Information Sciences 478 (2019) 606–626

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An approach to extracting complex knowledge patterns among concepts belonging to structured, semi-structured and unstructured sources in a data lake

Paolo Lo Giudice[a], Lorenzo Musarella[a], Giuseppe Sofo[b], Domenico Ursino[c,*]

[a] DIIES, Università "Mediterranea" di Reggio Calabria, Italy
[b] Deloitte & Touche, UK
[c] DII, Università Politecnica delle Marche, Italy

ARTICLE INFO

Article history:
Received 5 April 2018
Revised 19 November 2018
Accepted 22 November 2018
Available online 22 November 2018

Keywords:
Data lakes
Complex knowledge patterns
Network-based conceptual model
Structuring unstructured sources
Synonymies
Shortest paths

ABSTRACT

In this paper, we propose a new network-based model to uniformly represent the structured, semi-structured and unstructured sources of a data lake, which is one of the newest and most successful architectures proposed for managing big data. Then, we present a new approach to, at least partially, "structuring" unstructured sources. Finally, with the support of these two tools, we define a new approach to extracting complex knowledge patterns from the data stored in a data lake.

1. Introduction

In the last few years, the "big data phenomenon" is rapidly changing the research and technological "coordinates" of the information system area [2,46]. For instance, it is well known that data warehouses, generally handling structured and semi-structured data offline, are too complex and rigid to manage the wide amount and variety of rapidly evolving data sources of interest for a given organization, and the usage of more agile and flexible structures appears compulsory [9]. Data lakes are one of the most promising answers to this exigency. Differently from a data warehouse, a data lake uses a flat architecture (so that the insertion and the removal of a source can be easily performed). However, the agile and effective management of data stored therein is guaranteed by the presence of a rich set of extended metadata. These allow a very agile and easily configurable usage of the data stored in the data lake. For instance, if a given application requires the querying of some data sources, one could process available metadata to determine the portion of the involved data lake to examine.

One of the most radical changes caused by the big data phenomenon is the presence of a huge amount of unstructured data. As a matter of fact, it is esteemed that, currently, more than 80% of the information available on the Internet is unstructured [6]. In presence of unstructured data, all the approaches developed in the past for structured and semi-structured data must be "renewed", and the new approaches will be presumably much more complex than the old ones [20,42]. Think,

* Corresponding author.
E-mail addresses: paolo.lo.giudice@unirc.it (P. Lo Giudice), lorenzo.musarella@unirc.it (L. Musarella), gsofo@deloitte.it (G. Sofo), d.ursino@univpm.it (D. Ursino).

https://doi.org/10.1016/j.ins.2018.11.052
0020-0255/

for instance, of schema integration: unstructured sources do not have a representing schema and, often, only a set of keywords are given (or can be extracted) to represent the corresponding content [10].

This paper aims at providing a contribution in this setting. In particular, it proposes an approach to the extraction of complex knowledge patterns among concepts belonging to structured, semi-structured and unstructured sources in a data lake. Here, we use the term "complex knowledge pattern" to indicate an intensional relationship involving more concepts possibly belonging to different (and, presumably, heterogeneous) sources of a data lake. Formally speaking, in this paper, a complex knowledge pattern consists of a logic succession $\{x_1, x_2, \ldots, x_w\}$ of w objects such that there is a semantic relationship (specifically, a synonymy or a part-of relationship) linking the k^{th} and the $(k+1)^{th}$ objects $(1 \leq k \leq w-1)$ of the succession.

Our approach is network-based in that it represents all the data lake sources by means of suitable networks. As a matter of facts, networks are very flexible structures, which allow the modeling of almost all phenomena that researchers aim at investigating. For instance, they have been used in the past to uniformly represent data sources characterized by heterogeneous, both structured and semi-structured, formats [8]. In this paper, we also use networks to represent unstructured sources, which, as said before, do not have a representing schema. Furthermore, we propose a technique to construct a "structured representation" of the flat keywords generally used to represent unstructured data sources. This is a fundamental task because it highly facilitates the uniform management, through our network-based model, of structured, semi-structured and unstructured data sources.

Thanks to this uniform, network-based representation of the data lake sources, the extraction of complex knowledge patterns can be performed by exploiting graph-based tools. In particular, taking into consideration our definition of complex knowledge patterns, a possible approach for their derivation could consist in the construction of suitable paths going from the first node (i.e., x_1) to the last node (i.e., x_w) of the succession expressing the patterns. In this case, a user specifies the "seed objects" of the pattern (i.e., x_1 and x_w) and our approach finds a suitable path (if it exists) linking x_1 to x_w.

Since x_1 and x_w could belong to different sources, our approach should consider the possible presence of synonymies between concepts belonging to different sources, it should model these synonymies by means of a suitable form of arcs (cross arcs, or c-arcs), and should include both intra-source arcs (inner arcs, or i-arcs) and c-arcs in the path connecting x_1 to x_w and representing the complex knowledge pattern of interest.

Among all the possible paths connecting x_1 to x_w, our approach takes the shortest one (i.e., the one with the minimum number of c-arcs and, at the same number of c-arcs, the one with the minimum number of i-arcs). This choice is motivated by observing that a knowledge pattern should be as semantically homogeneous as possible. With this in mind, it is appropriate to reduce as much as possible the number of synonymies to be considered in the knowledge pattern from x_1 to x_w. This because a synonymy is weaker than an identity and, furthermore, it involves objects belonging to different sources which, inevitably, causes an "impairment" in the path going from x_1 to x_w. Moreover, there is a further, more technological reason leading to the choice of the shortest path. Indeed, it is presumable that, after a complex knowledge pattern has been defined and validated at the intensional level, one would like to recover the corresponding data at the extensional level. In this case, in a big data scenario, reducing the number of the sources to query would generally reduce the volume and the variety of the data to process and, hence, would increase the velocity at which the data of interest are retrieved and processed.

As it will be clear in the following, there are cases in which synonymies (and, hence, c-arcs) are not sufficient to find a complex knowledge pattern from x_1 to x_w. In these cases, our approach performs two further attempts in which it tries to involve string similarities first, and, if even these properties are not sufficient, part-whole relationships. If neither synonymies nor string similarities nor part-whole relationships allow the construction of a path from x_1 to x_w, our approach concludes that, in the data lake into consideration, a complex knowledge pattern from x_1 to x_w does not exist.

Summarizing, the main contributions of this paper are the following:

- It proposes a new network-based model to represent the structured, semi-structured and unstructured sources of a data lake.
- It proposes a new approach to, at least partially, "structuring" unstructured sources.
- It proposes a new approach to extracting complex knowledge patterns from the sources of a data lake.

This paper is structured as follows: in Section 2, we illustrate related literature. In Section 3, we present our network-based model for data lakes. In Section 4, we describe our approach to enriching the representation of unstructured data sources in such a way as to, at least partially, "structure" them. In Section 5, we present our approach to the extraction of complex knowledge patterns. In Section 6, we describe some case studies conceived to illustrate the various possible behaviors of our approach. In Section 7, we present a critical discussion of several aspects concerning our approach. Finally, in Section 8, we draw our conclusions.

2. Related literature

In the literature there is a strong agreement in the definition of data lake. For instance, Hai et al. [15] define data lakes as "big data repositories which store raw data and provide functionality for on-demand integration with the help of metadata descriptions". Terrizzano et al. [44] claim that "a data lake is a set of centralized repositories containing vast amounts of raw data (either structured or unstructured), described by metadata, organized into identifiable data

sets, and available on demand". Analogously, Miloslavskaya and Tolstoy [30] say that "a data lake refers to a massively scalable storage repository that holds a vast amount of raw data in its native format ($\ll$ as is $\gg$) until it is needed plus processing systems (engine) that can ingest data without compromising the data structure". Finally, Rangarajan et al. [35] say that "a data lake uses a flat architecture to store data in their raw format. Each data entity in the lake is associated with a unique identifier and a set of extended metadata, and consumers can use purpose-built schemas to query relevant data, which will result in a smaller set of data that can be analyzed to help answer a consumer's question". A step forward, but in the same direction, can be found in [29], where Malysiak-Mrozek et al. introduce the concept of Big Data Lake as "a central location in which users can store all their data in its native form, regardless of its source or format. Big data lake can be used as an environment for the development of in-depth analytics oriented toward fast decision making on the basis of raw data". Clearly, this strong agreement on the data lake definitions does not prevent the possibility to have very different architectures, management approaches and querying techniques in the data lake context, as we will see in the following.

The data lake paradigm requires each raw data to have associated a set of metadata. These represent a key component in the data lake architecture because they let data to be searchable and processed whenever this is necessary [45]. In [11], metadata are also used for bringing quality to a data lake. Here, Farid et al. present CLAMS, a system for discovering integrity constraints from raw data and metadata. To validate obtained results, CLAMS needs human intervention.

In [12], Farrugia et al. propose a data lake management approach that aims at extracting metadata from the Hive database. To reach its objective, it applies Social Network Analysis based techniques. Instead, in [41], iFuse, a data fusion platform that uses a Bayesian graphical model for both managing and querying a data lake, is proposed.

In the literature, there are many approaches to querying and managing both structured and semi-structured data (see [1,8,27,34], to cite a few of them). However, they are generally incapable of managing unstructured data and are not lightweight and flexible enough to be used in the new data lake context. Furthermore, most of the approaches used for representing unstructured data are limited to texts [37]. In order to address this issue, Chen and Co-workers [3,48] propose a generalized data model to represent unstructured data, a method to process it (called RAISE) and an associated SQL-like query language. Foley et al. [13] propose the usage of machine learning for managing and extracting information from unstructured data. They motivate this proposal by observing that, currently, unstructured data represent about the 80% of stored information and, therefore, they must be necessarily processed with a limited human intervention.

The extraction of Complex Knowledge Patterns (CKPs) is a topic widely investigated in the literature. This is due to the fact that CKPs can refer to a lot of research fields and, therefore, their extraction is a challenging issue in several research areas. Research concerning CKPs goes from keyword search and rank (see [7,14,16,22], just to cite a few approaches) to visual knowledge extraction [4,36]. In the literature, a huge variety of approaches to extracting CKPs has been proposed. Some of them are based on Network Analysis [47], others are centered on "questions and answers" mechanisms [18], further ones exploit Similarity Join [39], and so forth. Each family of approaches has its pros and cons, as well as its corresponding tools [40].

As for the approaches most related to ours, there are four main families that we need to investigate, namely: *(i)* extraction of keyword patterns; *(ii)* extraction of knowledge from structured sources; *(iii)* extraction of knowledge from heterogeneous sources; *(iv)* extraction of knowledge patterns through network analysis-based approaches.

As far as the first family is concerned, it is necessary to further differentiate the corresponding approaches. A first subfamily focuses on RDF analysis. In this context, several proposals can be found in the literature. For instance, in [7], approaches to keyword search inside RDF data are proposed. These approaches exploit user feedback to relax the search constraints and to identify a higher number of matches. Han et al. [16] build a bipartite graph from RDF data and aim at solving a graph assembly problem. Since this problem is NP-hard, they propose two heuristics for facing it. In [31], models letting knowledge patterns to be represented by means of RDF are investigated. The second sub-family, instead, aims at extracting keyword patterns in a graph database. In [17], He et al. propose BLINKS, a system consisting of an algorithm for bi-level indexing and a query processor useful for searching the top-k keywords in a graph. In [14], an engine for enumerating keywords and evaluating their search in a data graph is proposed. In the same way, in [22], EASE, a framework allowing indexing and keyword querying, is described.

As for the second family, most of the corresponding approaches are summarized in [25]. Here, the authors claim that, thanks to metadata, it is possible to think of a new, completely automated, approach.

As for the third family, in [5], Chen et al. provide an overview of techniques used in the literature to support keyword search in structured and semi-structured data. In [24], Liu et al. operate on semi-structured sources and try to make the extraction process as automated as possible. More recent approaches try to extract knowledge from heterogeneous sources. In fact, as evidenced in [23], the big data phenomenon led to the creation of a lot of heterogeneous sources that include unstructured data. These need to be integrated exactly as it was made before for structured data. Starting from this consideration, the authors analyze the most important challenges introduced by this new reality and present a unique query format taking this issue into account. In [36], Sadeghi et al. assert that, in order to have a user-friendly graph query engine, it is necessary to support different kinds of task, like synonymy detection and ontology usage. Based on this assertion, they propose a framework allowing these operations on data without schema or structure. In [39], Shang et al. argue that Similarity Join is a fundamental operation for clearing data and integrating different sources. It involves two big challenges, namely quantifying knowledge aware similarities and identifying similarity pairs efficiently. To address these issues, they propose a

new framework. Likewise, in [43], a system to integrate different sources through keyword search, and an evaluation system based on user feedback, are proposed.

The last family of approaches is based on network analysis. In [38], network communities and the apriori algorithm are used to identify rhythmic knowledge patterns of musical work. In [26], Lo Giudice et al. represent patent data as a network and, then, propose a new approach that analyzes this network for extracting CKPs about patent applicants. In [28], Mallat et al. propose a new formalism to represent a knowledge base through a network whose edges denote the semantic proximity between two or more concepts. This representation allows the discovery of association models among different concepts. In [19], Kargar and An propose an algorithm that uses the cliques in a graph for searching the keywords linked to a given input. According to what the authors claim, keyword search is necessary because it facilitates the identification of sub-graphs in a network.

3. A network-based model for data lakes

In this section, we illustrate our network-based model to represent and handle a data lake, which we will use in the rest of this paper.

In our model, a data lake DL is represented as a set of m data sources:

$DL = \{D_1, D_2, \cdots, D_m\}$

A data source $D_k \in DL$ is provided with a rich set $\mathcal{M}_k$ of metadata. We denote with $\mathcal{M}_{DL}$ the repository of the metadata of all the data sources of DL:

$\mathcal{M}_{DL} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m\}$

According to Oram [33], our model represents $\mathcal{M}_k$ by means of a triplet:

$\mathcal{M}_k = \langle \mathcal{M}_k^T, \mathcal{M}_k^O, \mathcal{M}_k^B \rangle$

Here:

- $\mathcal{M}_k^T$ denotes *technical metadata*. They represent the type, the format, the structure and the schema of the corresponding data. They are commonly provided by the source catalogue.
- $\mathcal{M}_k^O$ represents *operational metadata*. They include the source and target locations of the corresponding data, the associated file size, the number of their records, and so on. Usually, they are automatically generated by the technical framework handling the data lake.
- $\mathcal{M}_k^B$ indicates *business metadata*. They comprise the business names and descriptions assigned to data fields. They also cover business rules, which can become integrity constraints for the corresponding data source.

Since our approach focuses on the semantics of data sources, in this paper, we consider only business metadata. Indeed, they denote, at the intensional level, the information content stored in $\mathcal{M}_k$ and are those of interest for supporting the extraction of complex knowledge patterns from a data lake, which is our ultimate goal.

Our model adopts a notation typical of XML, JSON and many other semi-structured models to represent $\mathcal{M}_k^B$. According to this notation, Obj_k indicates the set of all the objects stored in $\mathcal{M}_k^B$. It consists of the union of three subsets:

$Obj_k = Att_k \cup Smp_k \cup Cmp_k$

Here:

- Att_k indicates the set of the attributes of $\mathcal{M}_k^B$;
- Smp_k represents the set of the simple elements of $\mathcal{M}_k^B$;
- Cmp_k denotes the set of the complex elements of $\mathcal{M}_k^B$.

Here, the meaning of the terms “attribute”, “simple element” and “complex element” is the one typical of semi-structured data models.

$\mathcal{M}_k^B$ can be also represented as a graph:

$\mathcal{M}_k^B = \langle N_k, A_k \rangle$

N_k is the set of the nodes of $\mathcal{M}_k^B$. There exists a node $n_{k_j} \in N_k$ for each object $o_{k_j} \in Obj_k$. According to the structure of Obj_k, N_k consists of the union of three subsets:

$N_k = N_k^{Att} \cup N_k^{Smp} \cup N_k^{Cmp}$

Here, N_k^{Att} (resp., N_k^{Smp}, N_k^{Cmp}) indicates the set of the nodes corresponding to Att_k (resp., Smp_k, Cmp_k). There is a biuniovocal correspondence between a node of N_k and an object of Obj_k. Therefore, in the following, we will use the two terms interchangeably.

Let x be a complex element of $\mathcal{M}_k^B$. Obj_{k_x} indicates the set of the objects directly contained in x, whereas $N_{k_x}^{Obj}$ denotes the set of the corresponding nodes. Furthermore, let x be a simple element of $\mathcal{M}_k^B$. Att_{k_x} represents the set of the attributes directly contained in x, whereas $N_{k_x}^{Att}$ denotes the set of the corresponding nodes.

A_k indicates the set of the arcs of $\mathcal{M}_k^B$. It consists of three subsets:

$A_k = A'_k \cup A''_k \cup A'''_k$

Here:

- $A'_k = \{(n_x, n_y) | n_x \in N_k^{Cmp}, n_y \in N_{n_x}^{Obj}\}$. This definition specifies that there is an arc from a complex element of $\mathcal{M}_k^B$ to each object directly contained in it.
- $A''_k = \{(n_x, n_y) | n_x \in N_k^{Smp}, n_y \in N_{n_x}^{Att}\}$. This definition specifies that there is an arc from a simple element of $\mathcal{M}_k^B$ to each attribute directly contained in it.
- $A'''_k = \{(n_x, n_y) | n_x \in N_k, n_y \in N_k, D_k$ is unstructured and there exists a correlation between n_x and $n_y\}$. The meaning of A'''_k will be clear after reading Section 4, where we illustrate our approach for "structuring" unstructured data.

Interestingly, our data lake formalization uses a model similar to the one adopted in [29]. Here, a data lake is defined as a pair $DL = \{V, M\}$, where V is a set of values in the data lake and M is a set of metadata describing the values of DL. In this definition, the authors introduce the concept of fully description in terms of attribute names and data types. This definition is similar to the components $\mathcal{M}_k^T$ and $\mathcal{M}_k^O$ of our model's metadata. However, the two approaches present several differences because our own also introduces the concept of business metadata (thus enriching the data description component), whereas the approach of Malysiak-Mrozek et al. [29] proceeds with the formal definition of the Extract, Process and Store (EPS) process (thus enriching the process description component).

4. Enriching the representation of unstructured data

Our network-based model for representing and handling a data lake is perfectly fitted for representing and managing semi-structured data because it has been designed having XML and JSON in mind. Clearly, it is sufficiently powerful to represent structured data. The highest difficulty regards unstructured data because it is worth avoiding a flat representation consisting of a simple element for each keyword provided to denote the source content. As a matter of fact, this kind of representation would make the reconciliation, and the next integration, of an unstructured source with the other (semi-structured and structured) ones of the data lake very difficult. Therefore, it is necessary to (at least partially) "structure" unstructured data.

Our approach to addressing this issue creates a complex element for representing the source as a whole and a simple element for each keyword. Furthermore, it adds an arc from the source to each of the simple elements. Initially, there is no arc between two simple elements. To determine the arcs to add, our approach exploits lexical and string similarities.

In particular, lexical similarity is considered by stating that there exists an arc from the node n_{k_1}, corresponding to the keyword k_1, to the node n_{k_2}, corresponding to the keyword k_2 (and vice versa), if k_1 and k_2 have at least one common lemma[1] in a suitable thesaurus. Taking the current trends into account, this thesaurus should be a multimedia one; for this purpose, in our experiments, we have adopted BabelNet [32]. When this pair of arcs has been added, n_{k_1} and n_{k_2} must be considered complex elements, instead of simple elements.

String similarity is applied by stating that there exists an arc from n_{k_1} to n_{k_2} (and vice versa) if the string similarity degree $kd(k_1, k_2)$, computed by applying a suitable string similarity metric on k_1 and k_2, is "sufficiently high" (see below). We have chosen N-Grams [21] as string similarity metric because we have experimentally seen that it provides the best results in our context. Also in this case, when this pair of arcs has been added, n_{k_1} and n_{k_2} change their types from simple elements to complex ones. Now, we illustrate in detail what "sufficiently high" means and how our approach operates. Let *KeySim* be the set of the string similarities for each pair of keywords of the source into consideration. Each record in *KeySim* has the form $\langle k_i, k_j, kd(k_i, k_j)\rangle$. Our approach first computes the maximum keyword similarity degree $kd_{\max}$ present in *KeySim*. Then, it examines each keyword similarity registered therein. Let $\langle k_1, k_2, kd(k_1, k_2)\rangle$ be one of these similarities. If $((kd(k_1, k_2) \geq th_k \cdot kd_{\max}) and\ .(kd(k_1, k_2) \geq th_{kmin}))$, which implies that the keyword similarity degree between k_1 and k_2 is among the highest ones in *KeySim* and that, in any case, it is higher than or equal to a minimum threshold, then an arc is added from n_{k_1} to n_{k_2}, and vice versa. We have experimentally set $th_k = 0.70$ and $th_{kmin} = 0.50$.

From this description, it emerges that, given two nodes n_{k_1} and n_{k_2}, corresponding to two keywords k_1 and k_2 of the unstructured source, four cases may exist, namely: *(1)* no arcs link n_{k_1} and n_{k_2}; *(2)* an arc derived from an object similarity links them; *(3)* an arc derived from a string similarity links them; *(4)* an arc derived from both an object and a string similarity links them.

In our approach, the component devoted to "structuring" unstructured data, which we are describing in this section, plays a key role. On the other hand, this last issue has been investigated in the recent past. For instance, in Section 3, we have cited the approach of Malysiak-Mrozek et al. [29] and we have seen that an important component of this approach is the EPS (Extract, Process and Store) process. The management of unstructured data is performed during the extract subtask of this process, when data are extracted from the data lake. This last is represented as a pair consisting of values and metadata. Instead, dataset schemas are dynamically defined, according to the "schema on read" approach. Starting from these three elements, the approach of Malysiak-Mrozek et al. [29] generate a rowset with n attributes.

To correctly interpret data and/or metadata of unstructured sources, in the construction of rowset, the approach of Malysiak-Mrozek et al. [29] use some transformation rules allowing the extraction and/or the correction of values and data acquired from the involved sources. To perform this task, the approach uses U-SQL and fuzzy logics. As a consequence of

[1] In this paper, we use the term "lemma" according to the meaning it has in BabelNet [32]. Here, given a term, its lemmas are other objects (terms, emoticons, etc.) semantically associated with it and, therefore, contributing to specify its meaning.

this way of proceeding, it can generate a tabular representation (i.e., the rowset) from unstructured data. The content of the rowset depends on the membership function associated with the fuzzy logic and on the possible constraints regarding it.

Our approach operates in a different way. Indeed, to perform structuring of unstructured sources, it leverages network analysis, as well as lexical and string similarities. In fact, unstructured sources are "structured" thanks to the addition of the arcs in the networks representing the sources themselves. These arcs can be created only when the similarity between nodes is higher than a certain degree. Interestingly, in both approaches, the final result of the structuring activity depends on a threshold.

The approach of Malysiak-Mrozek et al. [29] address the data variety issue by extending the operations that can be performed on unstructured sources by means of fuzzy techniques. These carry out the structuring task, and the consequent rowset creation, by means of an interface for the dataset extraction, which is unified and valid for all the sources. By contrast, our approach bases the structuring activity on business constraints involving the schemas of the data lake sources and on lexical and string similarities among the elements represented therein.

4.1. Example

Consider an unstructured data source consisting of a video about environment and pollution. As we said before, for each unstructured source, our approach begins from a set of keywords representing its content. In order to keep our description simple and clear, in this example, we assume that our video has a limited number of keywords, namely the ones shown in Fig. 1.

First, as we can see in Fig. 1(a), our approach constructs a graph having a node for each keyword. A further node is added to represent the video as a whole; nodes corresponding to keywords are colored in red, whereas the other one is colored in green. Following our strategy, in Fig. 1(b), we added an arc from the node representing the whole video to each node associated with a keyword. The next step consists of using BabelNet. In Fig. 1(c), we show two keywords ("Save" and "Protect") and the corresponding lemmas in BabelNet. Common lemmas (i.e., "keep" and "preserve") are in bold. Since "Save" and "Protect" have at least one common lemma, two arcs are added between the corresponding nodes in Fig. 1(d). These arcs are highlighted in blue in this figure and, due to layout reasons, we report only one arc with two arrows, instead of two arcs with one arrow. Each arc has a label representing the number of common lemmas between the corresponding keywords in BabelNet. After having added the new arcs, caused by the common lemmas present in BabelNet, we proceed by analyzing string similarities. In Fig. 1(e), we report the pairs of keywords that satisfy this feature. In Fig. 1(f), we add a pair of arcs for each pair of keywords of Fig. 1(e). Again, these arcs are highlighted in blue and, due to layout reasons, we report only one arc with two arrows, instead of two arcs with one arrow. Each arc has a label representing the string similarity degree (computed by means of N-Grams) between the corresponding keywords. Finally, in Fig. 1(g), we combine the arcs derived in the previous two steps. Clearly, it may happen that, for a pair of keywords (see, for instance, the keywords "garden" and "gardens"), two pairs of arcs have been generated, one in Fig. 1(d) and one in Fig. 1(f). In this case, in Fig. 1(g), we do not report two pairs of arcs; instead, we report only one pair, representing both of them. The label of this pair is obtained by merging the labels of the two corresponding pairs.

5. Extraction of complex knowledge patterns

5.1. General description of the approach

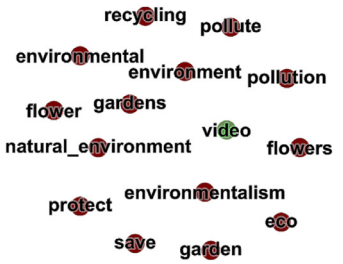
Our approach to the extraction of complex knowledge patterns operates on a data lake *DL* whose data sources are represented by means of the formalism described in Section 4.

It receives a dictionary *Syn* of synonymies involving the objects stored in the sources of *DL*. This dictionary could be a generic thesaurus, such as BabelNet [32], a domain-specific thesaurus, or a dictionary obtained by taking into account the structure and the semantics of the sources, which the corresponding objects refer to (such as the dictionaries produced by XIKE [8], MOMIS [1] or Cupid [27]). Furthermore, it receives two objects x_{i_h}, belonging to a source D_h of *DL*, and x_{j_q}, belonging to a source D_q of *DL*. x_{i_h} and x_{j_q} represent the base on which the complex knowledge pattern to extract should be constructed. As a matter of fact, we recall that, in this paper, a complex knowledge pattern consists of a logic succession $\{x_1, x_2, \ldots, x_w\}$ of w objects such that: *(i)* x_1 coincides with x_{i_h}; *(ii)* x_w coincides with x_{j_q}; *(iii)* there is a semantic relationship (specifically, a synonymy or a part-of relationship) linking the k^{th} and the $(k+1)^{th}$ objects ($1 \leq k \leq w-1$) of the succession.

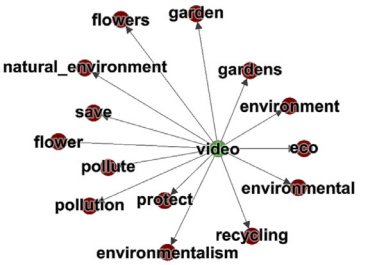
Before illustrating our approach in detail, it is necessary to introduce some preliminary concepts, namely the ones of i-arcs, c-arcs and pw-arcs. In Section 4, we have seen that, given a source D_k of *DL*, $\mathcal{M}_B^k = \langle N_k, A_k \rangle$ and $A_k = A'_k \cup A''_k \cup A'''_k$. All the arcs of A_k are internal to D_k; we call them *i-arcs* (i.e., internal arcs) in the following. Now, let x_{i_h} and x_{j_q} be two objects for which a synonymy exists in *Syn*. We call *c-arcs* (i.e., cross arcs) the arcs (n_{i_h}, n_{j_q}) and (n_{j_q}, n_{i_h}) corresponding to this synonymy. These arcs are extremely important because they allow the extraction, the processing and the management of information coming from different sources. Finally, given an arc $(n_{i_k}, n_{j_k}) \in A'_k \cup A''_k$, we call *pw-arc* (i.e., part-whole arc) the arc (n_{j_k}, n_{i_k}). The pw-arc (n_{j_k}, n_{i_k}) is the "reverse" arc of (n_{i_k}, n_{j_k}) because it starts from the part and ends to the whole[2]

[2] In order to keep the layout simple, in the graphical representation of our model, we explicitly show only the arcs from the whole to the parts; the corresponding pw-arcs are considered implicit.

(a)



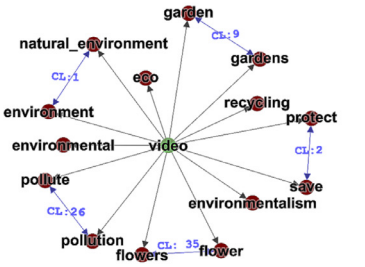
(b)



(c)

	Lemmas
Save	blown_save, bring_through, carry_through, conditional_salvation, deliver, economise, economize, file, hold_open, individual_salvation, kaligtasan, **keep**, keep_open, lay_aside, make_unnecessary, noar, noarus, **preserve**, proprietary_salvation, pull_through, redeem, relieve,rescued, salvage, salvation, sava, save, save_situation, save_up, saved, saving, savior, saviour, savus, sin_and_salvation, spare, special_salvation, svop, szva, tough_save, write
Protect	defend, guard, **keep**, **preserve**, protect, protect_(organisation), protect_(organization), protect_(political_organization), The_national_association_to_protect_children

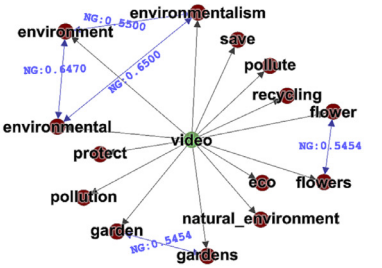
(d)



(e)

k_1	k_2	$kd(k_1, k_2)$
environmentalism	environmental	0.6500
environmental	environment	0.6470
environment	environmentalism	0.5500
flower	flowers	0.5454
gardens	garden	0.5454

(f)



(g)

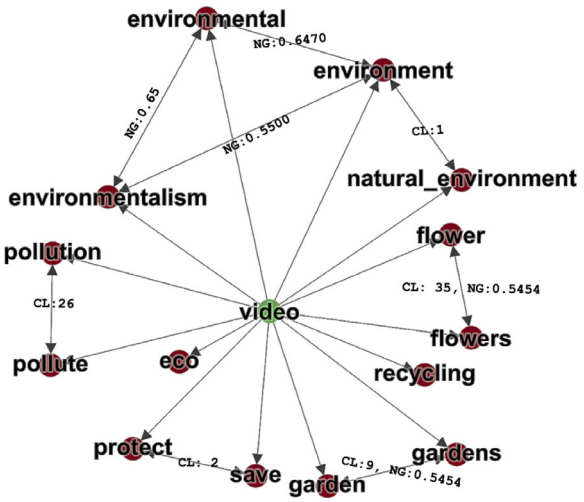


Fig. 1. Graphical representation of our approach to deriving a "structure" for an unstructured source.

The name of this arc clearly indicates its nature. As it is evident from the definition of A'_k and A''_k, the i-arc (n_{i_k}, n_{j_k}) specifies the existence of a part-of relationship, from the whole (n_{i_k}) to the part (n_{j_k}). The arc (n_{j_k}, n_{i_k}) is the reverse one going from the part to the whole.

Our approach operates as follows. Let n_{i_h} (resp., n_{j_q}) be the node corresponding to x_{i_h} (resp., x_{j_q}).

- If $h = q$, we have a trivial case and the complex knowledge pattern from n_{i_h} to n_{j_q} is obtained by selecting the set of the arcs belonging to the shortest path from n_{i_h} to n_{j_q}.
- If $h \neq q$, then c-arcs and pw-arcs must be considered. First, our approach determines the set of complex knowledge patterns each formed by a c-arc from n_{i_h} to a node n_{t_l} synonymous of n_{i_h}, followed by a complex knowledge pattern from n_{t_l} to n_{j_q}. Then, it determines the set of complex knowledge patterns each formed by an i-arc from n_{i_h} to a node n_{t_h}, being a part of n_{i_h}, followed by a complex knowledge pattern from n_{t_h} to n_{j_q}. If at least one of these two sets is not empty, it returns the complex knowledge pattern having the minimum number of c-arcs.

If both these sets are empty, then our approach performs a last attempt to find a complex knowledge pattern by considering pw-arcs having n_{i_h} as target, if they exist. In this case, it determines the set of complex knowledge patterns each formed by a pw-arc from n_{i_h} to a node n_{t_h} followed by a complex knowledge pattern from n_{t_h} to n_{j_q}. If this set is not empty, it returns the complex knowledge pattern having the minimum number of pw-arcs.

5.2. Technical details

As previously pointed out, our approach operates on a data lake *DL*. It receives a dictionary *Syn* of synonymies regarding the objects of *DL*, along with two objects x_{i_h}, belonging to a source D_h of *DL*, and x_{j_q}, belonging to a source D_q of *DL*. Let n_{i_h} (resp., n_{j_q}) be the node corresponding to x_{i_h} (resp., x_{j_q}), then the computation of $CKP(n_{i_h}, n_{j_q})$, i.e. of the complex knowledge pattern from n_{i_h} to n_{j_q}, is recursively performed as follows:

- If $h = q$, x_{i_h} and x_{j_q} belong to the same source and, therefore, n_{i_h} and n_{j_q} belong to the same network. In this case, the complex knowledge pattern $CKP(n_{i_h}, n_{j_q})$ from n_{i_h} to n_{j_q} is obtained by selecting the set of the arcs belonging to the shortest path from n_{i_h} to n_{j_q}. Any algorithm previously proposed in the literature for computing the shortest path between two nodes can be adopted.
- If $h \neq q$, then n_{i_h} and n_{j_q} belong to different networks.
  First, the set $NSynSet_{i_h}$ of the nodes corresponding to the objects synonymous of x_{i_h} in *Syn* is determined as:
  $NSynSet_{i_h} = \{n_{t_l} \mid (n_{i_h}, n_{t_l}) \in Syn\}$
  Then, the set $CKPSynSet(n_{i_h}, n_{j_q})$ of the complex knowledge patterns from n_{i_h} to n_{j_q} and passing through a node of $NSynSet_{i_h}$ is computed. Formally:
  $CKPSynSet(n_{i_h}, n_{j_q}) = \{SynCKP(n_{i_h}, n_{j_q}, n_{t_l}) | n_{t_l} \in NSynSet_{i_h}\}$
  where:
  $SynCKP(n_{i_h}, n_{j_q}, n_{t_l}) = \{(n_{i_h}, n_{t_l}) \cup CKP(n_{t_l}, n_{j_q})\}$
  After this, the set $NPartSet_{i_h}$ of the nodes representing a part of n_{i_h} (which, in this case, is the whole) is determined as:
  $NPartSet_{i_h} = \{n_{t_h} | (n_{i_h}, n_{t_h}) \in A'_h \cup A''_h\}$
  Then, the set $CKPPartSet(n_{i_h}, n_{j_q})$ of the complex knowledge patterns from n_{i_h} to n_{j_q} and passing through a node of $NPartSet_{i_h}$ is computed. Formally:
  $CKPPartSet(n_{i_h}, n_{j_q}) = \{PartCKP(n_{i_h}, n_{j_q}, n_{t_h}) | n_{t_h} \in NPartSet_{i_h}\}$
  where:
  $PartCKP(n_{i_h}, n_{j_q}, n_{t_h}) = \{(n_{i_h}, n_{t_h}) \cup CKP(n_{t_h}, n_{j_q})\}$
  If $CKPSynSet(n_{i_h}, n_{j_q}) \neq \emptyset$ and/or $CKPPartSet(n_{i_h}, n_{j_q}) \neq \emptyset$, our approach selects as $CKP(n_{i_h}, n_{j_q})$ the complex knowledge pattern having the minimum number of c-arcs. If more than one pattern exists with the same minimum number of c-arcs, it returns the one with the minimum number of i-arcs. If more than one pattern exists with these characteristics, it randomly selects one of them.
  If $CKPSynSet(n_{i_h}, n_{j_q}) = \emptyset$ and $CKPPartSet(n_{i_h}, n_{j_q}) = \emptyset$, then c-arcs are not sufficient to find a complex knowledge pattern from n_{i_h} to n_{j_q}. However, a last attempt to find such a pattern can be performed by exploiting a pw-arc having n_{i_h} as target, if it exists.
  In particular, let $NWholeSet_{i_h}$ be the set of the nodes of which n_{i_h} is a part. It is determined as:
  $NWholeSet_{i_h} = \{n_{t_h} | (n_{t_h}, n_{i_h}) \in A'_h \cup A''_h\}$
  Then, if $NWholeSet_{i_h} = \emptyset$, there is no possibility to find a complex knowledge pattern from n_{i_h} to n_{j_q}. Otherwise, the set $CKPWholeSet(n_{i_h}, n_{j_q})$ of the complex knowledge patterns between n_{i_h} and n_{j_q} and passing through a node of $NWholeSet_{i_h}$ is computed. Formally:
  $CKPWholeSet(n_{i_h}, n_{j_q}) = \{WholeCKP(n_{i_h}, n_{j_q}, n_{t_h}) | n_{t_h} \in NWholeSet_{i_h}\}$
  where:
  $WholeCKP(n_{i_h}, n_{j_q}, n_{t_h}) = \{(n_{i_h}, n_{t_h}) \cup CKP(n_{t_h}, n_{j_q})\}$
  Once $WholeCKP(n_{i_h}, n_{j_q}, n_{t_h})$ has been constructed, if it is not empty, our approach selects as $CKP(n_{i_h}, n_{j_q})$ the complex knowledge pattern having the minimum number of pw-arcs. If more than one pattern exists with the same minimum number of pw-arcs, it returns the one with the minimum number of c-arcs. If more than one pattern exists with these

characteristics, it selects the one with the minimum number of i-arcs. Finally, if more than one pattern exists with the same minimum number of i-arcs, it randomly selects one of them.

5.2.1. Computational complexity

As for the computational complexity of this approach, we can observe that:

- If $h = q$, any algorithm previously proposed in the literature for computing the shortest path between two nodes can be adopted. For instance, if the Dijkstra algorithm using a binary heap is implemented, the computational complexity of this step is $O(|A| \cdot \log |N|)$, where $|A|$ is the total number of arcs of the data lake and $|N|$ is the total number of its nodes.
- If $h \neq q$, in the worst case, it is necessary to determine the sets $NSynSet_{i_h}$, $NPartSet_{i_h}$ and $NWholeSet_{i_h}$ and, then, for each node of these sets, to compute the shortest path from n_i to n_j bounded to pass through it.
  Now, $|NSynSet_{i_h}|$ is $O(|DL|)$ because there could be at most one synonymous of a node for each source. $|NPartSet_{i_h}|$ is $O(|N_{\max}|)$, where $|N_{\max}|$ is the number of nodes of the largest source of the data lake. For the same reason, $|NWholeSet_{i_h}|$ is $O(|N_{\max}|)$.
  The complexity of the computation of the shortest path from n_i to n_j bounded to pass through a node is $O(|A| \cdot \log(|N|)$, if the Dijkstra algorithm with the support of the binary heap is applied.
  Therefore, in this case, the computational complexity of the algorithm is:
  $O(|A| \cdot \log |N|) \cdot O(max(|N_{\max}|, |DL|))$
  Now, since generally $|N_{\max}| \gg |DL|$, we have that the computational complexity of this step is:
  $O(|A| \cdot \log |N| \cdot |N_{\max}|)$

Since the computational complexity of the case $h \neq q$ is higher than the one of the case $h = q$, we can conclude that the overall computational complexity of our approach is $O(|A| \cdot \log |N| \cdot |N_{\max}|)$.

This computational complexity can be judged very good if we consider the problem to solve. Actually, one could say that it is high for real data lakes consisting of many sources. However, in these cases, we have that, in the reality, the corresponding graphs are very sparse and, therefore, $|A|$ is small. To better quantify this fact, in Section 7.3, we compare the theoretical and the real computation time of our approach against the number of nodes of the data lake.

6. Some case studies

In this section, we present some case studies devoted to illustrate the behavior of our approach in the various possible cases. To perform our test cases, we constructed a data lake consisting of 2 structured sources, 4 semi-structured sources (i.e., 2 XML sources and 2 JSON ones) and 4 unstructured sources (i.e., 2 books and 2 videos). All these sources store data about environment and pollution. To describe unstructured sources, we initially considered a set of keywords derived from Google Books, for books, and from YouTube, for videos. The interested reader can find the schemas, in case of structured and semi-structured sources, and the keywords, in case of unstructured sources, at the address http://www.barbiana20.unirc.it/dls/datasets/dl2. The password to type is "`za.12&;lq74:#`".

The case studies we present in this section involve five sources of our data lake. These sources are the following:

- *Climate*. This is a JSON source storing data about weather and climatic conditions of several places. In this dataset, space and time are expressed at several granularity levels. In particular, time is expressed in years, seasons and days, whereas space is expressed in countries and cities; these last ones have associated the corresponding latitude and longitude. The network representing *Climate* is reported in Fig. 2.
- *Energy*. This is a JSON source storing data about renewable and non-renewable energy sources used in the countries worldwide. Energy also stores data about the investments on the various kinds of energy source. The network representing *Energy* is reported in Fig. 3.
- *Environment disasters*. This is an XML source storing data about environment disasters (e.g., earthquakes, seaquakes, volcanic eruptions, etc.), the places where they happened, the damages caused by them, and so forth. The network representing *Environment disasters* is reported in Fig. 4.
- *Environment risks*. This is a book discussing about environment risks, their probabilities, their damages, etc. This is an unstructured source and, as such, it is represented by a set of keywords, which is reported in Table 1.
- *Air pollution*. This is a book focusing on air pollution, its causes, its consequences and the possible control strategies that can mitigate their impact on the environment. This is an unstructured source. Its keywords are reported in Table 2.

The first case study we are considering is very simple because the two "seed objects" of the complex knowledge pattern are `Energy` and `Population`, both belonging to the source *Energy*. Since both the source and the target node of the knowledge pattern belong to the same network, the pattern is obtained simply by computing the shortest path from `Energy` to `Population` in the network of Fig. 3. Actually, in this case, we have only one possible path, shown in Fig. 5. This path consists of 4 i-arcs, no c-arcs and no pw-arcs.

The second case study we are considering involves as "seed objects" `Position`, belonging to *Environment disasters*, and `Energy`, belonging to *Energy*. In this case, it is evident the necessity of passing through at least one c-arc because the two objects belong to different sources. One of the synonyms of the object `Position` is the object `Place`, belonging to the source *Energy*. As a consequence, it is possible to define at least one path, starting from `Position`, passing through

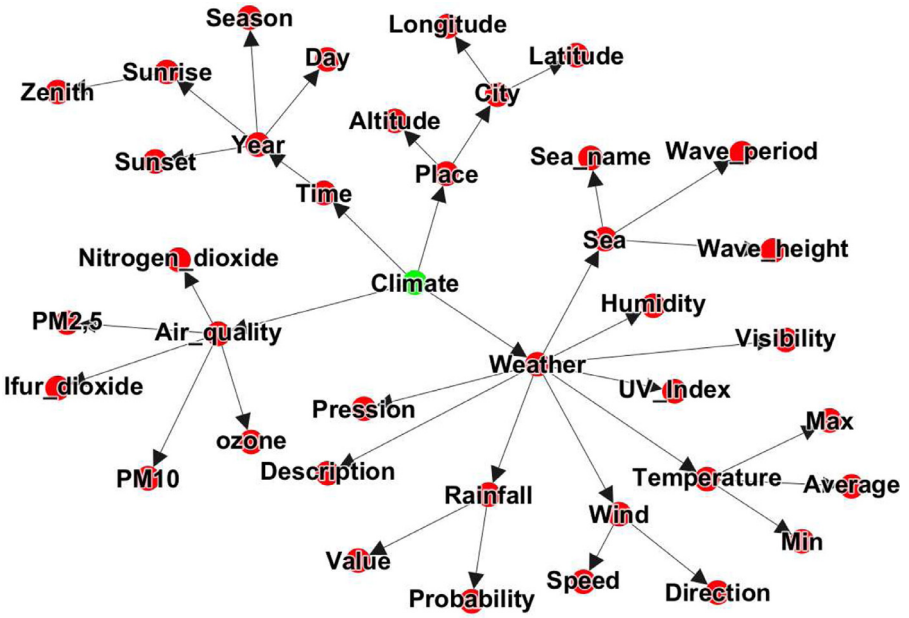


Fig. 2. The network corresponding to the source *Climate*.

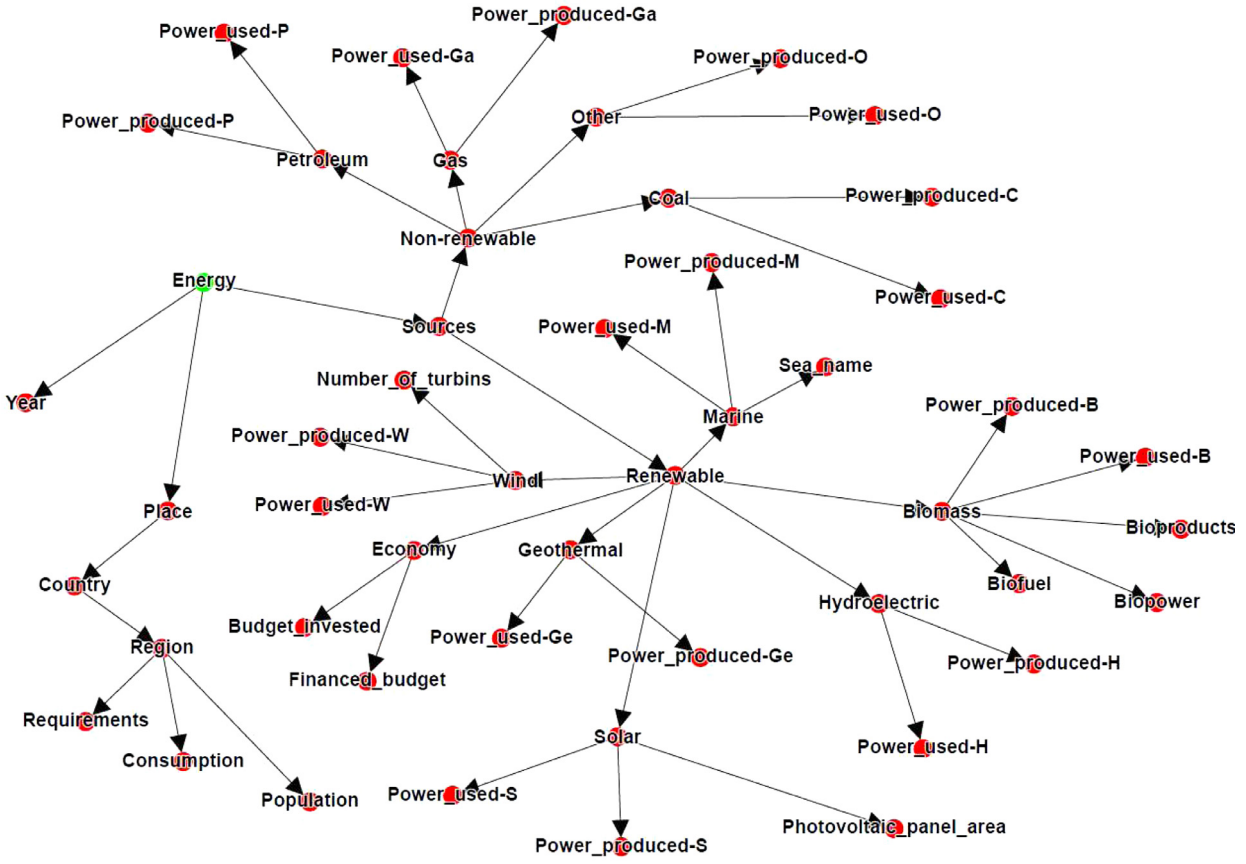


Fig. 3. The network corresponding to the source *Energy*.

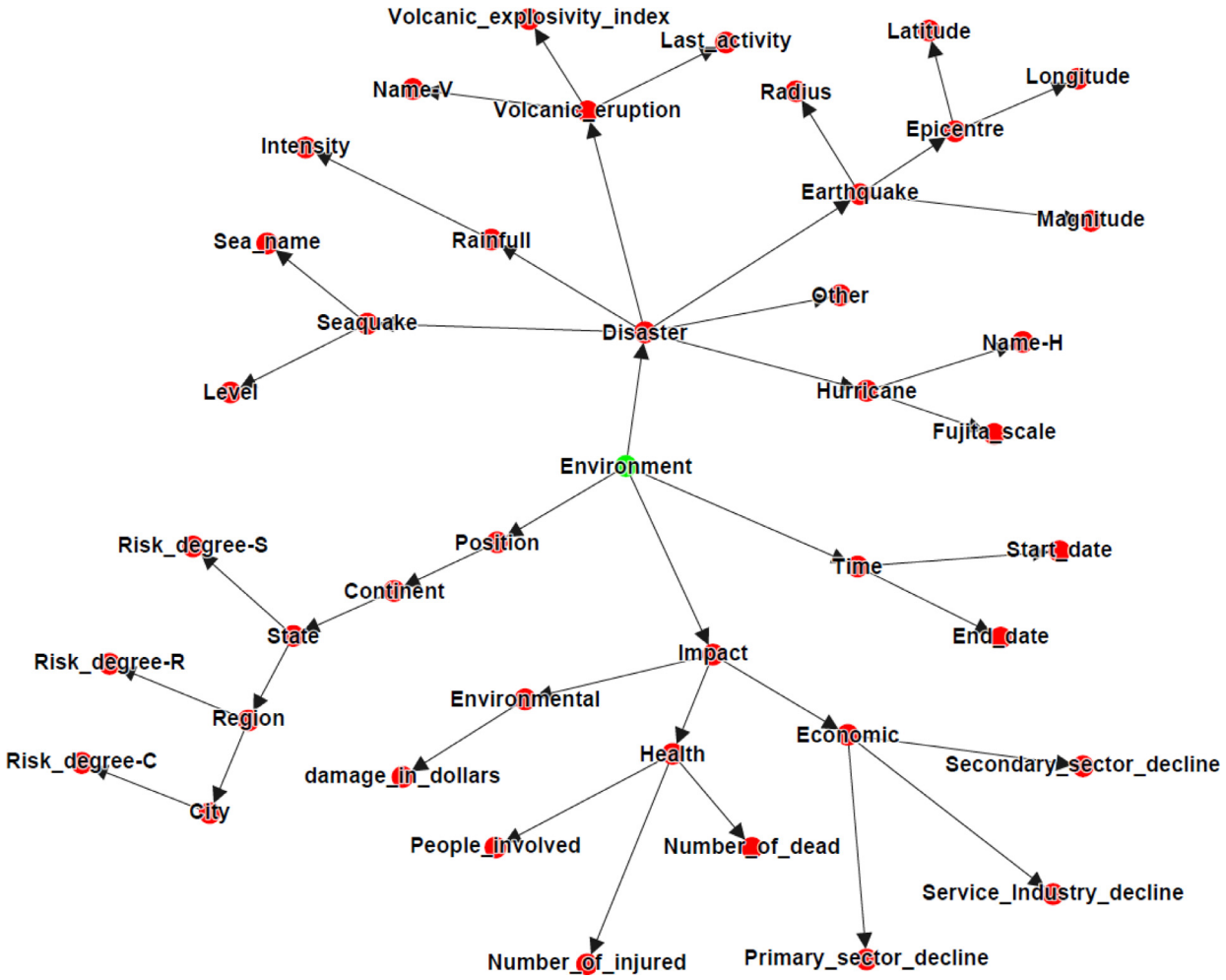


Fig. 4. The network corresponding to the source *Environment disasters*.

`Place` and reaching `Energy`. This path is shown in Fig. 6 and consists of 1 i-arc, 1 c-arc and no pw-arc. An alternative path would involve the nodes `Position` and `Continent` of *Environment disasters* and the nodes `Country`, `Place` and `Energy` of *Energy*. However, this path would consist of 3 i-arcs, 1 c-arc and no pw-arc and, clearly, it is not the shortest path. As a consequence, in this case, our approach returns the path shown in Fig. 6 as the complex knowledge pattern from `Position` to `Energy`.

The third case study we are considering involves, as "seed objects", `Fujita_scale` of *Environment disasters* and `Risk` of *Environment risks*. In this case, synonymies are not sufficient because there is no synonymy involving `Fujita_scale`. However, the "whole" node of `Fujita_scale` is `Hurricane` and there is a synonymy between `Hurricane` and `Tornado`. As a consequence, it is possible to define at least one path starting from `Fujita_scale`, passing through `Hurricane` and `Tornado` and ending to `Risk`. This path is shown in Fig. 7. It consists of 1 i-arc, 1 c-arc and 1 pw-arc. This is also the shortest path from `Fujita_scale` to `Risk` and, therefore, the complex knowledge pattern between these two nodes.

The fourth and last case study is the most complex one because it involves more alternative synonymies that could be selected, with the consequent need to determine the best one. The "seed objects" are `Risk_degree` of *Environment disasters* and `Emergency` of *Environment risks*. `Risk_degree` presents two synonymies in the dictionary; the former involves the object `Risk` of *Environment risks*; the latter regards the object `Risk` of *Air pollution*. As a consequence, at least two extremely different paths could exits. However, the path involving the node `Risk` of *Environment risks* can reach the target source through only 1 c-arc. The other one would need at least another c-arc to reach the target source. In particular, it should use the synonymy between `Risk` of *Air pollution* and `Hazard` of *Environment risks*. In Fig. 8, we report both these paths. The former involves the nodes `Risk_degree`, `Risk`, `Book` and `Emergency` and consists of 2 i-arcs and 1 c-arc. The latter involves the nodes `Risk_degree`, `Risk`, `Hazard`, `Book` and `Emergency` and consists of 2 i-arcs and 2 c-arcs. Clearly, the shortest path is the former, which is selected as the complex knowledge pattern between the two seed nodes.

Table 1
Keywords of the source *Environment risks*.

Keywords
absorptivecapacity, action, adjustments, adopted, agencies, agricultural, air, appraisal, areas, Bangladesh, Burton, capita, catastrophe, choice, coast, alcomprehensive, coping, cost, crops, damage, deaths, developingcountries, relief, drought, earthquake, economic, effects, effort, emergency, environment, environmental, estimated, evacuation, experience, extremeeventsfarmers, federa, Figure, flood, floodplain, forecasting, frequency, global, globalwarming, groups, hazardevent, hazardresearch, human, hurricane, impact, income, increase, individual, industrial, Kates, LabourBrigade, land, LiuLing, loss, magnitude, maize, major, measures, ment, million, naturaldisasters, naturalevents, naturalhazards, hazard, Nicaragua, occur, organization, pattern, people′scommunepercent, percent, population, possible, potential, prevent, protection, range, reduce, regions, risk, River, scale, scientific, social, society, SriLanka, storm, studies, threshold, tion, tornado, TristandaCunha, tropical, cyclone, TropicalStormAgnes, tsunami, UnitedKingdom, urban, vulnerable, warning, systems, zone, Managua, air, plant, disaster, airpollution, natural, Tanzania, TropicalStormAgnes.

Table 2
Keywords of the source *Air pollution*.

Keywords
acid, activatedsludge, activity, aerosol, airpollution, airquality, air, anaerobicdigestion, approach, aquatic, areas, Assesment, atmosphere, biofuels, carbon, catalyst, cause, chemical, chlorine, climatechange, combustion, concentrations, contaminated, cycle, CycleAssessment, deposition, diesel, dose, drinkingwater, ecosystem, effects, effluent, emissions, energy, EnvironmentAgency, European, EuropeanCommission, EuropeanUnion, exposure, Figure, fuel, gases, global, human, hydrocarbons, impacts, important, industrial, landfill, legislation, levels, London, major, materials, measures, models, monitoring, nanoparticles, nitrate, nitrogen, nitrogendioxide, nuisance, operation, organic, oxidation, oxygen, ozone, particles, PBDEs, PCBs, pesticides, plant, potential, radiation, radiativeforcing, radioactive, range, reaction, recycling, reduce, regulation, regulatory, release, response, result, risk, sewage, significant, sludge, soil, sources, species, standards, stratosphere, studies, substances, sulfurdioxide, surface, temperature, toxic, transport, treatment, typically, urban, vehicles, wastemanagement, ambient, biological, compounds, Directive, engine, example, increase, metals, petrol, reactor, eutrophication.

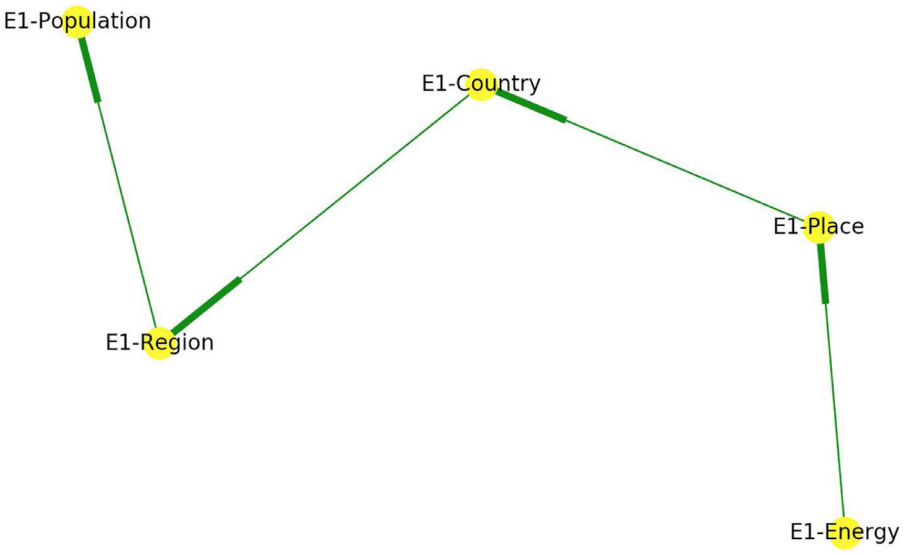


Fig. 5. Complex knowledge pattern from the node `Energy` to the node `Population` of the source *Energy*.

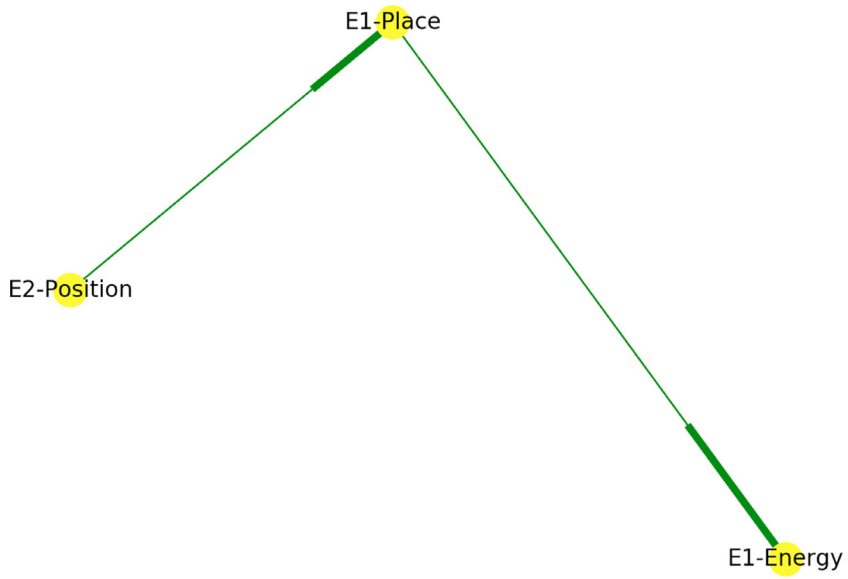


Fig. 6. Complex knowledge pattern from the node `Position` of the source *Environment disasters* to the node `Energy` of the source *Energy*.

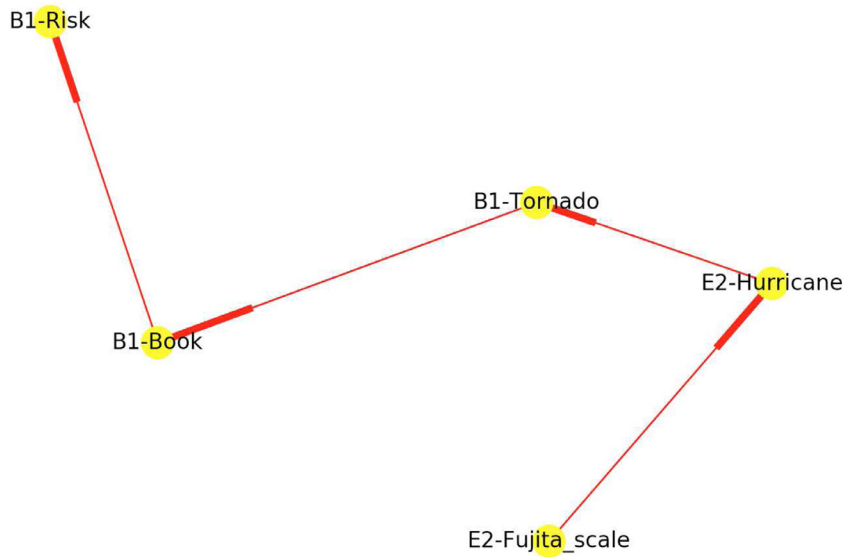


Fig. 7. Complex knowledge pattern from the node `Fujita_scale` of the source *Environment disasters* to the node `Risk` of the source *Environment risks*.

7. Discussion

This section is devoted to present a critical discussion of several aspects concerning our approach. It consists of four subsections. In the first, we present a comparison between our approach and the related ones. In the second, we evaluate the performance of our technique for structuring unstructured data. In the third, we evaluate the performance of our overall approach. Finally, in the fourth, we measure its efficiency for large datasets. To carry out the experiments described in this section, we used a server equipped with an Intel I7 Dual Core 5500U processor and 16GB of RAM with the Ubuntu 16.04.3 operating system. Clearly, especially for the last experiments, the capabilities of this server were limited. However, as we will see below, we were mostly interested to compare theoretical worst case response times with the real ones. Clearly, in real contexts, whenever necessary, much more powerful servers could be used to reduce the response time.

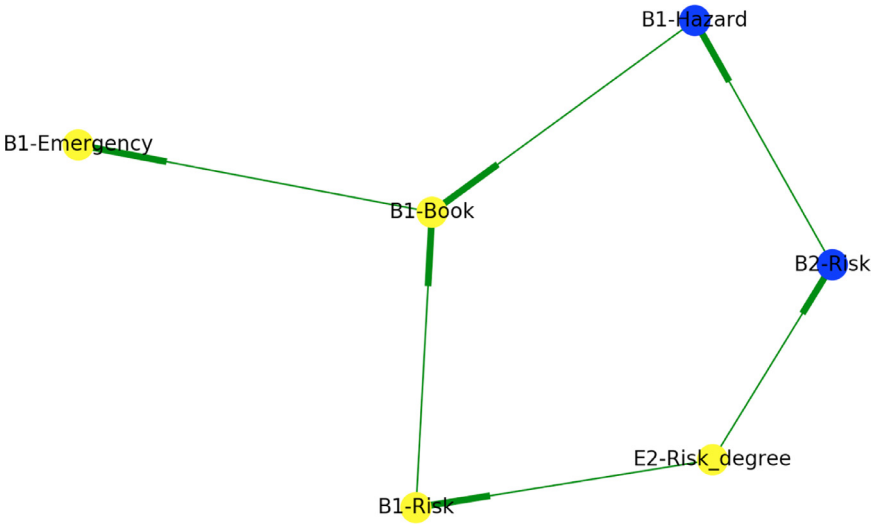


Fig. 8. Complex knowledge pattern from the node `Risk_degree` of the source *Environment disasters* to the node `Risk` of the source *Environment risks*.

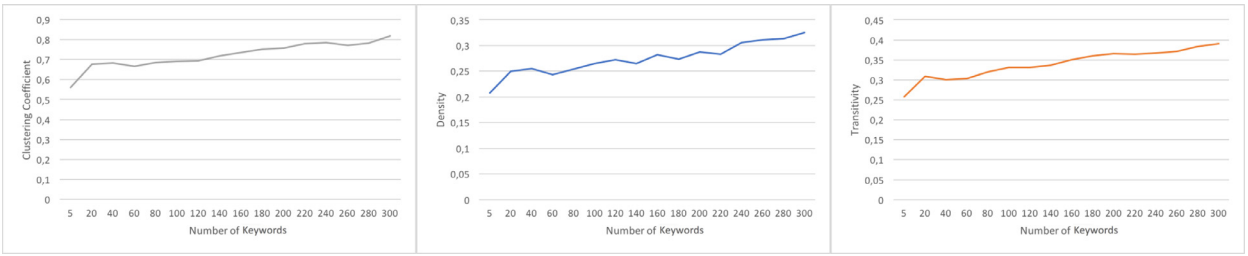


Fig. 9. Average clustering coefficient, density and transitivity of the network returned by our approach against the number of available keywords of the corresponding source.

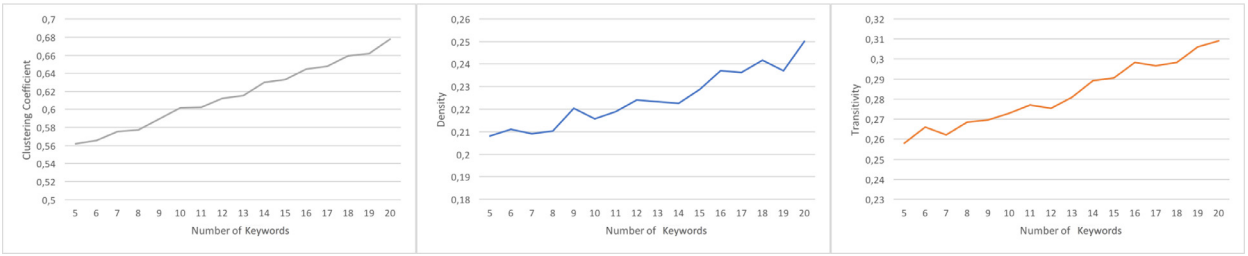


Fig. 10. A zoom of the graphs of Fig. 9 referred to the case in which the number of keywords ranges between 5 and 20.

7.1. Comparison between our approach and the related ones

In Section 2, we have seen that we can distinguish four main families of approaches that are most related to ours. Specifically, these approaches aim at extracting: *(i)* keyword patterns; *(ii)* knowledge from structured sources; *(iii)* knowledge from heterogeneous sources; *(iv)* knowledge patterns through network analysis-based techniques.

As for the first family, the corresponding approaches share with ours the objective, i.e. the extraction of some form of knowledge. However, the knowledge derived by them consists simply in keyword patterns. Furthermore, the techniques they leverage to carry out this task are different from ours, especially if we consider the sub-family operating on RDF data. A higher similarity can be found with the other sub-family, i.e., the one operating on graph databases [7].

As for the second family, the corresponding approaches present some analogies, but also some differences, with ours. In particular, both of them exploit metadata to perform the knowledge extraction task. However, the approaches of this second family operate only on structured sources, whereas our approach manages sources with disparate formats.

The approaches of the third family extract knowledge from heterogeneous (both structured and semi-structured) sources. For instance, the approach of Liu et al. [24] aim at querying heterogeneous data in fuzzy XML documents by using a tree-

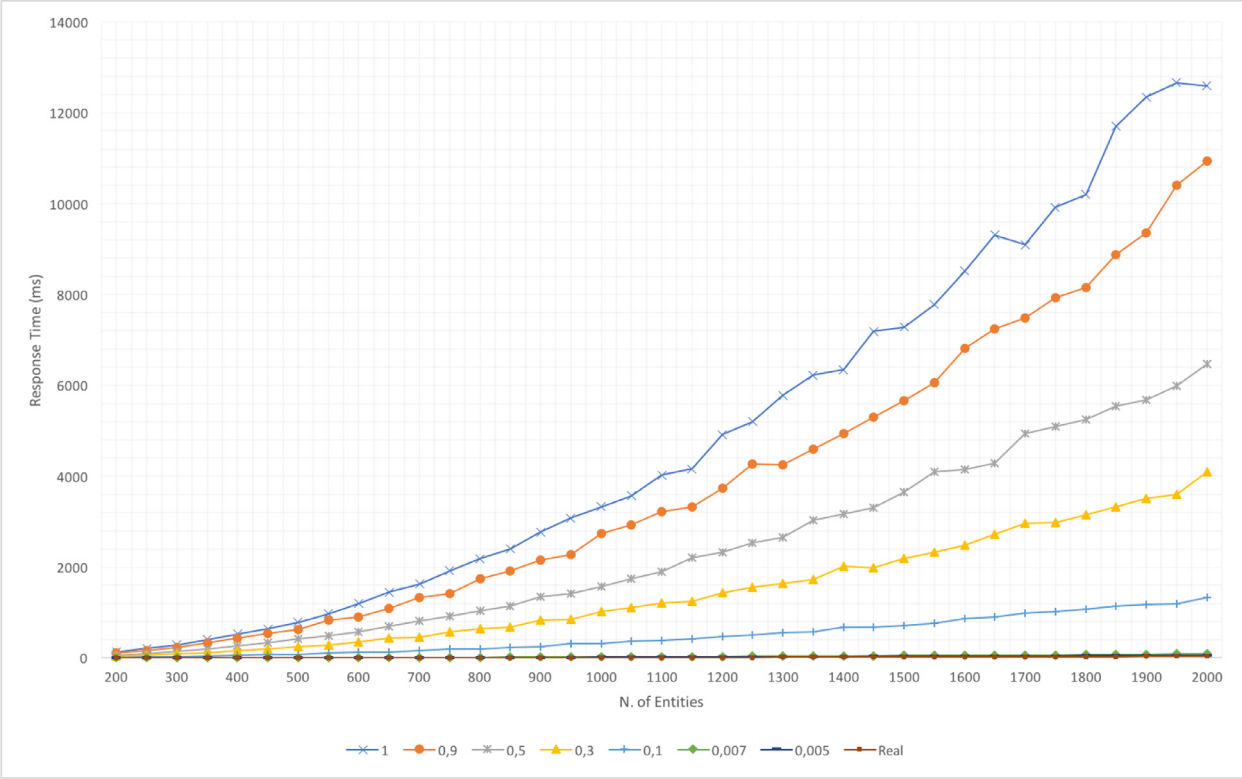


Fig. 11. Real and theoretical response time against data lake dimension and density.

pattern based algorithm. This approach has several differences with respect to ours. In fact, it focuses mainly on querying, whereas our approach considers the extraction of knowledge patterns. Furthermore, it operates on XML documents, whereas our approach operates on sources with different formats. Interestingly, the approach of Liu et al. [24] leverage a tree pattern-based algorithm, whereas ours exploits a graph pattern-based one. Another approach of this family is the one described in [39], which is based on Similarity Join. This approach and ours are similar in that both of them have a step in which a similarity detection task is performed. However, the approach of Shang et al. [39] need a support knowledge hierarchy, whereas our approach exploits one or more thesauruses. Furthermore, the data sources considered by the approach of Shang et al. [39] are just collections of objects (e.g., documents) and not a variegate data lake, which is the reference data structure for our approach.

The fourth family comprises network-based models and algorithms that exploit network analysis to extract knowledge patterns. One of these approaches is described in [38]. It operates in the context of Music Information Retrieval, which is actually quite far from the scenarios of interest to our approach. However, both this approach and ours share the usage of network to represent available data and of network analysis to extract the desired knowledge. The approach of Salazar and Zhao [38] focuses on non-traditional data sources, and, in this fact, is similar to ours. However, the source typology considered by it has a very specific nature, whereas the ones handled by our approach are numerous and are the most common ones encountered in the reality. Another approach belonging to the last family is the one described in [26]. This approach and ours present some analogies in that both of them use network analysis to extract knowledge of interest. However, the approach of Lo Giudice et al. [26] operate on only one kind of databases (e.g., relational ones) and focuses on a very specific topic, i.e., patent and applicant analysis. By contrast, our approach considers heterogeneous data formats and can operate on sources concerning different topics.

Other two approaches of this family that we have mentioned in Section 2 are the ones presented in [28] and [19]. Mallat et al. [28] propose a network-based formalism for representing available knowledge. In this formalism, nodes indicate concepts and arcs denote relationships between concepts. This representation coincides with the one adopted by our approach. The main difference between them consists in the fact that the approach of Mallat et al. [28] operate only on information organized in structured databases. This fact contributes to perform data investigation and formalization very easily but, on the other hand, it prevents from managing semi-structured and unstructured data. The approach of Kargar and An [19] aim at performing keyword search in a graph to facilitate the identification of sub-graphs. This approach and ours are similar in that both of them are network-based. However, they also present several differences. Indeed, the algorithm underlying the approach of Kargar and An [19] is centered on cliques, whereas the one underlying our approach is based on paths.

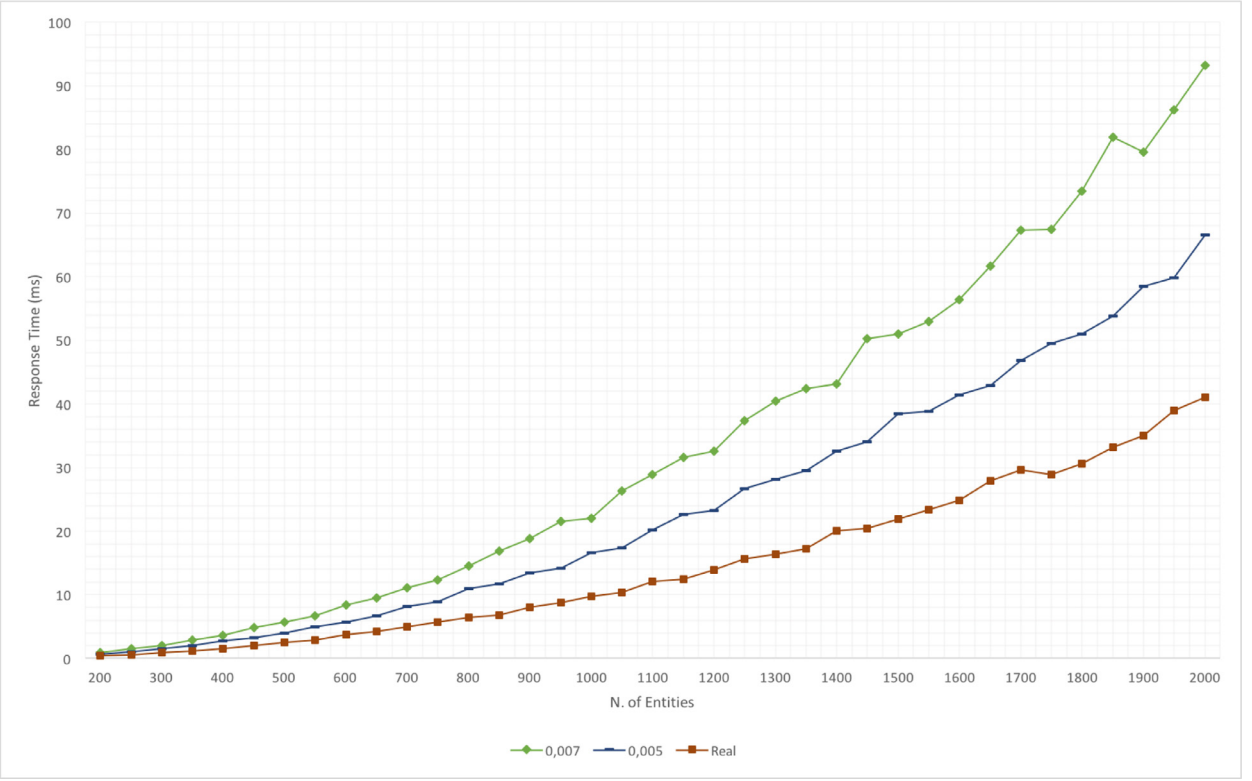


Fig. 12. Real and theoretical response time against data lake dimension and density (zoom of Fig. 11).

Furthermore, the approach of Kargar and An [19] focuse on keyword search, and the consequent subgraph identification, whereas ours aims at detecting knowledge patterns.

7.2. Evaluation of our approach to structuring unstructured data

One way to evaluate the performance of our approach to structuring unstructured sources consists of determining how much it is able to connect the concepts corresponding to the flat keywords commonly used to characterize unstructured sources. Given a network-based model, like ours, a logical way to quantify this feature is based on the exploitation of some measures tipically adopted in network analysis to quantify the structuring level of a network. These measures are: *(i)* average clustering coefficient, *(ii)* density, and *(iii)* transitivity. All of them range in the real interval [0,1]; the higher their value, the more structured the corresponding network.

We computed the values of these measures against the number of keywords representing unstructured sources. Obtained results are reported in Fig. 9, whereas in Fig. 10 we propose a "zoom" referred to the case in which the number of keywords ranges between 5 and 20.

From the analysis of these figures we can observe that our approach is really capable of structuring unstructured sources provided that the number of keywords representing each source is higher than a reasonable minimum value (i.e., 5). As long as the number of provided keywords increases, the values of all our structuring capability indicators increase, even if this increase is very slow.

In our opinion, the growth slowness, far from being a problem, is an indicator of correctness. Indeed, we must consider that we are trying to assign a structure to an originally unstructured source. Our approach can provide a certain structuring level but it cannot (and it must not) upset the original nature of the source, which is unstructured.

All these reasonings allow us to say that our approach to structuring unstructured sources presents a very satisfying behavior.

7.3. Performance of our overall approach

In Section 5.2.1, we have seen that the computational complexity of the extraction of complex knowledge patterns is $O(|A| \cdot \log|N| \cdot |N_{\max}|)$. We have also seen that this complexity can be judged very satisfactory, if we consider the problem to solve.

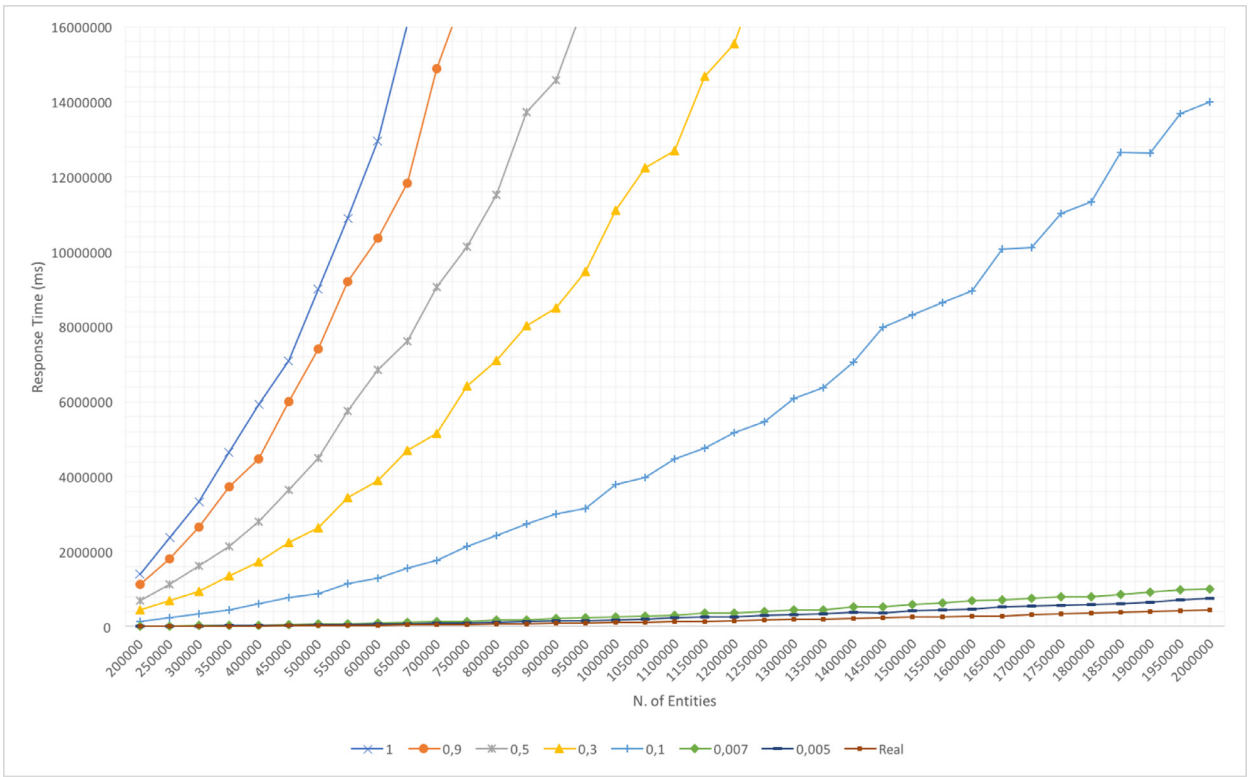


Fig. 13. Real and theoretical response time against dimension and density for large data lakes (Scenario 1).

However, in real data lakes, the number of involved sources is high and so, in principle, $|N|$ (and $|A|$, which is $O(|N|^2)$ could be very high. Nevertheless, in real situations, the number of relationships among attributes and elements is very small and, consequently, the corresponding networks are very sparse. As a consequence, $|A|$ should be very low, if compared with $|N|^2$, and, therefore, we were confident that, in real cases, the performance of our approach should be very good.

To verify this hypothesis we measured the response time of our approach when the number of involved nodes to examine increases; in particular, we measured the response time obtained by considering the theoretical computational complexity and the real response time. Obtained results are reported in Fig. 11, whereas in Fig. 12 we propose a "zoom" for those cases that in Fig. 11 appeared superimposed on the axis of abscissas. In these graphs, in the computation of the theoretical response time, we considered several values of graph density.

From the analysis of these figures, it clearly emerges that, in real cases, the response time of our approach is much smaller than the one determined by the worst case time complexity, even when the network density is low or very low. This fact leads our approach to work very well also in presence of large data lakes, provided that the corresponding networks are sparse or very sparse, which is the general condition that is found in practice. As a consequence, we can conclude that our hypothesis was true and, therefore, that our approach shows a good performance in real scenarios.

7.4. Efficiency of our overall approach for large data sets

In Section 5.2.1, we have seen that, from a theoretical point of view, in order to determine the computational complexity of our approach, we must consider two main scenarios, namely:

1. the detected path involves nodes of only one source, in which case the theoretical computational complexity is $O(|A| \cdot \log(|N|)$;
2. the detected path involves nodes of more sources, in which case the theoretical computational complexity is $O(|A| \cdot \log|N|) \cdot O(max(|N_{\max}|, |DL|))$.

Now, in presence of large data lakes, both $|N_{\max}|$ and $|DL|$ are much smaller than $|N|$; as a consequence, from a theoretical point of view, the two cases could be referred to a single one. However, since we aim at measuring the efficiency of our approach in the reality (and not only from a theoretical viewpoint), we prefer to keep the two cases separate and to verify if this hypothesis is also confirmed in practice.

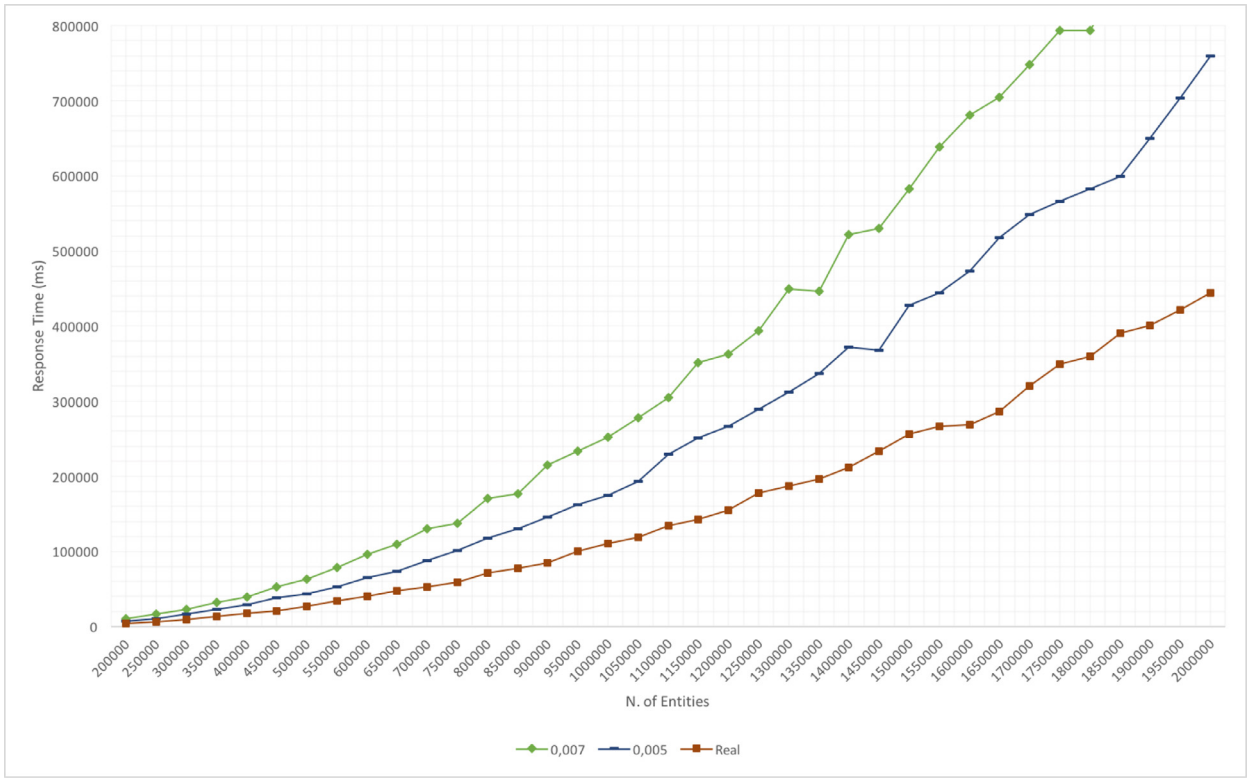


Fig. 14. Real and theoretical response time against dimension and density for large data lakes (zoom of Fig. 13).

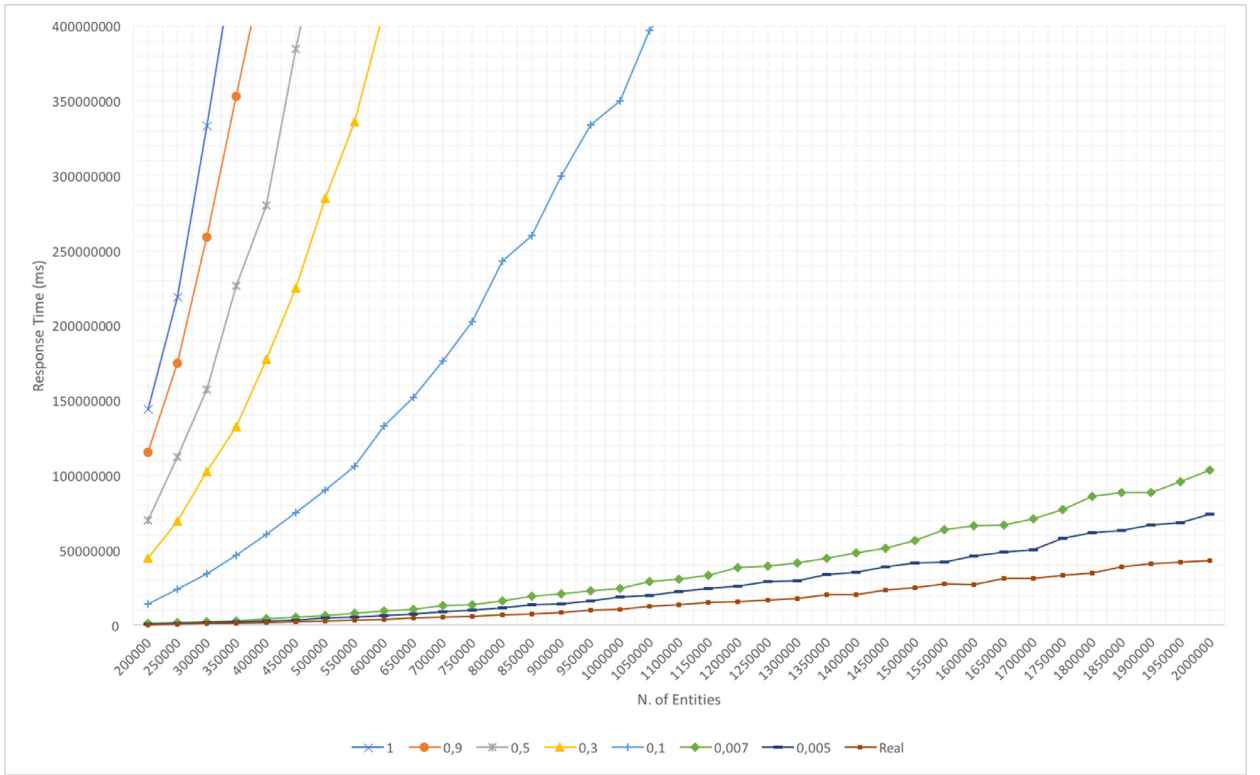


Fig. 15. Real and theoretical response time against dimension and density for large data lakes (Scenario 2).

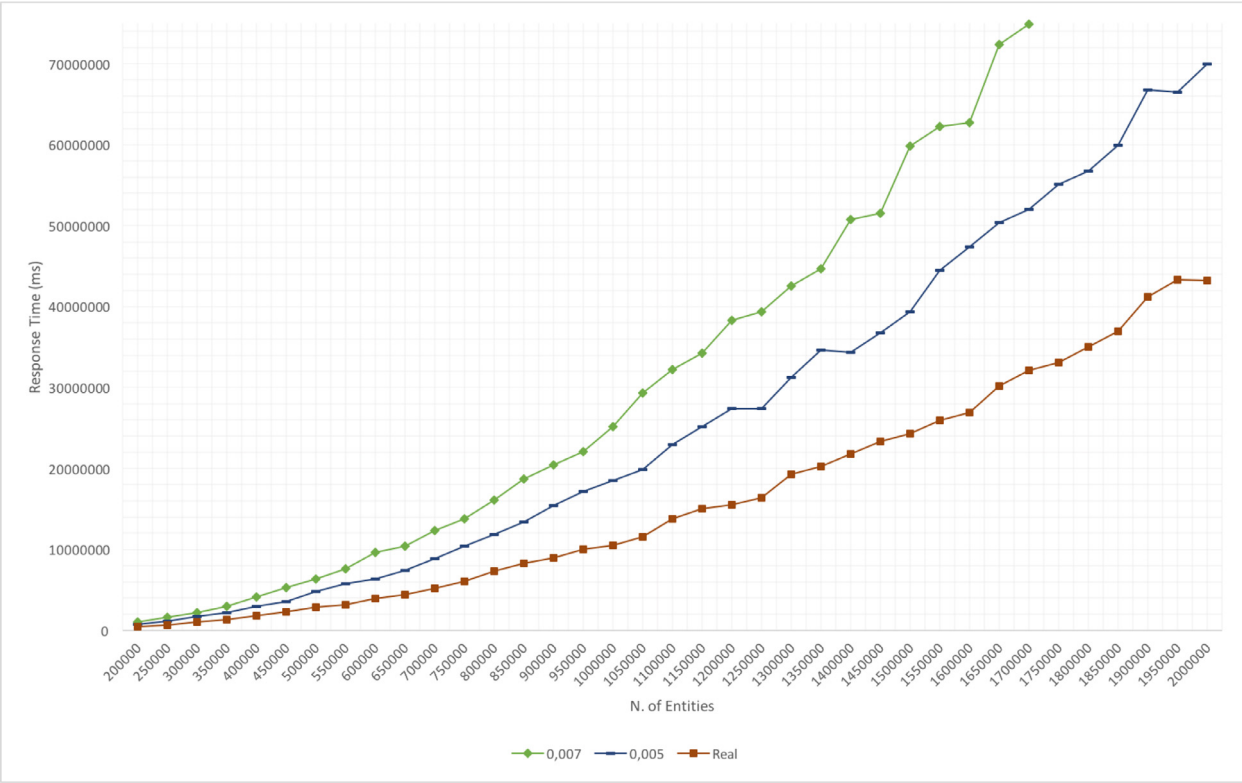


Fig. 16. Real and theoretical response time against data lake dimension and density for large data lakes (zoom of Fig. 15).

To carry out this experiment, we decided to repeat the tasks already performed in the previous one, but with a data lake having a number of nodes that is three orders of magnitude higher than the maximum one considered in the previous experiment. This number of nodes is clearly much higher than the ones we can currently meet in real situations. However, we preferred to put our approach under stress to see if, even in these extreme cases, it shows an acceptable behavior. Also in this case, we computed the response time of our approach against the number of nodes of the data lake and compared the response time obtained by considering the theoretical computational complexity against the real response time.

Obtained results are reported in Fig. 13, for the Scenario 1 mentioned above, and in Fig. 15, for the Scenario 2 considered previously. A "zoom" of these figures, limited to those cases that appeared superimposed on the axis of abscissas, are reported in Figs. 14 and 16, respectively. From the analysis of these figures we can observe that, in presence of very large data sets, the theoretical response time of our approach would make it not applicable for high values of density. Instead, our approach shows an acceptable response time for low values of density.

Actually, we have already seen that, in real cases, data lake density is very low. This is also witnessed by the trend of the real response time shown in Figs. 13–16, which is even better than the response time derived from the theoretical computational complexity obtained with a small density (i.e., 0.005). Interestingly, the trends of the real response time for Scenarios 1 and 2 are actually the same. The only difference regards the corresponding values that, in case of Scenario 2, are about two orders of magnitude higher than the ones shown in Scenario 1.

All the results described in this section, coupled with the fact that we stressed our approach in extreme cases generally not found in the current reality, lead us to conclude that our approach presents a very good efficiency, which makes it well suited also for large datasets.

8. Conclusion

In this paper, we have proposed a new network-based model to uniformly represent and handle structured, semi-structured and unstructured sources of a data lake. Then, we have presented a new approach to, at least partially, "structuring" unstructured sources. Furthermore, we have defined a new approach to extracting complex knowledge patterns from the sources of a data lake and we have presented some case studies showing the behavior of our approach in all the possible cases. Finally, we have compared our approach with the related ones and we have evaluated its performance.

This paper is not to be intended as an ending point. Instead, it could be the starting point of a new set of approaches specifically conceived to handling information systems in the new big data oriented scenario. For instance, we plan to define

new approaches to supporting the flexible and lightweight querying of the sources of a data lake, as well as new approaches to schema matching, schema mapping and data reconciliation and integration strongly oriented to unstructured data sources and data lakes.

References

[1] S. Bergamaschi, S. Castano, M. Vincini, D. Beneventano, Semantic integration and query of heterogeneous information sources, Data Knowl. Eng. 36(3) (2001) 215–249.
[2] C. Chen, C. Zhang, Data-intensive applications, challenges, techniques and technologies: a survey on Big Data, Inf. Sci. 275 (2014) 314–347.
[3] L. Chen, J. Shao, Z. Yu, J. Sun, F. Wu, Y. Zhuang, RAISE: a whole process modeling method for unstructured data management, in: Proc. of the International Conference on Multimedia Big Data (BigMM'15), IEEE, China National Conference Center, China, 2015, pp. 9–12.
[4] X. Chen, A. Shrivastava, A. Gupta, Neil: extracting visual knowledge from web data, in: Proc. of the International Conference on Computer Vision (ICCV'13), IEEE, Darling Harbour, Sydney, 2013, pp. 1409–1416.
[5] Y. Chen, W. Wang, Z. Liu, Keyword-based search and exploration on databases, in: Proc. of the International Conference on Data Engineering (ICDE'11), IEEE, Hannover, Germany, 2011, pp. 1380–1383.
[6] A. Corbellini, C. Mateos, A. Zunino, D. Godoy, S. Schiaffino, Persisting big-data: the NoSQL landscape, Inf. Syst. 63 (2017) 1–23. Elsevier.
[7] A. Dass, C. Aksoy, A. Dimitriou, D. Theodoratos, Relaxation of keyword pattern graphs on RDF Data, J. Web Eng. 16 (5–6) (2017) 363–398. Rinton Press, Incorporated.
[8] P. De Meo, G. Quattrone, G. Terracina, D. Ursino, Integration of XML schemas at various "severity" levels, Inf. Syst. 31(6) (2006) 397–434.
[9] F. Di Tria, E. Lefons, F. Tangorra, Cost-benefit analysis of data warehouse design methodologies, Inf. Syst. 63 (2017) 47–62. Elsevier.
[10] C. Diamantini, P. Lo Giudice, L. Musarella, D. Potena, E. Storti, D. Ursino, An approach to extracting thematic views from highly heterogeneous sources of a data lake, Atti del Ventiseiesimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati (SEBD'18), 2018. Castellaneta Marina (TA), Italy.
[11] M. Farid, A. Roatis, I. Ilyas, H. Hoffman, X. Chu, CLAMS: bringing quality to Data Lakes, in: Proc. of the International Conference on Management of Data (SIGMOD/PODS'16), ACM, San Francisco, CA, USA, 2016, pp. 2089–2092.
[12] A. Farrugia, R. Claxton, S. Thompson, Towards social network analytics for understanding and managing enterprise data lakes, in: Proc. of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM'16), IEEE, San Francisco, CA, USA, 2016, pp. 1213–1220.
[13] T. Foley, H. Hagen, G. Nielson, Visualizing and modeling unstructured data, Vis. Comput. 9 (8) (1993) 439–449. Springer.
[14] K. Golenberg, B. Kimelfeld, Y. Sagiv, Keyword proximity search in complex data graphs, in: Proc. of the International Conference on Management of data (SIGMOD/PODS'08), ACM, Vancouver, Canada, 2008, pp. 927–940.
[15] R. Hai, S. Geisler, C. Quix, Constance: an intelligent data lake system, in: Proc. of the International Conference on Management of Data (SIGMOD'16), ACM, San Francisco, CA, USA, 2016, pp. 2097–2100.
[16] S. Han, L. Zou, J. Yu, D. Zhao, Keyword search on RDF graphs-a query graph assembly approach, in: Proc. of the International Conference on Information and Knowledge Management (CIKM'17), ACM, Singapore, Singapore, 2017, pp. 227–236.
[17] H. He, H. Wang, J. Yang, P. Yu, BLINKS: ranked keyword searches on graphs, in: Proc. of the International Conference on Management of Data (SIGMOD/PODS'07), ACM, Beijing, China, 2007, pp. 305–316.
[18] F. Jebbor, L. Benhlima, Overview of knowledge extraction techniques in five question-answering systems, in: Proc. of the International Conference on Intelligent Systems: Theories and Applications (SITA'14), IEEE, Rabat, Morocco, 2014, pp. 1–8.
[19] M. Kargar, A. An, Keyword search in graphs: finding r-cliques, Proc. VLDB Endow. 4 (10) (2011) 681–692. VLDB Endowment.
[20] M. Karim, M. Cochez, O. Beyan, C. Ahmed, S. Decker, Mining maximal frequent patterns in transactional databases and dynamic data streams: a spark-based approach, Inf. Sci. 432 (2018) 278–300.
[21] G. Kondrak, N-gram similarity and distance, in: String Processing and Information Retrieval, 2005, pp. 115–126. Springer.
[22] G. Li, B. Ooi, J. Feng, J. Wang, L. Zhou, EASE: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data, in: Proc. of the International Conference on Management of Data (SIGMOD/PODS'08), ACM, Vancouver, Canada, 2008, pp. 903–914.
[23] C. Lin, J. Wang, C. Rong, Towards heterogeneous keyword search, in: Proc. of the ACM Turing 50th Celebration Conference-China (ACM TUR-C'17), ACM, Shanghai, China, 2017, p. 46.
[24] J. Liu, X. Zhang, L. Zhang, Tree pattern matching in heterogeneous fuzzy XML databases, Knowl. Based Syst. 122 (2017) 119–130. Elsevier.
[25] Z. Liu, P. Sun, Y. Chen, Structured search result differentiation, Proc. VLDB Endow. 2 (1) (2009) 313–324. VLDB Endowment.
[26] P. Lo Giudice, P. Russo, D. Ursino, A new Social Network Analysis-based approach to extracting knowledge patterns about research activities and hubs in a set of countries, Int J Bus Innov Res (2017). Inderscience. Forthcoming.
[27] J. Madhavan, P. Bernstein, E. Rahm, Generic schema matching with Cupid, in: Proc. of the International Conference on Very Large Data Bases (VLDB 2001), Morgan Kaufmann, Rome, Italy, 2001, pp. 49–58.
[28] S. Mallat, E. Hkiri, M. Maraoui, M. Zrigui, Semantic network formalism for knowledge representation: towards consideration of contextual information, Int. J. Semantic Web Inf. Syst. 11 (4) (2015) 64–85. IGI Global.
[29] B. Malysiak-Mrozek, M. Stabla, D. Mrozek, Soft and declarative fishing of information in Big Data Lake, IEEE Trans. Fuzzy Syst. 26 (5) (2018) 2732–2747.
[30] N. Miloslavskaya, A. Tolstoy, Big data, fast data and data lake concepts, Procedia Comput. Sci. 88 (2016) 300–305. Elsevier.
[31] S. Murugesh, A. Jaya, Representing natural language sentences in RDF graphs to derive knowledge patterns, in: Proc. of the International Conference on Data Engineering and Communication Technology (ICDECT'17), Springer, Maharashtra, India, 2017, pp. 701–707.
[32] R. Navigli, S. Ponzetto, BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artif. Intell. 193 (2012) 217–250. Elsevier.
[33] A. Oram, Managing the Data Lake, O'Reilly, Sebastopol, CA, USA, 2015.
[34] L. Palopoli, G. Terracina, D. Ursino, Experiences using DIKE, a system for supporting cooperative information system and data warehouse design, Inf. Syst. 28(7) (2003) 835–865.
[35] S. Rangarajan, H. Liu, H. Wang, C. Wang, Scalable architecture for personalized healthcare service recommendation using Big Data Lake, in: Service Research and Innovation, 2015, pp. 65–79. Springer.
[36] F. Sadeghi, S.K. Divvala, A. Farhadi, Viske: visual knowledge extraction and question answering by visual verification of relation phrases, in: Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPM'15), 2015, pp. 1456–1464. Boston, MA, USA.
[37] M. Sahlgren, J. Karlgren, Automatic bilingual lexicon acquisition using random indexing of parallel corpora, Nat. Lang. Eng. 11 (3) (2005) 327–341.
[38] A. Salazar, L. Zhao, Rhythmic pattern extraction by community detection in complex networks, in: Proc. of the International Conference on Intelligent Systems (BRACIS'14), IEEE, Sao Paulo, Brazil, 2014, pp. 396–401.
[39] Z. Shang, Y. Liu, G. Li, J. Feng, K-join: knowledge-aware similarity join, IEEE Trans. Knowl. Data Eng. 28 (12) (2016) 3293–3308. IEEE.
[40] A. Silva, C. Antunes, Constrained pattern mining in the new era, Knowl. Inf. Syst. 47 (3) (2016) 489–516. Springer.
[41] K. Singh, K. Paneri, A. Pandey, G. Gupta, G. Sharma, P. Agarwal, G. Shroff, Visual Bayesian fusion to navigate a data lake, in: Proc. of the International Conference on Information Fusion (FUSION'16), 2016, pp. 987–994. IEEE.
[42] D. Sun, G. Zhang, S. Yang, W. Zheng, S. Khan, K. Li, Re-Stream: real-time and energy-efficient resource scheduling in big data stream computing environments, Inf. Sci. 319 (2015) 92–112.
[43] P. Talukdar, Z. Ives, F. Pereira, Automatically incorporating new sources in keyword search-based data integration, in: Proc. of the International Conference on SIGMOD International Conference on Management of data (SIGMOD/PODS'10), ACM, Indianapolis, IN, USA, 2010, pp. 387–398.

[44] I. Terrizzano, P. Schwarz, M. Roth, J. Colino, Data wrangling: the challenging journey from the wild to the lake, in: Proc. of the International Conference on Innovative Data Systems Research (CIDR'15), 2015. Asilomar, CA, USA.
[45] C. Walker, H. Alrehamy, Personal data lake with data gravity pull, in: Proc. of the International Conference on Big Data and Cloud Computing (BDCloud'15), IEEE, Dalian, China, 2015, pp. 160–167.
[46] H. Wang, Z. Xu, H. Fujita, S. Liu, Towards felicitous decision making: an overview on challenges and trends of Big Data, Inf. Sci. 367 (2016) 747–765.
[47] Y. Yuan, G. Wang, L. Chen, B. Ning, Efficient pattern matching on big uncertain graphs, Inf. Sci. 339 (2016) 369–394. Elsevier.
[48] Y. Zhuang, Y. Wang, J. Shao, L. Chen, W. Lu, J. Sun, B. Wei, J. Wu, D-Ocean: an unstructured data management system for data ocean environment, Front. Comput. Sci. 10 (2) (2016) 353–369. Springer.