Distributed and Parallel Databases (2021) 39:891–937
https://doi.org/10.1007/s10619-021-07321-6

Semantic-based Big Data integration framework using scalable distributed ontology matching strategy

Imadeddine Mountasser[1] · Brahim Ouhbi[1] · Ferdaous Hdioud[2] · Bouchra Frikh[2]

Accepted: 6 January 2021 / Published online: 29 January 2021

Abstract
Nowadays, Big Data management has become a key basis for innovation, productivity growth, and competition. The correlated exploitation of data of this magnitude remains primordial to discover valuable insights and support decision making for domains of major interest. Furthermore, despite the complex aspects of Big Data environments, users are usually looking for a unified and appropriate view of this huge and heterogeneous data, to support the extraction of reliable and consistent knowledge. Thus, Big Data integration mechanisms must be considered to provide a uniform query interface, to mediate across large datasets and provide data scientists with a consistent integrated view suitable for analytical exploitations. Thus, this paper presents a semantic-based Big Data integration framework that relies on large-scale ontology matching and probabilistic-logical based assessment strategies. This framework applies optimization mechanisms and leverages parallel-computing paradigms (Hadoop and MapReduce) using commodity computational resources, to efficiently address the Big Data challenges and aspects. Several experiments were conducted and have proven the efficiency of this framework in terms of accuracy, performance, and scalability.

Keywords Big Data integration · Semantic-based integration · Distributed ontology matching · High performance computation · MapReduce paradigm · Probabilistic logical processing

1 Introduction

The rapid evolution of the Internet and the technological development of services have led to a proliferation of various heterogeneous data sources in a broad array of domains. Hence, extrapolating relevant insights on a specific domain (e.g. transport, energy, etc.) requires the correlated exploitation of multiple large data sets

✉ Imadeddine Mountasser
imountasser@gmail.com

Extended author information available on the last page of the article

(high volume), that are continuously collected from disparate sources (high velocity) by different data acquisition devices, with different formats (high variety) and various data quality levels (obscure veracity). Therefore, managing data of this magnitude (i.e. Big Data) is usually confronted with well-known challenges and practical obstacles [14, 43], which may undermine the effective building of data-driven applications devised to support decision-making and problem resolution. For instance, urban traffic control and optimization strategies require intelligent systems that harness information from diverse cross-domain data sources (i.e. traffic sensors, weather, social networking, etc.) to provide a deeper understanding of complex transportation cases, hence, supporting more coordinated and smarter planning and modeling of the transport sector.

To achieve this, the different engaged stakeholders (i.e. data scientists, planners, decision-makers) usually seek a unified and integrated view of these available data to support the extraction of reliable and consistent knowledge that would help them make better and smarter strategic and operational decisions [15, 43, 93]. Hence, given its important benefits in incorporating and combining multiple data sources, and considering the complex aspects of Big Data environments, Big Data integration has attracted growing interest [11, 14, 27]. Specifically, this operation is accentuated by the constraints related to the inherent Big Data characteristics, as they increase the complexity and undermine the real-time requirements of the integration process [77]. Moreover, the integration process is also affected by the heterogeneity and interoperability issues (i.e. the structural and the semantic conflicts between integrated data) [52]. Besides, the unpredicted character of the structure and content of data sources entails some agility of the data integration process to handle the dynamic aspects of the data sources. Furthermore, the integration process should, in particular, ensure the reliability and credibility of the integrated data in order to provide solid ground for their related exploitation scenarios, especially in large-scale environments.

In fact, several studies have been conducted to resolve the Big Data integration issue [11, 37, 45]. Among them, different approaches have been entirely oriented toward ontology-based data integration [13, 18, 19]. These approaches have claimed that ontologies grant an explicit and machine understanding conceptualization of a domain and offer a semantic model of the data sets under integration [19, 70]. By the same token, [20, 71] have stated that ontologies provide solutions to data heterogeneity and interoperability issues by allowing a cross-cutting meaning of terms, at various levels of formalization, and relationships between them. However, such data integration approaches usually rely on the nature of the data source and only tackle the variety aspect of Big Data. Furthermore, these data integration solutions do not consider the scalability, availability and performance aspects of the integration process, especially in large-scale context.

In this regard, this paper proposes a semantic-based Big Data integration framework that relies on a large-scale ontology matching strategy. This framework aims to overcome the Big Data challenges while ensuring scalability, availability, reliability and high performance. The scalability refers to the support of continuously increasing volumes of data, where a significant expansion or optimization of storage and computational resources (i.e. machines) is viable [64], whereas the availability

implies the ability to access and integrate data, even when machines failures occur. The reliability reflects the capacity of the integrated data to support its effective exploitation and avoid biased and wrong conclusions. In light of the above, this framework consists of three main steps:

1. Generating local ontologies from each data source: This step allows unifying the data model for each data source and addresses data heterogeneity and interoperability issues. It grants, thus, the ability to gather and integrate data from heterogeneous data sources despite their unpredicted content and form.
2. Hybrid large-scale ontology matching (HLSOM): This step integrates and combines these local ontologies in order to construct a shared global ontology, considered as the uniform view of the integrated data sources.
3. Probabilistic-logical based assessment: This step consists of handling the conflicts that arise during the preceding step (i.e. HLSOM), which may improve the reliability of the integration framework.

Fundamentally, this semantic-based integration framework uses optimization mechanisms and leverages parallel-computing paradigms (i.e. Hadoop and MapReduce), using commodity computational resources, to perform high computational complexity operations, efficiently. This will promote the scalability and availability aspects of this framework. On the other hand, this parallelism-enabled architecture allows performance gain during the integration process and overcomes machines failures issues. Ultimately, it is worth mentioning that the aim of this paper is beyond the development of new ontology learning approaches and will concentrate mainly on the implementation of the HLSOM process and the probabilistic-logic based assessment process.

The remainder of this paper is organized as follows. The related works drawing from the existing state-of-the-art of the associated fields, such as ontology-based data integration, ontology learning, and ontology matching approaches, are mainly reported and discussed, in Sect. 2, to highlight their weaknesses. Section 3 presents the overall architecture of the proposed integration framework and the detailed exposition of its modules. Section 4 presents and discusses the experimental results, in terms of the reliability and the performance of the proposed framework, on different datasets. The paper concludes with proposals for future works in Sect. 5.

2 Related work

2.1 Ontology-based data integration

Ontology-based data integration (OBDI) is an active field of research that has increasingly confronted with new challenges under the Big Data environments [13]. In fact, the ontology-based data integration as a Global-As-View (GAV) semantic-based integration approach [45], resides on the exploitation of ontologies to efficiently combine data or information from various heterogeneous sources, where the

global schema of data is replaced by the conceptual model of the domain, formally designed through a domain ontology [10, 63]. To this end, three main strategies are implemented, namely the single ontology, the multiple ontologies and the hybrid strategies [85].

For the single ontology-based integration approach, all data sources are semantically mapped to a global ontology as a shared vocabulary [16, 85]. This approach requires that all sources share the same view of the domain. However, it is time-consuming and difficult to devise, especially in a large scale context, as it requires a good understanding of the disparate data and may need an expert intervention to build a valid and consistent knowledge base (i.e. shared vocabulary). Moreover, this approach is not suitable for dynamic environments with changing data sources, as this may imply changes in the global ontology, affect the exiting mapping on data sources and alter the overall conceptualization of the domain, which is impracticable in Big Data scenarios. The multiple ontologies based approach, by contrast, aims to avoid expert intervention and allows the integration of data from different domains [16, 54]. First, this solution implies that each data source would apply a semantic lifting operation to describe the local data in terms of its own ontology. Then, the ontologies generated are mapped to each other by an inter-ontology mapping susceptible to be modified w.r.t. the dynamic change of data sources. Accordingly, this may allow multiple views on data inasmuch as that only a single ontology can be used for querying and exploiting the integrated knowledge base. Besides, the hybrid approach uses multiple local ontologies that would be aligned to a top-level shared ontology instead of defining correspondences between the other local ontologies [2, 16]. This top-level shared ontology may be a pre-existing domain-specific ontology or can be created from scratch. Hence, such an approach allows adding new data sources without altering existing mappings between local ontologies or the top-level ontology.

2.2 Ontology-based Big Data integration

Ontology-based data integration mainly relies on a three-pillar architecture, formed by the data sources, the global ontology, and the mapping between them. Early OBDI approaches exploit the single ontology strategy to represent available data sources. Vandecasteele and Napoli [87] exploit a geographical ontology for the correlation and aggregation of sensor data used to analyze abnormal ship behavior. By the same token, Curé et al. [18] propose an OBDI framework, in which NoSQL databases (MongoDB and Cassandra) are mapped to an OWL[1] ontology as target schema. Similarly, Kiran and Vijayakumar [41] have developed a semantic-based integration system for the HBase column-oriented NoSQL data store. Besides, Jirkovskỳ and Obitko [40] propose a semantic-based Big Data integration approach for the industrial automation domain. This approach deals with structural and semantic heterogeneity by the semi-manual construction of a shared ontology from

[1] http://www.w3.org/TR/owl-features/.

the preprocessed data sources. Then, the authors adopt some ontology matching systems to find correspondences between entities across all data sources, which ensures knowledge sharing.

Furthermore, Bansal and Kagemann [7] propose a semantic extract, transform and load (ETL) framework for Big Data integration. This framework produces a semantic model of the data sources under integration, then, generates semantically linked data (RDF triples), in conformity with the data model. For that, data is normalized and cleaned, in the Transform step, then their schema is analyzed manually. Each schema is mapped to an existing domain-specific ontology or to an ontology that has to be created from scratch. For data sources of multiple domains, multiple ontologies are required and ontology matching tools are invoked to specify common and related data fields. Likewise, Daraio et al. [19] exploit this approach to integrate heterogeneous data sources, including big scholarly data to support the assessment of research and to develop scientific policy models. García et al. [33] also propose an OBDI approach for web analytics in e-commerce. This approach exploits a single ontology as mediated schema to collect, integrate, and store web analytics data, from several sources of popular and commercial digital footprints. For that, several build-in wrappers are invoked to realize the mappings between data sources (RDB, JSON files, CSV files, etc.) and this ontology.

Additionally, Li et al. [46] propose a framework for ontology-based top-k global schema generation as a schema integration solution. The ontology provides a high-quality global schema from several relational schemas, with moderate user involvement, as a base-merging model. The strategy consists of converting each local relational schema to a local ontology, then, an ontology merging procedure is performed and top-k global ontologies are generated. Finally, the user selects a unique merged global ontology, which will be converted to a global schema. Abbes and Gargouri [1] propose an approach to build a modular ontology for Big Data integration in conjunction with the characteristics of Big Data. This approach leverages a MongoDB NoSQL database and takes advantage of modular ontologies by wrapping each data source to a MongoDB database, then, generating local ontologies from these databases by specifying a set of transformation rules to map MongoDB constructs to OWL ontologies. Finally, they propose a matching algorithm that allows the ontology modules to be automatically composed into a global ontology. Nadal et al. [58] propose a semantic-based Big Data integration by introducing a structured ontology that allows modeling and integrating evolving data from several data providers (using REST APIs). This approach governs the integration process by annotating it with information about the schema of the evolving sources. First, data sources are accessed via wrappers, which expose a relational schema, depict its RDF-based representation in the ontology, and define its appropriate mappings. Then, a query answering algorithm, that leverages the proposed ontology, is defined in order to translate a restricted subset of SPARQL queries over the ontology to queries over the sources.

From another perspective, ontology-based data access (OBDA) solutions were also adopted to integrate dynamic and voluminous data. Zamboulis et al. [92] propose an ontology-assisted data restructuring and transformation of XML heterogeneous data sources through using a semantic bridge between them. They also present

ontology-based data access to relational data sources to support the integration of heterogeneous medical databases under a predefined ontology. Moreover, Mezghani et al. [55] propose a semantic Big Data architecture as a Wearable KaaS (Knowledge as a Service) platform for smart management of scattered heterogeneous correlating medical data coming from wearable devices and to deal with their heterogeneity and scalability challenges. The main aim of this work is to provide a wearable healthcare ontology that ensures semantic interoperability and enables the aggregation of distributed heterogeneous data from wearables to make accurate health-related decisions and to generate new valuable information. Besides, Santipantakis et al. [70] propose the OBDAIR integration framework that follows the multiple ontologies approach to perform distributed retrieving, integrating, and reasoning tasks. The framework consists of creating multiple ontologies for the integrated data sources, extracting correspondences between concepts and individuals of each ontology unit, making RDF mappings of the relational databases using OBDA, and an ontology-mediated distributed data access for multiple sources cases.

Big Data integration is an important and complex process that is extremely affected by the challenges and aspects of Big Data. In the light of the above, several approaches have been conducted to resolve the integration issue by leveraging different ontology-based data integration strategies (i.e. single, multiple and hybrid). However, these data integration approaches suffer from several weaknesses, given the Big Data environment and its related requirements: They rely on the category of the data source, the ontology building procedure, and how to find the correspondences between them; and focus only on addressing the variety aspect of Big Data; They do not study the agility of data sources and their unpredicted structure and content (i.e. structured, unstructured or semi-structured); They do not use mechanisms to handle voluminous datasets; and they do not ensure the veracity of the integrated data. Furthermore, they do not consider the scalability, availability, and performance aspects of the integration process, which are especially essential in Big Data environments. Eventually, They adopt semi-automatic procedures to extract the mapping between local ontologies, which is certainly not suitable for large-scale context.

For that, the proposed semantic-based Big Data integration framework leverages the hybrid ontology-based integration approach and relies on a large-scale ontology matching strategy that creates the mapping between the created ontologies (from data sources) and the global ontology. This framework aims first to unify the data model for each data source and addresses data heterogeneity and interoperability issues, which allows integrating data from heterogeneous data sources despite their unpredicted content and form (i.e. Variety). It implements different optimization mechanisms and leverages parallel-computing paradigms to efficiently perform high computational complexity operations, which may tackle the Velocity challenge of Big Data, ensure that continuously increasing volumes of data (i.e. Volume) are supported where a significant expansion or optimization of storage and computational resources is viable (i.e. Scalability), and guarantee data integration even when machine failures occur (i.e. Availability). Moreover, the framework applies an assessment mechanism to ensure the reliability and credibility of the integrated data (i.e. Veracity) and to support its effective exploitation (i.e. Value). Accordingly, the shared global ontology is built by performing a large-scale ontology matching

process over local ontologies. Thus, it is more interesting to further investigate the ontology matching field in terms of the accuracy and performance aspects.

2.3 Large-scale ontology matching approaches

Fundamentally, this study would go beyond basic matching strategies (i.e. element-based and structure-based strategies) [30, 61, 76] and focus on strategies that optimize the matching process in large-scale context (such as reduction of matching space strategies and parallel matching strategies) while making an arrangement between accuracy and performance.

2.3.1 Reduction of matching space strategies

2.3.1.1 Pruning strategies Ontology matching algorithms may deal with large-scale ontologies, containing a huge number of entities. Therefore, pruning strategies can dynamically ignore parts of those ontologies and avoid comparing them during the matching process [30, 65]. In other words, on the basis of a further processing or a specified threshold, the process tries to correctly identify matching entity pairs by excluding highly dissimilar ones from the matching phase, which consequently improves the performance of the matching process by limiting the pairwise comparison of the whole entities.

Quick ontology matching [24] adopts this strategy by first extracting matching candidates based on entity labels, then evaluating structural properties only for highly similar ones gathered from the first constraint. Likewise, AROMA [21] tries to learn association rules by measuring, at each stage, the maximum implication intensity obtained by learning more specific rules. If the intensity value is below a threshold more specific rules are eliminated, which avoids any further comparisons. The approach proposed by Peukert et al. [62] uses filter operators within the matching process to eliminate dissimilar element pairs having a similarity below some threshold from intermediate match results. Similarly, Anchor-Flood strategy [73] uses anchors reduction mechanisms to handle large ontologies. First, the system analyzes the neighbors of each anchor (i.e predefined correspondence) and builds small segments candidates to be matched. Then, it iteratively explores neighbors until no new candidate pairs are founded or the collected entities are fully explored, which is the same concept adopted by Wang et al. [89]. Further works like LogMap [39] focus on index-based pruning by firstly indexing all the entities using their labels and their URIs, then, extracting matching candidates when each pair of entities is indexed together for different ontologies. Comparably, ServOmap [6] dynamically constructs an inverted index for each entity according to the features it holds. After that, the system computes the lexical-based similarity between entities using the previously built indexes. Finally, it computes context-based similarity only for those who have not been yet matched to any other entities during the first phase.

2.3.1.2 Partition-based matching strategies Since large-scale ontology matching issue is expressively related to the size and the complexity of matched ontologies,

partition-based strategies aim to perform splitting operations to obtain exhaustive and non-overlapping sub-ontologies with the intention to perform matching by parts and aggregate the independent partial matching results. Accordingly, partition-based matching involves four main stages: (i) partitioning the input ontologies into a set of small disjoint sub-ontologies, (ii) comparing these partitions with each other to determine those that worth to be fully matched, (iii) applying matching algorithms to determine local correspondences between partitions and (iv) aggregating the partial alignments to form the overall matching result.

COMA++ [23] conducts a partition-based matching strategy by first identifying fragments of two schemas and selecting the most similar ones to match. Likewise, Falcon-AO [36] proposes a divide-and-conquer strategy involving a structure-based agglomerative clustering approach to partition input ontologies into relatively small disjoint blocks, taking into account the internal cohesion of a block. Then, using predefined anchors, the system selects candidate blocks pairs to be further matched. Algergawy et al. [3] propose a clustering-based approach that consists of representing input ontologies as directed acyclic graphs, then, performs a structure-based clustering algorithm to split input ontologies into a set of disjoint sub-graphs in such a way that structurally similar nodes are placed in the same cluster while the nodes of different clusters are structurally dissimilar. Later, each cluster of the input ontologies is represented as a cluster of documents and the approach uses both the Vector Space Model and TF-IDF approaches to efficiently compute the similarity between cluster pairs. Finally, having similar clusters, a specific matching algorithm is applied to completely match the elements inside similar clusters. By contrast, SeeCOnt [4] aims to reduce the complexity of comparisons between entities within clusters by only comparing their clusters' seeds. The approach represents input ontologies as labeled directed graphs, then, a ranking function that quantifies the node importance is exploited to rank ontology entities. Those having the highest-ranking are selected to construct the clusters' heads. Later, remaining entities are specifically assigned to their appropriate clusters based on a membership function. Eventually, the Falcon-AO matching system is adapted to perform the matching process.

Unfortunately, the above-mentioned partition-based matching systems have several concerns regarding the efficiency of the partitioning processes, and the mechanisms adopted to identify the similar partitions to match, which may certainly impact their accuracy. In addition, these works do not consider non-taxonomic relations from input ontologies during the graph creations, which may affect the structural knowledge encoded on them and lead to create trimmed graphs. Furthermore, these works try to perform large-scale ontology matching without considering performance concerns, in terms of scalability and execution-time.

2.3.2 Parallel matching strategies

In fact, some research has been conducted in devising parallelism with regard to ontology matching issues [5, 35]. Specifically, the parallelism strategies apply two main kinds of parallel matching (inter and intra-matcher parallelization) to improve performance by using parallel and distributed infrastructure. The inter-matcher parallelism approach allows parallel execution of independently matching operations

(i.e. matchers) on a parallel platform (multiple cores of a single computing node or multiple nodes). However, the parallelization is limited by the number of independent matchers and their required memory when loading the overall ontology, which decreases the performance and expands the overall execution time of the ontology matching system. Also, matchers of dissimilar computational complexity may have largely different execution times, which harms and limits the achievable speedup, so that the slowest matcher determines the overall execution time. The intra-matcher parallelism approach, on the other hand, is more volatile and leads to several finer matching tasks with limited computational complexity, that can be performed in parallel with reduced memory requirements per task. The approach can be applied for sequential as well as independently executable matchers and can also be combined with inter-matcher parallelism scenarios. Therefore, Amin et al. [5] decouple the performance and accuracy aspects by implementing data parallelism over parallelism-enabled platforms for effectiveness-independent performance-gain during ontology matching. It aims to split complex ontologies into smaller subsets in order to preserve the parsing effort for future matching requests for the same ontologies. Nevertheless, these parallel-based approaches ignore and do not manage the node failure issues related to parallel platforms which affect the system performance whenever failed tasks are re-executed. Besides, they implement parallelism mechanisms for data storage level rather than computational level, thus, not benefiting from the computational advantages offered by parallel platforms.

Overall, several works were conducted in the large-scale ontology matching field. Even though, they do not consider scalability and availability aspects. They often carry out a matching process with high computational complexity, without taking performance aspects into consideration. On the other hand, the assessment and monitoring of the quality of the ontology matching process are usually impractical and mostly overlooked during these processes. For these reasons, using new optimization mechanisms and advanced parallel-computing paradigms on commodity computational resources, need to be more employed to achieve high-performance ontology matching.

3 Scalable semantic-based Big Data integration framework

3.1 Framework overview

Big Data comes from every imaginable source: user-generated data [8], machine-generated data [74], and official data (i.e. governmental and public authorities data) [84]. Indeed, this explosive growth of data is entirely reliant on different technologies, such as the Internet of Things, Cloud Computing, Internet, mobile devices, and various sensor technologies [95]. For these reasons, this paper introduces a semantic-based Big Data integration framework (SBDI), that considers Big Data features (4Vs) and aspects (i.e. scalability, availability, and high-performance) to support the extraction of reliable and consistent knowledge. This framework has a corpus formed by heterogeneous data sources as input and an OWL (Ontology Web Language) ontology as target. Choosing OWL as the ontology representation

language is determined because it is the recommended standard to represent ontologies according to the World Wide Web Consortium (W3C).[2] Thus, the integration framework (SBDI) is devised using three modules (see Fig. 1).

The local ontology building module (Sect. 3.2) which constructs local ontologies from each data source. This module (i.e. module 1 in Fig. 1) is responsible for gathering data from heterogeneous data sources, despite their unpredicted content and their type, and converting it to a common representation (i.e. OWL ontology) while preserving the autonomy of the initial sources. To this end, it harnesses several ontology learning strategies that differ depending on the nature of the input data sources. This semantic choice allows a cross-cutting meaning of terms, at various levels of formalization, and relationships between them. These local ontologies are stored as DL axioms in order to be mapped by the set of alignments derived through the HLSOM module. Overall, the local ontology building module provides a semantic data model that homogenizes data under integration, which resolves data heterogeneity and interoperability issues, and handles the 'Variety' challenge of Big Data.

The hybrid large-scale ontology matching module (HLSOM) (Sect. 3.3) is a three-layer module (i.e. module 2 in Fig. 1), which aims to extract the set of alignments between the local ontologies under integration. It is considered as the core of the semantic-based integration framework since it unveils the different mappings between the diverse entities of these local ontologies (i.e. classes and relationships between them), which are essential to construct the uniform view of the integrated data sources (i.e. the shared global ontology). Fundamentally, the HLSOM module consists of the resource extraction layer, which allows the parsing of local ontologies and the creation of personalized data subsets, with scalability-friendly data structures (i.e. lists and hashmaps), where each one is dedicated to satisfying some specific needs of the upcoming processes. Then, this layer serializes and persists these subsets in the HBase data store, that offers random real-time read/write access and allows parallelizing and distributing processing tasks over available storage resources [34]. Accordingly, this parallelism-based storage mechanism reduces the computational complexity of the remaining layers and improves their performance (see Sect. 3.3.1). Besides, this layer slightly handles the data and/or the schema evolution by dealing with each source separately from others and preserving their autonomy. For that, each evolution is propagated from the input ontology and applied, as updating operations, to their associated persisted data subsets. Overall, through this layer, the semantic-based integration framework tackles the 'Velocity' challenge of Big Data and improves the integration process performance, in terms of speedup and scalability.

Furthermore, in the ontology clustering layer, HLSOM performs a parallel clustering strategy to split the stored local ontologies (i.e. their duplicates) into a set of disjoint partitions, while preserving their internal encoded knowledge. To this end, the clustering strategy exploits certain metrics to rank the different ontologies entities and assign them to their appropriate clusters. Indeed, this operation allows the matching tasks to be performed by parts instead of matching the whole ontologies.

[2] https://www.w3.org/.

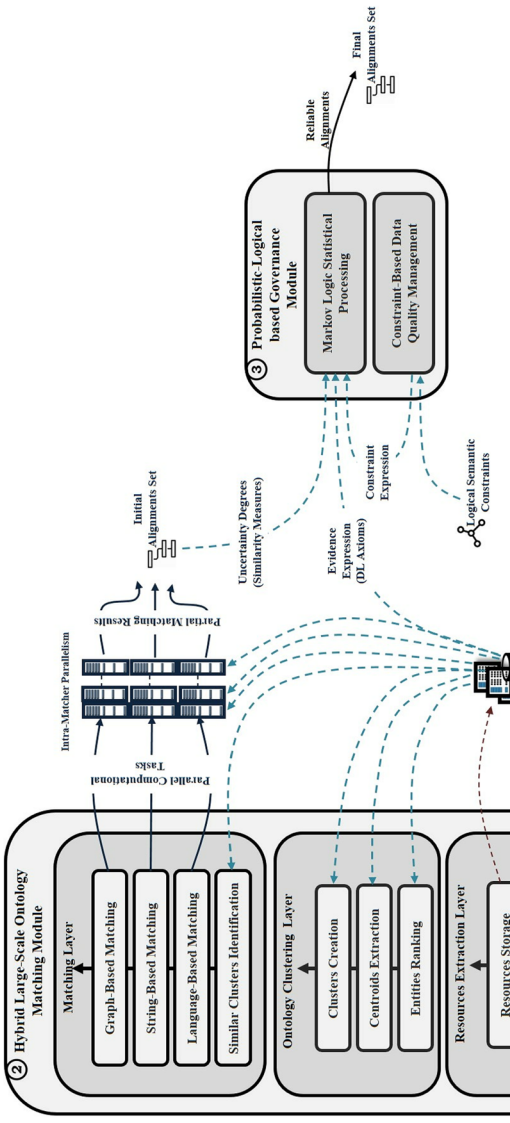


Fig. 1 Semantic-based Big Data integration framework global architecture

For that, advanced parallel-computing paradigms (i.e. MapReduce) are adopted to distribute these partial matching tasks over several commodity computational resources. Additionally, HLSOM performs a specific sequential combination of several matching strategies (within the matching layer), to provide more accurate ontology alignments by following the paradigm of matching space reduction. This layer incorporates, first, an initial stage that identifies semantically similar clusters, considered as candidates to be independently matched. This stage aims to ignore dissimilar clusters before performing the matching strategies which help improve the accuracy and performance of the matching layer and reduce its overhead. Moreover, each matching strategy relies on the intra-matcher parallelism approach (Sect. 2.3.2) under the Hadoop framework as a data parallelism-enabled platform and using the parallel programming technique (MapReduce) over distributed computational resources. Hence, this leads to several finer matching tasks, with limited computational complexity, that can be performed in parallel with reduced memory requirements per task. This parallel implementation impacts the real-time requirements of the Big Data integration process and promotes its scalability. Eventually, all the individual independent partial correspondences derived from each matching strategy are aggregated to form the final alignments set.

The probabilistic-logical based assessment module (Sect. 3.4) is dedicated to the refinement of the extracted alignments vis-a-vis the encoded knowledge of ontologies. For that, this module leverages the Markov logic network paradigm as a probabilistic-logical paradigm to reduce the incoherence and conflicts that arise during the HLSOM module. Thus, this module exploits the stored OWL-DL axioms derived from the created local ontologies (Sect. 3.2), to describe the encoded knowledge of these ontologies. Notably, OWL-DL supports the maximum expressiveness while retaining computational completeness and decidability. Overall, this module addresses the quality and accuracy of the semantic-based integration framework, which supports the appropriate and efficient exploitations of the shared global ontology.

3.2 Local ontology building module

The local ontology building module is responsible for transforming input data sources into local ontologies (Module 1 in Fig. 1). It provides a semantic data model that homogenizes data under integration through exploiting several ontology learning mechanisms to gather data from heterogeneous data sources, despite their content and their nature, and transform them into a common representation, i.e. OWL ontology, while preserving their autonomy. In fact, the ontology as a knowledge representation form plays a significant role in the promotion and deployment of the semantic web. It describes the different aspects of knowledge of a specific domain explicitly and formally. It provides a sound semantic ground and a cross-cutting meaning of terms (i.e. entities labels), at various levels of formalization, by defining concepts (i.e. classes) and named relationships linking concepts and their instances. This makes ontologies suitable to address the data heterogeneity and interoperability issues of overlapped domains. Nevertheless, the pervasive use of ontologies in

information sharing and knowledge management requests efficient and effective strategies for ontology development. Hence, the ontology learning (OL) paradigm [49] consists of the integration of a multitude of disciplines to facilitate the construction of ontologies [82]. In fact, OL refers to the automatic or semi-automatic discovery and creation of ontological knowledge using machine learning techniques. It also mitigates human-introduced biases and inconsistencies [94].

Most strategies of ontology learning are usually dependent on the nature of the data sources being exploited. For structured and semi-structured data (e.g. databases, JSON, and CSV files), the proposed integration framework follows the basic conversion rules from the relational databases to OWL [51], using their implementations in the RDBtoOnto tool [12]. In regards to unstructured data, such as text files, the proposed integration framework follows the ontology learning mechanisms provided by [25, 26]. Besides, ontology learning approaches which take NoSQL databases as input are extremely relative to the family of the input data store. They specify the mapping rules between the data store and the ontology representation language (OWL), to extract ontology's components (i.e. concepts, relations, axioms, domains, and ranges). In general, this integration framework follows the transformation rules from NoSQL data store to OWL ontology proposed by [1, 18, 41], given the data model of the data store. After all, these local ontologies are stored as OWL-DL axioms in order to be mapped by the set of alignments derived through the HLSOM module. They are also used to evaluate this latter via the probabilistic-logical based assessment module since they encompass the encoded knowledge of local ontologies.

3.3 Hybrid large-scale ontology matching module

3.3.1 Resources extraction layer

In the first stage, the resource extraction layer uses Jena[3] framework to parse the stored local ontologies and generate smaller, finer, and simpler personalized data subsets, with scalability-friendly data structures (e.g. lists and hashmaps) according to the requirements of the upcoming processes of the HLSOM module (such as ontology clustering, similar clusters identification and individual matching of similar ones) (Algorithm 1). This will address the memory issues experienced on some related works during the execution of ontology matching tasks (i.e. matchers) because they load the whole ontology as a resource even if they only require specific information from it. Next, this layer serializes and persists these data subsets in the HBase column-oriented data store which both offers random real-time read/write access and allows parallel processing tasks to be distributed over available storage resources [34].

As we can notice in the upcoming sections, this layer forms the nucleus around which the proposed integration framework is built, as it creates personalized data

[3] https://jena.apache.org/.

resources for each specific processing. For example, the graph-based matching strategy will load only the data resources that provide the list of nodes with their neighbors and their cotopic distances. In addition, this layer calculates all the metrics required (e.g. centralities measures, similarities, etc.) and encapsulates them for eventual exploitation. By the same token, when a new data source is introduced, the resource extraction layer generates the same data resources. Likewise, if new requirements are raised due to new specific processing, the resource extraction layer will again parse the input ontologies to generate and encapsulate the resources needed for the task requested. Moreover, this layer slightly handles the data and/or the schema evolution by dealing with each source separately from others and each evolution is propagated from the input ontology and applied, as updating operations, to their associated persisted data subsets.

Accordingly, for each processing, this layer not only preserves the repetitive parsing effort that would be conducted on the same ontologies but also provides the agility to combine multiple matching strategies and the flexibility to add new data sources to the integration framework. This layer also leverages parallelism-based storage mechanisms, which reduces the computational complexity of the remaining layers and improves their performance.

Algorithm 1: Resources Extraction Layer

```
   Input: Local Ontologies Collection O
   Output: Set of Resources for each Local Ontology
 1 Parsing input ontologies
 2 foreach collection O_i do
 3 |   Extract OWLClasses List(O_i)
 4 |   Extract Labels List (O_i)
 5 |   foreach OWLClass CL_j do
 6 |   |   Extract Neighbors HashMap of CL_j
 7 |   |   Extract Ascendants List of CL_j
 8 |   |   Extract Descendants List of CL_j
 9 |   |   Extract Instances List of CL_j
10 |   |   Compute Centrality Metrics of CL_j
11 |   end
12 end
13 Save Extracted Resources in HBase Data Store
```

3.3.2 Large-scale ontology clustering layer

Ontology matching is considered as a computationally-intensive process, usually performing a cartesian product of two candidate ontologies with quadratic computational complexity [86]. Hence, in large-scale ontology matching scenarios, partitioning input ontologies can be of a great benefit to building effective matching processes, as a strategy of reduction of the matching space (Sect. 2.3.1). This partitioning mechanism splits ontologies into a set of disjoint partitions and allows performing ontology matching tasks by parts. To this end, the HLSOM module adopts an entity-assignment parallel clustering approach to partition

large-scale ontologies into disjoint clusters taking into account their internal cohesion (see Fig. 2). First, the clustering strategy determines entities with the highest ranking as clusters centroids and assigns, then, the remaining entities to their convenient clusters according to some structural features (see Algorithm 2).

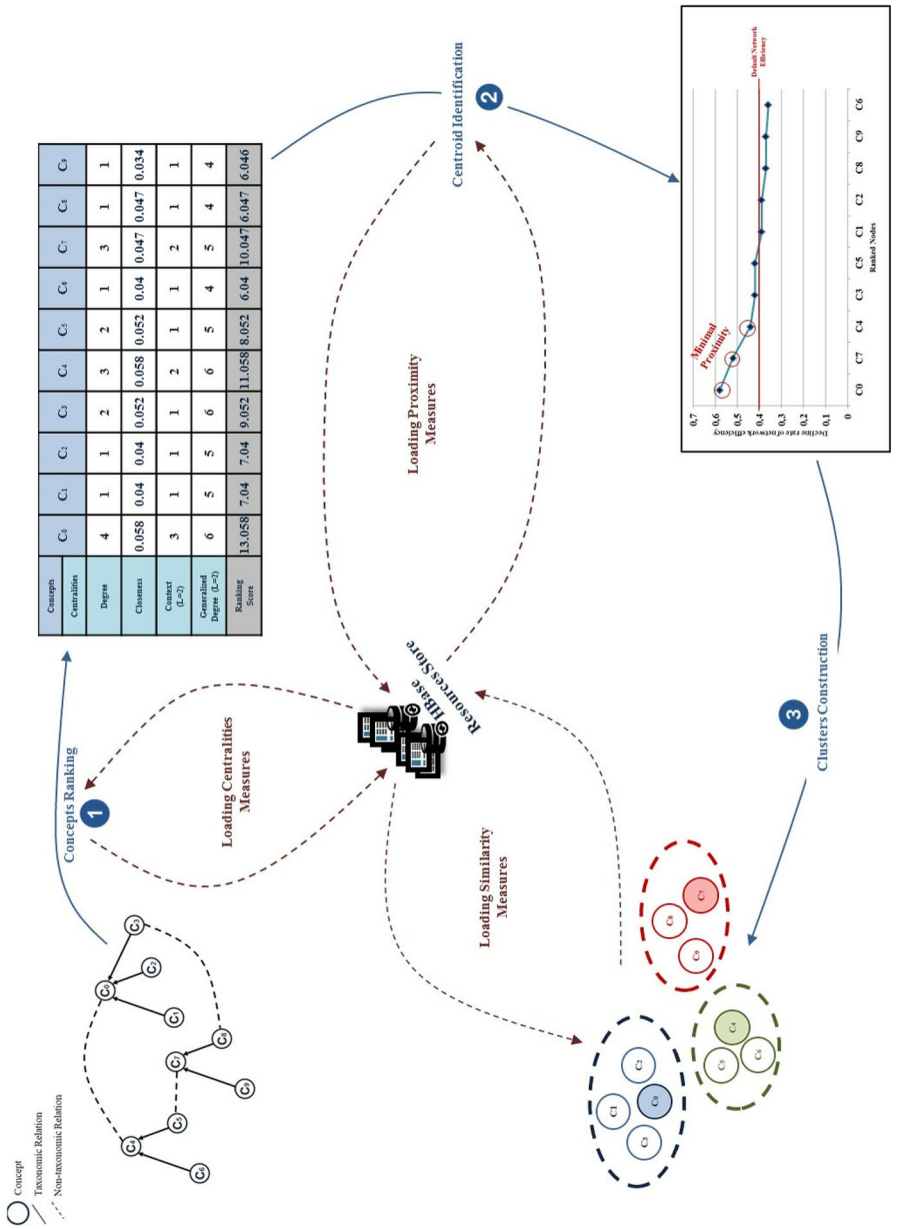
Entities ranking The current layer loads a copy of the stored ontologies, parses them and represents them as labeled directed graphs. Accordingly, the entities ranking phase aims to identify the nodes that are significantly important for each ontological graph. There are many metrics in the social network analysis field [9, 32, 42, 44, 48], that exploit the graph features to evaluate the importance of nodes while attributing a specific score to each node given its position in the graph [72]. Notably, the degree centrality metric, which is a local-based information metric, indicates the number of links directly connected to the node. It may, to some extent, describe the importance of nodes. Although, nodes having the same degree measure may not reflect the same importance in a complex network. Moreover, the betweenness centrality describes the capacities of nodes to control the information of networks. It aims to locate the bridge nodes, responsible for connecting important nodes and requires that information needs to be spread using the shortest way, which is not the case in most real networks. Yet, this metric is beyond the scope of our extent. Furthermore, the closeness centrality metric reflects the importance of nodes based on the distance between them. It is global-based information and a time-consuming metric since it measures the shortest path between any pair of nodes. Besides, the context centrality metric, which is a semi-local centrality metric evaluates the node importance using the node itself and its surrounding neighbors. Likewise, the relative generalized release of the degree centrality measures the node importance in the graph by redistributing the degree measure sum over its surrounding nodes, taking into account the global structure of the graph (i.e. the required depth) [17].

In view of the above, the HLSOM module, and especially the entities ranking layer, carries out a hybrid combination of local-based and global-based centrality metrics [53, 60] while making a tradeoff between accuracy and performance of the ranking operation. To this end, and based on the conducted experiments shown in Sect. 4, the centrality metrics adopted to perform entities ranking task are as follows:

Degree centrality This metric measures the number of links incident upon a node. In the case of a directed network, it calculates both the number of input and output links of a node. Indeed, the relationships between nodes can play a decisive role in concept ranking, since nodes having higher centrality degree are certainly more prominent than the others [53, 60].

$$Dc(N_i) = |Input_{Arcs}(N_i)| + |Output_{Arcs}(N_i)| \tag{1}$$

Closeness centrality This metric shows the importance of the closest nodes to the others in the graph. For each node, this metric calculates the cost to reach the others based on the shortest paths between the node and all other nodes in the graph. In general, the more the centrality value is high, the closer the node to all the others.



Concepts	C_3	C_1	C_7	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Centralities										
Degree	4	1	1	2	3	2	1	3	1	1
Closeness	0.058	0.04	0.04	0.052	0.058	0.052	0.04	0.047	0.047	0.034
Context (L=2)	3	1	1	1	2	1	1	2	1	1
Generalized Degree (L=2)	6	5	5	6	6	5	4	5	4	4
Ranking Score	13.058	7.04	7.04	9.052	11.058	8.052	6.04	10.047	6.047	6.046

Fig. 2 Large-scale ontology clustering layer

$$CLc(N_i) = \frac{1}{\sum_{N_j} Distance(N_i, N_j)} \tag{2}$$

where Distance(Ni, Nj) is the shortest path length between the nodes i and j in the graph.

Context centrality This metric takes into account the all-around structural (i.e. topological) context between nodes. It aims to calculate the number of surrounding neighbors (ancestors and descendants) of a node up to a specific depth (L). Having a large number of surrounding nodes compared to other nodes may reflect the importance of the node.

$$CXc(N_i) = \sum_{L} |Ancestor(N_i)| + |Descendant(N_i)| \tag{3}$$

Generalized degree centrality This metric aims to spread the degree centrality of node ancestors and descendants up to a specific level.

$$\begin{aligned} GDc(N_i, L) =& DegreeC(N_i) \\ &+ \sum_{L} GDc(Ancestor(N_i, L)) \\ &+ \sum_{L} GDc(Descendant(N_i, L)) \end{aligned} \tag{4}$$

where L is a predefined level of depth to calculate the metrics.

Besides, the ranking strategy incorporates the non-taxonomic relationships between nodes that are not tackled by existing partition-based matching approaches (Sect. 2.3.1). Indeed, including such relationships during the computation of centrality metrics may lead to more significant and pertinent entities ranking. In brief, the strategy combines the above metrics to perform an efficiently ranking process in terms of accuracy and performance. Furthermore, conversely to the related works, this step is improved by loading only the required resources, already created via the resources extraction layer, and by performing the ranking process in parallel over the distributed computational architecture, which enhances the performance of the HLSOM module. Thus, the relative ranking score for a given node is formulated as below:

$$\begin{aligned} Ranking_{score}(N_i) =& Dc(N_i) + Clc(N_i) \\ &+ CXc(N_i) + GDc(N_i) \end{aligned} \tag{5}$$

Centroid selection In fact, centroids selection is usually performed using these different approaches: random generation, buckshot approach, and ranking technique [69]. Therefore, this phase selects the most important nodes based on their ranking alongside their decline rate regarding the efficiency of the network (i.e. the graph). This latter reflects the network connectivity [48] in such a way that better connectivity implies better network efficiency. The network efficiency denotes that the damage of the network caused by deleting a node is equivalent to its importance. Thus, the network efficiency ρ_k can be formulated as follow:

$$\rho_k = \frac{1}{C(C-1)} \sum_{n_i \neq n_j \in N} \rho_{ij} \tag{6}$$

ρ_{ij} is the efficiency between n_i and n_j, $\rho_{ij} = \frac{1}{d_{ij}}$, d_{ij} is the shortest path between n_i and n_j, C and N are the number and the set of nodes without node k, respectively. Thus, the decline rate φ of network efficiency is defined as follow:

$$\varphi_k = 1 - \frac{\rho_k}{\rho_0} \tag{7}$$

ρ_k is the network efficiency affected by deleting the node k, whereas ρ_0 is the initial network efficiency. Having great φ_k means that the network connectivity is seriously destroyed which consequently reflects the importance of the removed node.

Afterward, the system calculates the structural proximity between them, so that only those having minimal proximity (i.e. far away nodes) are selected to construct clusters centroid. Let N_i, N_j be two nodes in a given ontological graph, the structural proximity between N_i and N_j is measured based on how closely they are related to their common super-class:

$$\begin{aligned} &Structural_{prox}(N_i, N_j) \\ &\quad = \frac{2 \times \max(depth(N_{ij}))}{\max(depth(N_i)) + \max(depth(N_j))} \end{aligned} \tag{8}$$

where N_{ij} is the common super-class of the nodes i and j in the graph.

Clusters Creation. This phase starts by affecting the direct surroundings nodes of each centroid to its corresponding cluster following the ontological graph topology (the graph architecture). Next, using the $Assign_{Sim}$ metric, it assigns the remaining nodes to their appropiate clusters. For that, each remaining node (i.e. entity label) is compared to the clusters centroid's, rather than performing a similarity comparison against the whole clusters entities, which reduces the comparison overhead and improves the computational performance of the HLSOM module.

$$\begin{aligned} Assign_{Sim}(C, N) = &Context_{Sim}(C, N) \\ &+ Semantic_{Sim}(C, N) \end{aligned} \tag{9}$$

The contextual similarity $Context_{Sim}$ exploits nodes and their parents and children, in such a way that whenever two nodes share several common nodes, their contextual similarity will be necessarily high. The contextual similarity between the centroid C and node N can be computed as follows:

$$Context_{Sim}(C, N) = \frac{\partial(C) \cap \partial(N)}{\sqrt{\partial(C) \times \partial(N)}} \tag{10}$$

where $\partial(C)$ and $\partial(N)$ represent the number of nodes surrounding the node C and N, respectively, including the node itself and $\partial(C) \cap \partial(N)$ is the number of common nodes between C and N.

On the other hand, the hierarchical semantic similarity $Semantic_{Sim}$ relies on the taxonomical features obtained through the following metric:

$$Semantic_{Sim}(C,N) = \frac{2 \times M_3}{M_1 + M_2 + 2 \times M_3} \tag{11}$$

where $M1$ and $M2$ are the numbers of ascendants from C and N to their common ancestor A, respectively, whereas M_3 is the number of descendants from the root of the taxonomy to the node A.

3.3.3 Large-scale ontology matching layer

The above clustering strategy allows us to perform ontology matching tasks over formerly partitioned ontologies following the MapReduce paradigm over a parallel and a distributed infrastructure. With this intention, this step determines, firstly, semantically comparable clusters candidates for the matching process. Accordingly, the entities of dissimilar clusters will be ignored from the matching process, which can lead to a reduced matching overhead and achieve better matching accuracy. Following that, we implement the matching process by combining several matching algorithms. These algorithms are parallelized and distributed over available computational resources, resulting in many finer matching tasks with limited computational complexity (see Fig. 3). Finally, all the partial individual results are aggregated to construct the alignments set.

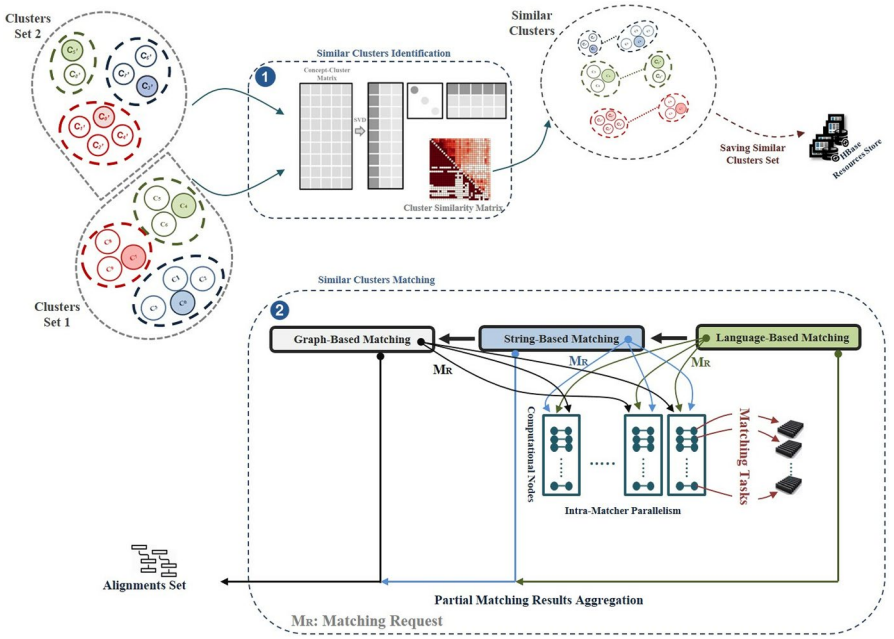


Fig. 3 Matching process between ontology partitions

Algorithm 2: Large-scale Ontology Clustering Layer

Input: Local Ontology O
Output: Set of Clusters of O

/* Entities Ranking */
$C \leftarrow Load\_OWLClasses(O)$
foreach $C_i \in C$ **do**
  $D_c \leftarrow get\_Degree\_Centrality(C_i)$
  $CL_c \leftarrow get\_Closeness\_Centrality(C_i)$
  $CX_c \leftarrow get\_Context\_Centrality(C_i)$
  $GD_c \leftarrow get\_GeneralizedDegree\_Centrality(C_i)$
  $Ranking_{score}(C_i) = D_c + CL_c + CX_c + GD_c$
  Rank C_i given its $Ranking_{score}$
end
/* Centroid Selection */
foreach $C_i \in Ranked_{list}$ **do**
  $\varphi(C_i) \leftarrow$ *calculate the decline rate if removing* C_i
  if $\varphi(C_i) \geq$ *default network efficiency* **then**
    Add C_i*to the set of important nodes* Set_{imp}
  end
end
foreach $N_i \in Set_{imp}$ **do**
  calculate the structural proximity between N_i *and the other nodes* $\in Set_{imp}$
  $centroid_{set} \leftarrow$ *nodes having min proximity*
end
/* Clusters Creation */
foreach $ctr_i \in centroid_{set}$ **do**
  create cluster for each ctr_i
  $neighbors_{List}(ctr_i) \leftarrow$ *load neighbors nodes of* ctr_i
  add $neighbors_{List}(ctr_i)$ *to the cluster*(ctr_i)
  $remaining_{nodes} \leftarrow \{C \setminus centroid_{set}\}$
  foreach $node_i \in remaining_{nodes}$ **do**
    assign $node_i$ *to cluster*(ctr_i) *given the max* $Assign_{Sim}$ *measure*
    if $node_i$ *affected* **then**
      remove $node_i$ *from* $remaining_{nodes}$
    end
  end
end
return clusters set

Similar clusters identification This phase determines the clusters susceptible to be matched independently. Similarly to Moawed et al. [57], this phase leverages the Latent Semantic Indexing (LSI) approach to identify semantically comparable clusters, which overcomes the limitations of some conducted strategies (Sect. 2.3.1). Indeed, instead of parsing the whole ontologies to extract similarities between nodes (which is a time-consuming operation), applying LSI aims to construct the similarity matrix between the clusters of nodes. The LSI strategy takes as input two sets of clusters and determines similar clusters across them . First, the node-cluster matrix is created to represent the elements of the first set of clusters. Then, a truncated SVD (singular decomposition value) is applied to

factor the node-cluster matrix into left singular vectors U (representing nodes), right singular vectors (representing clusters) V and keeping only the greatest k singular values (matrix S). Next, the second clusters set is also formulated according the reduced k-dimensional space (i.e. matrix S) to determine the singular vectors V′ representing clusters of the second set. Finally, the cosine similarity between V and V′ is invoked to identify comparable clusters and to create the cluster similarity matrix. Besides, it is important to notice that the LSI strategy is parallelized over distributed computational resources which significantly reduces its execution time.

Clusters matching Once identifying similar clusters, the matching phase performs a combination of different ontology matching strategies over them to find correspondences between their entities. Thus, for each cluster pairs, a parallel ontology matching strategy is considered as an individual matching request that should be resolved independently. For that, the ontology matching tasks are distributed among the participating computational nodes. Notably, where the number of matching tasks is greater than the number of computational nodes, this phase benefits from the resources management mechanisms provided by the computational architecture of the Hadoop ecosystem to align and schedule these tasks and minimize the matching space. Furthermore, each ontology matching task aims to discover alignments between clusters elements by implementing a similarity computation between their entities. Hence, these tasks are performed according to three main levels, notably, the linguistic, terminological, and structural.

In fact, several works that combine multiple ontology matching strategies have been conducted to meet the different issues related to the ontology matching field [22, 29, 79, 80]. More specifically, several ontology matching algorithms, with specific techniques and features, have been proposed, and each one of them has its strengths and weaknesses. Thus, to improve the accuracy of the HLSOM module, this phase primarily invokes the element-based strategies, since it determines more ontological alignments than structure-based strategies. For that, this phase sequentially combines the language-based, the string-based (i.e. element-based strategies), and the graph-based (i.e. structure-based strategies) algorithms over the clusters entities, in such a way that only alignments with similarities greater than a fixed threshold are extracted. In addition, following the mechanism of reducing the matching space, it is necessary to notice that for each pair of clusters candidate to be matched, the pair of entities of an extracted correspondence are marked in order to be omitted from the loaded resources prior to the next matching strategy. Accordingly, the charge of the next matching operations is reduced since they are only applied for unmatched entities that are not matched using the prior strategy. Thus, this reduction of the matching space will avoid redundant matching tasks and results and reduce the execution time of the HLSOM module.

Algorithm 3: Candidates Clusters Identification

Input: $Cluster_{Set1} = C_{11}, C_{12}, ..., C_{1n}$ *and* $Cluster_{Set2} = C_{21}, C_{22}, ..., C_{2m}$
Output: *Clusters Similarity Matrix* (C_{1i}, C_{2j})

`/* Cluster_Set1 Preparation */`

create node-clusters matrix M for Cluster_Set1 **foreach** $m_{ij} \in M$ **do**
  `/* compute log-entropy weights for` m_{lc} `*/`
  $c\_freq_{lc} \leftarrow$ *frequency of appearance of a label* l *in each cluster* c
  $gc\_freq_l \leftarrow$ *global frequency of a label* l *within the entire clusters set*
  $P_{lc} = c\_freq_{lc}/gc\_freq_l$
  $$\sum P_{lc} \times \log P_{lc}$$
  $m_{lc} = (1 + \frac{c}{\log n}) \times (\log c\_freq_{lc} + 1)$
end

`/* singular value decomposition and dimensionality reduction */`
Apply SVD technique to $M \Rightarrow M = USV^T$
Dimensionality reduction $\Rightarrow M_k = U_k S_k V_k^T$ `/* (k singular values) */`

`/* interpretation of the` $Cluster_{Set2}$ `according to the reduced space (k singular values) */`
Create reduced node-clusters matrix Q for Cluster_Set2 $Q_k = U'_k S_k V'^T_k$
`/* similarity calculating and ranking */`
foreach *column* $m_a \in V_k$ **do**
  foreach *column* $q_b \in V'_k$ **do**
    $Cluster\_Sim\_Mat[a][b] = \cos_{sim}(m_a, q_b)$
  end
end
Return $Cluster\_Sim\_Mat$
Save Clusters Similarity Matrix

The first matching strategy applied concerns the language-based matching. It applies natural language processing mechanisms (i.e. post-tagging and stemming) as intrinsic techniques to exploit the morphological features of entities labels by reducing each form of a term (i.e. entity label) to a standardized form (stem). This strategy is mainly adopted before running string-based strategies in order to reduce the matching space and improve the HLSOM results. Afterward, this phase benefits from linguistic resources (i.e. WordNet) [56] to compute the similarity between the stemmed labels. Indeed, each label is linked to its WordNet specific lexical category, called synset. This latter is organized into senses, thereby providing the synonyms, the hyponym/hypernym (i.e. Is-A), and the meronym/holonym (i.e. Part-Of) relationships of each label. As a result, this phase measures the similarity between labels by computing the similarity between their appropriate related senses. For instance, the sense similarity *Sim WordNet* between two labels is computed by the max of the sum of the similarity of their senses *Sim sense*:

$$Sim_{WordNet}(L_i, L_j) = \max \sum_{S_{in} \in S_i} \sum_{S_{jm} \in S_j} Sim_{sense}(S_{in}, S_{jm}) \tag{12}$$

where $Sim_{sense}(S_{in}, S_{jm})$ is the similarity between two senses.

The Sim_{sense} is calculated via the Wu–Palmer measure [91] depicted in Eq. 11. Subsequently, the extracted correspondences are saved and their corresponding entity labels are removed from the list of labels, of each clusters' pair. Then, the clusters matching phase proceeds to the string-based matching strategies.

String-based matching strategies are designed, in turn, to treat the labels of entities or their descriptions without considering their meaning and context. Hence, this phase will preserve only the correct correspondences extracted by string equality. In this phase, the Levenshtein, trigram and JaroWinkler metrics are adopted and combined to extract the most relevant alignments. Notice that these metrics are only applied to entities not yet matched to any other ones during the prior language-based matching algorithm.

Levenshtein distance: This measure computes the minimum number of "edit" operations allowing the transformation of the entity label L_1 into the entity label L_2. This measure is obtained by dividing this number by the minimum length of the two entity labels.

$$Lev_{Sim}(L_1, L_2) = \frac{\min Edit(L_1, L_2)}{|L_1|, |L_2|} \tag{13}$$

TriGram: The n-gram method is generally developed in the linguistic computing and probability fields. It interprets a contiguous sequence of n units from a string (i.e. entity label). For the sake of making a tradeoff between accuracy (i.e. reliability of the information provided by the extracted sequence) and performance (i.e. complexity of computations), this phase uses trigrams as follows:

$$TgramSim(L_1, L_2) = \frac{Trigram(L_1) \cap Trigram(L_2)}{\min |L_1|, |L_2| - 2} \tag{14}$$

where $Trigram(L_i)$ is the set of trigrams for the entity label L_i.

JaroWinkler distance: This metric measures the similarity between two entities labels in such a way that the longer the distance between them, the higher similar they are. It is calculated by the following formula:

$$d_w = d_j + \delta_p(1 - d_j) \tag{15}$$

with:

$$d_j = \frac{1}{3} \times \left(\frac{m}{|S_1|} + \frac{m}{|S_2|} + \frac{m - t}{m} \right) \tag{16}$$

with δ_p is the prefix's length; d_j is the Jaro's distance, m is the number of corresponding characters, $|S_i|$ is the length of the string i and t is the number of transpositions.

Besides, contrary to the element-level strategies, the structure-level ones consider both the ontology entities and their relations for extracting alignments. Among them, this phase applies the graph-based strategy which considers the input ontologies as labeled graphs. Thus, the similarity measurement between nodes (i.e. entities) is based on the taxonomic structure of the graph, in such a way that, if two nodes are similar, their neighbors must also be somehow similar. For that, this phase exploits the structural measure formerly mentioned on the cluster's creation step (see Eq. 10) to extract the correspondences between clusters pairs. Besides, this contextual measure is applied only to the clusters entities that were not yet matched to any other ones during the prior matching strategies.

Finally, the HLSOM module aggregates these independent partial results gathered from each matching strategy to form the alignments set that will be set between input ontologies in order to create the shared global ontology.

3.4 Probabilistic-logical based assessment module

3.4.1 Big Data integration quality issues

Certainly, data from several sources increases diversity, which is particularly significant for the Big Data representativeness and reliability issues. Thus, integrating large datasets needs automated strategies to manage and control the quality of the integrated data, to efficiently use them in decision-making processes. Generally speaking, data quality is among the most Big Data concerns. In fact, removing things like bias, abnormalities or inconsistencies are just some aspects that factor into improving the accuracy of Big Data. Therefore, and similarly to different approaches [31, 47, 88], the proposed Big Data integration framework leverages ontological mechanisms to address these quality concerns during the Big Data integration process. In other words, having accurate ontology alignments reflects the effective Big Data integration framework quality. However, after conducting several evaluation experiments of the HLSOM module over several ontological tracks, having reference alignments, we have noticed some conflicting alignments vis-a-vis the encoded semantic constraints on these ontologies. From another side, the assessment of the proposed integration framework in real-world scenarios and without reference assessment mechanisms remains extremely unachievable, which requires a dynamic and scalable strategy for conflict resolution and incoherence mitigation during the integration process, in which any helpful confidences and constraints can be considered.

To this end, the proposed semantic-based Bid Data integration framework incorporates a probabilistic-logical based module that dynamically improves the quality of the extracted alignments and evaluates their accuracy according to the encoded ontological knowledge. This module leverages both statistical-language processing and rule-based approaches via the Markov logic formalism. Notably, it harnesses well-defined semantics constraints (i.e. description logics) that guarantee that

alignment conditions are interpreted uniformly, in order to exclude logically inconsistent alignments between the entities and thus improve the overall accuracy of the integration process.

3.4.2 Ontologies through description logics

Generally, description logics (DL) are decidable fragments of first-order logic that are designed to describe the entities of the ontology in terms of complex logical expressions. Notably, ontology incorporates axioms of concepts, individuals, and roles that state the specific relations between these concepts. Let S_c, S_r, and S_a be the mutually disjoint sets of concepts, roles, and individuals of an ontology, respectively. Thus, the T-Box of the ontology is defined by the finite set of general concept inclusion axioms of the form $C \subset D$, where C and D $\in S_c$. Also, the finite set of role inclusion axioms having the form $R \subset R'$ define the R-Box of the ontology, where R and R' $\in S_r$. Besides, the A-Box of the ontology is defined via the finite set of assertions having the form $a : C$, $(a, b) : R$, $a \neq b$, where a and b $\in S_a$, R $\in S_r$ and C $\in S_c$.

Therefore, encoding ontologies in description logics is beneficial, as it allows specific expressions that can be used to describe necessary and sufficient constraints. It also enables inference engines to reason about ontological descriptions. Thus, we rely on the T-Box descriptions and their corresponding graphs to illustrate the evaluation issue. In fact, the assessment module involves both purely rule-based logical data and uncertain data. The former is defined by the DL axioms since they are known to be true, in such a way that they represent the encoded knowledge and the logical structure of the input ontologies and should not be altered by the matching process. On the other hand, the ontological alignments extracted (i.e. uncertain data) rely on the degree of confidence derived from the similarity measures calculated. Indeed, considering both the similarity between entities (as degrees of uncertainty) and the logical semantic constraints of ontologies (as known logical rules) requires specific formalism that provides the representation of both deterministic and uncertain aspects of the issue.

3.4.3 Alignments refinement using Markov logic

Markov logic is a knowledge representation formalism that combines first-order logic fragments as a declarative language with undirected Markov networks as probabilistic graphical models. Markov networks permit to efficiently handle uncertainty, whereas first-order logic fragments compactly represent a wide variety of knowledge. Thus, Markov logic network (MLN) [66] is considered as a set of first-order formulas with weights, in such a way that the more evidence that a formula is true, the higher the weight of that formula.

Indeed, the assessment strategy relies on representing DLs as Markov logic networks, by realizing the first-order translation of ontologies and incorporating the similarity scores of the alignments as the network weights. The translation links the concepts (i.e. ontology classes) to unary predicates and properties (i.e. taxonomic and non-taxonomic) to binary predicates which enables modeling some basic semantic inference rules and uses them in the probabilistic reasoning

Table 1 DL axioms and their predicates

DL Axiom	Predicate
$O_i \models D \subseteq E$	$sub_i(d, e)$
$O_i \models D \subseteq \neg E$	$dis_i(d, e)$
$O_i \models \exists R.\top \subseteq D$	$sub_i^{dom}(r, d)$
$O_i \models \exists R^{-1}.\top \subseteq D$	$sub_i^{ran}(r, d)$
$O_i \models \exists R.\top \supseteq D$	$sup_i^{dom}(r, d)$
$O_i \models \exists R^{-1}.\top \supseteq D$	$sup_i^{ran}(r, d)$
$O_i \models \exists R.\top \subseteq \neg D$	$dis_i^{dom}(r, d)$
$O_i \models \exists R^{-1}.\top \subseteq D$	$dis_i^{ran}(r, d)$
(E, D, A, c)	$\langle map(E, D), c\rangle$

process. This may cover the majority of alignments conflicts that can appear during the HLSOM module, especially for ontologies without complex axiom structures. Let D and $E \subseteq S_c$ and $R \subseteq S_r$, A is a correspondence between E and D with c which is a confidence value (i.e. similarity value). Table 1 illustrates some axioms with their relative predicates.

To sum up, this module enables compiling formal constraint-based semantic rules into predicate calculus while incorporating well-founded confidence values, which allows efficient conflict resolution mechanisms. First, this module introduces the observable predicates to model the structure of the similar matched clusters with respect to both concepts and properties. Then, it adds ground atoms of observable predicates to the set of hard formulas and makes them hold in every extracted alignment (i.e. the encoded knowledge in the ontologies is assumed to be true and should be maintained). Indeed, given the set of the observable predicates, the assessment module focuses on determining the state of the hidden predicates that maximize the a-posteriori probability of the corresponding possible world. In addition, the ground atoms of these hidden predicates are assigned to the weights specified by the matching similarity σ, notably, $(map(C, D), \sigma(C, D))$ and $(map(P, R), \sigma(P, R))$, where C and D $\subseteq S_c$ and R and P $\subseteq S_r$. At the same time, the module enforces consistency by adding constraints that model the conflicts. Hence, the following constraints are used to determine the state of the hidden predicates and add them to the set of formulas:

Alignments cardinality constraints These hard cardinality constraints restrict the alignment to be functional and one-to-one in the Markov logical framework.

$$map(x, y) \wedge map(x, z) \Rightarrow y = z \tag{17}$$

$$map(x, y) \wedge map(z, y) \Rightarrow x = z \tag{18}$$

Alignments coherence constraints Since incoherence occurs when the extracted alignments lead to logical conflicts, these coherence constraints are incorporated on the set of hard formulas:

$$dis_1(R,D) \wedge sub_2(R,D) \Rightarrow \neg(map_p(R,R\prime) \wedge map_c(D,E)) \tag{19}$$

$$dis_1^d(R,D) \wedge sub_2^d(R,D) \Rightarrow \neg(map_p(R,R\prime) \wedge map_c(D,E)) \tag{20}$$

where map_c and map_p represent respectively the mapping between concepts and properties.

To clarify, if any correspondence contains two concepts that subsume each other in the first model and that are disjoint at the same time in the second model (i.e. the first and the second input ontology that want to be matched), thus, it must be excluded. Notably, given two properties R and R′ and concepts D and E, we note that if (R, R′, ≡) and (D, E, ≡) were two alignments of properties and concepts, respectively, and given that $O_1 \models \exists R.\top \subseteq \neg D$ and $O_2 \models \exists R.\top \subseteq E$, therefore, we can notice the appearance of a logical conflict between $\exists R.\top \subseteq D$ and $\exists R.\top \subseteq \neg D$.

Stability constraints The stability signifies that a correspondence should not introduce new structural knowledge; e.g. if we have a correspondence between two concepts, it does not mean that we should also have correspondences between their children. These constraints prevent this type of correspondences by decreasing their probability.

$$sub_1(R,D) \wedge \neg sub_2(R',E) \Rightarrow (map_p(R,R\prime) \wedge map_c(D,E), w_1) \tag{21}$$

$$sub_1^d(R,D) \wedge \neg sub_2^d(R',E) \Rightarrow (map_p(R,R\prime) \wedge map_c(D,E), w_2) \tag{22}$$

where w_1 and w_2 are weights that render the correspondences that satisfy the formulas possible but less likely.

In other words, the hidden predicates interpret correspondences between the entities of ontologies whereas observable ones describe the predicates gathered from description logic statements. Thus, determining the most likely alignment of two ontologies relies on calculating the set of ground atoms of the hidden predicates that maximize the probability, taking into account the ground atoms of observable predicates and the ground formula. This can be performed by executing the MAP (maximum a-posteriori) inference over the ground Markov logic network. Hence, considered as an effective method for exact MAP inference in undirected graphical models (i.e Markov logic networks), this module leverages the integer linear programming (ILP) by applying the MAP inference engine RockIt [59] over our distributed architecture.

For instance, Fig. 4 presents an example of two partitions of ontologies, candidates for matching. Thus, the adopted matching strategy extracts the list of alignments (see Table 2) that will be refined by the probabilistic-logical based assessment module, where σ is the similarity measure between their entities ($\sigma \geq 0.5$):

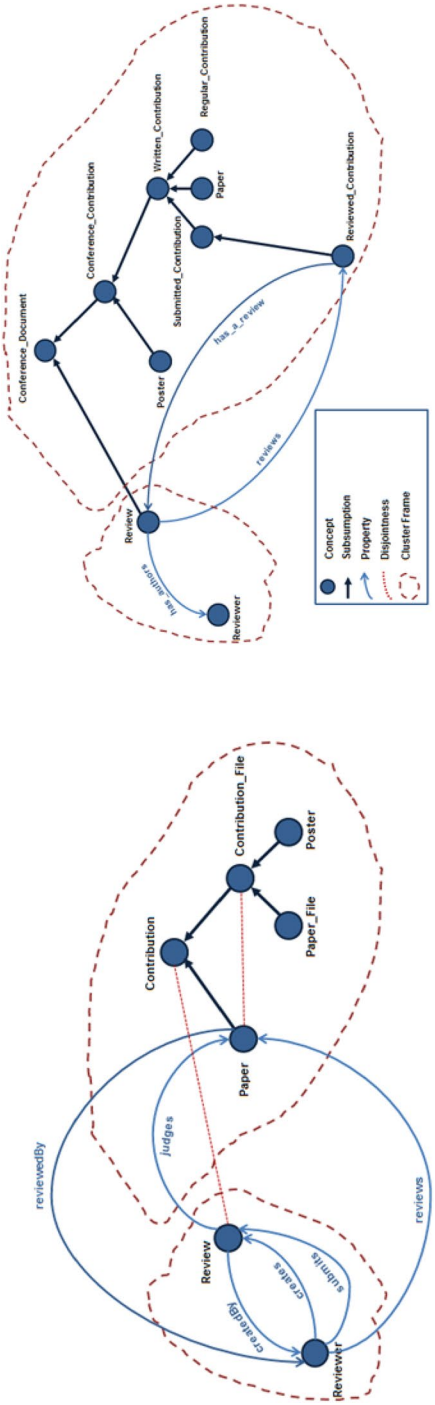


Fig. 4 Candidates partitions for the matching process

Table 2 Extracted alignments set ($\sigma \geq 0.5$)

Alignment	Similarity (σ)
$A_1 - map(Contribution, Conference\_Document)$	0.553
$A_2 - map(Contribution, Written\_Contribution)$	0.557
$A_3 - map(Contribution, Submitted\_Contribution)$	0.588
$A_4 - map(Contribution, Reviewed\_Contribution)$	0.608
$A_5 - map(Contribution\_File, Written\_Contribution)$	0.549
$A_6 - map(Paper, Paper)$	1
$A_7 - map(Paper\_File, Paper)$	0.722
$A_8 - map(Poster, Poster)$	1
$A_9 - map(Review, Review)$	1
$A_{10} - map(Reviewer, Reviewer)$	1
$A_{11} - map(reviews, reviews)$	1
$A_{12} - map(judges, reviews)$	0.678
$A_{13} - map(createdBy, hasAuthors)$	0.731

As an illustration, the consistency is enforced by applying the introduced constraints in such a way that, for the same entity, every alignment $map(F_1, F_2)$ that have positive entity assertion for the first individual (i.e. F_1) and a negative affirmation (i.e. F_2) for the second must be ignored from the obtained alignment results.

Cardinality constraints After induction, the refinement module excludes the alignment (A_7) even if it has a higher similarity value since *Paper_File* and *Paper* are two disjoint entities.

$$map(Paper\_File, Paper) \wedge map(Paper, Paper) \Rightarrow Paper\_File = Paper$$

Coherence constraints Similarly, using these constraints, the assessment module ignores the alignment (A_1).

$$dis_1(Review, Contribution) \wedge sub_2(Review, Conference\_Document) \Rightarrow \neg(map(Review, Review) \wedge map(Contribution, Conference\_Document))$$

Stability constraints These constraints leverage the fact that existing correspondences evidence should not introduce new structural knowledge. They render the alignments that satisfy the formulas possible but less likely. Thus, after induction, these constraints exclude respectively the alignments (A_3), (A_4) and ($A_1$1).

$$sub_1(Paper, Contribution) \wedge \\ \neg sub_2(Paper, Submitteded\_Contribution) \Rightarrow \\ map(Paper, Paper) \wedge \\ map(Contribution, Submitted\_Contribution)$$

$$sub_1(Paper, Contribution) \wedge \\ \neg sub_2(Paper, Reviewed\_Contribution) \Rightarrow \\ map(Paper, Paper) \wedge \\ map(Contribution, Reviewed\_Contribution)$$

$$sub_1(reviews, Reviewer) \wedge \\ \neg sub_2(reviews, Reviewer) \Rightarrow \\ map(reviews, reviews) \wedge \\ map(Reviewer, Reviewer)$$

4 Experimental evaluation

Since the focus was on implementing the HLSOM and the probabilistic-logical based assessment modules, this experimental evaluation is conducted over a collection of real-world ontologies provided by the OAEI (i.e Ontology Alignment Evaluation Initiative[4]), to evaluate the integration process in terms of accuracy and execution time. These ontologies are of various sizes and cover different magnitudes of the ontology matching issue. The conference dataset contains 16 ontologies from the conference organization domain, whereas, the anatomy dataset contains two large ontologies of human and mouse anatomy with respectively 3306 and 2746 concepts.

To this end, we have used various ontology matching approaches (Sect. 2.3) having computational complexity upper than $O(n^2)$. All the experiments were carried out using the Cloudera distribution Hadoop platform which is an open-source Apache Hadoop distribution, deployed over two distributed computational architectures. Cloudera provides a scalable, flexible, and integrated platform to manage massive amounts of data. It allows deploying and managing Apache Hadoop and its related projects. Furthermore, the integration framework uses HBase, which is a column-oriented data store built to run on top of the Hadoop Distributed File System (HDFS), to support random, real-time read/write access and allow using a larger in-memory cache, which reduces the execution time of the framework. Besides, the framework is implemented over two computational modes: (i) a pseudo-distributed mode over a single-node desktop

[4] http://oaei.ontologymatching.org.

PC incorporating multiple cores as a standalone cluster, equipped with Intel® Xeon® 5160 (3 GHz)*8 with 12 GB memory, Java 1.8 and Ubuntu 14.04 LTS; (ii) a fully-distributed mode over a cluster of 10 slave machines and one master machine. Each node is equipped with 3.4 GHz Intel(R) Core i3(R) with 4 GB memory, Java 1.8, and Ubuntu 14.04 LTS.

The Hadoop Distributed File System (HDFS) [75] is a scalable and distributed standalone file system designed as the core storage component of the Apache Hadoop ecosystem and the majority of its associated Big Data platforms [78, 90]. HDFS provides a highly reliable and scalable data storage system across a large set of low-cost commodity hardware [68]. HDFS is suitable for data-intensive applications, quickly data ingesting, and bulk processing that requires high throughput [81]. Typically, HDFS provides high scalability, reliability, availability, and protection against data loss caused by node failures [28]. Furthermore, HDFS achieves better storage efficiency (high throughput and network traffic reduction) [50, 75]. Also, it is agnostic of data storage format and allows storing and manipulating any data type using several read/write APIs.

HBase is an open-source column-oriented data store implemented to handle the Big Data storage requirements in the Apache project [34]. HBase runs on the top of HDFS and uses Apache Zookeeper [38] for storage cluster management. Moreover, HBase has exceptional support for read-intensive transactions [67], which is the main reason for its use in the HLSOM module (i.e. loading extracted data resources). Also, it is a high performance, scalable, distributed, and fault-tolerant storage system that offers random read-write access to Big Data [83].

In the first stage, we should get answers to the following issues:

- What is the best combination of centrality metrics that should be adopted to implement the entities' ranking phase?
- What is the convenient number of clusters centroids that should be selected to perform the ontology clustering phase?

Centrality metrics selection To this end, a series of experimental evaluations are carried out over two different large ontologies from the Anatomy dataset, namely the NCI human anatomy and the adult mouse anatomy. This test inspects the best centrality combination alternative. The results of the combinations of these centrality metrics on each ontology are shown in Figs. 5 and 6, where each bar describes one combination of them, DC refers to degree centrality, LC means the closeness centrality, CC is the context centrality and GDC is the generalized degree centrality. Hence, given the performed implementations, several combinations could achieve this objective. Accordingly, we select DC + LC + CC + GDC, as it outperforms the other combinations slightly while combining and benefiting from local-based and global-based information with affordable performance. In brief, this strategy conducts an efficiently ranking process in terms of accuracy and performance by loading only the required resources -already created via the resources extraction layer - and by performing a parallel ranking process over the distributed computational architecture.

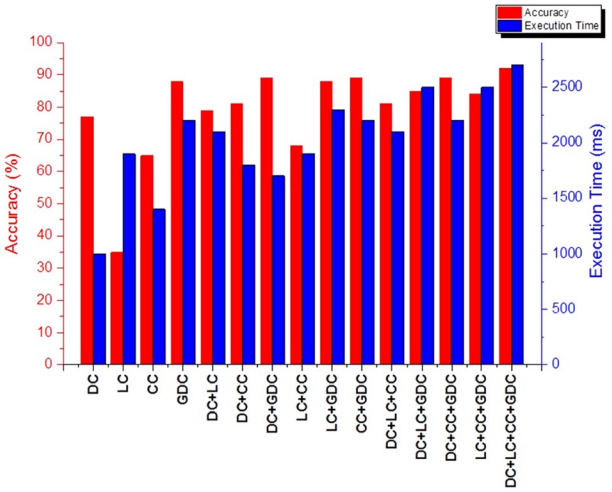


Fig. 5 Accuracy and execution time of the centrality metrics combinations on the NCI human anatomy ontology

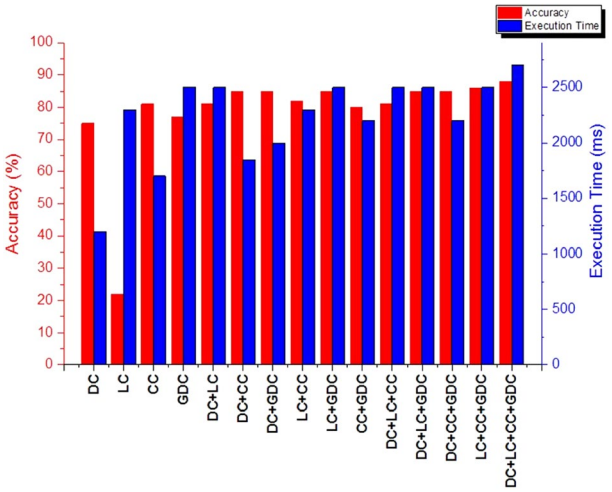


Fig. 6 Accuracy and execution time of the centrality metrics combinations on the the adult mouse anatomy ontology

Clusters centroids selection Given the large-scale nature of the input ontologies, the selection of the most important nodes, after being ranked by the combination of centrality measures as clusters centroids, remains impracticable. Therefore, this experiment adopts the decline rate of network efficiency as a solution to automatically select the most convenient number of clusters centroids. Figures 7 and 8 illustrates the set of nodes able to be centroids, for the two large

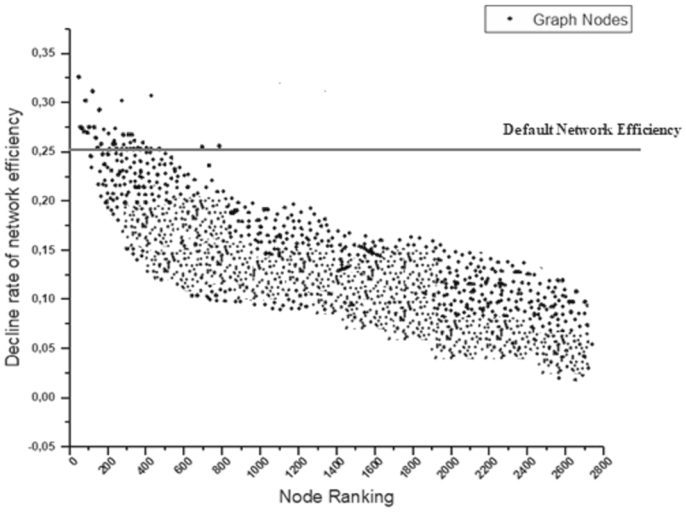


Fig. 7 Selected centroids for the adult mouse anatomy ontology

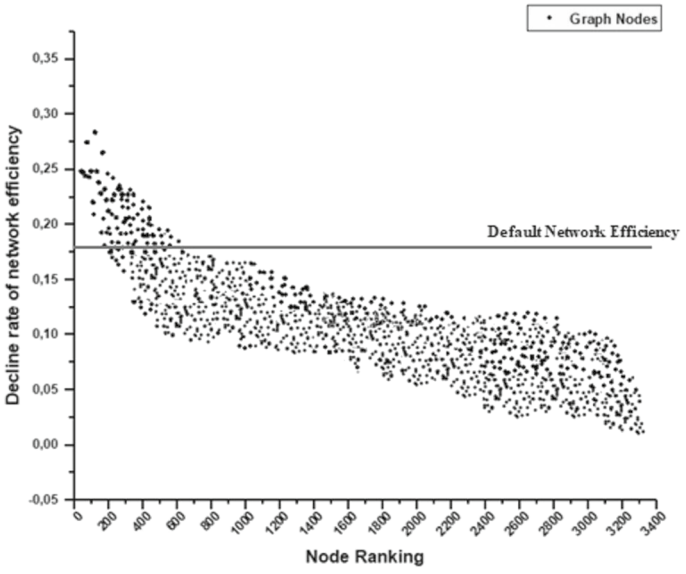


Fig. 8 Selected centroids for the NCI human anatomy

ontologies of the Anatomy track, which are those with a decline rate ratio on top of the default network efficiency. Then, from this set of nodes, only those having minimal structural proximity are selected as clusters centroids, which effectively smooth the large-scale clustering layer.

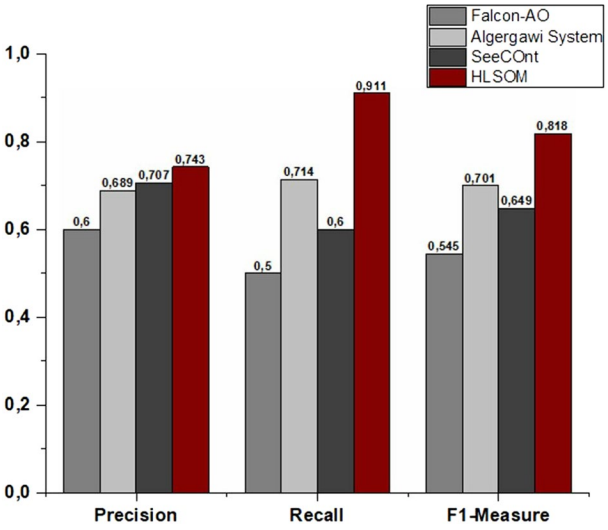


Fig. 9 Ontology matching results for the conference dataset

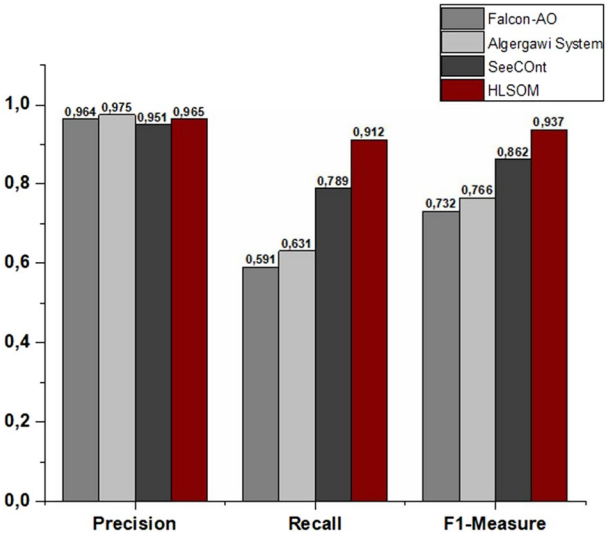


Fig. 10 Ontology matching results for the anatomy dataset

4.1 Accuracy evaluation using reference alignments

In this implementation, we validate the accuracy of the HLSOM module over a collection of real-world ontologies. First, we compare its efficiency, without using the probabilistic-logical based assessment module, against a set of recent large-scale ontology matching strategies (see Figs. 9 and 10) to prove its pertinence as

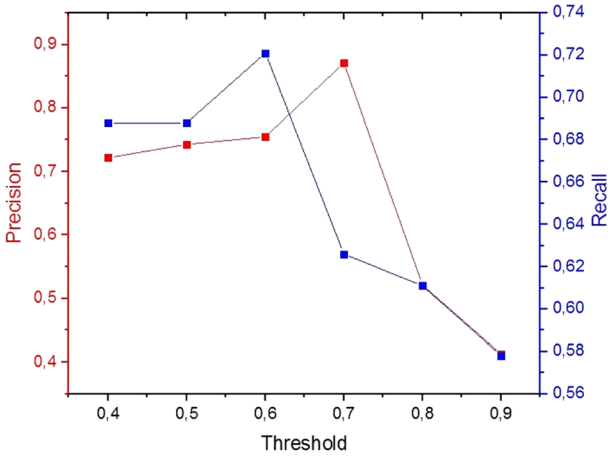


Fig. 11 Threshold variation impact for the conference dataset

a semantic-based Big Data integration strategy. The results show that the HLSOM module achieves high precision, recall, and F-measure than the other systems using the threshold ($\sigma = 0.6$). This threshold is selected based on a tradeoff between the accuracy and the recall of the matching process. In fact, the sequential combination of the different ontology matching strategies has improved the efficiency of the proposed HLSOM module, even if this accuracy is slightly lower than some ontology matching systems for the anatomy dataset due to the special features of its entity labels. Whereas, the adopted clustering strategy allows getting related entities in the same partition, which reduces the matching space and contributes to achieving a good recall.

On the other hand, Figs. 11 and 12 show that the number of extracted alignments decreases dramatically if we increase the predefined threshold, which consequently impacts the precision and recall of the matching process. Besides, after checking the gathered alignments, we found that the matching process neglected some pertinent alignments due to their low similarity for a selected great threshold. In contrast, by decreasing this latter, the matching process achieves some conflicting alignments vis-a-vis the semantic constraints encoded on input ontologies. For these reasons, the probabilistic-logical based assessment module is adopted after the HLSOM module to allow the extraction of the maximum number of alignments while ensuring their reliability.

4.2 Accuracy evaluation after probabilistic-logical based refinement

For large-scale context, the majority of ontology matching approaches try to improve the accuracy of their alignments by increasing the similarity thresholds. In contrast, significant alignments with low similarity value would be ignored, which may consequently impact the accuracy of the Big Data integration framework and undermine the appropriate exploitation of the created shared ontology. Accordingly,

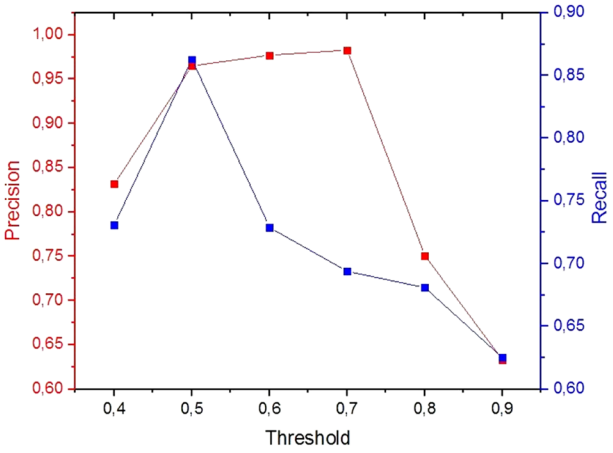


Fig. 12 Threshold variation impact for the anatomy dataset

the refinement module (i.e. the probabilistic-logical based assessment module) allows us to gather more alignments even if their similarity is low, and control them through various conflicts detection constraints.

First, we gather the alignments having similarity greater than a specific minimal threshold, as well as their corresponding similarity. Then, we apply different constraints (see Sect. 3.4.3) to enable conflict detection and to extract the most probable alignments. These constraints take advantage of the fact that existing correspondences evidence should not introduce new structural knowledge. Figures 13 and 14 demonstrate the improvement of the accuracy of the matching process, by providing only the accurate alignments that are consistent with the DLs of ontologies.

Furthermore, the refined matching strategy is compared with other matching systems (participating in previous OAEI competitions), over the Conference and Anatomy tracks, respectively. For simplicity of the chart, Figs. 15 and 16 present the F1-measure of each approach. We notice that the proposed refined matching approach outperforms the other systems, which proves its pertinence to be adopted as a semantic-based Big Data integration strategy.

4.3 Performance evaluation of the proposed integration framework

In this section, we present the results of the experimental evaluation of the performance and scalability of the proposed integration framework.

4.3.1 Evaluation criteria

Running time Running time is the time needed to create a shared global ontology. Thus, the running time is measured both for the pseudo-distributed and fully-distributed modes.

Running time The speedup is measured by the running time of an algorithm, ran on the smallest cluster, divided by the running time of the same algorithm, ran on

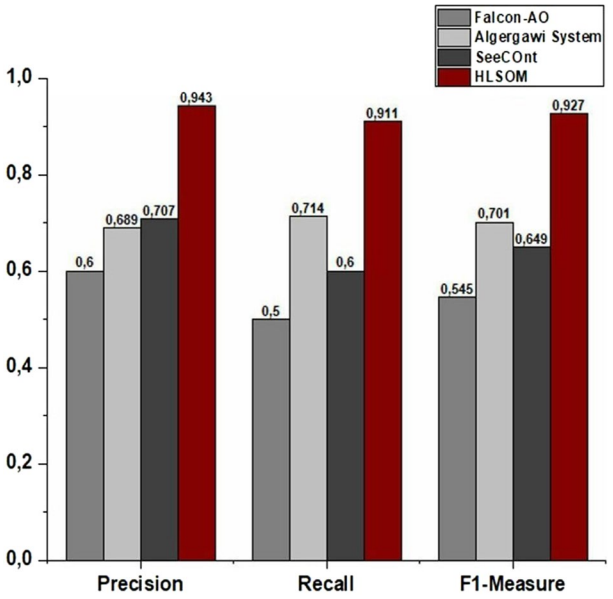


Fig. 13 Enhanced ontology matching results for the conference dataset

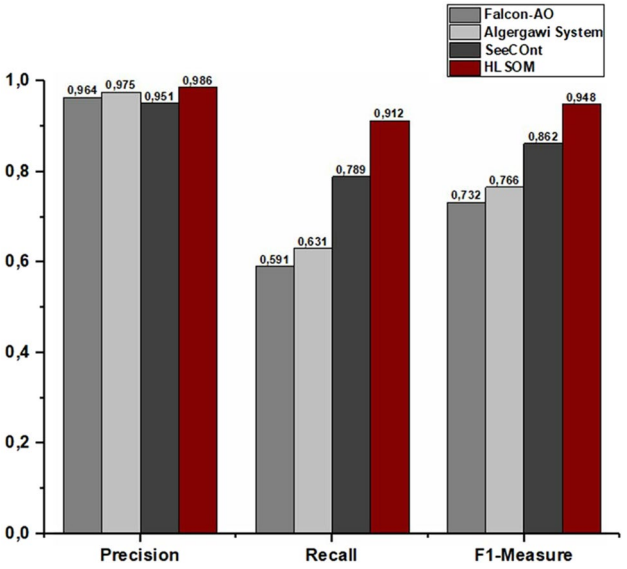


Fig. 14 Enhanced ontology matching results for the anatomy dataset

another cluster having more number of nodes (i.e. executors). The speedup measurement is as follows:

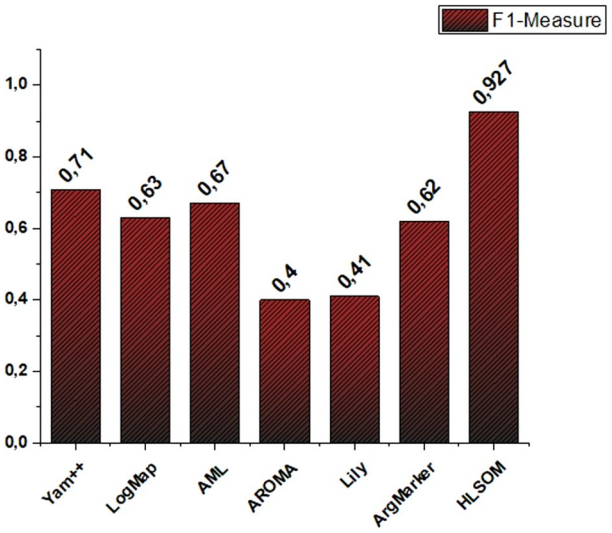


Fig. 15 Refined HLSOM comparison with leading systems participating in OAEI competitions for the conference dataset

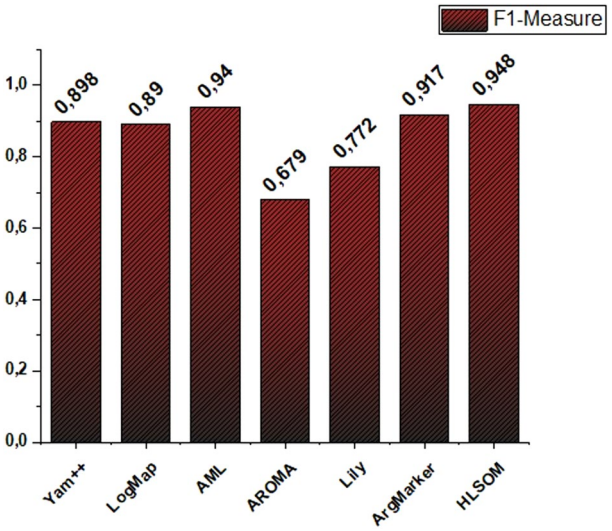


Fig. 16 Refined HLSOM comparison with leading systems participating in OAEI competitions for the anatomy dataset

$$S_p = \frac{T_k}{T_p} \tag{23}$$

where S_p indicates the speedup of the framework on a cluster with p nodes, T_k is the average running time of the framework on the smallest cluster that contains k

nodes, while T_p is the average running time of the framework ran on the cluster with p nodes. Indeed, an ideal speedup is increased linearly while increasing the number of used nodes ($S_p = p$).

Parallel Efficiency. The parallel efficiency metric denotes the efficiency of the framework while using increasing numbers of parallel processing elements (cores, nodes, etc.). Specifically, it measures the fraction of time for which a processor is usefully utilized (i.e. the speedup per processor). It is commonly defined as the speedup divided by the number of units of execution.

$$E_p = \frac{S_p}{p} \tag{24}$$

where E_p indicates the parallel efficiency of the framework, S_p is the speedup of the framework and p is the number of the units of execution. Indeed, the ideal efficiency is achieved when the speedup is ideal(i.e. $E_p = 1 = 100\%$.

Scalability. The scalability is the capacity of a parallel framework to increase its speedup in proportion to the number of execution units. It can be undermined by the overhead due to communications and the load-balancing of tasks distribution.

4.3.2 Pseudo-distributed mode

Following this mode, a single local machine with 8 available cores is used. Thus, to demonstrate the scalability, the proposed framework is executed on a cluster with one core and the overall running time was recorded. Then, the same experiment is executed on other clusters with increasing cores number. The speedup is measured via the formula 23, where the smallest cluster contains one single core.

As illustrated in Fig. 17, the differences in execution time between eight pseudo-clusters are not significant. The results show that the speedup starts improving as more cores are introduced. The framework takes 106 seconds to create the shared global ontology for the anatomy track. However, by monitoring the system during runtime, we notice that this latency is due to network limitations on one machine. Certainly, as the number of partitions grows, the system becomes I/O bound, meaning that its speed is limited by the speed of the input/output operations on that machine. Concerning the parallel efficiency, we observe that it decreases by increasing the number of used cores. This signifies that running the framework in a single machine, even in a pseudo-distributed mode, remains limited and does not improve the efficiency of the implemented modules. For these reasons, a fully-distributed implementation over a cluster of machines is performed, which yields more interesting results.

4.3.3 Fully-distributed mode

Under these settings, we implement the proposed framework over a cluster of several machines (one master and 10 slave nodes). Hence, we evaluate the scalability of the framework by varying the number of slave machines, given that the smallest

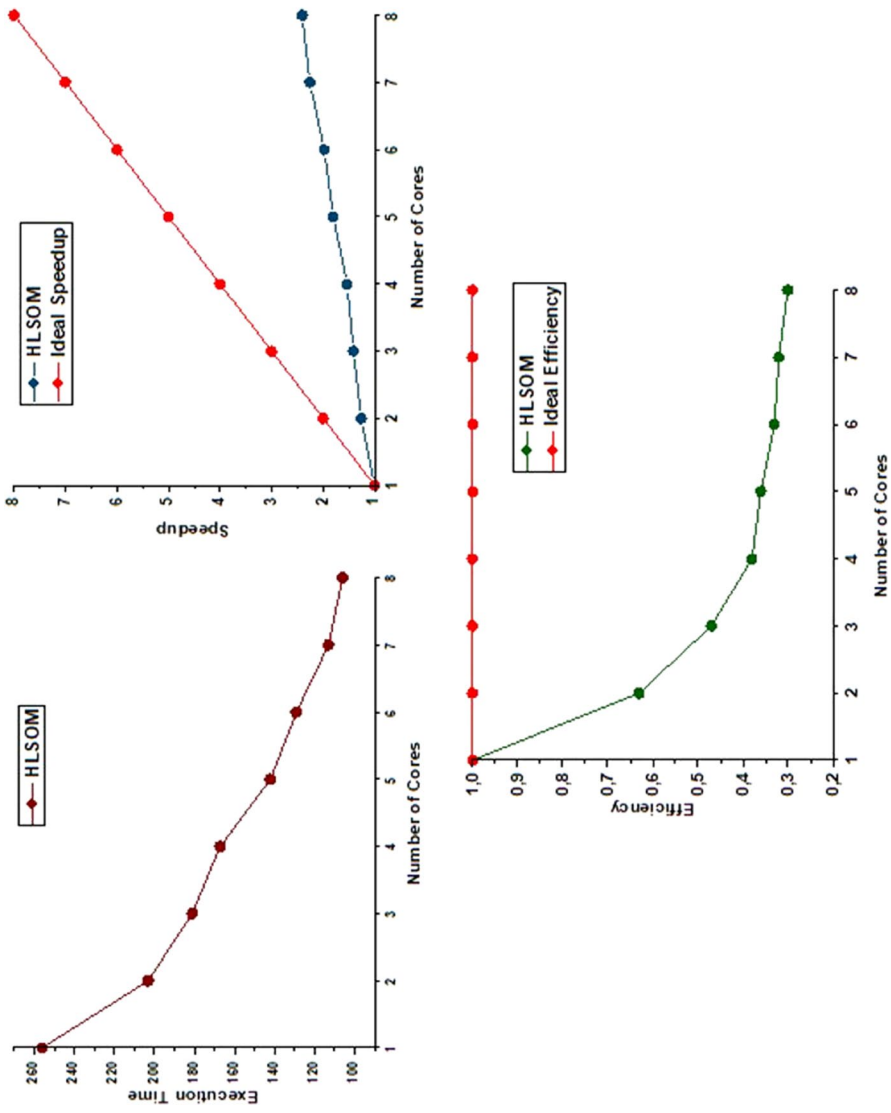


Fig. 17 Execution time, speedup and efficiency on pseudo-distributed mode

cluster is composed of one node. Then, we compute the speedup by comparing the running time of the different clusters using the formula 23.

Figure 18 presents a summary of the experimental results over different computing node clusters. Thus, the results show that the differences in running time between ten clusters yielded too much more favorable results compared to the pseudo-distributed implementation since the speedup increase linearly. Therefore, the more nodes are added for processing, the fastest the running time is achieved and the higher speedup is obtained. Furthermore, the speedup in a fully-distributed implementation is linear and gets closer to the ideal speedup. Moreover, the efficiency is higher, which signifies that the slave machines are effectively used to improve the scalability of the proposed framework. Overall, it is worth noting that the extracted scalability-friendly resources and the parallelism-driven implementation, using parallel-programming techniques over advanced distributed architectures, have contributed to enhancing the speedup and the efficiency of the framework.

5 Conclusions and future works

This paper presents a semantic-based Big Data integration framework that aims to provide a unified and integrated view of the available data, while considering Big Data features (4Vs) and aspects (i.e. scalability, availability, and high-performance), to support the extraction of reliable and consistent knowledge. This integration framework incorporates several modules, including the local ontology building module, the HLSOM module, and the probabilistic-logical based assessment module, implemented over distributed architectures. Therefore, it leverages Hadoop as a data parallelism-enabled platform and MapReduce as a parallel programming technique over distributed storage and computational resources. Experimental results over real-world ontologies and different implementation modes show significant accuracy, good runtime, high-performance, and high scalability in large-scale environments. This is achieved through scalability-friendly optimization mechanisms and the parallelism-driven implementation of the integration framework. Accordingly, future works will focus on using this semantic-based Big Data integration framework to build a domain-specific application for smart tourism destinations (STD). In fact, STD can be defined by the integrated engagement and efforts of different stakeholders at a destination, to collect, aggregate, and harness data derived from government/organizational sources, social networks connections, and physical infrastructure in conjunction with the use of advanced methodologies and technologies to transform that data into valuable insights for the experience enrichment and the business value co-creation to make better and smarter strategic and operational decisions. Therefore, the creation of a unified, reliable, and consistent knowledge as well as its exploitation in various querying and analytical operations will enable them to acquire additional and detailed information that assists them in the decision-making process.

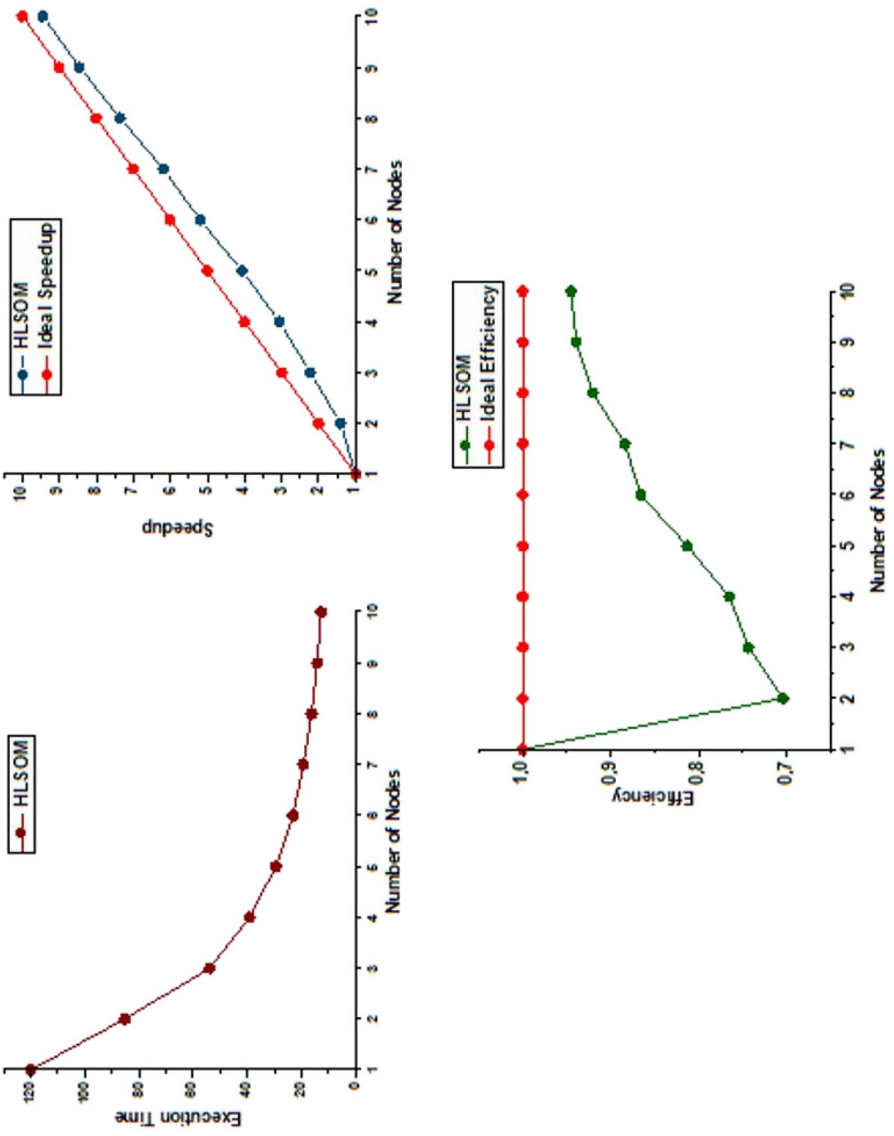


Fig. 18 Execution time, speedup and efficiency on fully-distributed mode

References

1. Abbes, H., Gargouri, F.: Mongodb-based modular ontology building for big data integration. J. Data Seman. **7**(1), 1–27 (2018)
2. Alasoud, A., Haarslev, V., Shiri, N.: A hybrid approach for ontology integration. In: Proceedings of the VLDB Workshop on Ontologies-based techniques for DataBases and Information Systems (ODBIS), Trondheim, Norway (2005)
3. Algergawy, A., Massmann, S., Rahm, E.: A clustering-based approach for large-scale ontology matching. In: Proceedings of the East European Conference on Advances in Databases and Information Systems, Springer, pp. 415–428 (2011)
4. Algergawy, A., Babalou, S., Kargar, M.J., Davarpanah, S.H.: Seecont: A new seeding-based clustering approach for ontology matching. In: Proceedings of the East European Conference on Advances in Databases and Information Systems, Springer, pp 245–258 (2015)
5. Amin, M.B., Khan, W.A., Lee, S., Kang, B.H.: Performance-based ontology matching. Appl. Intell. **43**(2), 356–385 (2015)
6. Ba, M., Diallo, G.: Large-scale biomedical ontology matching with servomap. IRBM **34**(1), 56–59 (2013)
7. Bansal, S.K., Kagemann, S.: Integrating big data: a semantic extract-transform-load framework. Computer **48**(3), 42–50 (2015)
8. Bello-Orgaz, G., Jung, J.J., Camacho, D.: Social big data: recent achievements and new challenges. Inf. Fus. **28**, 45–59 (2016)
9. Brandes, U., Borgatti, S.P., Freeman, L.C.: Maintaining the duality of closeness and betweenness centrality. Soc. Netw. **44**, 153–159 (2016)
10. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: the dl-lite family. J. Autom. Reason. **39**(3), 385–429 (2007)
11. Castano, S., Ferrara, A., Montanelli, S.: Matching techniques for data integration and exploration: from databases to big data. In: A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years, Springer, pp 61–76 (2018)
12. Cerbah, F.: Learning ontologies with deep class hierarchies by mining the content of relational databases. In: Advances in knowledge discovery and management, Springer, pp 271–286 (2010)
13. Cheatham, M., Pesquita, C.: Semantic data integration. In: Handbook of Big Data Technologies, Springer, pp 263–305 (2017)
14. Chen, J., Chen, Y., Du, X., Li, C., Lu, J., Zhao, S., Zhou, X.: Big data challenge: a data management perspective. Front. Comput. Sci. **7**(2), 157–164 (2013)
15. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mobile Netw. Appl. **19**(2), 171–209 (2014)
16. Cruz, I.F., Xiao, H.: The role of ontologies in data integration. Eng. Intell. Syst. Electr. Eng. Commun. **13**(4), 245 (2005)
17. Csató, L.: Measuring centrality by a generalization of degree. Central Eur. J. Oper. Res. **25**(4), 771–790 (2017)
18. Curé, O., Lamolle, M., Duc, C.L.: Ontology based data integration over document and column family oriented nosql. arXiv preprint arXiv:13072603 (2013)
19. Daraio, C., Lenzerini, M., Leporelli, C., Moed, H.F., Naggar, P., Bonaccorsi, A., Bartolucci, A.: Data integration for research and innovation policy: an ontology-based data management approach. Scientometrics **106**(2), 857–871 (2016a)
20. Daraio, C., Lenzerini, M., Leporelli, C., Naggar, P., Bonaccorsi, A., Bartolucci, A.: The advantages of an ontology-based data management approach: openness, interoperability and data quality. Scientometrics **108**(1), 441–455 (2016b)
21. David, J., Guillet, F., Briand, H.: Matching directories and owl ontologies with aroma. In: Proceedings of the 15th ACM international conference on Information and knowledge management, ACM, pp 830–831 (2006)
22. Djeddi, W.E., Khadir, M.T.: A novel approach using context-based measure for matching large scale ontologies. In: International Conference on Data Warehousing and Knowledge Discovery, Springer, pp 320–331 (2014)
23. Do, H.H., Rahm, E.: Matching large schemas: approaches and evaluation. Inf. Syst. **32**(6), 857–885 (2007)

24. Ehrig, M., Staab, S.: Qom–quick ontology mapping. In: Proceedings of the International Semantic Web Conference, Springer, pp 683–697 (2004)
25. El Idrissi Esserhrouchni, O., Frikh, B., Ouhbi, B.: Learning non-taxonomic relationships of financial ontology. In: Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, SCITEPRESS-Science and Technology Publications, Lda, pp 479–489 (2015)
26. El Idrissi, Esserhrouchni O., Frikh, B., Ouhbi, B., Ibrahim, I.K.: Learning domain taxonomies: the taxoline approach. Int. J. Web Inf. Syst. **13**(3), 281–301 (2017)
27. Emani, C.K., Cullot, N., Nicolle, C.: Understandable big data: a survey. Comput. Sci. Rev. **17**, 70–81 (2015)
28. Erraissi, A., Belangour, A.: Capturing hadoop storage big data layer meta-concepts. In: Proceedings of the International Conference on Advanced Intelligent Systems for Sustainable Development, Springer, pp 413–421 (2018)
29. Essayeh, A., Abed, M.: Towards ontology matching based system through terminological, structural and semantic level. Procedia Comput. Sci. **60**, 403–412 (2015)
30. Euzenat, J., Shvaiko, P., et al.: Ontology Matching, vol. 18. Springer, New York (2007)
31. Fürber, C., Hepp, M.: Towards a vocabulary for data quality management in semantic web architectures. In: Proceedings of the 1st International Workshop on Linked Web Data Management, ACM, pp 1–8 (2011)
32. Gao, C., Wei, D., Hu, Y., Mahadevan, S., Deng, Y.: A modified evidential methodology of identifying influential nodes in weighted networks. Phys. A Stat. Mech. Appl. **392**(21), 5490–5500 (2013)
33. García, MdMR, García-Nieto, J., Aldana-Montes, J.F.: An ontology-based data integration approach for web analytics in e-commerce. Expert Syst. Appl. **63**, 20–34 (2016)
34. George, L.: HBase: The Definitive Guide: Random Access to Your Planet-size Data. O'Reilly Media Inc, Newton (2011)
35. Gross, A., Hartung, M., Kirsten, T., Rahm, E.: On matching large life science ontologies in parallel. In: Proceedings of the International Conference on Data Integration in the Life Sciences, Springer, pp 35–49 (2010)
36. Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: a divide-and-conquer approach. Data Knowl. Eng. **67**(1), 140–160 (2008)
37. Hui, J., Li, L., Zhang, Z.: Integration of big data: a survey. In: Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators. pp. 101–121. Springer (2018)
38. Hunt, P., Konar, M., Junqueira, F.P., Reed, B.: Zookeeper: Wait-free coordination for internet-scale systems. In: Proceedings of the USENIX Annual Technical Conference, Boston, vol 8 (2010)
39. Jiménez-Ruiz, E., Grau ,B.C.: Logmap: Logic-based and scalable ontology matching. In: Proceedings of the International Semantic Web Conference, Springer, pp 273–288 (2011)
40. Jirkovskỳ, V., Obitko, M.: Semantic heterogeneity reduction for big data in industrial automation. In: Proceedings of the ITAT (2014)
41. Kiran, V., Vijayakumar, R.: Ontology based data integration of nosql datastores. In: Proceedings of the Industrial and Information Systems (ICIIS), 2014 9th International Conference on, IEEE, pp 1–6 (2014)
42. Klein, D.: Centrality measure in graphs. J. Math. Chem. **47**(4), 1209–1223 (2010)
43. Krishnan, K.: Data Warehousing in the Age of Big Data. Newnes, Oxford (2013)
44. Landherr, A., Friedl, B., Heidemann, J.: A critical review of centrality measures in social networks. Bus. Inf. Syst. Eng. **2**(6), 371–385 (2010)
45. Lenzerini, M.: Data integration: A theoretical perspective. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM, pp 233–246 (2002)
46. Li, L., Wei, Y., Tian, F.: A framework for ontology-based top-k global schema generation. J. Data Seman. **6**(1), 31–53 (2017)
47. Liaw, S.T., Rahimi, A., Ray, P., Taggart, J., Dennis, S., de Lusignan, S., Jalaludin, B., Yeo, A., Talaei-Khoei, A.: Towards an ontology for data quality in integrated chronic disease management: a realist review of the literature. Int. J. Med. Inf. **82**(1), 10–24 (2013)
48. Liu, J., Xiong, Q., Shi, W., Shi, X., Wang, K.: Evaluating the importance of nodes in complex networks. Phys. A Stat. Mech. Appl. **452**, 209–219 (2016)
49. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intell. Syst. **16**(2), 72–79 (2001)
50. Mailavaram, A., Rani, B.P.: Big data: scalability storage. In: Innovations in Computer Science and Engineering, Springer, pp 473–481 (2019)

51. Mallede, W.Y., Marir, F., Vassilev, V.T.: Algorithms for mapping rdb schema to rdf for facilitating access to deep web. In: Proceedings of the First International Conference on Building and Exploring Web Based Environments, pp 32–41 (2013)
52. Malucelli, A., da Costa Oliveira, E.: Ontology-services to facilitate agents' interoperability. In: Proceedings of the Pacific Rim International Workshop on Multi-Agents, Springer, pp 170–181 (2003)
53. Marsden, P.V.: Network centrality, measures of, 2nd edn. International Encyclopedia of the Social and Behavioral Sciences (2015)
54. Mena, E., Illarramendi, A., Kashyap, V., Sheth, A.P.: Observer: an approach for query processing in global information systems based on interoperation across pre-existing ontologies. Distrib. Parallel Databases **8**(2), 223–271 (2000)
55. Mezghani, E., Exposito, E., Drira, K., Da Silveira, M., Pruski, C.: A semantic big data platform for integrating heterogeneous wearable data in healthcare. J. Med. Syst. **39**(12), 185 (2015)
56. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
57. Moawed, S., Algergawy, A., Sarhan, A., Eldosouky, A., Saake, G.: A latent semantic indexing-based approach to determine similar clusters in large-scale schema matching. In: New Trends in Databases and Information Systems, Springer, pp 267–276 (2014)
58. Nadal, S., Romero, O., Abelló, A., Vassiliadis, P., Vansummeren, S.: An integration-oriented ontology to govern evolution in big data ecosystems. Inf. Syst. **79**, 3–19 (2019)
59. Noessner, J., Niepert, M., Stuckenschmidt, H.: Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In: Proceedings of the AAAI Workshop: Statistical Relational Artificial Intelligence (2013)
60. Oldham, S., Fulcher, B., Parkes, L., Arnatkeviciute, A., Suo, C., Fornito, A.: Consistency and differences between centrality measures across distinct classes of networks. arXiv preprint arXiv:180502375 (2018)
61. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: a literature review. Expert Syst. Appl. **42**(2), 949–971 (2015)
62. Peukert, E., Berthold, H., Rahm, E.: Rewrite techniques for performance optimization of schema matching processes. In: Proceedings of the 13th International Conference on Extending Database Technology, ACM, pp 453–464 (2010)
63. Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Linking data to ontologies. In: Journal on data semantics X, Springer, pp 133–173 (2008)
64. Putnik, G., Sluga, A., ElMaraghy, H., Teti, R., Koren, Y., Tolio, T., Hon, B.: Scalability in manufacturing systems design and operation: state-of-the-art and future developments roadmap. CIRP Ann. **62**(2), 751–774 (2013)
65. Rahm, E.: Towards large-scale schema and ontology matching. In: Schema matching and mapping, Springer, pp 3–27 (2011)
66. Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. **62**(1–2), 107–136 (2006)
67. Ruflin, N., Burkhart, H., Rizzotti, S.: Social-data storage-systems. In: Databases and social networks, ACM, pp 7–12 (2011)
68. Sakr, S.: Big Data 2.0 Processing Systems: A Survey. Springer, New York (2016)
69. Sandhya, N., Sekar, M.R.: Analysis of variant approaches for initial centroid selection in k-means clustering algorithm. In: Smart Computing and Informatics, Springer, pp 109–121 (2018)
70. Santipantakis, G., Kotis, K., Vouros, G.A.: Obdair: ontology-based distributed framework for accessing, integrating and reasoning with data in disparate data sources. Expert Syst. Appl. **90**, 464–483 (2017)
71. Schneider, T., Hashemi, A., Bennett, M., Brady, M., Casanave, C., Graves, H., Gruninger, M., Guarino, N., Levenchuk, A., Lucier, E., et al.: Ontology for big systems: the ontology summit 2012 communique. Appl. Ontol. **7**(3), 357–371 (2012)
72. Schuhmacher, M., Ponzetto, S.P.: Ranking entities in a large semantic network. In: Proceedings of the European Semantic Web Conference, Springer, pp 254–258 (2014)
73. Seddiqui, M.H., Aono, M.: An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. Web Seman. **7**(4), 344–356 (2009)
74. Sezer, O.B., Dogdu, E., Ozbayoglu, M., Onal, A.: An extended iot framework with semantics, big data, and analytics. In: Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), IEEE, pp 1849–1856 (2016)
75. Shvachko, K., Kuang, H., Radia, S., Chansler, R., et al.: The hadoop distributed file system. MSST **10**, 1–10 (2010)
76. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1), 158–176 (2013)

77. Siddiqa, A., Hashem, I.A.T., Yaqoob, I., Marjani, M., Shamshirband, S., Gani, A., Nasaruddin, F.: A survey of big data management: taxonomy and state-of-the-art. J. Netw. Comput. Appl. **71**, 151–166 (2016)
78. Siddiqa, A., Karim, A., Gani, A.: Big data storage technologies: a survey. Front. Inf. Technol. Electron. Eng. **18**(8), 1040–1070 (2017)
79. Song, F., Zacharewicz, G., Chen, D.: An analytic aggregation-based ontology alignment approach with multiple matchers. In: Advanced Techniques for Knowledge Engineering and Innovative Applications, Springer, pp 143–159 (2013)
80. Steyskal, S., Polleres, A.: Mix'n'match: An alternative approach for combining ontology matchers. In: Proceedings of the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems", Springer, pp 555–563 (2013)
81. Strohbach, M., Daubert, J., Ravkin, H., Lischka, M.: Big data storage. In: New horizons for a data-driven economy, pp. 119–141. Springer, Cham (2016)
82. Sure, Y., Staab, S., Studer, R.: Methodology for development and employment of ontology based knowledge management applications. ACM Sigmod. Record. **31**(4), 18–23 (2002)
83. Taylor, R.C.: An overview of the hadoop/mapreduce/hbase framework and its current applications in bioinformatics. In: BMC bioinformatics, BioMed Central, vol 11, p S1 (2010)
84. Thorsby, J., Stowers, G.N., Wolslegel, K., Tumbuan, E.: Understanding the content and features of open data portals in american cities. Government Inf. Q. **34**(1), 53–61 (2017)
85. Uzdanaviciute, V., Butleris, R.: Ontology-based foundations for data integration. In: Proceedings of the BUSTECH The First International Conference on Business Intelligence and Technology, Citeseer, pp 34–39 (2011)
86. Van Hage, W.R., Katrenko, S., Schreiber, G.: A method to combine linguistic ontology-mapping techniques. In: Proceedings of the International Semantic Web Conference, Springer, pp 732–744 (2005)
87. Vandecasteele, A., Napoli, A.: Spatial ontologies for detecting abnormal maritime behaviour. In: Proceedings of the OCEANS 2012 MTS/IEEE Yeosu Conference: The Living Ocean and Coast-Diversity of Resources and Sustainable Activities, IEEE-Institute of Electrical and Electronics Engineers, pp 7–pages (2012)
88. Wand, Y., Wang, R.Y.: Anchoring data quality dimensions in ontological foundations. Commun. ACM **39**(11), 86–95 (1996)
89. Wang, P., Zhou, Y., Xu, B.: Matching large ontologies based on reduction anchors. In: Proceedings of the IJCAI, pp 2343–2348 (2011)
90. White, T.: Hadoop: The Definitive Guide. O'Reilly Media Inc., Newton (2012)
91. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp 133–138 (1994)
92. Zamboulis, L., Poulovassilis, A., Wang, J.: Ontology-assisted data transformation and integration. In: Proceedings of the ODBIS, pp 29–36 (2008)
93. Zhou, K., Fu, C., Yang, S.: Big data driven smart energy management: from big data to big insights. Renew. Sustain. Energy Rev. **56**, 215–225 (2016)
94. Zhou, L.: Ontology learning: state of the art and open issues. Inf. Technol. Manage. **8**(3), 241–252 (2007)
95. Zhu, X., Song, B., Ni, Y., Ren, Y., Li, R.: Business Trends in the Digital Era: Evolution of Theories and Applications. Springer, New York (2016)

Authors and Affiliations

Imadeddine Mountasser[1] · Brahim Ouhbi[1] · Ferdaous Hdioud[2] · Bouchra Frikh[2]

Brahim Ouhbi
ouhbib@yahoo.co.uk

Ferdaous Hdioud
ferdaous.hdioud@usmba.ac.ma

Bouchra Frikh
bouchra.frikh@usmba.ac.ma

[1] National Higher School of Arts and Crafts, Industrial Engineering and Productivity Department, Moulay Ismaïl University, Meknes, Morocco

[2] Higher School of Technology, Computer Science Department, Sidi Mohamed Ben Abdellah University, Fez, Morocco