Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-018-0536-1

SemiLD: mediator-based framework for keyword search over semi-structured and linked data

Mohamed Kettouch[1] · **Cristina Luca**[1] · **Mike Hobbs**[1]

Received: 11 December 2016 / Revised: 12 October 2018 / Accepted: 12 October 2018 /
Published online: 03 November 2018

Abstract
Linked Data initiative has completely changed the procedure of sharing knowledge over the Web. It primarily aimed at improving the interoperability and semantics of the data published, by following a set of recommendations. Still, many data sources, which have a significant value, have not migrated to this new data space and continue to publish semi-structured data. Thus, new challenges arise in accessing and integrating the two data sources and models. This paper explores and identifies some of the major challenges, such as the continuous expansion and dynamism of a heterogeneous and an autonomous yet connected web of data, and addresses them by proposing SemiLD, a mediator-based framework to integrate on-the-fly heterogeneous semi-structured and Linked Data sources. The approach is implemented into a highly automated keyword search system that retrieves its input from various SPARQL endpoints and web APIs. The evaluation of the system illustrates the high precision, performance and recall of the contributed approach.

Keywords Semantic web · Data integration · Linked data · Semi-structured data · Keyword search · Schema integration · Information retrieval

1 Introduction

Today, there is a large amount of information stored in distributed, independent and autonomous data sources. In many cases, these data sources can be used complementarily, in other words, the data scattered in these sources can be connected in order to find the targeted result. The user is required to manually perform this process by querying one source at a time. Thus, Berners-Lee (1999) proposed to introduce a new technology that allows new possibilities

✉ Mohamed Kettouch
mohamed.kettouch@anglia.ac.uk

Cristina Luca
cristina.luca@anglia.ac.uk

Mike Hobbs
mike.hobbs@anglia.ac.uk

[1] Department of Computing and Technology, Anglia Ruskin University, Cambridge, UK

in processing and connecting data on the web. Linked Data (LD) is the materialisation of his idea. It is a paradigm that lowers the barriers and facilitates the publishing of interlinked structured data based on four recommendations described in his Web architecture note "Linked Data" (Berners-Lee 2006).

The reduction in the number of restrictions in publishing and accessing information in the global information space has completely changed the procedure of sharing knowledge over the World Wide Web (Bizer et al. 2009). As a result, a transformation of the Web from a global information space of linked documents to a web of linked data has begun to take place (Haase et al. 2010). Since then, the amount of the available structured data has seen a dramatic growth and the LD cloud has significantly expanded in various domains. Linked Open Data (LOD) diagram (Cyganiak and Jentzsch 2014) illustrates the wide use of Linked Data in different domains and areas e.g. science, governmental and statistical data, people, etc.

Along with LD growth, Web APIs, another popular method of accessing data in the web, is still expanding. The APIs that allow third party access to functions and data, have opened up knowledge resources and generated new opportunities for applications to utilise, combine and re-propose information. Along with offering the possibility of mashing up content from different Web data sources (Bizer et al. 2009), most of the results were expressed in a semi-structured format, particularly Extensible Markup Language (XML) or JavaScript Object Notation (JSON), that can be automatically exchanged and computed (see Fig. 1). This has led to a boost in their volume (Verborgh et al. 2015) and in the availability of semi-structured data on the web. Still, Web APIs have many limitations, including the inflexibility in adding new sources once the interface is designed.

1.1 Research questions and objectives

Although the LD initiative has shown many semantic and navigational capabilities, there are several new challenges that arise. Accessing LD sources is not as usable as the publishing

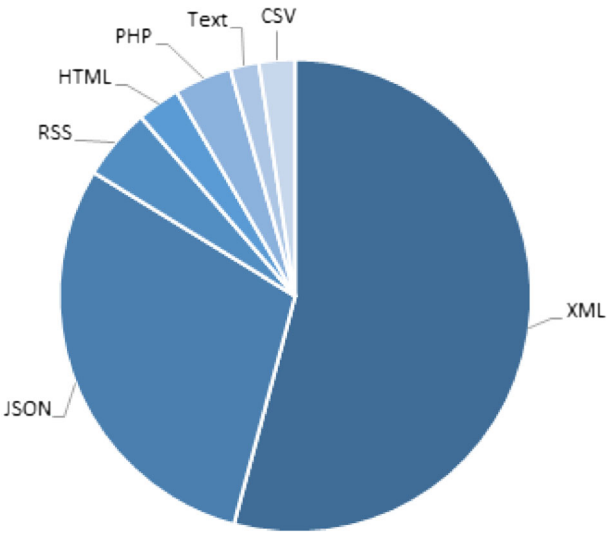

Fig. 1 Web API data formats in 2011 according to programmableweb.com

part (Zhao and Ichise 2013; Ciobanu et al. 2015), and can exclude ordinary users, without a technical background in SPARQL and the related technologies, to contribute to or to consume LD. In addition, a considerable amount of valuable semi-structured data, provided by the Web APIs, has not been migrated. To address these challenges, this paper identifies and responds to the following questions:

- How to enable the users to search and access semi-structured and structured data, particularly Linked Data, through a transparent interface?
- How can an integration system sustain the dynamism of the continuous expansion and changes in the Web of Data?

The research contributions stemming from addressing these questions are as follows:

- An automatic method to create a global schema that has the ability to expand automatically;
- Embedding a novel Schema Matching technique that combines semantic web technologies, such as ontologies, with similarity measurements, to map automatically between LD datasets and the global schema, and between semi-structured data and the global schema;
- Accommodating LD datasets changes without any human intervention;

1.2 Current challenges of data integration

Data integration is the process of providing homogenous access to a set of heterogeneous and autonomous sources (Calì et al. 2004). It differs from data federation in that it does not only collect and combine data from different sources, but rather offers a logically unified view. Data integration is still considered as a very challenging task despite being a major research subject in Web Semantics for more than 35 years (Kalja et al. 2014; Jarke et al. 2014). The rise of new data models and representations, such as the Resource Description Framework (RDF), as well as new challenges in terms of precision and performance suggest that this research needs to be (re)explored. The following are the main challenges that SemiLD addresses:

- **Decentralisation and Autonomy of the Sources:** data integration systems are required to deliver up-to-date information. This cannot be achieved without addressing the dynamism of the relation between the mediated schemas, or the central repository, and the sources.
- **Heterogeneity:** many different and incompatible data and knowledge description formats exist due to legacy systems and the increasing variety of new approaches. Various types of heterogeneity can occur at many levels, including structural, syntax or semantic mismatch, the access method, language, or protocol (Macura 2014).
- **Usability for end users:** it does not only indicate that the system ought to be easy to operate, but also to provide sufficient information for the users to appropriately interpret the outputs. The most usable method for accessing data sources, arguably, is keyword search (Freitas et al. 2012), as no pre-knowledge is required to use it.
- **Expressivity** is defined as the ability to "query datasets by referencing elements in the data model structure" (Freitas et al. 2012). A system can be considered expressive if it helps the users to make their queries more specific and structured to provide more details in querying data sources (Freitas et al. 2012). Although domain dependency is a limitation in Semantic Web applications in general, defining and limiting the domain can reduce the semantic conflicts; therefore, it increases the expressivity and the ability

to perform complex semantic interpretation and inference (Kaufmann and Bernstein 2010).
- **Adaptivity and the degree of Automation** are, in this context, the flexibility in accommodating the continuous changes in data structures and models. It is an essential criterion, particularly when LD namespaces are amongst the sources due to the increasing expansion of the web of data. It is also crucial to offer the possibility to add further sources in the future. Offering tailored integration views (Ziegler and Dittrich 2007) for specific sources, or case studies, can be seen as a limitation.
- **Privacy** of the data accessed is a recent major issue in this area. Since most of the integration systems are designed to be used by the public, the solutions should not include, mine or index sources with personal data and other sensitive information. Privacy is a big issue that can be avoided by including only publicly accessed sources, such as Web APIs and SPARQL endpoints.

This paper is organised as follows: Section 2 gives a summary of related works along with some examples and definitions. In Section 3, the system architecture is presented and detailed. Section 4 presents a testing of the implementation of the presented approach. Then, Section 5 proposes an evaluation of the precision and the performance of the system as well as a theoretical comparison against other solutions based on the identified challenges. Section 6 states the limitations of the proposed approach and the envisaged future works. Finally, conclusions are drawn in Section 7.

2 Related work

This section explains the context of this work by reviewing existing approaches to the problems of data integration and homogeneity reconciliation. It also highlights the contributions that have been made comparing to the search systems. Three broad classes can be identified according to the search method:

2.1 Document-centric search

Also referred to by universal search. The document-centric search consists mainly of the popular search engines, such as: Google, Yahoo and Bing. At the time they were introduced, many technological breakthroughs were able to connect the users with the one data source that contains the information they need. Although they were arguably an essential factor of the success of the Web, currently they are not sufficient to respond to all users queries. Data, originated from various sources, can be used complementarily to respond to the users' needs, and this feature is not supported by current search engines. Whilst they are able to perform federated search upon autonomous repositories, and some of them extend their search to cover "non-document" repositories, they still lack the ability to interpret and join the information retrieved semantically. Therefore, in the proposed approach, a global schema is constructed containing all the information related to the keyword search, integrated from various and heterogeneous relevant sources.

Other approaches focus on tackling the heterogeneity between semi-structured documents. The approach proposed in Abelló et al. (2018) addresses this issue by creating a common inexact schema based on the characteristics and resemblance between well formed semi-structured documents (valid XML documents in the context of the proposed approach). The paper took into consideration the fact that the schema of semi-structured data

is self-describing and schema-less as well as the problem of overfitting. Although the end result is good both in terms of efficiency and performance, the rules upon which the algorithms are built and the validity requirements of semi-structured documents do not always conform with graph data, such as Linked Data.

2.2 Semantic search and question answering (QA)

This group of systems includes most of the Natural Language Processing (NLP) systems. A knowledge base (Collarana et al. 2016), or unambiguous ontology, Lopez et al. (2013) is commonly used in this type of system to process the query and to integrate the outputs. Their main limitation is revealed when addressing large open-domain sources where the system ontology, or the knowledge base, is unable to disambiguate the query. The frequent challenge tackled by this class of systems is the heterogeneity within the LD, whereas semi-structured data sources, accessed through web APIs, are not considered. PowerAqua (Lopez et al. 2011) (evolved from AquaLog Lopez et al. 2007) is one instance of a QA solution that proposes the use of multiple ontologies that will be selected according to the user query. This concept is utilised in SemiLD as the authors agree with the fact that it is not possible to select in advance which of the vocabularies or ontologies will be needed to answer a query. The term heterogeneity is used in PowerAqua, and in most of the similar approaches in this category, to refer to ambiguity and the discrepancy in describing resources in LD, that may lead to multiple, yet not similar, results for one query. The authors proposed in Umbrich (2012) a Linked Data query engine that extends SPARQL with lightweight features for reasoning and optimisation in order to discover additional data sources and thus generating further relevant results. The triple pattern and logic that this approach is built on is not present in other data structures (including semi-structured data).

Heterogeneity in this paper has a wider meaning, as discussed in Section 1.2, being the difference of the structure, syntax, the access method, language, and protocol not only internally between LD datasets, but externally with semi-structured sources.

2.3 Hybrid search

Hybrid search has the ability to take in various data structures to respond to the users' query (Usbeck et al. 2015; Morbidoni et al. 2008). This category of systems can be studied from many perspectives. In this section, the main aspect discussed is the method used in integrating different data structures.

Semantic Web Integration Middleware (SWIM) uses query mediation and provides tools to "view data as virtual RDF" (Koffina et al. 2006). The middleware publish, or re-publish, XML and Relational databases (RDB) as RDF. Resource Query Language (RQL) queries are then composed and optimised according to the RDF views and the mappings constructed. The paper (Le-Phuoc et al. 2012) illustrates Linked Stream Middleware (LSM), a middleware system to integrate time-dependent data, or sensor data, with the LOD cloud. It unifies and publishes stream raw data, coming from different sources, as Linked stream data before finally executing SPARQL (Protocol and RDF Query Language) queries over them. The system uses wrappers in the data acquisition layer to collect the data from different sensor devices and publish them into a unified format. Having the data stored in a Linked Data layer, it will be then accessed via two types of query engines: a standard SPARQL query processor and Continuous Query Evaluation over Linked Streams (CQELS) engine processor.

In Vincini et al. (2013), the authors described the Mediator/Wrapper based architecture for integrating semi-structured and structured data. Mediator envirOnment for Multiple

Information Sources (MOMIS) uses its own description, definition, languages and a thesaurus for extracting, defining and storing the information inputted and retrieved from the sources. In their semi-automatic methodology, they follow GaV (Global as View) paradigm to express the global schema following to local schema.

In Talukdar et al. (2010), the authors tried to address the challenge of accommodating new datasets in a data integration system. They highlighted the difficulty of automatically discovering new relevant information and the semantic conversion of datasets in a data integration system. However, the approach they proposed, named Q system, focuses only on relational databases, and defines incoming data as new tables in a particular database. They incorporated a "registration service" via which they can add new tables or data sources either manually by the user or using a Web crawler that extracts tables from the Web. Q system is based on keyword search techniques, ranked answers and user feedback to integrate relational database tables. Integration in Q System is defined as a union of queries with more weight given to the more relevant. At the end, the system returns the query result to the user, who will be able to give feedback to the system by rating whether a result tuple is relevant.

The authors identified in Pánek (2015) that the challenge of building a global schema in a virtual data integration is the building of a solid metadata. Hence, they came up with a concept called "the catalog" which contains all the information needed for integration heterogeneous datasets. The admin needs, however, to create a new wrapper and an entry into the catalog every time a new source is added. The approach proposed by Pánek (2015) is based on REST architecture in both showing the result and administrating the catalog. REST architecture, however, imposes new limitation including the decreasing the complexity and the expressivity of the queries that the user can run.

FuhSen (Collarana et al. 2016) is a keyword search platform that federates data from heterogeneous sources. Although, strictly speaking, it is not an integration approach, but it is the closest to the context of the proposed approach. It is a usable and adaptable solution within its scope of interest, which is crime data information. As part of the integration process, the application uses many components including their own vocabulary named "OntoFuhSen". The vocabulary is used as an exchange and an intermediate language between the other parts of the system. In its current status, it is tailored to accommodate information about their targeted data, being information about a person, a product or an organisation. In spite of the fact that the vocabulary can be extended, its location in the system suggests that many other changes will need to be made as well. Wrappers (adapters) are one of the components connected to the vocabulary. They are designed to extract data from the source outputs. In FuhSen, every data source is associated with a collection of wrappers.

The SemiLD framework falls into this category of search systems. It is a hybrid search system with data integration system at the core. It takes two data models as input, semi-structured and LD sources, in order to offer a homogeneous and transparent access to non-expert users. SemiLD uses keyword input, which is user-friendly and hides the complexity of the integration and querying process from the user. It compensates the lack of expressivity of keywords by offering a very rich post-filtering feature that gives the possibility of refining the results using all the available properties and their values (see Section 4.3). But more significantly, it provides the ability for an integrated view to sustain the continuous and frequent changes in the web of data, while preserving the autonomy of the participated sources. The framework consists of six subsystem that were developed to address separate data integration sub-problems (Kettouch et al. 2015a, b, 2017; Fatima et al. 2014). They were part of a vision that is based on combining the use of property matching techniques, such as semantic similarity finding for mapping between sources and the global schemas, and

semantic web technologies, such as ontologies and the global schema in formulating the query and accommodating the results respectively. The aim is to achieve a high degree of automation, precision and recall.

As introduced previously, data integration has been undergoing intense study as a database problem for many years and, as a result, many tools and solutions have been proposed. The review presented in this section puts this proposal into context by comparing it with existing systems that have either a similar application area or the same input model. For instance, Keymantic (Bergamaschi et al. 2010) is a keyword-based search approach in data integration systems that requires no prior knowledge of the instances of the database. The only data sources queried in Keymantic, however, are relational databases.

3 The new SemiLD mediator-based approach

The distributed and the autonomous nature of LD sources make it unlikely to sustain the use of one model in representing data in a particular domain. Each source has its specificities, conditions, and a different vision on the way to expand. Hence, maintaining the mapping between LD links represent a challenging task, given they connect two vocabularies, models or views that are managed and situated in separate locations and regularly changing. This dynamism of the relations between the integration system and the sources, and the continuous expansion of the Web of Data, along with data freshness requirements, suggests that a solution would need to integrate data virtually on-the-fly. SemiLD combines the use of ontologies, to obtain high precision, with an unsupervised property matching as part of the interlinking subsystem, to achieve a high degree of automation while targeting large scale data.

The distributed and autonomous nature of semantic web sources imposes more challenges to schema mapping and leads to heterogeneous terminologies. Multiple ontologies and vocabularies can be utilised to represent similar information in a particular domain. On the other hand, it is observed and argued in Kettouch et al. (2015a) that although different dispersed RDF datasets, describing data in the same domain, may not be exactly identical, they overlap in the semantics of their properties. It is also validated in this paper in Section 4.1, where Fig. 5 shows that the semantic of Linked Data properties stabilises before their syntax, after retrieving a representative sample. This statement is as valid for LD sources as for semi-structured data sources. Semi-structured data is frequently described using XML or JSON technologies, where tags arguably play the same role as properties in RDF.

To exploit this semantic matching, the modular approach proposed, illustrated in Fig. 2, uses a global schema, as part of a mediated architecture, that has the ability to learn and expand automatically. The global schema is an XML file that contains all the possible and potential properties (tags) that the system may retrieve in running a query in a specific domain. The domain is defined upfront as it is not in the scope of this paper to recognise it automatically. By using a schema integration module, the system is able to map different semi-structured and LD sources with the global schema, in a specific domain, using the semantic similarity between the properties. The ontologies (or vocabularies) are used in the system to support the formulating of the query and clustering the results.

SemiLD modular architecture consists of four(4) main components: user interface, ontologies, query engine distributor and schema integration, as shown in Fig. 2, that are described in details along with the algorithms used in the following subsections. Figure 3 presents the flowchart of SemiLD.

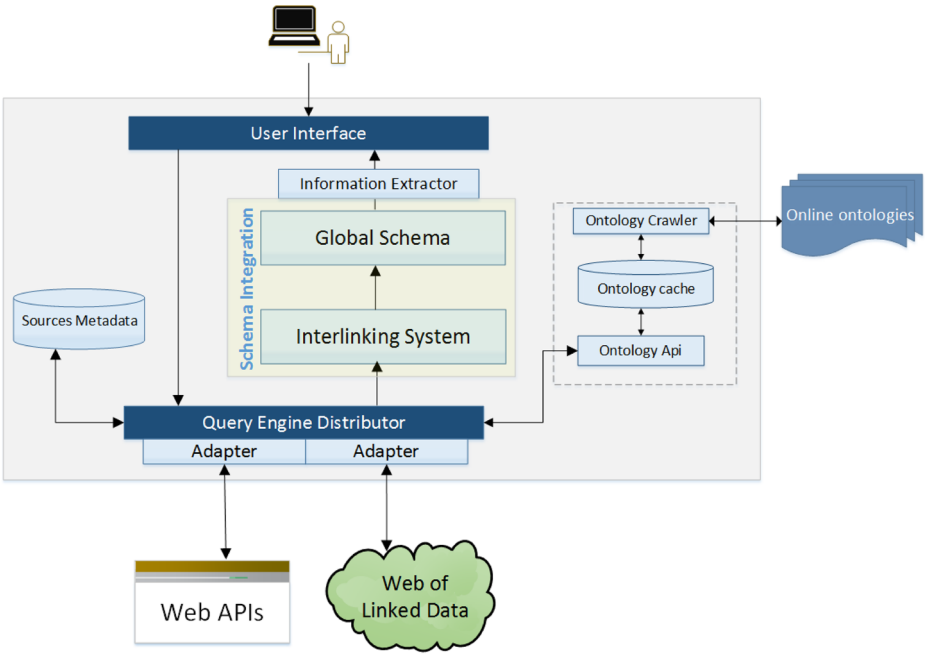

Fig. 2 General architecture of SemiLD

3.1 User interface

The user interface is a requisite component in order to allow non-expert users to benefit from the service provided. Recent search engines, particularly where RDF data is amongst the sources, may contribute to the writing of the queries by adapting it to the vocabulary or the structure used. It is different from the classic environments where developers write queries knowing the search requirements, data model and the query language. Thus, having a query language on its own may not be sufficient to benefit from an effective search system. It depends on the structure of the data being processed and the function of the system that users are interacting with.

To hide the complexity of SPARQL queries and API HTTP requests, the proposed interface allows keywords as an input. As stated in Section 1.2, keyword searching is considered a very usable mean of accessing information on the internet (Freitas et al. 2012). Their limitations, however, are centred on the expressivity as little information is provided in the query. This problem has been addressed in the implementation by adding optional and domain-related textfields that can match the structure of some of the sources. This will enable adding additional information to the query. The usability is also addressed in the interface by providing a feature to filter the list of the results by all the properties that are available in global schema.

The output of this module is a keyword, which is used to retrieve the ontologies needed in the query engine module. Then, this keyword is embedded in the formulated query. Additional and optional keywords can be provided by the user in the user interface to support

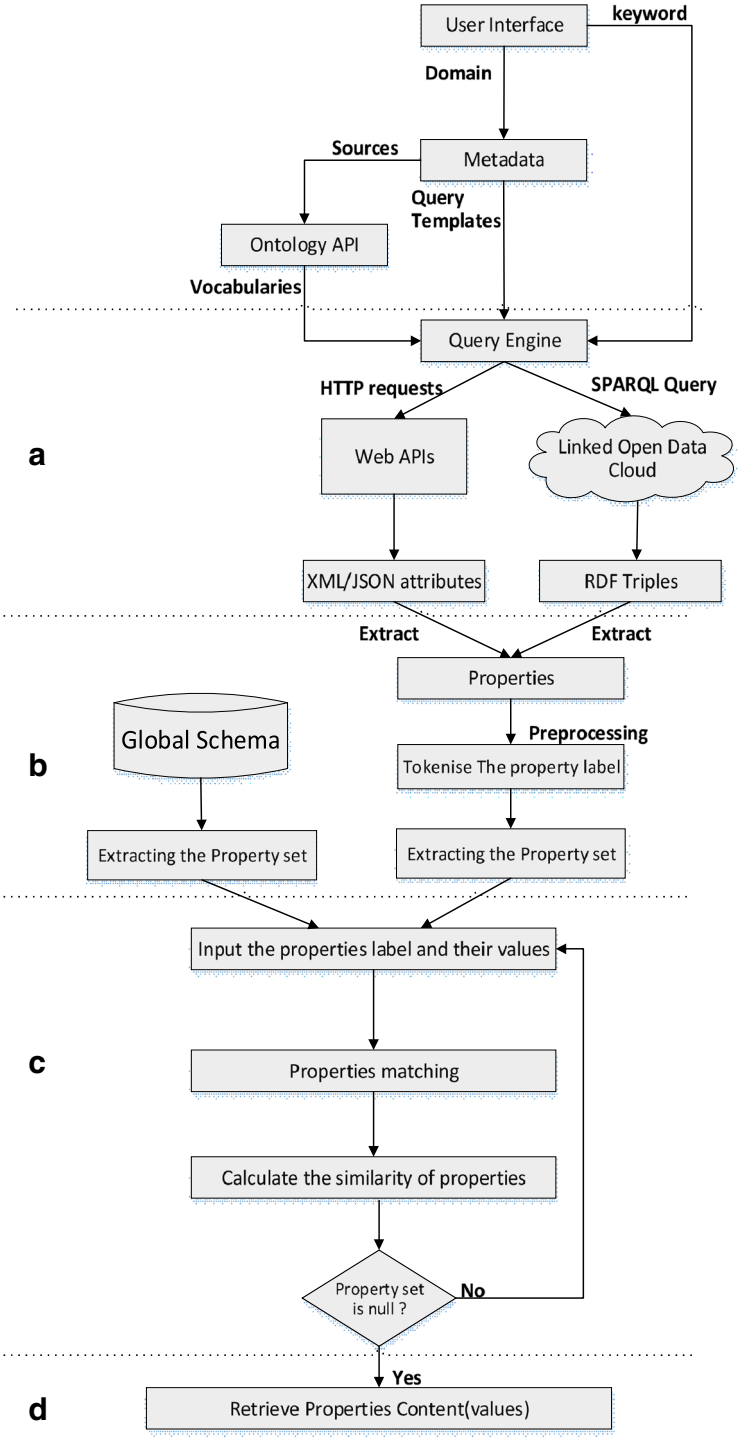

Fig. 3 Flowchart of SemiLD

the query and increase the expressivity. These keywords, however, frequently do not match the structure of all the Web APIs considered. For example: Web APIs for movies require a single keyword of the title, but some of them also permit to send more information, such as: the year and the gender, as an optional inputs. In the interface, a text field for the year can be used with a limited list of sources, but it is not utilised in the processing nor in the retrieving of ontologies.

In this approach, the expressivity is addressed by offering the possibility to post-filter the results according to all the properties of all the sources. For every property, it provides a list of the potential values to be used as filters. Thanks to the global schema that reconciles between the heterogeneity of the properties of different sources, the data is transfered to a uniform repository file.

3.2 Ontologies(Vocabularies)

Ontology is defined as a formal specification of a shared conceptualisation of some domain knowledge (Gruber 1993). It is a flexible, extensible, and scalable mechanism to describe and structure stored information (Ramis et al. 2014). The information is encoded in ontologies in the form of concepts and properties linked via semantic relations. SemiLD uses ontologies for two tasks in the system:

- To support the query engine distributor in forming and structuring the query against LD sources by allocating the appropriate vocabulary.
- To cluster the outputs according to the domain (defined upfront) in a domain-independent application of the approach to ensure an accurate and conflict-free schema integration.

The ontology module consists of many parts, rather than one central ontologies repository, that crawl for the ontologies and vocabularies of the LD namespaces considered, in order to support and sustain their continuous changes. The following three components are part of the ontology module:

3.2.1 Ontology crawler

The Ontology Crawler is responsible for caching online ontologies and checking regularly for updates. Using the list of the saved ontologies and their paths, the module checks regularly not only for the existence of the file containing the ontology, but also for the date of creation compared to the online version. The crawler is key to maintaining the high performance, precision and efficiency in search mining "by semantically discovering, formatting, and indexing information" (Dong and Hussain 2014). The Algorithm for the ontology crawler is drawn from Fatima et al. (2014) and adapted to a data integration scenario in Kettouch et al. (2015b).

3.2.2 The ontology cache

The ontology cache is a repository of ontologies that have been retrieved from running previous queries. This eliminates the time that the system would take to mine for ontologies online. A simple indexer is built-in the ontology cache module that classifies the ontologies URIs according to the keyword and LD source requested.

3.2.3 Ontology API

The API is an intermediate between the query engine distributor and the rest of ontology modules of the system. After receiving a keyword from the engine distributor it considers the sources then outputs a list of tags and vocabularies needed to form the query.

3.3 Query engine distributor

The query engine distributor is the central subsystem where all the information is gathered from the ontology module and the metadata store in order to perform the mapping in both directions. Algorithm 1 illustrates the overall functioning of the query engine distributor. Having received the information needed, the system prepares the query to be sent to the sources (top-down part). In the the second phase, the query engine parses the outputs retrieved from the sources. The results are then organised into a list of datasets in order to facilitate the next step, the interlinking.

Algorithm 1 Query Engine

Input: Keyword, Metadata, ontology_index_file
Output: Result-file

▷ top-down direction / Query decomposition and formation
for each source in Metadata.source_list() **do**
Query_structure= OntologyAPI(source, keyword)
Load (Adapter)
Connect (metadata.sourceLocation())
Query = PrepareQuery (Keyword, Query_structure)
Result-file = Execute (Query)
end for each
▷ bottom-up direction / Results parsing and preparation
Input: Result-file
Output: outputs: collection of parsed results

Results = Parse (Result-file)
rows-count = Results.length()
outputs.setSize(rows-count)
for i=0 to rows-count **do**
outputs(i).setcontent(Results.getrow(i).getcontent())
end for

return outputs

For every semi-structured source, an instance of an adapter is created to process its queries and outputs. It contains pre-defined functions to identify the source type and to send, parse and organise the information. Two roles are assigned to this module: first, it establishes the connection with the API URI or the SPARQL endpoint of the data source; second, it gathers all the information to reformulate the query and distribute it over the local schemas. Having retrieved the files containing the results from the sources and identifying both the format and the model, the adapters do the reverse process, splitting the results into a list of datasets.

3.4 Schema integration

The structural reconciliation is a decisive stage in virtual data integration. Schema Integration is the operation of mapping between the structures of two, or more datasets originating from the same or different, dispersed data sources. The model and format of the datasets may differ as a result. The result of this module is a "common schema" (Cai and Yates 2013) that is able to map with the participated data sources. The common schema, in this paper, is referred to as the global schema.

The schema Integration in SemiLD is an extension of previous work (Kettouch et al. 2017). It consists of two components:

3.4.1 Global schema

The global schema is the module responsible for representing the outputs of the different sources, adapted and organised in the Query Engine Distributor, into a single and uniform temporary storage. The global schema also has the role of preparing the results to be displayed by parsing, extracting and organising them into a list of instances. XML is the format used to fulfil this task due to its effectiveness and popularity in information exchange, along with the simplicity and the availability of the tools manipulating this data language.

The global schema is created when the system is first developed, and it is updated on a time-lapse basis to verify whether a new source has been added or the structure of existing sources is modified or extended.

The Algorithms 3 and 4 utilise a method, described in Algorithm 2, to extract the semantically distinct properties between two sets of properties. The method utilise UMBC EBIQUITY-CORE (Han et al. 2013) (see Section 3.4.2) to measure the semantic distance between two property labels.

Algorithm 2 SemanticDistinction

```
Input:  set1, set2: PropertiesSets
        threshold
Output: P: PropertiesSet

  sizeSet1 = size (set1)
  sizeSet2 = size (set2)
  for i=0 And i < sizeSet1 do
      for j=0 And j < sizeSet2 do
          distance = SemanticDistance (set1[i], set2[j]);
          if distance > threshold then
              P.addAttribute(set1[i])
          else
              i++; j++;
          end if
      end for
  end for

  return P;
```

The global schema is formed by extracting all the semantically distinct properties of all the sources considered. This will force a semantic overlap between the global schema and all the sources considered. It is designed to make the properties of every source a semantic subset of the properties of global schema, as demonstrated in Algorithm 3.

Algorithm 3 The creation of the global schema

Input: S: DataSources
ps_x: Properties of a Source x
Output: G: GlobalSchema

```
while S.hasNext() do
    if S_0 (The first Source) then
        ps_0 = extractProperties(S_0)
        G.addAttributes(ps_0)
    else
        G.AddAttributes(SemanticDistinction(ps_x, G) )
    end if
end while
```

The properties extraction differs between the semi-structured and Linked Data. For semi-structured data, retrieved via Web APIs, the properties are extracted by processing one result, since all datasets share the same properties. Whereas in LD sources, not all the datasets in a particular namespace share the same structure. Therefore, the properties are extracted through a separate process that takes into consideration many datasets. LD datasets use various vocabularies in representing their data. Yet, many of these vocabularies, particularly when they are related to the same domain, share many semantically identical properties. This will create redundancy in the global schema which lead to conflicts in generating the mapping rules in the semantic matching. Therefore, the system retrieves and processes the properties from the first N results (that varies according to the LD source) (see Section 4.1) in a particular domain. The system goes on extracting the semantically distinct properties from the properties of the outputs and then incrementally comparing them with what has been already found in order to update the global schema, as illustrated in Algorithm 4. Finally, the global schema contains the semantically distinct properties of all the sources considered.

Algorithm 4 Properties extraction in an LD source

Input: pr_x: Properties retrieved of a result x
T: Threshold of stabilisation
Output: P: Semantically distinct properties of an LD source
N: Number of results needed to extract all the semantically distinct properties

```
while pr.hasNext() do
    if x==0 then
        P.addProperties(pr_0)
    else
        P.AddAttributes(SemanticDistinction(pr_x, P) )
        if SemanticDistinction(pr_x, P) - SemanticDistinction(pr_{x-1}, P) < T then
            N = x
            break
        end if
    end if
end while
```

3.4.2 Interlinking subsystem

Having the global schema created with a guarantee of a semantic overlap with all the participant sources, the interlinking system matches semantically the properties of the retrieved results with it. Data interlinking is a technique that discovers the counterparts of a single real world object that may have properties recorded in one or more sources (Nguyen et al. 2012). In the context of this paper, however, it is used as a complementary tool to the integration approach. Its main role is to automatically generate mapping rules between the predicates (properties) of the retrieved sources and the global schema. The final output of this system is a list of mapping rules.

The system is drawn from two components of a previous work (Kettouch et al. 2015a). They are part of an asymmetric and unsupervised approach to compute the semantic similarity between resources in the LD web. It would be almost impractical to process the integral heterogeneous web of Linked Data using a manual or supervised (based on conceiving training set) approach due to the prior knowledge or the resources needed to apply such techniques. Furthermore, the critical response time required by the nature of the process, data access and search, does not allow sufficient time to learn every time a query is executed. The common drawback of unsupervised algorithms, however, is the high computational cost required to implement it.

The flowchart in Fig. 3 describes the interlinking module proposed. It can be seen that the dataset goes through many stages before the properties matches are identified and the content is re-allocated from the sources to the global schema. These stages can be organized and grouped in three main phases as follows:

- **Preparing the datasets** The section (A) in the flowchart in Fig. 3 corresponds to the query engine distributor (Section 3.3). It highlights that the subsystem starts when the results are retrieved from all sources. A pre-processing step (see Section B in Fig. 3) is then performed to extract the labels of the resources (in this case the predicate) that are described using URIs, which is its last part according to Linked Data principles (Berners-Lee 2006). The label of the predicate (or the property) is tokenised in order to optimise the measurement of semantic similarity in the next stages.

  Example: the output of the URI http://dbpedia.org/ontology/foundingDate after the pre-processing is "founding Date". In Addition, the properties with one character are excluded, as no semantics can be derived from them.
- **Properties Matching** The distance between the properties of the source dataset and the target dataset is measured by calculating the semantic similarity of their labels (see Section C in Fig. 3), using the semantic text similarity system: UMBC EBIQUITY-CORE (Han et al. 2013). UMBC tool is constructed by combining LSA word similarity and WordNet knowledge. It concentrates on the semantics of the word but not its lexical category, which makes it a typical mean similarity measurement for our system, since the available vocabularies for describing vary between nouns and verbs. UMBC also provides a web API whereby external systems can retrieve the similarity between two texts without the necessity of going through the re-implementation of the approach. The following URL is a prototype of UMBC API:

  http://swoogle.umbc.edu/SimService/GetSimilarity?
  operation=api&phrase1=SourcePropertyLabel&
  phrase2=TargetPropertyLabel

UMBC similarity tool is implemented in SemiLD in order to eliminate the connection time to the API every time a similarity matching is required. It also helps in evaluating the genuine performance of the system.

– **Instances integration** This is the final stage of the integration process. Having the list of the matched properties between the Global Schema and each of the target datasets, the system extracts their content (instance) (see Section D in Fig. 3). The instances are then transferred to their counterpart in the global schema using the generated mapping rules.

4 Implementation and testing

This section presents a test scenario of the implementation of the proposed architecture. SemiLD is a mediator-based integration system that takes into consideration semi-structured and Linked Data. It is implemented into a keyword search in a collection manager for movies. The user searches for information about movies in four heterogeneous sources, being:

– Two LD sources: DBpedia,[1] LinkedMDB[2];
– Two semi-structured sources: OMDB,[3] TMDB.[4]

The system then fetches all the available information and displays it in an interactive interface. Many graphical forms and features are provided to improve the user experience to hide complexity. To the best of our knowledge, the architecture presented in this paper and its implementation are unique. The implementation is written in Java with Jena libraries, along with other several tools to parse JSON and XML files.

The global schema and the interlinking module are the key components in this system. Their creation and processing is completely independent from any model or domain. The ontology module is also able to access a variety of models to formulate the query. The only component that is domain-dependent, and can arguably be subject to amendments, is the user interface and the list of the semi-structured sources considered.

Due to the nature of the task, it is difficult to propose a benchmark or a specific methodology in evaluating systems integrating LD with semi-structured sources (Pfaff and Krcmar 2014). Thus, comparability in this type of system is frequently carried out theoretically. The following evaluation is designed to measure the genuine precision, recall and performance of this specific approach, followed by a theoretical comparison with the previously mentioned related works.

4.1 The creation of the structure of the global schema

The aim of this section is to show the significant difference between the number of properties that are syntactically unsimilar and semantically different, as well as identify the 'N' number for this implementation. The N number, discussed in Section 3.4.1, represents the

[1] DBpedia.org
[2] LinkedMDB.org
[3] OMDB.com
[4] TMDB.org

Fig. 4 An example of a SPARQL query to count the syntactically distinct properties in DBpedia

```
select count(distinct ?p) where {
  ?s ?p ?o .
  FILTER (?s = ?film)
  {
    select ?film {
    ?film a <http://schema.org/Movie>.
    }
  limit 100
  }
}
```

number of the first results needed to extract all the semantically distinct properties. First, a query is run to count the syntactically distinct properties on the SPARQL DBpedia endpoint, which is a multi-domain LD source that uses various and heterogeneous vocabularies in describing datasets in the same domain. Then, the semantic distance is measured between these properties in order to extract the semantically distinct properties and count them. The N number is identified when the number of the semantically distinct properties becomes steady.

Figure 4 is an example of a SPARQL query that counts (and retrieves by removing count) the syntactically distinct properties that the first 200 movie datasets contain. It is an adaptable query for all LD endpoints supporting subquery feature (SPARQL 1.1), where only vocabularies used change. In this example, it is expressed to work on the DBpedia endpoint.

For other LD endpoints that have not been updated and still running SPARQL 1.0, such as LinkedMDB (at the time this paper is written), the system uses the Jena framework to nest the results of a query within another query. The version of the SPARQL endpoint is included in the Metadata to decide automatically which of the two predefined means will be used.

After applying Algorithm 2 (see Section 3.4.1) to extract and count the number of the semantically distinct properties, the line chart in Fig. 5 is generated. It illustrates the discrepancy between the numbers of the syntactically and semantically distinct properties according to the number of the results retrieved.

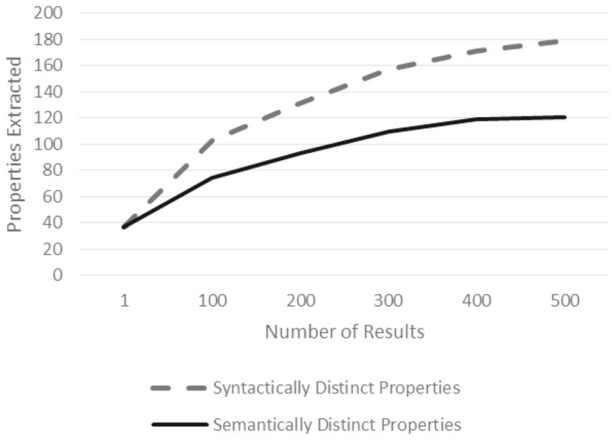

Fig. 5 The discrepancy between the syntaxes and semantic unsimilarity

Table 1 Metadata example

The source	API URL / SPARQL endpoint	Requested results	SPARQL 1.1
DBpedia	http://dbpedia.org/sparql	4	1
LinkedMDB	linkedmdb.org/sparql	4	0
IMDB	http://api.themoviedb.org/3/search/movie	3	N/A
OMDB	http://www.omdbapi.com/	3	N/A

The diagram shows the noticeable gap between the numbers of the two types of properties. It can be concluded from this chart that the vocabularies describing datasets in the same domain share roughly the same properties semantically. The N number that can be extracted from this diagram is 500.

The structure of the Global Schema is created when the system is first developed, and it is updated in the background on a time lapse basis, similar to a "cron job" (on a time lapse basis through a scheduled process). Since the time of the creation of the Global Schema is not part of the response time, the N number can be set to a maximum and a "safer" value that ensure all the properties are recalled from all sources. Thus, it does not affect the adaptivity nor the degree of the automation, as it does not run every time the system is queried, and nor does it need to be changed when a new LD source is added.

4.2 Testing of the metadata

This is the only module in the system that is predefined manually. This repository indicates the links to the SPARQL endpoints or Web APIs of the sources to be queried (see Table 1). In addition, the metadata gives the users the possibility to choose the number of the results desired from each of the sources considered, accessed through an interface (see Fig. 6). It is also important, as discussed in Section 4.1, to determine whether the version of SPARQL supported in the endpoint is SPARQL 1.1 or lower.

4.3 Testing of the user interface

Figure 7 shows the dynamic and interactive feature that allows the users to filter the results according to the properties of the Global Schema, along with the possible values extracted from the sources. These features are essential in this keyword search to rectify their lack of expressivity without affecting the usability, by assisting the users in finding the requested result. The system also keeps track of the originated source of every result.

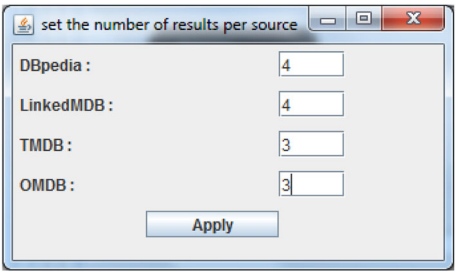

Fig. 6 Example of a metadata repository

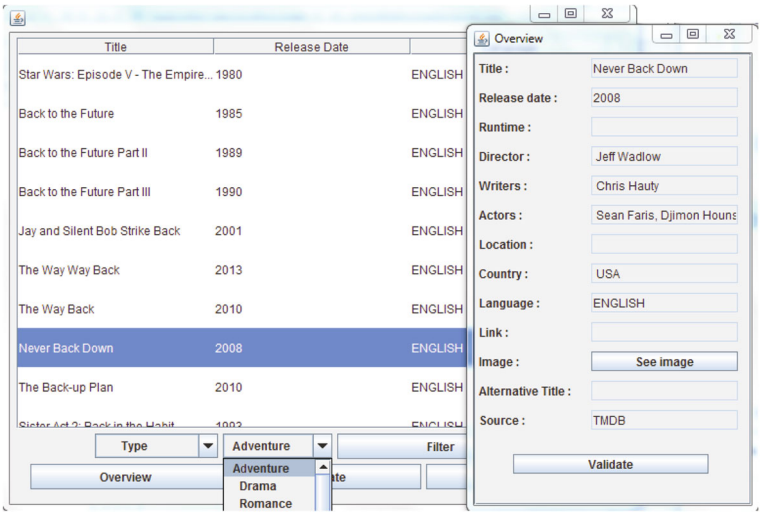

Fig. 7 Results visualisation

5 Evaluation of the system

As part of the evaluation, the system is queried and tested using common keywords, such as:"best". Figure 8 is a SPARQL query to search a keyword in an LD source.

SPARQL endpoints offer various formats in expressing the results. To illustrate the heterogeneity, RDF is the format outputted from LD sources, and JSON and XML are the formats expressing the results of the semi-structured sources.

S1, S2, S3, S4 in Tables 2 and 3 refer to DBPedia, LinkedMDB, TMDB, OMDB sources respectively.

Table 2 evaluates the process of the creation of the global schema according to the number of first results considered. It shows that DBpedia has a considerable semantic overlap between the properties of the vocabularies describing its datasets. More importantly, there is a noticeable overlap with the properties of the four sources considered. For example, for N= 500, the system extracted 199 semantically distinct properties from all sources; however, 35 were deleted as they have the same semantics of some of 164 properties previously retrieved. It can be seen that DBpedia and LinkedMDB are more general than the rest of semi-structured sources. In this case, they contain all the properties available in IMDB and TMDB.

Fig. 8 An example of a SPARQL query to count the syntactically distinct properties in LinkedMDB

```
PREFIX mdb: <PATH/data.linkedmdb.org/resource/movie/>
PREFIX rdfs: <PATH/w3.org/2000/01/rdf-schema#>
PREFIX dc: <PATH/purl.org/dc/terms/>

select * where {
  ?title dc:title  ?keyword.
  filter(REGEX(?keyword, ''best'',''i''))
}
limit 4
```

Table 2 The number of the semantically distinct properties extracted per source

Number of results	Syntactically distinct properties				Semantically distinct properties				Global schema
	S1	S2	S3	S4	S1	S2	S3	S4	
100	103	36	20	13	74	36	20	13	112
200	131	41	20	13	93	41	20	13	136
300	157	43	20	13	110	43	20	13	153
400	171	47	20	13	119	45	20	13	164
500	179	47	20	13	121	45	20	13	164

Table 3 illustrates the precision and recall of the system. The keywords are ordered from the most to the least general. The common keywords generate the maximum number of results; whereas the specific words do not occur in many movies, less results are returned. The precision is calculated by dividing the properties matched (the number of mapping rules generated) by the semantically distinct properties retrieved. The global schema generated from processing the first 500 results is the one utilised in this table.

Table 4 is an example of a portion of the results retrieved. The predicates Runtime, Director and Year are included to show the differences between the sources and the results retrieved in terms of the available information.

Figures 9 and 10 validate the performance and the recall of the approach. Due to the limited number of results that the keyword "best" can retrieve, it is changed to a more common keyword "in" to reach a higher number of results. Figure 9 shows the insignificance of the delay caused by increasing the number of the results requested in the first search operation. In Fig. 10, the diagram presents a comparison between the performances of SemiLD against FuhSen. Although the scenario in FuhSen upon which the line chart is generated is different, the performance included is based on 10 wrappers, which is the optimal that FuhSen can achieve.

Table 5 presents a theoretical comparison of SemiLD, against the related systems. As discussed in Section 1.2, adaptivity, in this context, refers to the ability to add new sources automatically in order to increase the scale of the amount of data retrieved, while not addressing a subset of the available sources in the targeted data structure(s).

In contrast to FuhSen, SemiLD does not utilise any pre-defined vocabulary or language. The global schema is the intermediate step that accommodates the results from the datasets. It is constructed automatically according to the sources considered. More importantly, the approach generates mapping rules between the global schema and the sources, that guarantees the integration of the data. Furthermore, SemiLD has only two classes of adapters, one for semi-structured source and the other for LD sources, which are used to access rather than extract. They are not modelled to conform to the structure of the sources. Instead, for every approach an instance of adapter is created, which receives all the information needed from a metadata file. The latter is the only module that contains a minimal amount of preloaded information needed about the sources for the system to function.

Finally, Table 6 indicates which of SemiLD module(s) are responsible for addressing each of the challenges listed in Section 1.2.

The privacy in this approach is addressed by excluding relational databases and other tools that may give the users access to non-public data. The system can be only used to search through different Web APIs and SPARQL endpoints, which represent an alternative gate to an available public data.

Table 3 Mapping precision of SemiLD

Keyword	Results requested per source	Results available					Properties retrieved				Properties matched with Global Schema				Overall precision
		S1	S2	S3	S4	overall	S1	S2	S3	S4	S1	S2	S3	S4	
In	50	50	50	50	50	200	56	36	20	13	56	36	20	13	1
In	200	200	200	200	200	800	94	41	20	13	85	41	20	13	0.97
Best	50	50	5	50	50	152	59	21	20	13	57	21	20	13	0.99
Best	200	124	5	200	200	529	83	21	20	13	71	21	20	13	0.96
London	200	85	4	200	200	489	79	22	20	13	72	22	20	13	0.97
Steve	200	15	0	200	200	415	57	–	20	13	56	–	20	13	0.99

Table 4 Example of the results retrieved by running a search using the keyword "best"

Title	Runtime	Director	Year	Source
My Best Friend's Wedding	105	P. J. Hogan	1997	LinkedMDB
The Best Man	102	Franklin Schaffner	1964	LinkedMDB
My Best Friend's Birthday	–	Quentin Tarantino	1987	LinkedMDB
The Best of Insomniac	–	Nick McKinney	2003	LinkedMDB
O Despertar da Besta	–	Jose Mojica Marins	1983	DBpedia
Best Wishes for Tomorrow	110	Takashi_Koizumi	2008	DBpedia
Best Player	98	Damon Santostefano	2011	DBpedia
The Best Exotic Marigold Hotel	–	–	2011	OMDB
The Best Offer	–	–	2013	OMDB
Best	–	Mary McGuckian	2000	TMDB
Best of the Best	–	Andrew Lau Wai-Keung	1996	TMDB
Best of the Best	–	Herman Yau	1992	TMDB

6 Limitations and future works

The primary aim of SemiLD is to improve the degree of automation and to sustain the dynamism of LD space in an integration system. Addressing such a challenge comes at a cost. The data integration system that aims at adapting itself to unpredictable and unknown future changes of LD datasets cannot rely on and employ the structure of these datasets in the reconciliation process. This approach means that there is a limitation for SemiLD in the schema integration module, where the textual form of the properties recovered are semantically processed. This causes problems if the labels of the properties are encoded in a semantically meaningless syntax. But in the context of this application, where the sources are web services and LD sources, the datasets generally are expressed in a mining and parsing friendly form as they have been designed to facilitate data exchange with other

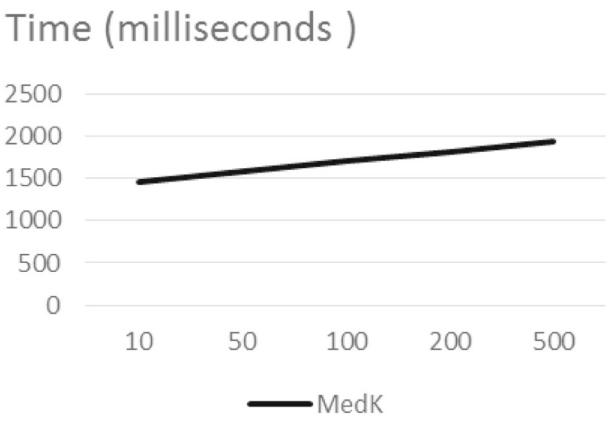

Fig. 9 The processing time according to the number of the results searched

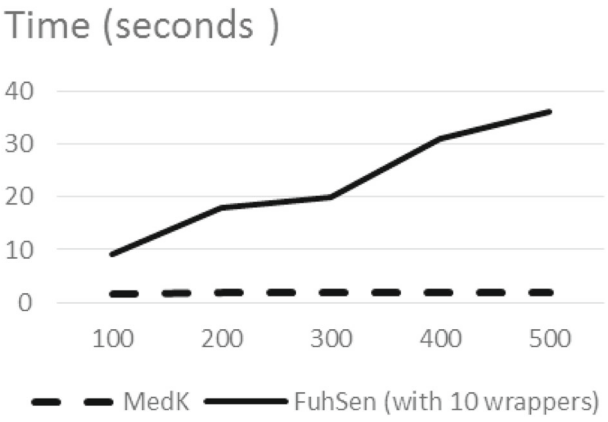

Fig. 10 Comparison between the processing time of SemiLD and FuhSen

services or applications. This problem does not affect the other modules, particularly the query engine distributor since it is not dependent on the syntax and rather utilises a dynamic selection of vocabularies mechanism.

As pointed out in Section 1.2, there are not, as yet, any well-established measures for qualitative evaluation of data integration approaches taking into account different data models. Unlike ontology matching, or data federation to a lesser extent, it is challenging "to make data of different types of benchmarks comparable with each other" due to the lack of a common description or a parameter that can be measured upon (Pfaff and Krcmar 2014). The comparison is frequently carried out theoretically taking into account approaches that have not necessarily presented explicit results. The other method, which the authors tried to adopt in this paper, is to evaluate the results based on a manually pre-constructed "gold standard" or ground truth to assess their quality. Still, this method is limited to the scale of data or number of scenarios that can be verified manually and is unable to provide a common and objective ground for direct comparison.

Further work will address the limitations and explore opportunities to use this approach in different areas.

Table 5 Mapping precision of SemiLD

	Approach	Query expressiveness	Adaptivity	Up-to-date data	Generic	Semantic
Google	Federated	Keywords, NL	Yes	Yes	Yes	No
PowerAqua	Centralised	NL	No	Yes	Yes	Yes
SWIM	Centralised	SPARQL/RQL	No	No	Yes	Yes
LSM	Centralised	SPARQL/CQELS	No	No	No	Yes
MOMIS	Federated	n/a	No	n/a	Yes	Yes
FuhSen	Federated	Keywords	No	Yes	No	Yes
SemiLD	Centralised	Keywords	Yes	Yes	Yes	Yes

Table 6 The modules responsible for adressing each of the challanges

Challenge	Module(s)
Decentralisation and autonomy of the sources	All modules except the user interface
Heterogeneity	All modules except the user interface
Usability for end users	The user interface
Expressivity	The user interface
Adaptivity and the degree of automation	All modules except the user interface
Privacy	Addressed by including publicly accessed sources

7 Conclusions

In the last decade, researchers in the semantic web community have been designing tools and architectures to integrate heterogeneous data originated from distributed sources. Technologies, such as RDF, were created in response to the increased adoption of LD paradigm, have enabled new data spaces and concept descriptors to define an increasing complex and heterogeneous web of data. Other types of data that existed previously, such as semi-structured, still hold a significant value in many areas. To bridge between the two data spaces, this paper proposed a mediator-based approach that offered a homogeneous and transparent access to these sources. The idea behind the system presented in this paper is to create a more general global schema in order to force an overlap with the participating sources. It is composed by retrieving all the semantically distinct properties of both LD sources and semi-structured sources. Then, using the interlinking module, the mapping rules are generated automatically. The data originated from the heterogeneous sources is parsed and re-organised in the global schema to be finally displayed in an interactive interface for the user. The implementation of this approach is a keyword search engine, embedded in a Movie Collection Manager, that takes into consideration all the challenges and the criteria stated in the paper. The results confirm the performance, the precision and recall of the approach presented that are (similar or) better than comparable approaches, while providing a user friendly interface.

References

Abelló, A., de Palol, X., Hacid, M.S. (2018). Approximating the schema of a set of documents by means of resemblance. *Journal on Data Semantics*, *7*(2), 87–105. https://doi.org/10.1007/s13740-018-0088-0.

Bergamaschi, S., Domnori, E., Guerra, F., Orsini, M., Lado, R.T., Velegrakis, Y. (2010). Keymantic: semantic keyword-based searching in data integration systems. *Proceedings of the VLDB Endowment*, *3*(1-2), 1637–1640.

Berners-Lee, T. (1999). Weaving the Web. Harper.

Berners-Lee, T. (2006). Linked data. http://www.w3.org/DesignIssues/LinkedData.html. Accessed: 04 Jan 2016.

Bizer, C., Heath, T., Berners-Lee, T. (2009). Linked data-the story so far. *International Journal on Semantic Web and Information Systems*, *5*(3), 1–22.

Cai, Q., & Yates, A. (2013). Large-scale semantic parsing via schema matching and lexicon extension. In *ACL (1)* (pp. 423–433). Citeseer.

Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M. (2004). Data integration under integrity constraints. *Information Systems*, *29*(2), 147–163. https://doi.org/10.1016/S0306-4379(03)00050-4.

Ciobanu, G., Horne, R., Sassone, V. (2015). Minimal type inference for linked data consumers. *Journal of Logical and Algebraic Methods in Programming*, *84*(4), 485–504. https://doi.org/10.1016/j.jlamp.2014.12.005.

Collarana, D., Lange, C., Auer, S., Grangel-González, I. (2016). Fuhsen: a platform for federated, rdf-based hybrid search. In *The 16th international conference on web engineering (ICWE2016)*.

Cyganiak, R., & Jentzsch, A. (2014). Linking open data cloud. http://lod-cloud.net/versions/2014-08-30/lod-cloud.svg.

Dong, H., & Hussain, F.K. (2014). Self-adaptive semantic focused crawler for mining services information discovery. *IEEE Transactions on Industrial Informatics*, *10*(2), 1616–1626. https://doi.org/10.1109/TII.2012.2234472.

Fatima, A., Luca, C., Wilson, G. (2014). User experience and efficiency for semantic search engine. In *2014 International conference on optimization of electrical and electronic equipment (OPTIM)* (pp. 924–929). IEEE.

Freitas, A., Curry, E., Oliveira, J.G., Riain, S.O. (2012). Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends. *Internet Computing IEEE*, *16*(1), 24–33.

Gruber, T.R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220. https://doi.org/10.1006/knac.1993.1008.

Haase, P., Mathäß, T., Ziller, M. (2010). An evaluation of approaches to federated query processing over linked data. In *Proceedings of the 6th international conference on semantic systems 2010 SRC* (pp. 5:1–5:9. https://doi.org/10.1145/1839707.1839713.

Han, L., Kashyap, A., Finin, T., Mayfield, J., Weese, J. (2013). *UMBC EBIQUITY-CORE: semantic textual similarity systems* (Vol 44). Atlanta.

Jarke, M., Jeusfeld, M., Quix, C. (2014). Data-centric intelligent information integration from concepts to automation. *Journal of Intelligent Information Systems*, *43*(3), 437–462.

Kalja, A., Haav, H.M., Robal, T. (2014). *Databases and information systems VIII: selected papers from the eleventh international baltic conference, DB&IS 2014*. IOS Press.

Kaufmann, E., & Bernstein, A. (2010). Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Web Semantics: Science, Services and Agents on the World Wide Web*, *8*(4), 377–393. https://doi.org/10.1016/j.websem.2010.06.001.

Kettouch, M.S., Luca, C., Hobbs, M. (2015a). An interlinking approach based on domain recognition for linked data. In *2015 IEEE 13th International conference on industrial informatics (INDIN)* (pp. 488–491). IEEE.

Kettouch, M.S., Luca, C., Hobbs, M., Fatima, A. (2015b). Data integration approach for semi-structured and structured data (linked data). In *2015 IEEE 13th international conference on industrial informatics (INDIN)* (pp. 820–825). IEEE.

Kettouch, M.S., Luca, C., Hobbs, M., Dascalu, S. (2017). Using semantic similarity for schema matching of semi-structured and linked data. In *2017 Internet technologies and applications (ITA)* (pp. 128–133). https://doi.org/10.1109/ITECHA.2017.8101923.

Koffina, I., Serfiotis, G., Christophides, V., Tannen, V. (2006). Mediating RDF/S queries to relational and XML sources. *International Journal on Semantic Web and Information Systems*, *2*(4), 68–92.

Le-Phuoc, D., Nguyen-Mau, H.Q., Parreira, J.X., Hauswirth, M. (2012). A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web*, *16*, 42–51. https://doi.org/10.1016/j.websem.2012.06.003.

Lopez, V., Uren, V., Motta, E., Pasin, M. (2007). AquaLog: an ontology-driven question answering system for organizational semantic intranets. *Web Semantics: Science, Services and Agents on the World Wide Web*, *5*(2), 72–105. https://doi.org/10.1016/j.websem.2007.03.003.

Lopez, V., Fernández, M., Motta, E., Stieler, N. (2011). Poweraqua: supporting users in querying and exploring the semantic web. *Semantic Web*, *3*(3), 249–265.

Lopez, V., Unger, C., Cimiano, P., Motta, E. (2013). Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, *21*, 3–13. https://doi.org/10.1016/j.websem.2013.05.006.

Macura, M. (2014). Integration of data from heterogeneous sources. *Computer Science*, *15*(2), 109–132.

Morbidoni, C., Le Phuoc, D., Polleres, A., Samwald, M., Tummarello, G. (2008). *The semantic web: research and applications, lecture notes in computer science* Vol. 5021. Berlin: Springer.

Nguyen, K., Ichise, R., Le, B. (2012). SLINT: a schema-independent linked data interlinking system. *Ontology Matching*, 1–12.

Pánek, O. (2015). Integration of heterogeneous data sources based on a catalog of master entities. Diploma thesis, Czech Technical University, in Prague.

Pfaff, M., & Krcmar, H. (2014). Semantic integration of semi-structured distributed data in the domain of IT benchmarking - towards a domain specific ontology. In *Proceedings of the 16th international conference on enterprise information systems* (pp. 320–324). https://doi.org/10.5220/0004969303200324.

Ramis, B., Gonzalez, L., Iarovyi, S., Lobov, A., Martinez Lastra, J., Vyatkin, V., Dai, W. (2014). Knowledge-based web service integration for industrial automation. In *EEE International conference on industrial informatics* (pp. 733–739). IEEE, https://doi.org/10.1109/INDIN.2014.6945604.

Talukdar, P.P., Ives, Z.G., Pereira, F. (2010). Automatically incorporating new sources in keyword search-based data integration. In *Proceedings of the 2010 ACM SIGMOD international conference on management of data* (pp. 387–398). ACM.

Umbrich, J. (2012). A hybrid framework for querying linked data dynamically. PhD thesis.

Usbeck, R., Ngonga Ngomo, A., Bühmann, L., Unger, C. (2015). Hawk-hybrid question answering over linked data. In *12th extended semantic web conference*.

Verborgh, R., Steiner, T., Van de Walle, R., Gabarro, J. (2015). Linked data and linked apis: similarities, differences, and challenges. In Simperl, E., Norton, B., Mladenic, D., Della Valle, E., Fundulaki, I., Passant, A., Troncy, R. (Eds.) *The semantic web: ESWC 2012 satellite events* (pp. 272–284). Berlin: Springer.

Vincini, M., Beneventano, D., Bergamaschi, S. (2013). Semantic integration of heterogeneous data sources in the momis data transformation system. *Journal of Universal Computer Science*, *19*(13), 1986–2012.

Zhao, L., & Ichise, R. (2013). Integrating ontologies using ontology learning approach. https://doi.org/10.1587/transinf.E96.D.40.

Ziegler, P., & Dittrich, K.R. (2007). Data integration-problems, approaches, and perspectives. In *Conceptual modelling in information systems engineering* (pp. 39–58). Springer.