and QALD-7, the previous procedure requires updating replies in the testing set to compile the current Wikidata version (July 2020). We use the updated version of the QALD-7 training dataset[6] and we share it online to encourage further comparison.

Metrics. We follow the standard evaluation metrics for the end-to-end KGQA task, i.e., we report precision (P) and recall (R) and F-measure (F1) at a micro and macro level.

[5] Wikibase-sdk: https://www.npmjs.com/package/wikibase-sdk

[6] QALD-7 training set updated to July 2020 Wikidata status `qald-7-train-en-wikidata-July2020Version.json`

7.1.2. Results

Configuration options, the DBpedia case. We compare results achieved by (manual and auto-configured) DBpedia Alexa skills[7]. GERBIL returns precision, recall and F-measure at micro and macro level, as reported in the metrics paragraph. Table 8 reports the comparison of results achieved by KGQA systems joining the QALD-9 challenge as presented by the challenge report and the DBpedia Alexa skill results computed by GERBIL. Regardless of the considered configuration approach, we achieve the best results in all the metrics, and we obtain results from 2 to 6 times better than the second-best result obtained by the participants in the challenge (Q1). As the auto-generated VA extensions achieve better results than other systems joining the QALD challenge, it means that the proposed generator creates VA extensions that are competitive with systems proposed in the literature. As systems joining the challenge do not imply the exploitation of a VA technology, this achievement also implies that the proposed approach of KGQA over VA succeeds in achieving results competitive and better than alternative approaches for KGQA. The achieved results are justified by i) the exploitation of structured NL questions and ii) the possibility to tune the VA extension initialization according to specific needs by a fine-grained control. End-users can add data of interest in the configuration file, for instance, resources required by the testing dataset that the previous coarse-grained entity selection has not included. While the manually configured VA extension obtains optimal results due to the user's full control, the auto-configured DBpedia Alexa skill provides lay-users with a good starting point to be used with or without manual refinement (Q2).

Table 8

Manual VS auto-configured DBpedia Alexa skills and systems joined the QALD-9 challenge. Best results are highlighted in bold.

Tool	Micro results			Macro results		
	P	R	F1	P	R	F1
ELON	0.095	0.002	0.003	0.049	0.053	0.050
QASystem	0.039	0.021	0.027	0.097	0.116	0.098
TeBaQA	0.163	0.011	0.020	0.129	0.134	0.130
wdaqua	0.033	0.026	0.029	0.261	0.267	0.250
gAnswer	0.095	0.056	0.070	0.293	0.327	0.298
Auto-c.	**0.991**	0.197	0.328	0.369	0.358	0.354
Manual c.	0.990	**0.284**	**0.441**	**0.683**	**0.677**	**0.678**

Table 9

Dictionary based VS API-driven WikiSkills (WSs) on QALD-7. Best results are highlighted in bold.

	Micro-results			Macro-results		
	P	R	F1	P	R	F1
Dict. WS	**0.989**	**0.946**	**0.967**	**0.736**	**0.747**	**0.739**
API WS	0.954	0.262	0.412	0.664	0.677	0.669

Linking approach, the Wikidata case. The QALD-7 training set contains 100 questions, but 4 questions cannot be more answered. We reply to 76/96 questions, while the remaining 20 questions correspond to not supported patterns. Table 9 reports results of the auto-generated WikiSkills over QALD-7.

Not surprisingly, the dictionary-based linking approach is more precise than the API-driven approach as a dictionary gives the possibility to tune and customize the order and the priority in the URIs list attached to the same entity or predicate. For instance, the term Paris might be attached to the French capital and VIPs whose name is Paris, such as Paris Hilton. If the VA extension is designed to be used as a museum virtual guide, the dictionary based configuration can attach a higher priority to Paris as a city instead of other interpretations. This mechanism cannot be performed in the API-based configuration. Even if the dictionary represents a static snapshot of the KG content, it can be exploited both in the entity and relation linking task. On the contrary, it is required to verify if APIs offer both linking mechanisms. The dictionary based linking approach is also KG-agnostic, i.e., it is independent of any external service. It only requires configuration time and extra storage in the back-end but guarantees direct and immediate (without execution time) access to URIs. Moreover, the dictionary-based solution is general enough to enable the VA extension back-end configuration with any KGs without any constraint.

7.2. Users Experience in a Controlled Environment

This section assesses the usability of an auto-generated Alexa skill according to delegates of HETOR[8], a CH association of the Campania region in Italy, and it behaves as a preliminary usability assessment of the proposed approach in a controlled environment. The

[7] Manual configured and auto-configured DBpedia Alexa skill results, respectively: http://gerbil-qa.aksw.org/gerbil/experiment?id=202012170018 and http://gerbil-qa.aksw.org/gerbil/experiment?id=202012170019

[8] http://www.hetor.it

HETOR project collects and makes available as Open Data the Open Heritage published by the National Institutions, such as ISTAT, MIBACT, MIUR and Campania Region (Italy), and the Open Heritage that can be created by the citizens themselves, concerning their own territories, improving the quality and quantity of Open Data available at a local and national level. HETOR mainly collaborates with schools to study and preserve the historical and collective memory of local CH. In the context of their activities with schools, they organize co-creation sections for encouraging learners to familiarise themselves with CH, collect information about the CH to preserve and model it as tabular data by caring about the correct terminology. It requires to familiarize with terms and their definition, hierarchy of concepts, mastering synonyms and analogies. Thus, during the activities, learners usually ask mentors questions like *What is the meaning of geolocalization?*, *Can you define a point of interest?*, *What do you mean by year of foundation?* In this context, the HETOR group has the real need to address a plethora of requests posed by each group to clarify terminology. The situation is even worse during the COVID-19 as activities were performed online and there was a limited possibility to clarify all the doubts due to the lesson settings and also due to the wider exploitation of asynchronous activities that required learners to do without continuous support from moderators. We proposed to the HETOR group to think about the possibility to use a VA extension generated by the proposed approach configured to query a thesaurus, in particular the UNESCO thesaurus, and test the usability of the VA extension in the first person.

Participants and Setting. 5 delegates of the HETOR project joined the usability evaluation of the UNESCO Alexa skill generated by the proposed generator, corresponding to the one described in Section 6. The evaluation took place remotely due to the COVID-19. As the VA extension has not already been published on the Alexa store, we deployed the VA extension on the Alexa developer console and asked participants to interact with the textual interface. We behaved as moderators, while we asked participants to formulate questions and we collected their thought and reactions.

Protocol. The performed protocol involved:

- an introductory overview of the objective of the user experience evaluation, the queried source by looking at the UNESCO Thesaurus browsing interface[9], the setting of the evaluation, and inspecting the presentation of the VA extension which introduce itself by pronouncing "*Hi!! Welcome to the UNESCO personal assistant! Ask me your curiosities, such as: Can you define digital heritage? What is the narrower of cultural heritage? What is broader of churches? What is related to digitization?*";
- the assignment of a collection of tasks to each participant posing questions such as *The definition of Digitisation*, *The definition of CH*, *The specializations of digital heritage*, *The generalization of digital heritage*, *The terms related to CH*. For each task, participants are encouraged to identify the pattern to pose the related questions and collect replies returned by the Alexa skill.

Data collection. At the end of the evaluation, the moderator asked for the fulfillment of a final questionnaire to evaluate i) users' satisfaction based on a Standard Usability Survey (SUS [48]) and ii) their interest in using and proposing the tool by a Behavioural Intentions (BI) survey. The questions of the BI survey are i) "*I will use this approach in the future*"; ii) "*I will recommend others to use the proposed approach.*" and users can use a 5-point scale to reply. Moreover, the moderator annotates all the comments and observations raised during the evaluation.

Results. The proposed approach achieved a SUS score of 80, close to the higher step, interpreted as a great appreciation of the proposed tool and the propensity to propose it to others. The latter result is verified by the BI survey which achieved a mean score of 4.6.

Besides the tasks explicitly assigned by the moderator, participants started posing queries on their own, asking for *The generalization of mosques and synagogues*, *the definition of amphitheater*, *the generalization of Catalan or Gothic*, *generalization of painting* and *specialization of fine arts*.

Question templates. It resulted naturally to pose questions according to specific templates as it is the traditional approach used to query Alexa and its VA extensions, but it requires training to learn the supported templates. Participants inquired the moderator asking for the other supported patterns, besides the ones tested in the UNESCO Alexa skill, and they were almost satisfied with the covered templates. In particular, they asked for further details on numerical filters, quantita-

[9] http://vocabularies.unesco.org/browser/thesaurus/en/

tive queries, mechanisms to retrieve images, and they assessed that the query to retrieve object proprieties is the easiest and the more natural one. However, they considered tailoring utterances according to the target user a crucial aspect. For instance, educational contexts may require further simplify the used terminology and adopt a wider used way to formulate similar questions. Participants suggested integrating the definition intent with utterances such as *What is the meaning of X?*. We also discussed if a keyword-based search might result in a dirtier but quicker way to retrieve information. Further study in this direction should be performed to verify the expressiveness capability of a keyword-based querying mechanism.

Target age. The proposed mechanism is perceived as a powerful approach above all for young people that are more and more accustomed to query VAs to perform daily tasks. Participants observed that it seems also particular adapt with very young learners, also in the pre-scholar phase, as vocal commands represent the unique approach that they can use as they cannot already write commands. Similarly, this approach might be critical for learners with disabilities that prevent them from typing questions or adopting textual interfaces, that may result too difficult for blind people or people with limited range of motion.

The role plaid by the data source. Queried data sources play a crucial role is the effectiveness of the resulting VA extension. For instance, the UNESCO Thesaurus is too generic for domain-specific questions and CH experts also disagree with some of the reported definitions. As an example, they are surprised by the taxonomy proposed by the UNESCO for the CH concept, as they expect the well-known taxonomy based on tangible, intangible and natural heritage. As data modeling impacts also the VA extension utterances, it is crucial to evaluate the naturalness of the resulting questions.

Application contexts. The HETOR group really appreciated the proposed approach as a way to provide learners with continuous support to master terminology about CH and become familiar with related concepts. Learners are less and less accustomed to consult a dictionary to look for the right terminology. The proposed approach might revive the interest in familiarising and disambiguate terms and enrich the personal vocabulary. As in the described activities, this proposal has interesting implications for groups of works to retrieve thematic information and images.

Discussing about alternative application contexts, the proposed mechanism seems to be particularly useful in guided tours, to guarantee personalized interactions, guided by curiosities avoiding boring prepackaged presentations of artworks and points of interest. VA extensions as virtual guides can overcome the lack of interest in the entire exhibition and too detailed descriptions, lack of personalization in terms of tour duration, interests, curiosities. It also solves the linguistic gap between visitors and personnel. It enables the possibility to perform tours to the desired speed with the possibility to repeat unclear passages without bothering others. If it may be an interesting alternative to audio guides already available in museums, it might be revolutionary for city tours to explore monuments or points of interest spread in a city or minor realities, such as small villages.

7.3. Impact and utility according to end-users

This section discusses the perceived impact and utility from the end-user perspective. We proposed an online survey to collect opinions and suggestions. We do not limit to experts in the CH field, but also try to involve CH lovers to take also their opinion into account. Moreover, it is worth noting that we do not limit this survey to experts in the field as we are assessing the perceived impact and utility from the end-users side meaning that we need to collect opinions by inquiring potential users of the resulting VA extensions.

Participants and Setting. 86 people joined the online survey that was administered for one week, from September 15th to September 22nd, 2021. All the participants spontaneously joined the survey in an anonymous form. According to general information collected by the survey, 73 people are (very) interested in CH by rating their interest at least as 4 out of 5 and 24 of them are experts in CH by rating their expertise in CH at least as 4 out of 5. By looking at people that consider themselves as experts in the CH, they have limited expertise in Computer Science, stressing that it is crucial to provide the CH community with tools that do not take for granted their competencies in programming, query languages, and computer science.

Data gathering and Survey Outline. The survey has been administered in English and in Italian and its content is described in Table 10 that reports questions, reply format, and the rationale behind each posed question. The survey is structured in three main sections: i) general information about participants expertise and interest in CH, the spread of VAs within the CH community interpreted as people that are either experts or interested in CH, alternative means used to query

Table 10

Impact and utility survey outline

Question	**Reply format**	**Question role**
	General information	
Your interest in the CH.	1-5	User profiling
Your expertise in the CH.	1-5	User profiling
Your expertise in computer science.	1-5	User profiling
Used Virtual Assistants	None Alexa Google Assistant Others More than one	Spread of VAs within the CH community
Frequency of Virtual Assistant device usage.	Never Rarely - Less than once a week Sometimes - 3 times a week Always - Everyday	Spread of VAs within the CH community.
Have you ever looked for CH information?	Yes/No	Alternative exploitation means
If so, used device and application.	Free text	Alternative exploitation means
	Impact and Utility to query Cultural Heritage by Virtual Assistants	
In which context does the proposed approach may be useful?	Library In museums as virtual guides As learning assistant at school No utility	Perceived utility in terms of application contexts
Application CH context advantaged by VA	Free text	Application context
To what extent VAs can spread the CH?	1-5	Perceived impact
Example of queries you are interested in	Free text	Intent coverage
Are there activities performable only by VAs?	Yes/No/Maybe	Perceived utility
If so, which one?	Free text	Perceived utility
Are there activities improved by VAs?	Yes/No/Maybe	Perceived utility
If so, which one?	Free text	Perceived utility
	Suggestions and Comments	
Any suggestion	Free text	Collection of suggestions
Any comment	Free text	Collection of comments and feedback

and explore CH; ii) questions concerning the perceived utility in terms of application contexts and as an alternative to traditional CH exploitation means, the perceived impact achieved by spreading CH data by VAs while queries users are interested in are both useful to assess the intent coverage and to collect ways users naturally pose questions; iii) finally, general suggestions and comments as a free text.

Results. This paragraph reports most common replies and comments that raised interesting considerations concerning the proposed approach in the field of CH.

Current exploitation means. More than half of the participants that assessed to be interested in CH by rating their interest at least as 4 out of 5 (56%) queried and looked for CH data, at least once, by rarely exploiting websites dedicated to the CH of interest (in 2 out of 43 cases) or bibliographic sources (in 3 out of 43 cases) and by mainly googling it (in 28 out of 43 cases). It is interesting to notice that in 10 out of 43 cases, CH lovers already exploit VAs to fulfil their curiosities.

The diffusion of VAs within CH lovers and experts is also confirmed by results reported in Table 11 that summarises the most used VA providers and the fre-

Table 11

Diffusion of VAs in the CH community

	Used VAs						Usage frequency			
	Tot.	**None**	**Alexa**	**Google Assistant**	**Others**	**More than one**	**Never**	**Rarely**	**Sometimes**	**Always**
Interested in CH	73	18	18	20	8	9	18	29	20	6
CH experts	24	6	6	6	4	2	6	8	6	4

quency of their usage. Most of the participants have their favorite VA and usually get stick to it without experiencing multiple providers. If a single provider is chosen, Google Assistant appears to be the preferred one. VA usage is still limited to few days a week meaning that there are still barriers in the wide exploitation of VAs in this community and it requires overcoming skepticism, perhaps, leveraging on curiosity connected to novelty or by demonstrating to the potential users about utility and potentialities.

If compared with traditional text-based interfaces, by asking for activities that can be performed **only** with VAs, participants recall the potentialities to use VAs with users with disabilities, such as blindness, or situations that impede the usage of a keyword to type questions. It seems to be particularly useful at school during teamwork. By asking for activities that can benefit from the usage of VAs, users underlined the advantage to pose questions rapidly, interactive consultation of sources, simplify lookup operations.

Application contexts. Considering the entire set of replies, independently from participants' interest and expertise in CH, just in one case, a participant cannot see the potentialities of the proposed approach, while all the other ones selected at least a CH application context that might take advantage of VAs. We first asked participants to choose among a set of options, i.e., in libraries to help to look up books, as a virtual guide in museums, and as a learning assistant at school. Fig. 12 reports participants' opinion whose seem to be convinced that our proposal is a promising approach as a virtual guide in museums.

Furthermore, we also asked users to think about any other application context that can take advantage of VAs. In 39 out of 67 cases, users see the potentialities to adopt the proposed approach as a virtual guide not only in a museum but to guide visitors while wandering in an unknown city, above all while visiting small villages, unconventional destinations, or cities with low population density and high cultural impact, dense of archaeological parks or churches. An interesting consideration has been proposed by more than one participant that assessed that the proposed approach is particularly useful when there is no possibility to type requests, for instance, while driving. Moreover, a user also suggested thinking about the exploitation of VAs in a virtual museum by simulating a real tour also in terms of tour guide. In 14 out of 67 cases, users state that it might result in a promising individual learning tool at university to learn about terminology and clarify doubts while preparing exams or scientific contributions, at school to disambiguate terms, at home to deeper knowledge and awareness about CH, for young learners to overcome limits posed by textual interfaces and to leveraging on their curiosity. Moreover, further considerations concern the inclusiveness of the proposed approach, able to overcome disabilities related to limited usability of textual interfaces or blindness. Users also proposed VAs as support in superintendence offices, in archives to guide the lookup phase, as support in offices, and (surprisingly!) in hospitals.

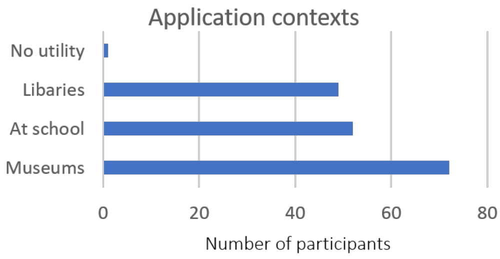


Fig. 12. Application contexts rated by survey participants.

Utility and impact. We explicitly asked users to assess using a 5-Likert scale the perceived impact of using VAs as a means to spread the culture, interest, and awareness about CH. Fig. 13 graphically represents the resulting perceived impact demonstrating that most of the people think that VAs have the potentiality to wider spread the interest about CH, possibly by leveraging of curiosity of the novelty or providing and immediate access to data of interest.

Analysis topics users are generally interested in addressing by the posed questions, they inquire about point of interest's curiosities and details, such as

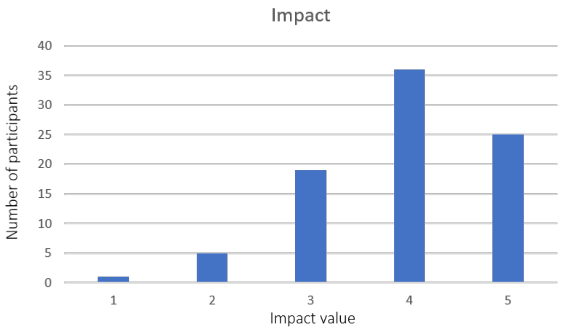


Fig. 13. Perceived impact of the capability of VAs to spread the culture of CH.

“*Which is the history of monument x?*”, “*Which museums are located in x?*”, “*When x happened?*”, “*What is the aim of x?*”, “*Who is the author of x?*”, “*What is the historical context of x?*”, “*Which artists authored any painting hosted in the museum x?*”, “*Where is x?*”, “*Which is the architectural style of x?*”

Participants expect to use the proposed approach to plan a trip. Thus, besides collecting cultural information, users are also interested in collecting practical information about points of interest, such as the accessibility posing questions like “*Which is the ticket for visiting x?*”, “*Does x support an all-inclusive ticket?*”,“*What is the opening time of x?*”, “*How do visitors rate place x?*”, “*What can I visit in city x?*”, “*Which are the most important artworks hosted in x?*”, “*Where can I visit x?*”, “*Are there events in x?*”, ‘*Which is the most important point of interest in x?*”, “*Which are the most famous x?*”, “*Where is the oldest surviving monument of the world?*”.

Some of the participants simulate a conversation with a thesaurus, by clarifying terms and terminology. Moreover, learners can be interested either in looking for details about the queried sources to assess their reliability or retrieving the list of sources containing information about a given topic. As an alternative, according to the school level and subject, learners might be interested in specific information, such as “*Who is restoring x?*”, “*Who is curator of x?*”, “*Is x curated by UNESCO?*”, “*How was called x before date d?*”, ‘*How tall is x?*”. Learners might be interested in the story of the past, such as “*Did Romans took baths at the sea?*”, “*Who was x?*”, “*What is authored/discovered/invented by x?*” and vice versa “*What did x author/discover/invent?*”, “*In which occasion x has been build?*”, ‘*Why x is famous?*”, “*Which artist influenced x?*”, “*Which is the art movement of x?*”, “*Which are the most important artworks authored in x?*”, “*What characterise x?*”, “*How many artworks have been authored by x?*”.

Many participants treat the proposed mechanism as an approach to fulfill general curiosities, such as “*Is there any legend behind x?*”. An interesting aspect that emerged by the collected replies is that VAs interpreted as vocal assistants can easily perform storytelling and can be queried to tell a random event or curiosity about an artist or a monument, can narrate “*How was city x before event x happened?*”, “*Tell me the story of x*”, “*Can you describe x?*”, “*Talk about x.*”, “*Give me further details about x*”, “*Tell me curiosities about x.*”

Even if the most common questions concerns tangible CH, both movable, such as artworks, coins, and documents, and immovable, such as churches, monuments, and castles, users are also interested in curiosities related to intangible CH, such as the folklore of local traditions, such as “*What are the traditional dances of x?*”, music, such as “*Which musician has the most albums?*”, events, co-occurrences of terms in books and manuscripts, such as “*Does x discuss about x?*”, “*How many times x talk about y?*”.

While in most of the cases participants posed punctual questions, some of them used their imaginary VA extension to explore available data. For instance, “*Give me all the titles/authors belonging to the topic/category x.*”, “*Give me artworks related to x*”.

Besides textual replies, users are also interested in visualizing photos and videos, such as “*Show me x*” to retrieve examples of presbyters or a building plan, or “*Show me video related to x*” where x might be an event or a monument.

In most of the cases, participants formulated complete questions, while rarely they posed commands to collect information. Looking at the way questions have been formulated, the proposed intents successfully reply to most of them. The proposed approach misses the fulfillment of complex queries, which are quite rare in this questionnaire. For instance, currently, it cannot deal with questions like “*How long did it take the building of x?*” if it implies computing the difference between the foundation and the completion date, composed questions like “*Give me artworks painted by the same author of x*” requires users to split them into two queries as demonstrated in the performance evaluation.

General comments. As general suggestions, participants noticed that even if our proposal is particularly suitable for the CH field, its usage might be hypothesis in any application context, such as Public Administrations and hospitals. It is crucial to care about queried sources in terms of coverage of topics and reliability. As the interaction model is strictly connected to the target user group, the usability of the resulting VA extension should be carefully checked to tune the way questions can be posed and to make the interaction as natural as possible and limit the error rate. It might be interesting either to introduce a playful component or to evaluate the combination of VAs and virtual exhibitions. Participants also suggested to automatically merge information from multiple sources letting users save time in querying individual sources.

Participants mainly used the comment question to compliments about the project, assessing that *the project has enormous potential*, *it guides digital transitions to our country*, *we are in a modern world and everything is going to be connected with technology, CH should not remain out of this*, *extremely versatile as it might be applied to any application context.*

7.4. *Impact and utility according to CH data curators*

This section discusses the perceived impact and utility from the CH data curators' perspective. We proposed an online survey to collect opinions and suggestions and we administered it to two different groups of CH experts who are either modeling or are planning to model their data as KGs. This survey collects opinions and comments of potential users of the generator who might decide to propose the resulting VA extensions as a data exploitation mean.

Participants and Setting. 5 people joined the online survey belonging to two different groups of CH experts. While 3 of them are delegates of the HETOR project, the other 2 researchers belong to a research group of Medieval Philosophy of the University of Salerno. The HETOR group mainly models tangible CH concerning local and national CH as tabular data releasing them according to the Open Data directive. They are planning to expose their data as KGs in the future. On the other side, the research group of philosophers is designing an ontology to model their collection of medieval manuscripts by representing co-occurrence of terms, philosophical concept interpretation, philosophical movements, both concerning Greek and Latin culture. The two research projects spontaneously joined the survey. They are composed of people interested and experts of CH.

Data gathering and survey outline. The survey has been administered in English and in Italian and its content is described in Table 12 that reports questions, reply format, and the rationale behind each posed question. The survey is structured in three main sections: i) general information about participants expertise and interest in CH, the spread of VAs within the CH community interpreted as people that are either experts or interested in CH, alternative means used to query and explore CH; ii) the interest in making their data accessible by VAs; iii) finally, general suggestions and comments as a free text.

Results. This paragraph reports the most common replies and comments that raised interesting considerations related to the exploitation of the proposed approach in the field of CH.

Current exploitation means. The HETOR group performs analysis on their data by using query builders and data visualization approaches, mainly via SPOD[10], a social platform for Open Data that offers co-creation rooms to produce Open Data as tabular data, data analysis, and data visualizations means to explore and exploit data. The used mechanism supports users in exploring, visualizing, and interpreting data, but requires expertise in data analysis and take time to have a fast insight on available data. They feel that a VA extension might be a powerful approach to have an immediate insight of data, without limits on the dataset size and without requiring specific competencies.

The group of Medieval Philosophy are modeling history of philosophy data and related metadata by ontologies and plan to materialize the related KGs in the next future and to make them accessible to all. Even if planned exploitation tools are still under investigation, they hypothesize to exploit data in data visualization approaches to guide users in interpreting data. They see potentialities to make them accessible by VAs, reacting with curiosity and enthusiasm, but mainly focusing on actual data to improve their accessibility. They are a bit skeptical about making metadata accessible by VAs as they cannot already foresee an application context of interest as only experts are usually interested in metadata, according to their opinion.

It is worth noting that in both groups, participants stated that their working groups have no competencies in querying languages, such as SPARQL. Thus, it is crucial to provide this community with CH data exploitation tools not requiring technical competencies.

[10] http://spod.routetopa.eu

Table 12

Interest in making CH data accessible by VAs survey outline.

Question	**Reply format**	**Question role**
General information		
Your interest in the CH.	1-5	User profiling
Your expertise in the CH.	1-5	User profiling
Your expertise in computer science.	1-5	User profiling
Used Virtual Assistants	None Alexa Google Assistant Others More than one	Spread of VAs within the CH community
Frequency of Virtual Assistant device usage.	Never Rarely - Less than once a week Sometimes - 3 times a week Always - Everyday	Spread of VAs within the CH community.
Used device and application to access CH data.	Free text	Alternative exploitation means
Impact and Utility to make CH data exploitable by Virtual Assistants		
Are you modelling data as KGs?	Yes/No/Maybe	Info about available data
Modelled data	Free text	Info about available data
Expertise in SPARQL in your working group	Yes/No/Maybe	Competences in CH groups
Do you plan to make your data accessible to others?	Yes/No/Maybe	Interest in data exploitation means
Which task do you plan to perform on your data?	Free text	Application context
Impact of VAs to spread CH data	1-5	Impact of VAs
Reaction to the proposed approach	Sceptical Suprised Euphoric Neutral Entusiastic Curious	Perceived impact
Reaction justification	Free text	Reaction to our proposal
Foreseen potentialities	Free text	Reaction to our proposal
Foreseen obstacles	Free text	Reaction to our proposal
Example of queries on your data	Free text	Intent coverage
Would you think about VAs as data exploitation means?	Yes/No/Maybe	Perceived utility
If so, which one?	Free text	Perceived utility
Suggestions and Comments		
Any suggestion	Free text	Collection of suggestions
Any comment	Free text	Collection of comments and feedback

Application contexts. According to data published by the HETOR group, they are interested in retrieving artworks information, such as location, author, date posing questions like "*How many x are in y?*" as a general question to quantify castles in Campania, or museums in Italy, or churches in Naples; "*Which is the construction year of x?*", "*Where is x?*", "*Which are artworks authored by x?*". They also hypothesize to query a VA extension to obtain terms definitions and disambiguation, such as "*What is meaning of x?*". Our proposal is perceived as a promising approach at school to familiarise with terms and concepts also during remote sessions, in museums, or in city tours as virtual guides.

Impact and utility. Participants assess that they would be delighted to query data by pronouncing questions instead of datasheets and query builders. Moreover, they assess that in their opinion the impact of making CH data accessible by VAs might be very high (grade 5 out 5). They reacted with enthusiasm to our proposal and are curious about the consequent applications. They foresee great potentialities given the possibility to spread the interest and the usage to a vast public without constraints on the age and without requiring any technical competence. They only see refrains by people that are still skeptical about the extensive use of technologies, but for sure it might be useful to engage young CH lovers to deepen their awareness and expertise in CH.

General comments. Participants suggested to introduce the possibility to query multiple data sources at a time; tune the interaction model according to the target group and the planned application context; carefully check the used source in terms of accuracy and reliability. They explicitly stated that they foresee potentialities in this project and it would be extremely useful in the CH field, according to their opinion.

8. Conclusions

We propose a general-purpose approach to perform KGQA by VA, and we embed it into a community shared software framework to generate VA extensions by requiring minimum/no programming and query language competencies. Our proposal may have a significant *impact* as it may unlock the Semantic Web technologies potentialities by bringing KGs in everyone "*pocket*". This play on words underlines that the proposed system generates VA extensions that can be also accessed by a smartphone and stress that the presented mechanism offers the opportunity to make KGs queryable by everyone, without asking for any specific technical competence, where "everyone's pocket" is a metaphorical alternative to "everyone means".

Besides its general-purpose nature, we considered it particularly suitable for the CH community for different reasons. First, the CH community heavily invested in publishing data as KGs, as demonstrated by the survey detailed in Section 3. Consequently, we believe that it is useful to provide them with tools and approaches to easily exploit the vast amount of available data. Second, CH lovers are usually provided with tools and interfaces to explore results of data exploitation means, such as virtual exhibitions, data visualization tools, question answering applications, but they are rarely moved to the position of active curator of a mechanism to access data. Thus, the proposed generator moves the CH community in the position of generating their QA tools able to query any data source of interest provided with a working SPARQL endpoint. Thus, librarians can query their book archive, musicians can pose queries on music collections, art gallery curators can provide visitors with a virtual guide able to reply to questions instead of reproducing standard tracks narrating artifacts details. It is the first attempt in the literature to empower lay-users to create personalized and ready-to-be-use VA extensions.

We propose a *reusable* prototype of VA extensions generator to query any KG. In its actual open-source release (v1.0), we allow the building of Alexa extensions, and we aim to provide support for the Google Assistant. It is important noting that we followed all the best practices in software *design* (e.g., abstraction and modularity) to guarantee *technical quality* and make the generator fully extensible.

The proposed community-shared software framework is *available* on GitHub[1] with an open-source license. The ISISLab research lab of our Department will maintain the code and drive the evolution. We aim to extend the set of supported patterns by formulating iterative queries with consecutive refinements. Moreover, we plan to further evaluate our software framework's usability and user perception in real settings.

References

[1] O.-M. Machidon, A. Tavčar, M. Gams and M. Duguleană, CulturalERICA: A conversational agent improving the exploration of European cultural heritage, *Journal of Cultural Heritage* **41** (2020), pp. 152–165.

[2] D. Agostino, M. Arnaboldi and A. Lampis, Italian state museums during the COVID-19 crisis: from onsite closure to online openness, *Museum Management and Curatorship* **35**(4) (2020), pp. 362–372.

[3] P. Seetharaman, Business models shifts: Impact of Covid-19, *International Journal of Information Management* **54** (2020), pp. 102173.

[4] S. Keesara, A. Jonas and K. Schulman, Covid-19 and Health Care's Digital Revolution, *New England Journal of Medicine* **382**(23) (2020), pp. e82.

[5] A.M. Sullivan, Cultural Heritage & New Media: A Future for the Past, 15 J. Marshall Rev. Intell. Prop. L. 604 (2016), *UIC Review of Intellectual Property Law* **15**(3) (2016), 11.

[6] V. de Boer, J. Wielemaker, J. van Gent, M. Hildebrand, A. Isaac, J. van Ossenbruggen and G. Schreiber, Supporting LD Production for Cultural Heritage Institutes: The Amsterdam Museum Case Study, in: *Semantic Web*, 2012, pp. 733–747.

[7] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semantic Web* **8** (2016), pp. 489–508.

[8] B. Haslhofer and A. Isaac, data. europeana. eu: The europeana linked open data pilot, in: *Int. Conf. on Dublin Core and Metadata Applications*, 2011, pp. 94–104.

[9] E. Hyvönen, Publishing and Using Cultural Heritage Linked Data on the Semantic Web, *Synthesis Lectures on Semantic Web: Theory and Technology* **2** (2012), pp. 1–159.

[10] H. Vargas, C. Buil-Aranda, A. Hogan and C. López, *RDF Explorer: A Visual SPARQL Query Builder*, Springer International Publishing, 2019, pp. 647–663.

[11] P. Bellini, P. Nesi and A. Venturi, Linked open graph: Browsing multiple SPARQL entry points to build your own LOD views, *Journal of Visual Languages & Computing* **25**(6) (2014), pp. 703–716.

[12] D. Damljanovic, M. Agatonovic and H. Cunningham, Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-based Lookup Through the User Interaction, in: *The Semantic Web: Research and Application*, 2010, pp. 106–120.

[13] S. Ferré, SQUALL: The expressiveness of SPARQL 1.1 made available as a controlled natural language, *Data & Knowledge Engineering* **94** (2014), pp. 163–188.

[14] A.-C. Ngonga Ngomo, L. Bühmann, C. Unger, J. Lehmann and D. Gerber, Sorry, i Don't Speak SPARQL: Translating SPARQL Queries into Natural Language, in: *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 977–988.

[15] E. Kaufmann and A. Bernstein, How useful are natural language interfaces to the semantic web for casual end-users?, in: *The Semantic Web*, 2007, pp. 281–294.

[16] P. Cimiano and S. Kopp, Accessing the Web of Data through embodied virtual characters, *Semantic Web* **1** (2010), pp. 83–88.

[17] V.W. Anelli, T.D. Noia, E.D. Sciascio and A. Ragone, Anna: A Virtual Assistant to Interact with Puglia Digital Library, in: *27th Italian Symposium on Advanced Database Systems*, 2019.

[18] S. Cuomo, G. Colecchia, V. Cola and U. Chirico, A virtual assistant in cultural heritage scenarios, *Concurrency and Computation: Practice and Experience* **33**(3) (2021), pp. e5331.

[19] L. Hirschman and R. Gaizauskas, Natural language question answering: the view from here, *natural language engineering* **7**(4) (2001), 275.

[20] A.M.N. Allam and M.H. Haggag, The question answering systems: A survey, *International Journal of Research and Reviews in Information Sciences (IJRRIS)* **2**(3) (2012).

[21] B. Cuteri, K. Reale and F. Ricca, A Logic-Based Question Answering System for Cultural Heritage, in: *Logics in Artificial Intelligence*, 2019, pp. 526–541.

[22] V. Lopez, V. Uren, M. Sabou and E. Motta, Is Question Answering fit for the Semantic Web?: A survey, *Semantic Web* **2** (2011), pp. 125–155.

[23] F. Benamara, Cooperative question answering in restricted domains: the WEBCOOP experiment, in: *Proceedings of the conference on question answering in restricted domains*, 2004, pp. 31–38.

[24] D. Diefenbach, P.H. Migliatti, O. Qawasmeh, V. Lully, K. Singh and P. 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*, 2019, pp. 3507–3510.

[25] D. Sorokin and I. Gurevych, End-to-End Representation Learning for Question Answering with Weak Supervision, in: *Semantic Web Challenges*, 2017, pp. 70–83.

[26] L. Zou, R. Huang, H. Wang, J.X. Yu, W. He and D. Zhao, Natural Language Question Answering over RDF: A Graph Data Driven Approach, in: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, 2014, pp. 313–324.

[27] R.D. Donato, M. Garofalo, D. Malandrino, M.A. Pellegrino, A. Petta and V. Scarano, QueDI: From Knowledge Graph Querying to Data Visualization, in: *Semantic Systems. In the Era of Knowledge Graphs - 16th International Conference on Semantic Systems, SEMANTiCS*, 2020, pp. 70–86.

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

[29] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber and P. Cimiano, Template-based question answering over RDF data, in: *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 639–648.

[30] P. Haase, A. Nikolov, J. Trame, A. Kozlov and D.M. Herzig, Alexa, Ask Wikidata! Voice Interaction with Knowledge Graphs using Amazon Alexa, in: *International Semantic Web Conference*, 2017.

[31] J. Krishnan, P. Coronado and T. Reed, SEVA: A Systems Engineer's Virtual Assistant., in: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*, 2019.

[32] M. Lombardi, F. Pascale and D. Santaniello, An application for Cultural Heritage using a Chatbot, in: *2019 2nd International Conference on Computer Applications Information Security (ICCAIS)*, 2019, pp. 1–5.

[33] G. Pilato, G. Vassallo, A. Augello, M. Vasile and S. Gaglio, Expert Chat-Bots for Cultural Heritage, in: *IX Convegno della Associazione Italiana Intelligenza Artificiale Proc. of. Workshop Interazione e Comunicazione Visuale nei Beni Culturali*, 2004, p. 15.

[34] J.P. McCrae, LOD Cloud, 2007, Last access March 2020. https://lod-cloud.net/.

[35] Open Knowledge International, The free data management platform to publish and register datasets, 2006, last access March, 2021. https://datahub.io/.

[36] M. Doerr, *Ontologies for Cultural Heritage*, Springer Berlin Heidelberg, 2009, pp. 463–486, In Handbook on Ontologies.

[37] S. Vakulenko, J.D. Fernandez Garcia, A. Polleres, M. de Rijke and M. Cochez, Message passing for complex question answering over knowledge graphs, in: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1431–1440.

[38] D. Diefenbach, J. Giménez-García, A. Both, K. Singh and P. Maret, QAnswer KG: Designing a Portable Question Answering System over RDF Data, in: *The Semantic Web*, 2020, pp. 429–445.

[39] W3C, LinkedData, 2016, Last access March, 2021. https://www.w3.org/wiki/LinkedData.

[40] A. Developer, Build Skills with the Alexa Skills Kit, 2014, Last access March, 2021. https://developer.amazon.com/en-US/docs/alexa/ask-overviews/build-skills-with-the-alexa-skills-kit.html.

[41] Semantic Computing Research Group (SeCo), Mapping Manuscript Migrations, 2017, last access March, 2021. https://mappingmanuscriptmigrations.org/.

[42] Museum of Fine Arts Budapest, Open Linked data from the Museum of Fine Arts Budapest, 2016, last access March, 2021. http://data.szepmuveszeti.hu.

[43] Y. Raimond, DBTune classical, 2007, last access March, 2021. http://dbtune.org/classical/.

[44] Scottish Governmentl, Open access to Scotland's official statistics, 2010, last access March, 2021. https://statistics.gov.scot.

[45] UNESCO, The UNESCO thesaurus, 1977, last access May 2020. http://vocabularies.unesco.org/.

[46] R. Usbeck, R.H. Gusmita, A.N. Ngomo and M. Saleem, 9th Challenge on Question Answering over Linked Data (QALD-9), in: *Proc. of the 9th Question Answering over Linked Data challenge co-located with 17th International Semantic Web Conference (ISWC)*, 2018, pp. 58–64.

[47] R. Usbeck, M. Röder, M. Hoffmann, F. Conrads, J. Huthmann, A.-C. Ngonga-Ngomo, C. Demmler and C. Unger, Benchmarking question answering systems, *Semantic Web* **10**(2) (2019), 293–304.

[48] J.R. Lewis and J. Sauro, The Factor Structure of the System Usability Scale, in: *International conference on human centered design*, 2009, pp. 94–103.

[49] The UNESCO Thesaurus, What is meant by cultural heritage?, 2017, Last access March, 2021. http://www.unesco.org/new/en/culture/themes/illicit-trafficking-of-cultural-property/unesco-database-of-national-cultural-heritage-laws/frequently-asked-questions/definition-of-the-cultural-heritage/.

[50] W3C - World Wide Web Consortium, SPARQL Query Language for RDF, 2008, Last access March, 2021. https://www.w3.org/TR/rdf-sparql-query/.

[51] M. Koho, E. Heino, P. Leskinen, E. Ikkala, M. Tamper, K. Apajalahti, J. Tuominen, E. Mäkelä and E. Hyvönen, WarSampo Knowledge Graph on Zenodo, Zenodo, 2019.