Chapter 7
Querying and Searching Heterogeneous Knowledge Graphs in Real-time Linked Dataspaces

André Freitas, Seán O'Riáin, and Edward Curry

Keywords Knowledge graphs · Query processing · Data search · Best-effort · Dataspace

7.1 Introduction

As the volume and variety of data sources within a dataspace grow, it becomes a semantically heterogeneous and distributed environment; this presents a significant challenge to querying the dataspace. Approaches used for querying siloed databases fail within large dataspaces because users do not have an a priori understanding of all the available datasets. This chapter investigates the main challenges in constructing query and search services for knowledge graphs within a linked dataspace. Search and query services within a linked dataspace do not follow a one-size-fits-all approach and utilise a range of different techniques to support different characteristics of data sources and user needs.

This chapter is structured as follows: Section 7.2 explores the difference between querying and searching knowledge graphs in a Real-time Linked Dataspace and details the high-level functionality needed by the search and query service. Section 7.3 introduces search and query over dataspaces, discusses the challenges with data heterogeneity in a dataspace, and identifies the core requirement for the search and query service. State-of-the-art analysis of existing approaches to searching and querying is provided in Sect. 7.4. Section 7.5 details an analysis of the emerging design features for creating schema-agnostics query mechanisms, and the chapter concludes in Sect. 7.6.

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_7

7.2 Querying and Searching in Real-time Linked Dataspaces

Driven by the adoption of the Internet of Things (IoT), smart environments are enabling data-driven intelligent systems that are transforming our everyday world, from the digitisation of traditional infrastructure (smart energy, water and mobility), the revolution of industrial sectors (smart autonomous cyber-physical systems, autonomous vehicles, and Industry 4.0), to changes in how our society operates (smart government and cities). To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among intelligent systems.

7.2.1 Real-time Linked Dataspaces

A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management that is distinct from current approaches. The dataspace approach recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. In this chapter, we focus on the search and query support services of the RLD.

Dataspaces assume that the querying capability of the participants in the dataspace is not equal, and they do not assume the support of any specific standards to support data sharing. By building on web (URIs and HTTP) and semantic web standards (such as the Resource Description Framework and RDF Schema [RDFS]), and vocabularies, RLD can effectively reduce barriers to data publication, consumption, and reuse within a dataspace. Participants in a linked dataspace expose their

data as Knowledge Graphs (KGs), which can be interlinked and integrated with other datasets, creating an interlinked dataspace.

RLDs and generic dataspaces share more commonalities than differences, and the analysis provided in this chapter is relevant to both. In this chapter, the scope of the search and query service is mainly focused on non-streaming data sources with the search and query of live data streams discussed in Part III of this book. The discussion in this chapter builds on our earlier analysis of querying heterogeneous linked data sources [111] by contextualising the challenges within Real-time Linked Dataspaces [4].

7.2.2 Knowledge Graphs

Knowledge graphs pose challenges inherent to querying highly heterogeneous and distributed data. To query, data users must first be aware of which datasets potentially contain the data they want and what data model describes these datasets, before using this information to create structured queries. This query paradigm is deeply attached to the traditional perspective of structured queries over databases and does not suit the heterogeneity, distributiveness, or scale we expect from the datasets and KGs within a linked dataspace. It is impractical to expect users to have a previous understanding of the structure and location of datasets within the linked dataspace. Letting users expressively query relationships in the data while abstracting them from the underlying data model is a fundamental problem for massive data consumption, which, if not addressed, will limit the utility of dataspaces for consumers.

Consider a journalist compiling a list of facts regarding public personalities and their family connections. The journalist can express his or her information needs as natural language queries, such as "Who are the children of Marie Curie married to?" Document search engines cannot currently provide a level of query interpretation that could point directly to the final answer. With a traditional search engine, the journalist must navigate through the links and read the content of each candidate page the search engine returns.

The information that can answer this query may be available in the linked dataspace. However, to access it, users must know the location and structure of relevant datasets and the syntax of the query language. There exists a semantic gap between the user's information need, which is expressed in a generic natural language query and the data representation in the target dataset. The query's terms and structure differ from the data representation in the dataset. The provision of intuitive and flexible query mechanisms that can approximate users from an unconstrained amount of data represents a fundamental challenge of querying knowledge graphs in a linked dataspace.

7.2.3 *Searching Versus Querying*

Query mechanism for structured data which supports data consumers with expressive queries (queries which can make use of the conceptual structure behind the database and the supported database operations) and abstracts them away from the representation (being schema-agnostic) is an active research challenge.

The simplicity and intuitiveness of search engine interfaces, where users search the web using keyword queries, was a crucial element in the widespread adoption of web search engines and in the process of maximising the value of the information available on the web. On the other side of the spectrum, from the perspective of structured/semi-structured data consumption, users expect precise and expressive queries. In this scenario, most users query data with the help of structured query languages such as SQL or SPARQL. In a large-scale data scenario, structured query approaches do not thoroughly address all search and query usability requirements from all categories of users (such as being accessible to casual users and supporting lower query construction times for expert users).

With the web, users have recognised search to be a first-class activity. The search paradigm used in the web, however, cannot be directly transported for querying structured data. Keyword search over data does not provide the desired expressivity, while traditional structured query mechanisms have poor usability. Query expressivity and usability are two dimensions of database querying which define trade-off behaviour. Different categories of query/search approaches have emerged, targeting the trade-off between usability and expressivity (see Fig. 7.1) and have achieved some level of success.

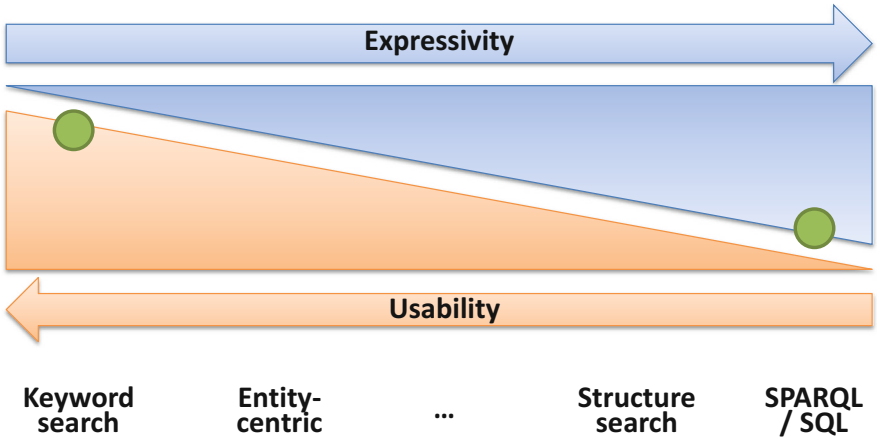

Fig. 7.1 The expressivity–usability trade-off for querying over structured data. The green dots indicate that an ideal query mechanism must provide both high expressivity and high usability [111]. Adapted from [161]

7.2.4 Search and Query Service Pay-As-You-Go Service Levels

The objective of the Search and Query service is to help developers, data scientists, and users to find relevant datasets within the dataspace. Users can navigate the dataspace by entities (if supported), or by performing a search or query on the datasets. A key challenge in developing search and query services over heterogeneous sources in a dataspace is the expressivity–usability trade-off. An ideal dataspace query mechanism must provide both high expressivity and high usability. As data sources are more tightly integrated into the dataspace, and move towards forming a knowledge graph, the search and query service can offer more sophisticated functionality.

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The functionality of the search and query service follows the five star pay-as-you-go model (detailed in Chap. 4) of the RLD. The search and query service offers the following levels of functionality:

1 Star **Browsing:** Browsing of the datasets available in the dataspace catalog.

2 Stars **Keyword Search:** Basic keyword search of the sources within the dataspace.

3 Stars **Structure Search:** A structured search is when the dataset has been indexed by the search service to enable entity-centric searches over the data and structure of the dataset.

4 Stars **Structured Queries:** Structured queries are possible where the data source supports a SPARQL interface, or the data source has been loaded into the local RDF store of the query service. In order to write a structured query (which can be entity-centric), the user must understand the underlying schema of the data.

5 Stars **Schema-Agnostic Question Answering:** A best-effort entity-centric natural language interface to the dataspaces knowledge graph that allows users to ask questions without understanding the underlying schema.

The provision of search and query services within a linked dataspace does not follow a one-size-fits-all approach with a range of different techniques used to support the different characteristics of the data sources and user needs. Running queries over heterogeneous data sources is a particularly challenging proposition that is an active area of research. The two initial levels, Browsing and Keyword search, are well-understood techniques that have readily available solutions. The third level is structured queries where keyword queries' expressivity is enhanced to include the structure of the data. This is achieved by extending existing inverted list indexes to represent structural information present in datasets. The next level is structured queries where SPARQL query support is provided over data sources via endpoints

on the data source or by importing data into the local RDF store. These approaches are suited to creating queries over homogenous and well-formed schemas that the user understands. Working with heterogeneous sources at large scales requires a different approach with the ability for the query mechanism to be agnostic to the underlying diverse schemas. At the high-end of the search and querying service for the RLD is schema-agnostic question answering over the interlinked knowledge graph that simplifies user–data interaction. A famous example of this emerging style of data interaction is the IBM Watson Question Answering (QA) system which competed in the television game show Jeopardy or the Apple Siri virtual assistant.

7.3 Search and Query over Heterogeneous Data

The vocabulary problem for databases is a consequence of data heterogeneity [162], that is, the multiple realisations in which data can be represented. Even if given the same task, different database designers can materialise the same domain into a database using different lexical expressions, conceptualisations, data models, data formats, or record granularities [162]. This intrinsic variability in the construction of a database defines a fundamental level of data heterogeneity between different databases.

Similarly, there is an intrinsic heterogeneity between a specific data representation and the data consumer's mental representation of a domain. If asked to materialise their information needs as free queries (e.g. using natural language), data consumers would be likely to use different terms and structures in the query formulation, a fact which is supported by Furnas et al. [163]. The intrinsic heterogeneity is mediated by the role of phenomena intrinsic to natural languages such as synonymy, ambiguity, and vagueness. The vocabulary problem is a concrete instance of the syntactic and semantic barriers in the knowledge boundaries identified in the Knowledge Value Ecosystem (KVE) Framework that exist when sharing knowledge among systems. These boundaries to knowledge sharing are discussed in more detail in Chap. 2.

7.3.1 Data Heterogeneity

Data heterogeneity becomes a more immediate concern as users start to query data/KGs from different datasets built by independent parties. This is the scenario faced by dataspaces (and Knowledge Graphs) where one starts to move from a centralised schema and data model (where data is integrated under a single representation model) to a decentralised scenario where data from different schemas and data models are brought together into a different data consumption context [2, 74]. The concept of data heterogeneity can be examined within different dimensions:

- *Conceptual Model Heterogeneity:* Different domains can be conceptualised using different abstractions and lexical expressions, which are dependent on the intended use behind the database and the background of the individuals modelling the domain (the KVE knowledge boundary). Given a modelling task with a minimum level of complexity, it is unlikely that two independent parties will generate identical conceptual models [162, 163]. Semantic heterogeneity emerges as a central concern in a dataspace when, data from multiple datasets, developed by different third-parties, need to be accessed and processed in a different context. Conceptual model heterogeneity includes distinct classes of differences which define the conceptual gap.
- *Format Heterogeneity:* Covers different formatting assumptions for values. This dimension covers the notational and measurement units' differences. Examples of value types dependent on data format are currency, numerical values, and date-time values. Abbreviations and acronyms are also included in this category.
- *Data Model Heterogeneity:* Data models provide the syntactical model in which different data objects are represented. Different data sources can be represented using different data models (the KVE knowledge boundary). Examples of data models include the relational model RDF and eXtensible Markup Language (XML), among others.

The three data heterogeneity dimensions are orthogonal and impact the reconciliation of model dimensions between different databases/KGs and the ability of users to query a data source. The more significant the gap between the two models (data, format or conceptual), the larger is the cost of querying or data integration.

The abstraction of users from the conceptual database model is intrinsically connected with the provision of a principled semantic matching mechanism to cross the conceptual gap between the user query and the data representation. Query mechanisms with the ability to automatically bridge the gap between the user and database conceptual models are described as schema-agnostic or vocabulary-independent queries. A motivational scenario example is introduced below.

7.3.2 Motivational Scenario

Suppose a user has an information need expressed as the natural language query 'Who are the children of Marie Curie married to?' (Fig. 7.2). The person has access to different databases/KGs within a dataspace which contain data that can help to address the information need. However, the data representations inside the target databases do not match the vocabulary and structure of the natural language query.

Figure 7.2 depicts an example of the semantic gap between the example user query and possible representations for the knowledge graphs supporting answers for the query. In (a), 'child' and 'married to' in the query map to 'Child' and 'Spouse' in the knowledge graph; in (b), these query terms map to 'motherOf' and 'wifeOf'

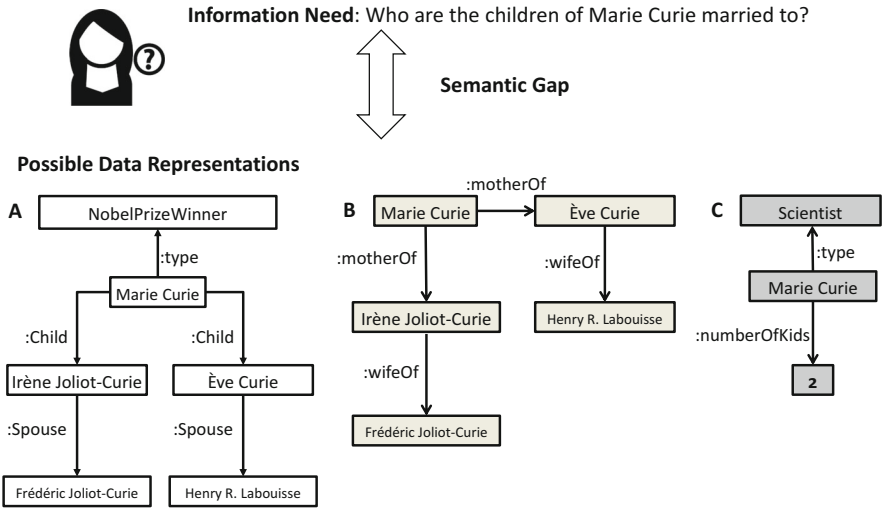

Fig. 7.2 Example of user information requirement expressed as a natural language query and possible knowledge graph representations in different conceptual models

respectively; while in (c), the query information related to 'child' is given by the predicate 'numberOfKids' representing an aggregation in (c), not fully mapped to the query information need.

To address query-data alignments, it is necessary to provide a query mechanism which can support a semantic matching which copes with the semantic gap between the user query and the knowledge graph representation.

7.3.3 Core Requirements for Search and Query

The dimensions of semantic heterogeneity are at the centre of the search and query challenge within dataspaces and addressing them directly can define the semantic matching requirements to provide robust search and query mechanisms. However, in addition to the requirements related to the semantic matching, search and query approaches need to satisfy requirements common to all search and query mechanisms. These requirements are used as qualitative dimensions to evaluate the effectiveness of search and query approaches:

- *High Usability and Low Query Construction Time:* Support for a simple and intuitive interface for experts and casual users.
- *High Expressivity:* Queries referencing structural elements and constraints in the dataset (relationships, paths) should be supported, as well as operations over the data (e.g. aggregations, conditions).

- *Accurate and Comprehensive Semantic Matching:* Ability to provide a principled semantic matching addressing all the dimensions of the semantic heterogeneity problem (abstraction, conceptual, compositional, functional) with high precision and recall.
- *Low Setup and Maintenance Effort:* Easily transportable across datasets/knowledge graphs without significant manual adaptation effort. The query mechanism should be able to work under an open domain and across multiple domains. Databases should be indexed with a minimum level of manual adaptations in the construction of supporting semantic resources used in the semantic matching.
- *Interactive Search and Low Query-Execution Time:* Minimisation of user interaction/feedback effort in the query process. Users should get answers with interactive response times for most of the queries. An interactive query execution time is contrasted with a batch query execution time (seconds vs. minutes).
- *High Scalability:* The query approach should scale to large datasets/knowledge graphs both in query execution and indexing construction time.

With a clear understanding of the challenges and requirements that need to be overcome, we now examine state-of-the-art approaches for searching and querying heterogeneous data.

7.4 State-of-the-Art Analysis

Three high-level categories of approaches for querying heterogeneous data within dataspaces exist: (1) approaches employing strategies inherited from the Information Retrieval (IR) space in which keyword search is mixed with elements from structure queries, (2) approaches focusing on natural language queries, and (3) structured SPARQL queries over distributed datasets. Leveraging existing work [111] we focus on the usability and semantic matching problems, thus analysing approaches from the first two categories.

7.4.1 Information Retrieval Approaches

We can categorise IR approaches according to the index type, which includes entity-centric search approaches and structure search approaches. Although both types provide hybrid search interfaces that merge keyword search with dataset structure elements, only structure search targets indexing strategies focus on addressing the expressivity–usability trade-off at the index construction level.

Entity-Centric Search
Entity-centric approaches let users search for entities (instances and classes) in datasets, employing Vector Space Model (VSM) variations to index those entities. Existing approaches range from less expressive queries, based on keyword search

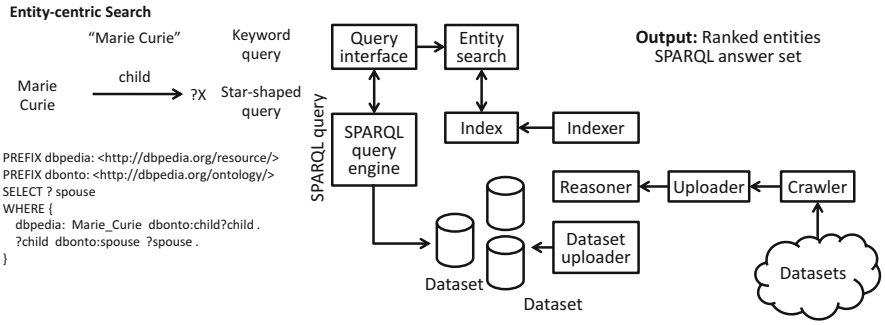

Fig. 7.3 High-level architecture components for Sindice (entity-centric search) [111]

over textual information associated with the dataset entities, to star-shaped queries and hybrid queries (i.e. queries mixing keyword search, and structured queries centred on an entity).

The Semantic Web Search Engine (SWSE) is a search and query service that implements an architecture with components for crawling, integrating, indexing, querying, and navigating over multiple data sources [164]. The system architecture's main components include query processing, ranking, an index manager, and an internal data store (YARS2), which focuses on scalability issues to enable federated queries over linked data. SWSE uses an approach called ReConRank to rank entities [164]; this approach adapts the PageRank algorithm to work over RDF datasets, propagating dataset-level scores—computed from interlinking patterns—to data-level entities. The Scalable Authoritative OWL Reasoner (SAOR) provides an RDFS and a partial Web Ontology Language (OWL) reasoning engine to address scalability issues [164]. SAOR applies reasoning only on dataset fragments supported by an authoritative ontological definition.

Sindice is a search and query service for the linked data web that ranks entities according to the incidence of keywords associated with them [165]. It uses a node-labelled tree model to represent the relationship among datasets, entities, attributes, and values. Similar to SWSE, Sindice provides a comprehensive entity-centric search and indexing approach. Figure 7.3 depicts Sindice's architecture.

The SPARK [166] approach provides a ranking solution for translating keyword-based queries to low complexity SPARQL queries, targeting low complexity RDF datasets. The SPARK is based on three basic steps: term mapping, query graph construction, and query ranking.

Entity-centric search approaches have developed comprehensive data management strategies for linked data on the web, providing the infrastructure for managing the complete crawl–index–search cycle. These approaches also developed services complementary to the entity-centric search process that let users either visually explore (via Visinav [164] and Sigma [165]) or execute full structured SPARQL queries over the crawled data. Entity-centric approaches avoid significant changes in

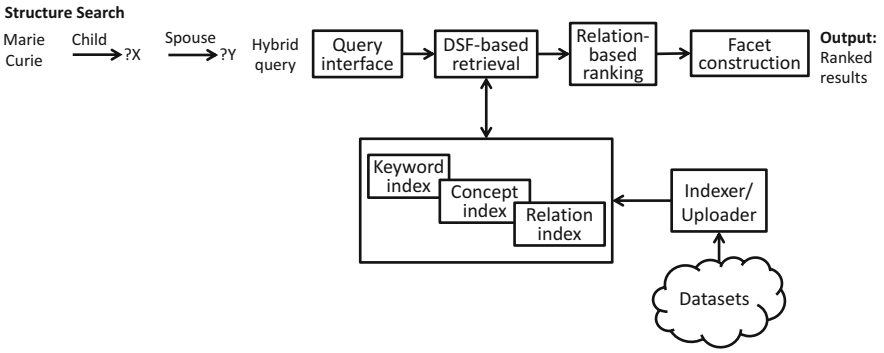

Fig. 7.4 High-level architecture components for Semplore (structure search) [111]

standard indexing strategies, inheriting index and search optimisation mechanisms present in existing VSM frameworks. These approaches have avoided tackling the expressivity–usability trade-off by aggregating multiple query interfaces; in practice, to execute expressive queries, users must be aware of the vocabularies behind the datasets. Also, most entity-centric approaches have only limited evaluation in terms of the search result quality.

Structure Search

Structure search engines improve keyword queries' expressivity, extending existing inverted list indexes to represent structural information present in datasets. The main difference between entity-centric search and structure search is that the latter improves query expressivity with support from the extended index.

The search engine Semplore [167] uses a hybrid query formalism that combines a keyword search with structured queries (i.e. a subset of SPARQL). Semplore uses position-based indexing to index relations and join triples. It relies on three types of inverted indexes: keyword, concept, and relation. Semplore also explores user feedback strategies for improving search, providing a faceted and navigational interface. Figure 7.4 depicts Semplore's high-level architecture. Xin Dong and Alon Halevy propose an approach for indexing triples to enable queries that combine keywords and dataset structure elements [168]. To provide a more flexible semantic matching, the authors propose four structured index types based on the introduction of additional structural information and semantic enrichment in the inverted lists. Taxonomies associated with the dataset vocabularies are used as a semantic enrichment strategy.

Structure search approaches target the expressivity–usability trade-off by modifying and extending traditional inverted index structures. They introduce a limited level of semantic matching by considering the terminology-level information present in datasets or by enriching the index with related terms using WordNet. No comprehensive evaluation of the search results' quality exists, making it unclear how these approaches perform in addressing the expressivity–usability trade-off.

7.4.2 *Natural Language Approaches*

Approaches in the literature based on natural language queries target query mechanisms with high usability and expressivity. Although some approaches focus on the question answering (QA) problem, in which, similar to databases, precise answers are expected as the output. Others focus on a best-effort scenario that returns a ranked list of results.

Question Answering
The investigation of QA systems focuses on the problem of allowing users to query data using natural language queries. As opposed to IR techniques' best-effort nature, QA systems target crisp answers, as with structured queries over databases. Work on QA approaches investigates the interpretation of users' information requirement that is expressed as natural language queries, applying Natural Language Processing (NLP) techniques to parse queries and match them with dataset structures. Substantial research efforts have focused on this problem. We look at two works on open domain linked data.

PowerAqua is a QA system that uses PowerMap, a hybrid matching algorithm comprising terminology-level and structural schema-matching techniques with the assistance of large-scale ontological or lexical resources [169]. In addition to the ontology structure, PowerMap uses WordNet-based similarity approaches as a semantic approximation strategy.

Exploring user interaction techniques, FREyA is a QA system that employs feedback and clarification dialogs to resolve ambiguities and improve the domain lexicon with users' help [170]. Compared to PowerAqua, FREyA delegates a large part of the semantic matching and disambiguation process to users. User feedback enriches the semantic matching process by allowing manual entries of query-vocabulary mappings. Figure 7.5 depicts FREyA's high-level architecture.

TBSL [172] exploits both natural language and information retrieval techniques and explores corpus-based patterns to support schema-agnosticism. TBSL relies on parsing the user question to produce a query template. The core rationale behind the approach is that the linguistic structure of a question together with well-defined expressions in the context of QA over structured data (such as more than and the

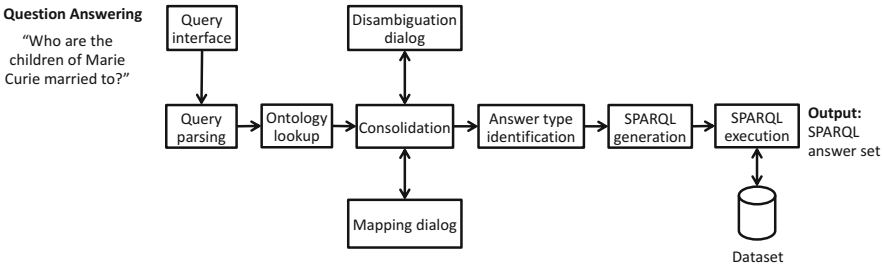

Fig. 7.5 High-level architecture components for FREyA (question answering) [111]

most) define a domain-independent structure for the query, which then needs to be filled in with domain-specific vocabulary elements.

Compared to IR-based approaches, QA approaches aim toward more sophisticated semantic matching techniques because they target queries with high expressivity and do not assume users are aware of the dataset representations (high usability). In contrast to entity-centric and structure search approaches, QA systems have a strong tradition of evaluating the quality of results and have concentrated less on performance and scalability issues. Traditionally, QA approaches have focused on limited semantic matching (WordNet-based) strategies, making them unable to cope with high levels of heterogeneity. Most QA approaches apply limited semantic matching techniques (e.g. synonymic, taxonomic similarity) for matching query terms to dataset terms. Also, they depend on resources that are manually created (WordNet) and difficult to expand across different domains.

Best-Effort Natural Language Interfaces

More recent approaches aim to merge natural language queries' expressivity and usability with IR models' scalability and best-effort nature, targeting a best-effort natural language search mechanism. As in QA systems, users can still enter full natural language queries; however, instead of targeting crisp answers, these approaches return an approximate ranked list of results.

The Treo natural language query mechanism for linked data uses semantic relatedness measures derived from Wikipedia to match query terms to dataset terms [171]. The use of semantic relatedness measures allows the quantification of the semantic proximity between two terms, using semantic information which is embedded in large textual resources available on the web such as Wikipedia. Wikipedia-based semantic relatedness measures address previous limitations of WordNet-based semantic matching. Treo's approach combines entity search, spreading activation search, and semantic relatedness to navigate over the linked data/knowledge graph, semantically matching the parsed user query to the data representation in the datasets. Figure 7.6 depicts Treo's components.

The principles of the Treo approach are generalised by constructing a distributional semantic space (T-Space) for linked datasets [121]. The T-Space is built using a distributional semantic model based on statistical semantic information derived from Wikipedia. This model enables flexible semantic matching in the search

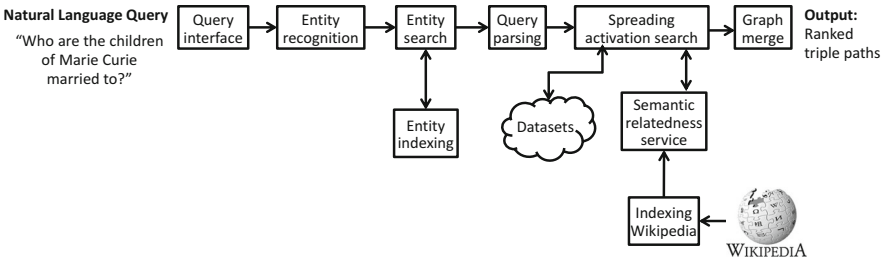

Fig. 7.6 High-level architecture components for Treo (best-effort natural language) [111]

process. The definition of the T-Space provides a principled representation of datasets focused on addressing the expressivity–usability trade-off.

7.4.3 Discussion

Table 7.1 lists how each category addresses the key requirements for search and query over heterogeneous knowledge graph within a linked dataspace. The practical relevance of a search and query mechanism lies in the fact that structured data is a fundamental component of data sources where the effort associated with accessing this structured data is still significant and heavily mediated by database experts. The dissolution of the expressiveness/usability trade-off is the goal of schema-agnostic query approaches (see Fig. 7.7) that provide a semantic matching approach which enables the alignment or semantic mapping of the data consumer's query to the database conceptual model elements. Based on the analysis in Table 7.1, we can see that the Treo approach meets most of the requirements identified. Best-effort natural language search approaches provide a robust semantic matching approach. However, they relax expectations in terms of query results, delegating the results' final assessment to end users.

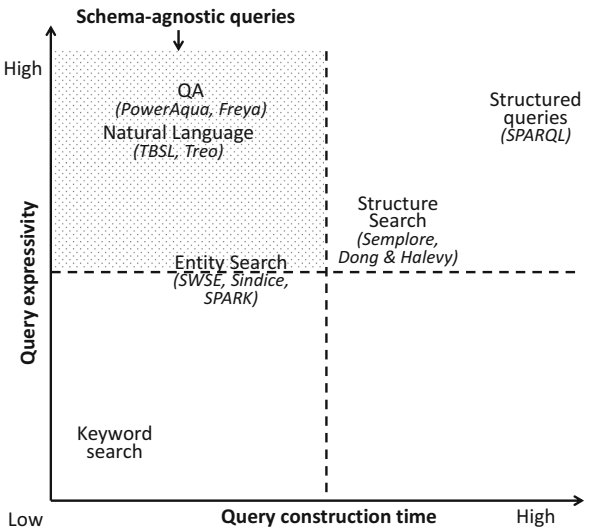

Fig. 7.7 Query expressivity vs. Query construction time quadrant. Schema-agnostic queries allow both high expressivity and low query construction time

Table 7.1 Coverage of approaches to address dataspace query requirements

Approaches	Query requirements						
	Level of schema agnosticism	High query expressivity	High scalability	Interactive search and low query execution time	Accurate and comprehensive semantic matching	Low setup and maintainability effort	High usability and low query construction time
Information retrieval							
Entity-centric: SWSE/ Visinav [164]	−	+−	++	++	−	++	+−
Entity-centric: Sindice/ Sigma [165]	−	+−	++	++	−	++	+−
Entity-centric: SPARK [166]	+	+−	NE	NE	+	++	++
Structure indexes: Semplore [167]	+−	+−	++	++	−	++	+−
Structure indexes: [168]	+−	+−	+	++	+	++	++
Natural language approaches							
QA systems: PowerAqua [169]	+	+	+	+	+	+	++
QA systems: Freya [170]	+	+	NE	+	+	−	+
Natural language search: TBSL [172]	+	+	+	+	+	+	++
Natural language search Treo [173], Treo T-Space [121]	++	++	+	+	++	++	++
Structured queries							
SPARQL	−	++	++	NA	NA	−−	−−

++ requirement dimension is well covered
+ requirement dimension is partially covered with positive
+− there is an attempt to address requirement dimension, but the solution is not effective
− requirement dimension is poorly covered
−− requirement dimension is very poorly covered
NA requirement dimension is not addressed or focused on the research
NE dimension dependent on evaluation is either poorly or not evaluated

7.5 Design Features for Schema-Agnostic Queries

Leveraging existing work [111] we analysed the current approaches to determine the design features present in search and query mechanisms over heterogeneous data to determine the key features needed for a knowledge graph query mechanism for RLDs. The result of this analysis is presented in Table 7.2, where we can see five key design features emerging as clear trends for the creation of search and query services over heterogeneous knowledge graphs [111].

Query Type Entity-centric search, keyword-based search, natural language queries, and structured SPARQL queries represent complementary search and query services that might suit users in different tasks and purposes. Search and query platforms should explore this complementary aspect regarding heterogeneous data to enable users to switch among different search and query strategies. SWSE and Sindice explore this trend; however, the availability of natural language queries is a key feature not present in these systems. As part of the search and query features, users should be able to explore, understand, and refine search results by relying on navigational, browsing, and filtering capabilities integrated into the process (this functionality is present in SWSE, Sindice, and Semplore).

For many years, the difficulties associated with the hard constraints of the question answering problem have overshadowed the potential for applying NLP techniques for queries. NLP has developed a large set of techniques and tools for parsing and analysing users' information needs expressed as natural language queries. Different flavours of syntactic parsers, morphological analysers, and named entity recognition techniques are widely and effectively employed in QA systems and natural language search interfaces (e.g. PowerAqua, FREyA, Treo, and Treo T-Space). Recently, the efficacy of NLP techniques was demonstrated in the IBM Watson system [174], which outperformed a human contestant in a "Jeopardy" challenge. Watson heavily leverages standard NLP techniques to build a complex information extraction and search pipeline. Search and query mechanisms can explore NLP techniques to provide expressive and intuitive query interfaces.

Disambiguation The presence of ambiguity and incomplete information is intrinsic to the search and query process. As already explored in systems such as FREyA and Semplore, user feedback can help resolve ambiguities, enrich an application's semantic model, and filter and post-process results. Providing a supporting context around the answers can help users assess the data's correctness. In the Treo approach, the path in the dataset generated during the querying process provides contextual information for users. A best-effort approach can live together with database operations, such as aggregations, via data filtering mechanisms that let users remove incorrect entries from the results (e.g. using the associated type information).

Ranking In "If You Have Too Much Data, then 'Good Enough' Is Good Enough" [76], Pat Helland summarises the mindset shift that must occur in heterogeneous and distributed data environments, where many still expect the accurate and crisp results

Table 7.2 Design features of schema-agnostic query mechanisms for heterogeneous knowledge graphs

	Design features					
Approaches	Query type	Disambiguation	Ranking	Semantic approximation	Supporting knowledge bases/ linguistic resource	Performance/ scalability mechanism
Information retrieval						
Entity-centric: SWSE/Visinav [164]	Keyword/SPARQL	None	Modified TF/IDF + link-based	None	None	Inverted index
Entity-centric: Sindice/Sigma [165])	Keyword/Star-shaped	None	Modified TF/IDF + link-based	None	None	Inverted index
Entity-centric: SPARK [166]	Keyword-based	None	Yes	Dataset term expansion/ edit distance, substring matching	WordNet	
Structure indexes: Semplore [167]	Keyword with structural information (single-atom queries, path queries, star-shaped queries, entity queries, and tree-shaped queries)	Facet-based	Yes	Dataset term expansion (taxonomical enrichment)	None	Inverted index (position index)
Structure indexes: [168]	Keyword with structure information	None	Yes	Query/dataset term expansion (synonyms)	WordNet	Inverted index
Natural language approaches						
QA systems: PowerAqua [169]	Full natural language	None	Based on the similarity scores	WordNet-based (hypernym, hyponym, synonym) semantic/	WordNet	NA

(continued)

Table 7.2 (continued)

	Design features					
Approaches	Query type	Disambiguation	Ranking	Semantic approximation	Supporting knowledge bases/ linguistic resource	Performance/ scalability mechanism
				string similarity, based on taxonomical relations in the data		
QA systems: Freya [170]	Full natural language	Disambiguation dialog	Yes	Query/dataset term expansion, manual lexicon enrichment	WordNet	NA
Natural language search: TBSL [172]	Full natural language		Yes	Corpus pattern mining	WordNet, Corpus-based	Yes
Natural language search: Treo [173], Treo T-Space [121]	Full natural language	User feedback	Based on distributional semantic relatedness measure	Context-based distributional semantic approximation	Wikipedia-based ESA model	Distributional-relational structured vector space model

typical for siloed databases. This trade-off is discussed in more detail in Chap. 3. The challenge of building query solutions with high usability and expressivity in a dataspace is coping with the data's semantic heterogeneity; this demands to relax our expectations of the results into a best-effort solution. Ranked lists of results in which users can assess those results' suitability are widely used in document search engines; web users have been extensively exposed to this approach and are thus familiar with best-effort search models. However, although document search engines can potentially return a long list of candidate documents, the best-effort query [171, 175] and ranking [176, 177] mechanisms for dataspaces should leverage the structure and types present in the data to target more concise answer sets. A number of dataspace search and query approaches leverage associations and relations to rank results [87, 168, 178–180].

Semantic Approximation The difficulty in effectively providing a robust semantic matching solution has been associated with a level of semantic interpretation that depends on fundamental and hard problems in artificial intelligence, such as common-sense knowledge representation and reasoning. Dataspace query approaches have considered both synonyms [181] and similarity [182] within the matching process. Recently, distributional semantic approaches have emerged as solutions to provide robust semantic matching by leveraging the use of semantic information embedded in large amounts of web corpora.

Distributional semantic models assume that the context surrounding a given word in a text provides essential information about its meaning [183]. Distributional semantics focus on constructing a semantic representation of a word based on the statistical distribution of word co-occurrence in texts. The availability of high-volume and comprehensive web corpora has made distributional semantic models a promising approach for building and representing meaning. However, the simplification of distributional semantic models implies some constraints on its use as a semantic representation. Distributional semantic models are suitable for computing semantic relatedness, which can act as a best-effort solution for providing robust semantic matching solutions for linked data queries (present in the Treo T-Space system).

Supporting Knowledge Bases/Linguistic Resources The availability of large amounts of unstructured text and structured data on the web can help to bootstrap a level of semantic interpretation based on available open and domain-specific knowledge. It is possible to address the volume of unstructured text corpora necessary to build distributional semantic models by using comprehensive knowledge sources available on the web, such as Wikipedia (present in the Treo and Treo T-Space systems). In addition, it is possible to use the semantically rich entity structure of data sources such as DBpedia (http://dbpedia.org), YAGO (www.mpi-inf.mpg.de/yago-naga/yago/), and Freebase (www.freebase.com) as a general-purpose entity and entity typing system that can easily integrate to the target datasets to provide a minimum level of structured common-sense knowledge, and which can later be used to improve semantic interpretation and tractability. RDF's standardised graph-based format facilitates the reuse and integration of existing data sources into target datasets.

7.6 Summary

The emergence of heterogeneous and distributed data environments such as the web of data, knowledge graphs, and dataspaces, in contrast to small controlled schema databases, fundamentally shifts how users search and query data. Approaches used for searching and querying siloed databases fail within these large-scale heterogeneous data environments because users do not have an a priori understanding of all the available datasets. This chapter investigates the main challenges in constructing a query and search service for knowledge graphs within a dataspace. The search and query services within a dataspace do not follow a one-size-fits-all approach and utilise a range of different techniques from keyword search to structured queries and question answering to support different characteristics of data sources, and user needs in the dataspace. Our analysis of the state of the art shows that existing approaches based on IR and natural language query interfaces have complementary design features, which, if combined, can provide schema-agnostics solutions to the usability and semantic matching challenges of querying large-scale heterogeneous data.