An approach to extracting topic-guided views from the sources of a data lake

An approach to extracting topic-guided views from the sources of a data lake

Claudia Diamantini[1], Paolo Lo Giudice[2], Domenico Potena[1], Emanuele Storti[1] and Domenico Ursino[1]

[1] DII, Polytechnic University of Marche
[2] DIIES, University "Mediterranea" of Reggio Calabria,
{c.diamantini@univpm.it; paolo.lo.giudice@unirc.it; d.potena@univpm.it; e.storti@univpm.it; d.ursino@univpm.it}

Abstract

In the last years, data lakes are emerging as an effective and an efficient support for information and knowledge extraction from a huge amount of highly heterogeneous and quickly changing data sources. Data lake management requires the definition of new techniques, very different from the ones adopted for data warehouses in the past. In this scenario, one of the most challenging issues to address consists in the extraction of topic-guided (i.e., thematic) views from the (very heterogeneous and often unstructured) sources of a data lake. In this paper, we propose a new network-based model to uniformly represent structured, semi-structured and unstructured sources of a data lake. Then, we present a new approach to, at least partially, "structuring" unstructured data. Finally, we define a technique to extract topic-guided views from the sources of a data lake, based on similarity and other semantic relationships among source metadata.

Keywords: Data Lakes, Unstructuted Data Sources, Metadata Management, Thematic Views, Semantic Similarities, DBpedia

1 Introduction

In the last years, data lakes have emerged as an effective and efficient answer to the problem of extracting information and knowledge from a huge amount of highly heterogeneous and quickly changing data sources (Fang 2015).

Data lake management requires the definition of new techniques, very different from the ones adopted for data warehouses in the past. These techniques may exploit the large set of metadata always supplied with data lakes, which represent their core and the main tool allowing them to be a very competitive framework in the big data era. In such a way, it is possible to guarantee an effective and efficient management of data source interoperability. As a proof of this, the main data lake companies are performing several efforts in this direction (see, for instance, the metadata organization proposed by Zaloni, one of the market leaders in the data lake field (Oram 2015)). For this reason, the definition of new models and paradigms for metadata representation and management represents an open problem in the data lake research field.

The extraction of thematic (or topic-guided) views from data sources is one of the main issues to address in a scenario comprising many data sources extremely heterogeneous in their format, structure and semantics (Aversano et al. 2010). This consists in the construction of views concerning one or more topics of interest for the user, obtained by extracting and merging data coming from different sources. The problem has been largely investigated in the past for structured and semi-structured data sources stored in a data warehouse (Wu et al. 2009; Castano and Antonellis 1999; Palopoli et al. 2000, 2003a,c,b), and this witnesses its extreme relevance. However, it is esteemed that, currently, more than 80% of data sources are unstructured (Corbellini et al. 2017). As a consequence, it is just this type of source that represents the main actor of the big data scenario and, consequently, of data lakes.

In this paper, we aim at providing a contribution in this setting. Indeed, we propose a supervised approach to extracting thematic views from highly heterogeneous sources of a data lake. Our approach represents all the data lake sources by means of a suitable network model. Indeed, networks are very flexible structures that allow the modeling of almost all phenomena that researchers aim at investigating (Bouadjenek et al. 2016). Our model starts from the considerations and the ideas proposed by data lake companies. In particular, it starts from the general metadata classification also used by Zaloni (Oram 2015). In this classification, metadata are divided in three categories, namely: *(i) business metadata*, which include business rules; *(ii) operational metadata*, which include information automatically generated during data processing; *(iii) technical metadata*, which include information about data format and schema. However, it complements them with new ideas and, being based on network theory and semantics-driven approaches, it can benefit from all the results already found in these fields. As a consequence, it can allow a large variety of sophisticated tasks that the currently adopted metadata models do not guarantee. For instance, it allows the definition of a structure for unstructured data. Thanks to this uniform representation of the data lake sources, the extraction of thematic views from them can be performed by exploiting graph-based tools. We define "supervised" our approach because it requires the user to specify the set T of topics that should be present in the thematic view(s) to extract.

Our approach consists of two steps. The former is mainly based on the structure of involved sources. It exploits several notions typical of (social) network analysis, such as the one of ego network, which actually represents its core. An ego network is a network consisting of a focal node, called ego, and the nodes, called alters, whom ego is directly connected to. The ego network comprises the ties from the ego to the alters and the ones, if any, between the alters. The usage of ego network is a key feature of our approach. This concept is inherited from Social Network Analysis. In our case, it is justified by several past studies in Cooperative Information Systems that showed that the neighborhood of a concept plays a key role in defining its meaning (Palopoli et al. 2003a, 2001; De Meo et al. 2006). These studies are strictly related to the concept of homophily in Social Network Analysis (McPherson et al. 2001). The latter step exploits a knowledge repository, which is used to discover new relationships, other than synonymies, among metadata, with the purpose to refine the integration of different thematic views obtained after the first step. In this step, our approach relies on DBpedia[1], a project aiming to extract structured content from the information created in the Wikipedia project.

From the previous description it emerges that the main contribution of this paper is threefold. Specifically:

- We introduce a new approach to "structuring" unstructured data. As for this aspect, we observe that, certainly, in the past literature, several approaches to extract keywords from unstructured data have been presented. See, for instance, RAKE - Rapid Automatic Keyword Extraction - (Rose et al. 2010), LDA - Latent Dirichlet Allocation - (Blei et al. 2003), YAKE! - Yet Another Keyword Extractor - (Campos et al. 2020) and TopicRank (Bougouin et al. 2013), just to cite a few of them. However, all of them return a "flat" list of keywords. By contrast our approach returns a hierarchy, similar to the one characterizing semi-structured sources, which makes it possible to uniformly handle unstructured sources along with semi-structured and structured ones.

- The approach to thematic view extraction we are proposing in this paper has been specifically conceived to operate on current data lakes where, due to the number and the dimension of present data sources, efficiency is a key issue. Indeed, as we will see in Subsection 6.6, the computation time of our approach is small. Along with efficiency, the proposed approach can extend to our, very complex, reference scenario some important features that, in the past, were guaranteed only by approaches operating on structured and semi-structured data. These last ones were characterized by a much higher computational complexity than our approach. In particular, the features mentioned above are:

  - it is very effective in enriching the global view obtained after the integration of two or more separate views, as we will see in Subsection 6.5;
  - it is capable of creating thematic views that put together homogeneous information even if this last is sparse among several data sources, as we will show in Subsection 6.3;
  - the information contained in the thematic views returned by our approach is very cohesive, as we will see in Subsection 6.2.

- Our approach to thematic view extraction uses ego-networks as starting points for thematic view extraction operations. As it will be clear in the following, this choice allows the concept of interest to be put at the centre of the view extraction task and to focus the efforts in a targeted manner. Furthermore, ego networks allow us to decide the desired connection strength in the thematic view extracted. Indeed, by changing the level of neighborhood considered in the ego network, it is possible to determine the minimum strength of the relationships existing between the elements of the thematic view and the corresponding core.

This paper is organized as follows: in Section 2, we present related literature. In Section 3, first we describe a unifying model for data lake representation; then, we present a technique to partially structuring unstructured sources. In Section 4, we illustrate our approach to thematic view extraction.

[1] http://dbpedia.org/

In Section 5, we present an example case. In Section 6, we describe several tests that we performed to evaluate our approach. Finally, in Section 7, we draw our conclusions and have a look to future developments of our research efforts in this field.

2 Related Literature

The new data lake scenario is characterized by several peculiarities that make it very different from the data warehouse paradigm (Hai et al. 2016, 2018). In particular, differently from data warehouses, data lakes: *(i)* store raw data in its native format (this could be structured, semi-structured and unstructured); *(ii)* retain all the data; *(iii)* store data irrespective of volume and variety; *(iv)* can be handled by several kinds of users and not only by business professionals; *(v)* have data that only transforms when needed; *(vi)* perform configurations and re-configurations when required, in a highly agile way; *(vii)* allow low-storage and economical reporting and analysis tasks. Hence, to operate in the data lake scenario, it is necessary to adapt (if possible) the old algorithms conceived for data warehouses or to define new approaches capable of handling and taking advantage of the specificities of this new paradigm.

2.1 Approaches to metadata classification

In the literature, several metadata classifications have been proposed in the past. For instance, Bilalli et al. (2016) present a tree-based classification. They split metadata into several categories, illustrate a conceptual schema of the metadata repository and use RDF for metadata modeling. RDF stands for Resource Description Framework (Lassila et al. 1998). It is a framework conceived to describing resources and favoring data interchange on the Web. It facilitates data merging even if the underling schemas differ. The strength of this model is undoubtedly its richness, whereas its weakness is its complexity that cannot guarantee a fast processing of the corresponding data.

A metadata model well-suited for data lakes is proposed by Oram (2015). This is also the model adopted by Zaloni[2], one of the main commercial leaders in the data lake field. It divides metadata based on their generation time or on the meaning and information they bring. In this latter case, metadata can be classified in three categories, namely operational, technical and business metadata. As will be clear in the following, our metadata model starts from this, but it goes much further. In particular, we argue that the three classes are not independent from each other because there are several intersections of them. Some of these intersections are particularly expressive and important; for them, it provides a network-based representation rich enough to allow several interesting, but, at the same time, not excessively complex, tasks in such a way as to prevent a slow processing.

Several metadata models and frameworks are widely adopted by the Linked Data (Heath and Bizer 2011) community (e.g., DCMI Metadata Terms and VoID). DCMI Metadata Terms (Dublin Core Metadata Initiative 2012) is a set of metadata vocabularies and technical specifications maintained by the Dublin Core Metadata Initiative. It includes generic metadata, represented as RDF properties, on dataset creation, access, data provenance, structure and format. A subset was also published as ANSI/NISO and ISO standards and as IETC RFC. The Vocabulary of Interlinked Datasets (VoID)

[2] https://www.zaloni.com/

(Keith et al. 2011) is an RDF Schema vocabulary that provides terms and patterns for describing RDF datasets. It is intended as a bridge between the publishers and the users of RDF data. It focuses on: *(i) general metadata*, following the Dublin Core model; *(ii) access metadata*, describing how RDF data can be accessed by means of several protocols; *(iii) structural metadata*, representing the structure and the schema of datasets, mostly used for supporting querying and data integration for a variety of scenarios (both in private or public sectors (Mouzakitis et al. 2017)).

2.2 Approaches to thematic view extraction

As for the structuring of unstructured sources, their querying and the extraction of thematic views from them, most approaches presented in the past literature do not completely fit the data lake paradigm. As a matter of fact, although researchers are increasingly focusing on issues concerning unstructured data and on costs for its management (see, for instance, Hamadou and Ghozzi (2018); Hai et al. (2018); Brackenbury et al. (2018); Klettke et al. (2017); Chen et al. (2016)), the amount of work to be done in this context appears considerable.

As far as the thematic view[3] extraction is concerned, Castano and Antonellis (1999) propose some techniques for building views on semi-structured data sources based on some expected queries, the analysis is ontology-driven and leads to the construction of reconciled views of the sources. Other researchers focus on materialized views and, specifically, on throughput and execution time. They a-priori define a set of well-known views and, then, materialize them. Finally, they show that the complexity of the problem depends on the possibility that views store all the tuples satisfying the corresponding definition. Two surveys on this issue can be found in Halevy (2001) and Abiteboul and Duschka (1998). Wu et al. (2009) investigate the same problem but they focus on XML sources. They adopt a model based on inverted lists and holistic algorithms which together have been established as the prominent technique for evaluating queries on large persistent XML data. Bidoit et al. (2018) propose an approach to statically and dynamically partition a large XML document, so as to distribute the computing load among the machines of a MapReduce cluster. The approaches of Wang and Yu (2012) and Hai et al. (2018) address the same issue by means of query rewriting. Specifically, the authors of Wang and Yu (2012) transform a query Q into a set of new queries, evaluate them, and, then, merge the corresponding answers to construct the materialized answer to Q. The approach proposed by Hai et al. (2018), arguing that one of the most important tasks of data lakes is to provide a unified querying interface, exploits logic-based methods for data integration exploiting declarative mappings with a scalable big data query processing system. Bachtarzi and Bachtarzi (2015) propose an approach to constructing materialized views for heterogeneous databases; it is based on a model-driven technique for views definition and requires the presence of a static context along with the pre-computation of some queries.

Another family of approaches exploits materialized views to perform tree pattern querying (Wang et al. 2011) and graph pattern queries (Fan et al. 2016). In both approaches the authors present a rewriting algorithm to find the best approximate answers. Unfortunately, all these approaches are well-suited for structured and semi-structured data, whereas they are not scalable and lightweight enough

[3] Recall that, in database context, a view is the result of a query or a more complex extraction process that can be exploited by users for further computations.

to be used in a dynamic context or with unstructured data. An interesting advance in this area can be found in Singh and Singh (2016). Here, the authors propose an incremental approach to addressing the graph pattern query problem on both static and dynamic real-life data graphs. Furthermore, they propose an efficient polynomial algorithm to generate the maximal contained rewriting, whenever it exists. Other kinds of view are investigated in Biskup and Embley (2003) and Aversano et al. (2010). In particular, Biskup and Embley (2003) propose a framework to extract information from heterogeneous sources for particular predefined target views conceptually specified through ontologies. Then the target is mapped to sources and expressed in the same modeling language. The authors adopt a formal foundation to prove that when a source has a valid interpretation, the generated mapping produces a valid interpretation for the part of the target loaded from the source. Instead, Aversano et al. (2010) use virtual views to access heterogeneous data sources without knowing many details of them. For this purpose, it creates virtual views of the sources themselves. The proposed approach provides features for the automatic schema matching and schema merging. It exploits both syntax-based and semantic-based techniques for performing this task.

Finally, semantic-based approaches have long been used to drive data integration in databases and data warehouses (García-Moya et al. 2013; Janjua et al. 2013). More recently, in the context of big data, a formal semantics has been specifically exploited to address issues concerning data variety/heterogeneity, data inconsistency and data quality in such a way as to increase understandability (Hitzler and Janowicz 2013; Debattista et al. 2014; Konstantinou et al. 2017; Mouttham et al. 2012). The proposed approaches identify interoperability deficiencies and find different solutions to address it. The solutions are mainly based on self-contained graphs and data wrangling. The ultimate purpose is the efficient integration and management of both structured and unstructured data sources by aligning data silos and better managing evolving data models. For instance, in Hai et al. (2016), the authors discuss a data lake system with a semantic metadata matching component for ontology modeling, attribute annotation, record linkage and semantic enrichment. Farid et al. (2016) present a system to discover and enforce expressive integrity constraints from data lakes. Similarly to what happens in our approach, knowledge graphs, e.g. based on RDF, are used to drive integration. To reach their objectives, these techniques usually rely on information extraction tools, e.g., Open Calais[4] or KAYAK (Maccioni and Torlone 2018), that may assist in linking metadata to uniform vocabularies (e.g., ontologies or knowledge repositories, such as DBpedia).

Starting from the previous description, we can identify some features that may characterize thematic view extraction approaches. These are: *(i)* the adoption of a materialized view; *(ii)* the exploitation of a holistic algorithm; *(iii)* the usage of inverted lists; *(iv)* the adoption of an ontology; *(v)* the exploitation of tree patterns; *(vi)* the usage of query rewriting; *(vii)* the adoption of a multidimensional data model. In Table 1, we present a classification of the approaches described above based on these features.

[4] `http://www.opencalais.com`

Paper	*Mat. View*	*Holistic Alg.*	*Inv. Lists*	*Ontology*	*Tree Pattern*	*Query Rewr.*	*Multid. Data Model*
Castano and Antonellis (1999)				x			
Halevy (2001)	x						
Abiteboul and Duschka (1998)	x						
Wu et al. (2009)		x	x				
Wang and Yu (2012)	x				x	x	
Bachtarzi and Bachtarzi (2015)	x					x	
Wang et al. (2011)					x		
Fan et al. (2016)	x				x		
Biskup and Embley (2003)	x			x			
Singh and Singh (2016)	x					x	
Aversano et al. (2010)	x			x		x	
García-Moya et al. (2013)							x
Debattista et al. (2014)							x
Konstantinou et al. (2017)		x					
Mouttham et al. (2012)				x			

Table 1: Classification of the thematic view extraction approaches based on the features considered by them.

3 Preliminaries

In this section, we present some preliminary concepts and techniques necessary to understand our approach to extracting thematic views. In particular, we illustrate our model to represent data lake sources and our approach to “structuring” unstructured data. Actually, as we pointed out in the Introduction, these are two further (even if collateral) contributions of our paper.

3.1 A unifying model for representing the metadata of data lake sources

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

Our model represents a data lake DL 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: $\mathcal{M}_{DL} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_m\}$.

3.1.1 Metadata classification

Following what it is said in Oram (2015), metadata can be divided into three categories, namely:

1. *Business metadata*, which include business rules (e.g., the upper and lower limit of a particular field, integrity constraints);
2. *Operational metadata*, which include information automatically generated during data processing (e.g., data quality, data provenance, executed jobs);
3. *Technical metadata*, which include information about data format and schema.

Based on this classification, we represent $\mathcal{M}_k$ as the union of three sets $\mathcal{M}_k^B \cup \mathcal{M}_k^O \cup \mathcal{M}_k^T$, related to business, operational and technical metadata, respectively.

As an advancement of the model of Oram (2015), we observe that these three subsets are intersected with each other (as shown in Figure 1). For instance, since business metadata contain all business rules that are mainly expressed in terms of data fields, and since the data schema is included in the technical metadata, we can conclude that data fields represent the perfect intersection between these two subsets. Analogously, technical metadata contain the data type and length, the possibility that a field can be NULL or auto-incrementing, the number of records, the data format and some dump information. These last three things are in common with operational metadata, which contain information like sources and target location, and the file size as well. Finally, the intersection between operational and business metadata represents information about the dataset license, the hosting server and so forth (e.g. see the DCMI Metadata Terms).

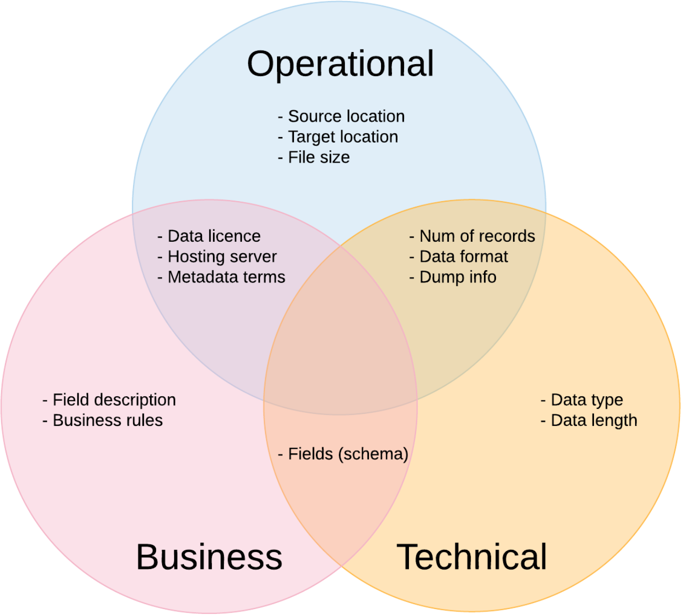


Figure 1: Metadata classification.

In this paper, we focus on business and technical metadata. Indeed, they denote, at the intensional level, the information content stored in the data lake sources and are those of interest for supporting most tasks, including the ones described here. In particular, we focus on the intersection of the two categories, which contains the data fields, both domain description and technical details. For instance, in a structured database, this intersection contains the attributes of the tables. Instead, in a semi-structured data source, it consists of the names of the (complex or simple) elements and attributes of the schema. Finally, in an unstructured source, it could consist of a set of keywords generally used to give an idea of the source content.

3.1.2 A network-based model for business and technical metadata

We indicate by $\mathcal{M}_k^{BT}$ the intersection between $\mathcal{M}_k^B$ and $\mathcal{M}_k^T$. We denote by $\mathcal{O}_k$ the set of all the objects stored in $\mathcal{M}_k^{BT}$. The concept of "object" depends on the data source typology. For instance, in a relational database, objects denote its tables and their attributes. In an XML document or in a JSON one, objects include complex/simple elements and their attributes.

In order to represent $\mathcal{M}_k^{BT}$, our model relies on a suitable directed graph $G_k^{BT} = \langle N_k, A_k, \Omega_k \rangle$.

Definition 3.1 Given a set of labels Λ, a *labeled direct graph* $G = \langle N, A, \Omega \rangle$ is a graph such that:

- N is the set of nodes;
- A is the set of arcs (n_s, n_t) from $n_s \in N$ to $n_t \in N$;
- $\Omega : A \rightarrow \Lambda$ is a mapping function s.t. $\Omega(a) = l \in \Lambda$ is the label of the arc a. □

For each object $o_{k_j} \in \mathcal{O}_k$ there exists a node $n_{k_j} \in N_k$. As there is a one-to-one correspondence between a node of N_k and an object of $\mathcal{O}_k$, in the following, we will use the two terms interchangeably. The label $l_{k_i} = \Omega(a_{k_i} = (n_s, n_t))$ represents the kind of relationship occurring between n_s and n_t. In this work, we consider the following kinds of relationships:

- *Structural relationship*: it is used to represent the relationship between an object and its sub-objects or between an object and the other ones structurally linked to it. For instance, it is used to indicate the relationship between a relational table and its attributes, or foreign keys relationships between tables, or, again, between a JSON complex object and its simple objects, or, finally, between a simple object and its attributes. In order to refer to a uniform representation of different structured sources (e.g., relational databases and XML documents), hereby we denote all structural relationships by the same label "`contains`".
- *Definition relationship*: it is represented by the label "`lemmaOf`" and denotes that the target node is a lemma included in the source's definition (or gloss). Again, its usage will be clear in Section 3.2.
- *Similarity relationship*: it is denoted by the label "`similarTo`" and represents a form of similarity between two objects. We will see an example of its semantics and usage in Section 3.2.

Many more relations could be defined by relying on more expressive dictionaries and/or defining sub-properties (e.g., specific types of containment, or specific types of similarities), especially when dealing with specific application domains. For the sake of generality, however, the approach described in this paper sticks on such a minimal set, which is capable to account for both structural aspects ("contains"), glossaries/definitions ("lemmaOf", useful especially for unstructured sources) and similarity.

Also, it is worth pointing out that our model enables a scalable and flexible representation and management of the metadata of heterogeneous data lake sources. Indeed, adding a new data source only requires the extraction of its metadata and their conversion to our model. Furthermore, the integration of metadata regarding different data sources can be simply performed by adding suitable arcs between the nodes for which there exists some relationship.

Similarly, G_k^{BT} can be extended with external knowledge graphs (e.g., DBpedia). In the following, we refer to an extension of G_k^{BT} as $G_k^{Ext} = G_k^{BT} \cup G^E$, where G^E is an external knowledge graph. An arc from a node of G_k^{BT} to its corresponding node in G^E will be labeled as "`externalSource_X`", where `X` is the name of the external knowledge graph at hand.

3.2 Defining a structure for unstructured sources

Based on a generic graph representation, our model is perfectly fitted for representing and managing both structured and semi-structured data sources. The highest difficulty regards unstructured data because it is worth avoiding a flat representation. Indeed, a trivial way to represent this kind of data sources consists in a set of simple elements, one 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 carrying out this task consists of four phases, namely: *(1)* creation of nodes; *(2)* extraction of lexical similarities; *(3)* extraction of string similarities; *(4)* merging of similar nodes. We describe these phases below.

- *Phase 1.* As first step, our approach creates a node representing the source as a whole and a node for each keyword. Furthermore, it adds an arc (labeled "`contains`") from the node associated with the source to any node corresponding to a keyword[5]. Initially, there is no arc between two keywords. To determine the arcs to add, the next phases are necessary.

- *Phase 2.* The goal of this phase is to handle lexical similarities. For this purpose it leverages a suitable thesaurus. Taking the current trends into account, this should be a multimedia thesaurus; for this purpose, in our experiments, we have adopted BabelNet (Navigli and Ponzetto 2012). In particular, for each node n_{k_1} of the graph, corresponding to the keyword k_1, our approach adds a set of nodes representing its lemmas[6]. Then, for each lemma, we add an arc with label "`lemmaOf`" linking n_{k_1} to the node representing the lemma. A lexical similarity exists between two nodes k_1 and k_2 if they have at least one common lemma in the thesaurus. In this case, our approach adds an arc (with label "`similarTo`") from the node n_{k_1} to the node n_{k_2}, and vice versa.

- *Phase 3.* Here, our approach derives similarities between keywords. A similarity between two keywords k_1 and k_2 exists 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). In this case, it adds an arc from n_{k_1} to n_{k_2} and an arc from n_{k_2} to n_{k_1}. Both of them have "`similarTo`" as label. We have chosen N-Grams (Kondrak 2005) as string similarity metric because we have experimentally seen that it provides the best results in our context.

  Now, we illustrate in detail what "sufficiently high" means and how our approach operates. Let $StrSim$ be the set of the string similarities for each pair of keywords of the source into consideration. Each record in $StrSim$ has the form $\langle k_i, k_j, kd(k_i, k_j)\rangle$. Our approach first computes the maximum string similarity degree kd_{max} in $StrSim$. Then, it examines each string similarity registered therein. If $((kd(k_i, k_j) \geq th_k \cdot kd_{max})$ and $(kd(k_i, k_j) \geq th_{kmin}))$, which implies that

[5] Here and in the following, to make the presentation smoother, we use the term "source" (resp., "keyword") to denote both the source (resp., a keyword) and the corresponding node associated with it.

[6] In this paper, we use the term "lemma" according to the meaning it has in BabelNet (Navigli and Ponzetto 2012). Here, given a term, its lemmas are other objects (terms, emoticons, etc.) that contribute to specify its meaning.

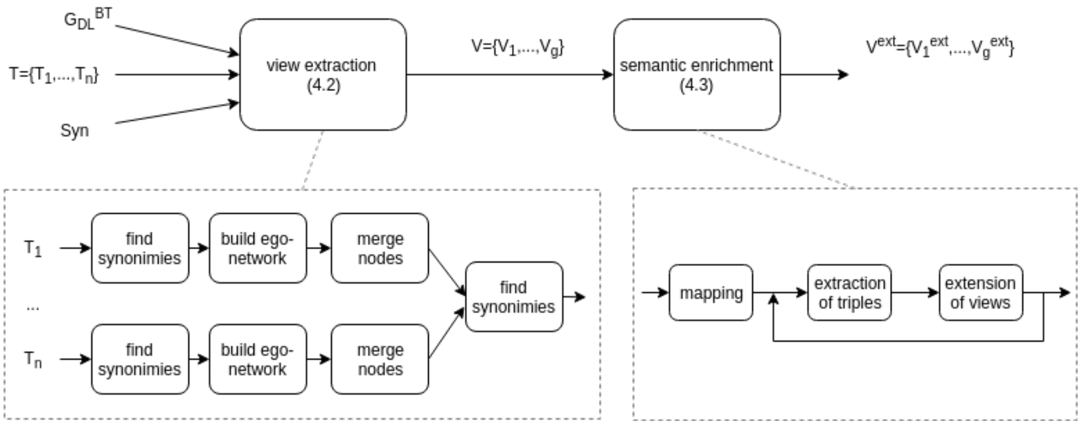


Figure 2: Overview of the approach to extracting thematic views.

the string similarity degree between k_i and k_j is among the highest ones in $StrSim$ and that, in any case, it is higher than or equal to a minimum threshold, then it concludes that there exists a similarity between n_{k_i} and n_{k_j}. We have experimentally set $th_k = 0.70$ and $th_{kmin} = 0.50$.

- *Phase 4.* This phase is devoted to merge similar nodes into a unique one. If an arc with label "`similarTo`" between two nodes n_{k_1} and n_{k_2} exists, our approach merges these nodes into a unique one, which inherits all the incoming and outgoing arcs of n_{k_1} and n_{k_2}. Finally, if two or more arcs from n_{k_1} to n_{k_2} with the same label exist, our approach merges them and returns only one arc with the same label[7].

4 An approach to extracting thematic views

Our approach to extracting thematic views operates on a data lake DL whose data sources are represented by means of the model described in Section 3.1. It consists of two steps: *view extraction* and *semantic enrichment*, as also summarized in Figure 2. The former is mainly based on the structure of the sources at hand, the latter mainly focuses on the corresponding semantics. Before describing and formalizing these two steps, we must introduce some background notions.

4.1 Background

Definition 4.1 Given a labeled direct graph $G = \langle N, A, \Omega \rangle$ and a node $n \in N$, *an Ego Network* $E = \langle N_E, A_E, \Omega_E \rangle$ is a subgraph of G such that:

- $A_E = \{a_E = (n_s, n_t) \mid (n_s = n \wedge (n, n_t) \in A) \vee ((n, n_s) \in A \wedge (n, n_t) \in A)\}$;
- $N_E = \{n_E \mid (n, n_E) \in A_E\} \cup \{n\}$;
- $\Omega_E(a_E) = \Omega(a_E), \forall a_E \in A_E$ □

[7] Note that Phases 2 and 4 could be merged into a unique one, avoiding to define arcs with label "`lemmaOf`". Here, we maintain these arcs and both phases to keep the information about similarity between nodes for future uses.

The node n is the *ego* of E, whereas the other nodes are the *alters*. The function $ego(E)$ returns the ego E, whereas $alter(E)$ returns the set of the alters of E, i.e. $alter(E) = N_E \setminus ego(E)$. The function $en(n, G)$ returns the ego network E focused on n (i.e., $n = ego(E)$) from G.

Definition 4.2 Given two ego networks $E_1 = \langle N_1, A_1, \Omega_1 \rangle$ and $E_2 = \langle N_2, A_2, \Omega_2 \rangle$, *the graft* of E_2 in E_1 is the ego network $E_g = \langle N_g, A_g, \Omega_g \rangle$ such that:

- $N_g = N_1 \cup N_2 \setminus \{ego(E_2)\}$;
- $A_g = A_1 \cup \{(n_s, n_t) \mid (n_s = ego(E_1) \wedge (ego(E_2), n_t) \in A_2) \vee ((n_s, n_t) \in A_2 \wedge n_s, n_t \in alter(E_2))\}$;
- $\Omega_g(a) = \begin{cases} \Omega_1(a), & if\ a \in A_1 \\ \Omega_2(a), & if\ a \in A_2 \\ \Omega_2(ego(E_2), n_t), & if\ a = (ego(E_1), n_t) \wedge (ego(E_2), n_t) \in A_2 \end{cases}$

□

The function $graft(E_1, E_2)$ returns a new ego network E_g, which is the graft of the ego network E_2 in the ego network E_1. The ego of E_g is equal to the ego of E_1.

Definition 4.3 Given a labeled direct graph $G = \langle N, A, \Omega \rangle$, *the merge of* $n_y \in N$ *and* $n_x \in N$ is a labeled direct graph $G_m = \langle N_m, A_m, \Omega_m \rangle$ such that:

- $N_m = N \setminus \{n_y\}$;
- $A_m = A \cup \{(n_s, n_x) \mid (n_s, n_y) \in A \wedge n_s \neq n_x\} \cup \{(n_x, n_t) \mid (n_y, n_t) \in A \wedge n_t \neq n_x\} \setminus \{(n_s, n_t) \mid (n_s = n_y \vee n_t = n_y) \wedge (n_s, n_t) \in A\}$
- $\Omega_m(a) = \begin{cases} \Omega(n_s, n_y), & a = (n_s, n_x) \wedge (n_s, n_y) \in A \\ \Omega(n_y, n_t), & a = (n_x, n_t) \wedge (n_y, n_t) \in A \\ \Omega(a), & otherwise \end{cases}$

□

The function $mergeNodes(G, n_1, n_2)$ returns a new graph G_m obtained from G by merging the nodes n_1 and n_2. If G is an ego network and $n_1, n_2 \in alter(G)$, then G_m is an ego network.

4.2 View extraction

At the beginning, our approach to extracting views from a data lake DL requires a set $T = \{T_1, T_2, \cdots, T_l\}$ of topics, representing the themes of interest for the user, and a dictionary Syn of synonyms involving the objects stored in the sources of DL. This could be a generic thesaurus, such as BabelNet (Navigli and Ponzetto 2012), 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 (De Meo et al. 2006), MOMIS (Bergamaschi et al. 2001) or Cupid (Madhavan et al. 2001). Algorithm 1 describes the pseudo-code of our approach.

Let T_i be a topic of T. Let $Obj_i = \{o_{i_1}, o_{i_2}, \cdots, o_{i_q}\}$ be the set of the objects in DL that are synonymous (according to Syn) of T_i. Let $N_i = \{n_{i_1}, n_{i_2}, \cdots, n_{i_q}\}$ be the nodes corresponding to Obj_i. First, our approach constructs the ego networks $E_{i_1}, E_{i_2}, \cdots, E_{i_q}$ having $n_{i_1}, n_{i_2}, \cdots, n_{i_q}$ as the corresponding egos (Algorithm 1, Steps 2-5). Then, it merges all the egos into a unique node n_i. In this way, it obtains a unique ego network E_i, corresponding to T_i, from $E_{i_1}, E_{i_2}, \cdots, E_{i_q}$ (Algorithm 1, Steps 7-9).

If a synonymy exists (according to Syn) between two alters belonging to different ego networks, then these are merged into a unique node and the corresponding arcs linking them to the ego n_i are merged into a unique arc (Algorithm 1, Steps 10-14).

The previous task is performed for each $T_i \in T$, so that, at the end, we have a set $E = \{E_1, E_2, \cdots, E_l\}$ of l ego networks.

At this point, all the nodes belonging to different ego networks in E that are synonyms (according to Syn) are merged into a unique node. At the end, a (potentially disconnected) graph V is obtained. V consists of g $(1 \leq g \leq l)$ connected graphs $\{V_1, \cdots, V_g\}$, which represent potential views (Algorithm 1, Steps 17-22).

If $g = 1$, then there exists a unique thematic view comprising all the topics required by the user. Otherwise, more views exist, each comprising some (but not all) of the topics of interest for the user.

Algorithm 1 Pseudo-code describing the first step of our approach (View extraction)

```
Input:
    let $DL$ be a data lake consisting of a set $\{D_1, D_2, \cdots, D_m\}$ of sources;
    let $G_k^{BT} = \langle N_k^{BT}, A_k^{BT}, \Omega_k^{BT} \rangle$ be the network-based representation of $D_k \in DL$;
    let $G_{DL}^{BT} = \{G_1^{BT}, G_2^{BT}, \cdots, G_m^{BT}\}$;
    let $T = \{T_1, T_2, \cdots, T_l\}$ be the set of topics;
    let $s(n_x, n_y)$ be true iff $n_x$ and $n_y$ are synonyms according to $Syn$.
Output:
    the set $V$ of views

 1: for each $\mathrm{T}_i \in \mathrm{T}$ do
 2:     find $N_i = \{n_{i_1}, \cdots, n_{i_q}\}$ such that $\forall n_{i_x} \in N_i \exists G_x^{BT} \in G_{DL}^{BT} \wedge n_{i_x} \in N_x^{BT} \wedge s(T_i, n_{i_x}) == true$
 3:     for each $n_{i_x} \in N_i$ do
 4:         $E_{i_x} = en(n_{i_x}, G_x^{BT})$
 5:     end for
 6:     $E_i = E_{i_1}$
 7:     for $k := 2$ to $q$ do
 8:         $E_i = graft(E_i, E_{i_k})$
 9:     end for
10:     for each $n_x, n_y \in alter(E_i)$ do
11:         if $s(n_x, n_y) == true$ then
12:             $E_i = mergeNodes(E_i, n_x, n_y)$
13:         end if
14:     end for
15: end for
16: $E = \{E_1, E_2, \cdots, E_l\}$
17: $V = \langle N_v, A_v, \Omega_v \rangle = \bigcup_i E_i$
18: for each $n_x, n_y \in N_v$ do
19:     if $s(n_x, n_y) == true$ then
20:         $V = mergeNodes(V, n_x, n_y)$
21:     end if
22: end for
```

4.3 Semantic enrichment

This part starts by constructing the graph V_i^{Ext} obtained by extending the view V_i with an external knowledge graph G^E semantically enriching V_i. Any suitable external graph, or set of graphs, can be used for this purpose, e.g. dictionaries, glossaries, ontologies. In this paper, we rely on DBpedia, a project aiming to extract structured content from the information created in the Wikipedia project and now including more than 4.5 million terms represented as a knowledge graph in RDF.

For this purpose, first each node n_{i_j} of V_i is linked to the corresponding entry $n_{i_j}^E \in G^E$ through an arc with label "`externalSource_DBpedia`". In our scenario, such a DBpedia node $n_{i_j}^E$ is already specified in the BabelNet entry corresponding to n_{i_j} (or to any of its synonyms in Syn)[8].

Then, for each $n_{i_j}^E$ considered above, it retrieves all the related concepts. In DBpedia, knowledge is structured according to the Linked Data principles (Heath and Bizer 2011), i.e. as an RDF graph built by triples. Each triple $\langle s(ubject), p(roperty), o(bject)\rangle$ states that a subject s has a property p, whose value is an object o. Therefore, retrieving the related concepts for a given element x implies finding all the triples where x is either the subject or the object.

The procedure to extend a view $V_i \in V$ (see Algorithm 2) consists of the following tasks:

- *Mapping*: for each node $n_{i_j} \in V_i$, its corresponding DBpedia entry $n_{i_j}^E$ is found. An arc from n_{i_j} to $n_{i_j}^E$ with label "`externalSource_DBpedia`" is added to V_i (Algorithm 2, Steps 1-8);

- *Extraction of triples*: all the related triples $\langle n_{i_j}^E, p, o\rangle$ and $\langle s, p, n_{i_j}^E\rangle$, i.e., all the triples in which $n_{i_j}^E$ is either the subject or the object, are retrieved (Algorithm 2, Step 12);

- *Extension of views*: for each retrieved triple $\langle n_{i_j}^E, p, o\rangle$ (resp., $\langle s, p, n_{i_j}^E\rangle$), V_i is extended: *(i)* by defining a node (if not already existing) for the object o (resp., s), and *(ii)* by drawing an arc from $n_{i_j}^E$ to o (resp., from s to $n_{i_j}^E$) labeled as p (Algorithm 2, Steps 13-24).

The second and third tasks are recursively repeated for each new added node. The procedure stops after a given number of iterations, defined in such a way as to limit the length of the external incoming and outcoming paths of the nodes of V_i. The longer the path, the weaker the semantic link between nodes.

The enrichment procedure is performed for all the views of V. It is particularly important if $|V| > 1$ because the new derived relationships could help to merge the thematic views that have not been merged during the view extraction phase. In particular, let $V_i \in V$ and $V_h \in V$ be two views of V, and let V_i^{Ext} and V_h^{Ext} be the extended views corresponding to them. If there exist two nodes $n_{i_j} \in V_i^{Ext}$ ad $n_{h_k} \in V_h^{Ext}$ such that $n_{i_j} = n_{h_k}$[9], then they can be merged in one node; in this way, V_i^{Ext} and V_h^{Ext} become connected.

After all equal nodes of the views of V have been merged, all the views of V could be either merged in one view or not. In the former case, the process terminates with success. Otherwise, it is possible to conclude that no thematic view comprising all the topics specified by the user can be found. In this

[8] Whenever this does not happen, the mapping can be automatically provided by the DBpedia Lookup Service (`http://wiki.dbpedia.org/projects/dbpedia-lookup`).

[9] Here, two nodes are assumed to be equal if the corresponding names coincide.

Algorithm 2 Pseudo-code describing the second step of our approach (Semantic enrichment)

Input:

Let V be the set of views obtained at the end of *view extraction*;
let G_E be an external knowledge graph;
let $NumIter$ be the number of iterations;
let $\mathit{findExternalNode}(n_{i_j}, G^E)$ be a function that, given a node n_{i_j}, finds the corresponding node in the external graph G^E;
let $\mathit{findTriples}(n^E_{i_j})$ be a function that retrieves from G^E all RDF triples where the node $n^E_{i_j}$ is either subject or object.

Output:

the set V^{Ext} of views

```
for each $V_i = \langle N_{v_i}, A_{v_i}, \Omega_{v_i} \rangle \in V$ do
    for each $n_{i_j} \in N_{v_i}$ do
        $n^E_{i_j} = \mathit{findExternalNode}(n_{i_j}, G^E)$
        $N_{v_i} = N_{v_i} \cup \{n^E_{i_j}\}$
        $A_{v_i} = A_{v_i} \cup \{a' = \langle(n_{i_j}, n^E_{i_j})\}$
        $\Omega_{v_i}(a') =$"externalSource$_{G^E}$"
    end for
end for
for $k := 1$ to $NumIter$ do
    for each $V_i = \langle N_{v_i}, A_{v_i}, \Omega_{v_i} \rangle \in V$ do
        $V_i^{Ext} = \langle N^{Ext}_{v_i}, A^{Ext}_{v_i}, \Omega^{Ext}_{v_i} \rangle = V_i$
        for each $n_{i_j} \in N^{Ext}_{v_i}$ do
            $Triples = \mathit{findTriples}(n_{i_j})$
            for each $tr_k \in Triples$ do
                if $tr_k == \langle n_{i_j}, p, o \rangle$ then
                    $N^{Ext}_{v_i} = N^{Ext}_{v_i} \cup \{o\}$
                    $A^{Ext}_{v_i} = A^{Ext}_{v_i} \cup \{a' = (n_{i_j}, o)\}$
                    $\Omega^{Ext}_{v_i}(a') = p$
                end if
                if $tr_k == \langle s, p, n_{i_j} \rangle$ then
                    $N^{Ext}_{v_i} = N^{Ext}_{v_i} \cup \{s\}$
                    $A^{Ext}_{v_i} = A^{Ext}_{v_i} \cup \{a' = (s, n_{i_j})\}$
                    $\Omega^{Ext}_{v_i}(a') = p$
                end if
            end for
        end for
    end for
end for
```

last case, our approach still returns the enriched views of V and leaves the user the choice to accept of reject them.

The quality of the integration task strictly depends on: *(i)* the type of the properties of the properties paths linking the merged views, *(ii)* their length, and *(iii)* their overall cost. For each of these aspects, the user can tune specific parameters. As for *(i)*, we manually evaluated DBpedia properties in order to assess their meaningfulness when used to link concepts belonging to different views. As a result, we evaluated some of the properties as non-significant and we filtered out them from the procedure. For instance, among them we discarded http://dbpedia.org/ontology/country because it is a property used to link a concept to its nation, and, as such, its semantics is too loose as it could connect any two concepts referring to the same country. If necessary, in making our decisions we could also use the results of semi-automatic approaches performing machine learning based rankings (e.g., in Dessi and Atzori (2016)). As for *(ii)*, we count the number of properties. Finally, as for *(iii)*, we consider the overall cost of each property in the path. In our tests, we always referred to a cost equal to 1, with the exception of the http://www.w3.org/2002/07/owl#sameAs property, which states that two concepts are equivalent, and has weight 0.

Properties to be filtered out, as well as property weights, can be customized through a configuration file.

5 An example case

In this section, we present an example case aiming to show the various steps of our approach. Here, we consider: *(i)* a structured source, called *Weather Conditions* (W, in short); *(ii)* two semi-structured sources, called *Climate* (C, in short) and *Environment* (E, in short); *(iii)* an unstructured source, called *Environment Video* (V, in short), consisting of a YouTube video. The E/R schema of W and the XML Schemas of C and E are not reported here for space limitations. However, the interested reader can find them at the address `http://daisy.dii.univpm.it/dl/datasets/thematicViews`. The keywords of V are: $garden$, $flower$, $rain$, $save$, $earth$, $tips$, $recycle$, $aurora$, $planet$, $garbage$, $pollution$, $region$, $life$, $plastic$, $metropolis$, $environment$, $nature$, $wave$, eco, $weather$, $simple$, $fineparticle$, $climate$, $ocean$, $environmentawareness$, $educational$, $reduce$, $power$, $bike$.

By applying the approach mentioned in Section 3, we obtain the corresponding representations in our network-based model, shown in Figures 3, 4 and 5[10].

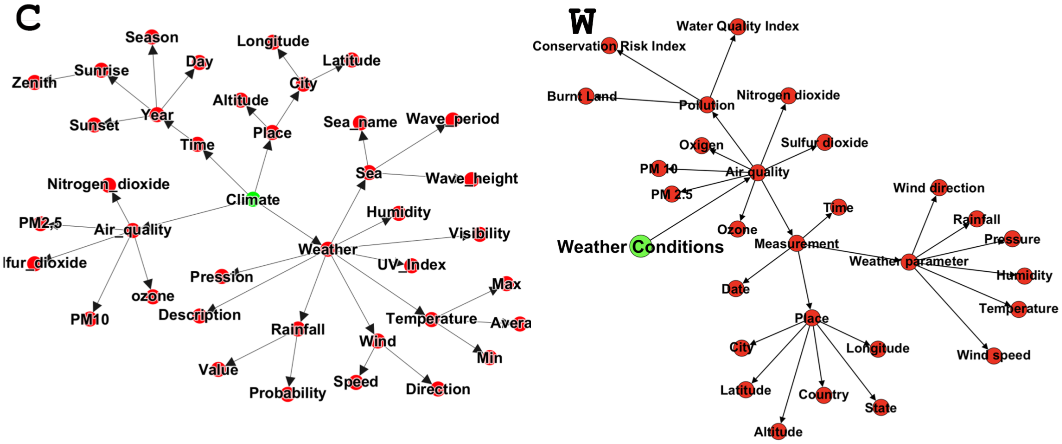


Figure 3: Network-based representations of *Climate* and *Weather Conditions*.

Assume, now, that a user specifies the following set T of topics of her interest: $T = \{Ocean, Area\}$. First, our approach detects those terms (and, then, those objects) in the four sources that are synonyms of $Ocean$ and $Area$. As for $Ocean$, the only synonym present in the sources is Sea; as a consequence, Obj_1 comprises the node $Ocean$ of the source V ($V.Ocean$[11]) and the node Sea of the source C ($C.Sea$). An analogous activity is performed for $Area$. At the end of this task we have that $Obj_1 = \{V.Ocean, C.Sea\}$ and $Obj_2 = \{W.Place, C.Place, V.Region, E.Location\}$.

[10] In Figures 3 and 4, we do not show the arc labels for the sources C, W and E because all of them are "contains" and their presence would have complicated the layout unnecessarily.

[11] Hereafter, we use the notation $S.o$ to indicate the object o of the source S.

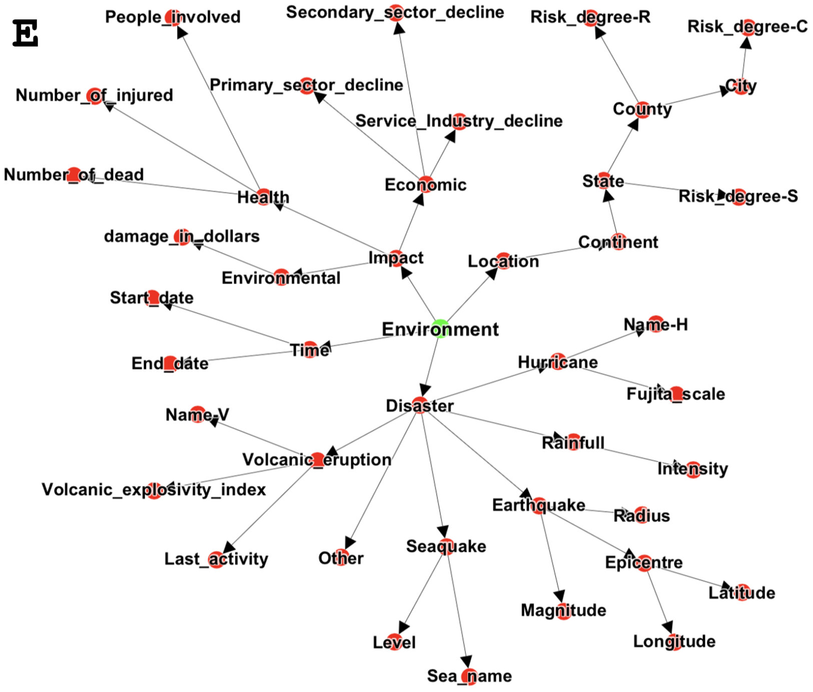


Figure 4: Network-based representations of *Environment*.

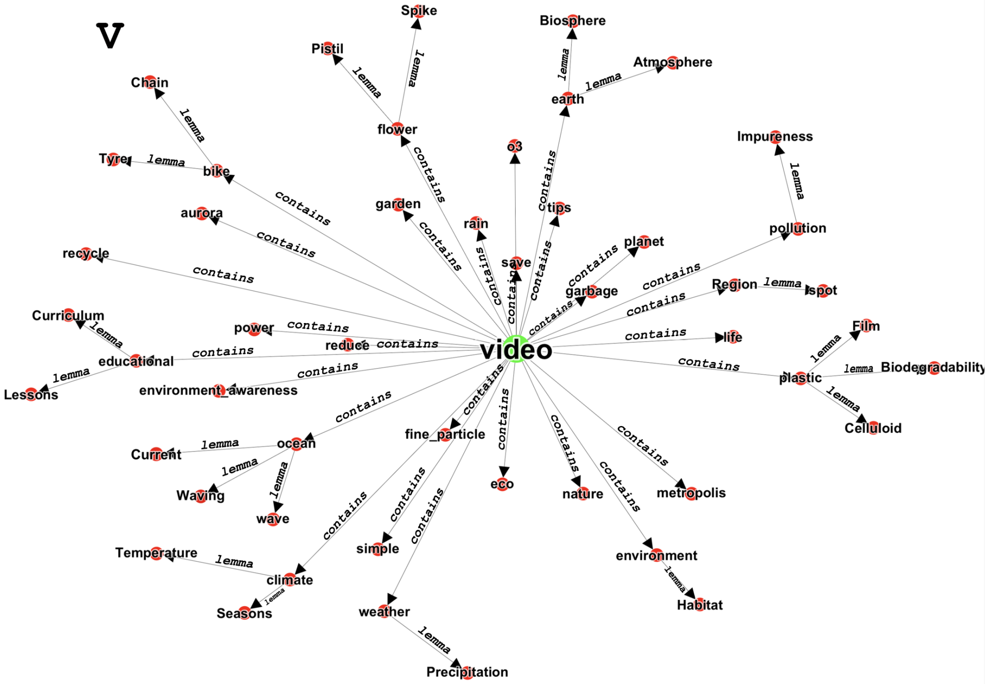


Figure 5: Network-based representations of *Environment Video*.

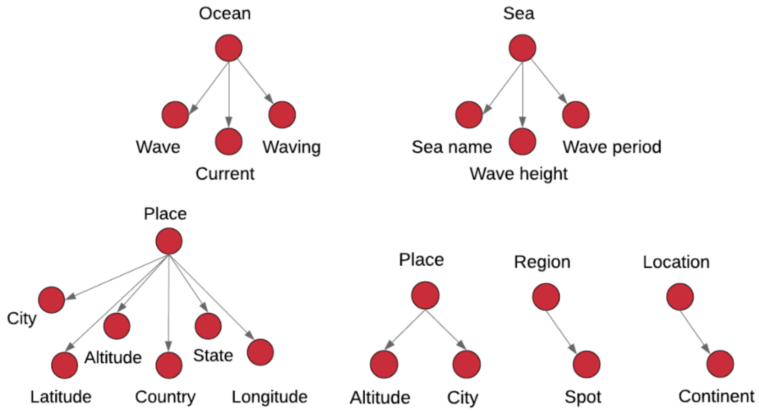
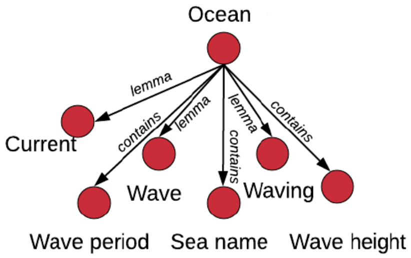


Figure 6: Ego networks corresponding to *V.Ocean*, *C.Sea*, *W.Place*, *C.Place*, *V.Region* and *E.Location*.

Our approach proceeds by constructing the ego networks corresponding to the objects of Obj_1 and Obj_2. They are reported in Figure 6[12].

Now, consider the ego networks corresponding to *V.Ocean* and *C.Sea*. Our approach merges the two egos into a unique node. Then, it verifies whether further synonyms exist between the alters. Since none of these synonyms exists, it returns the ego network shown at the left of Figure 7. The same task is performed for the ego networks corresponding to *W.Place*, *C.Place*, *V.Region* and *E.Location*. In particular, first the four egos are merged. Then, synonyms between the alters *W.City* and *C.City* and the alters *W.Altitude* and *C.Altitude* are retrieved. Based on this, *W.City* and *C.City* are merged in one node, *W.Altitude* and *C.Altitude* are merged in another node, the arcs linking the ego to *W.City* and *C.City* are merged in one arc and the ones linking the ego to *W.Altitude* and *C.Altitude* are merged in another arc. In this way, the ego network shown at the right of Figure 7 is returned. At this point, there are two ego networks, E_{Ocean} and E_{Area}, each corresponding to one of the terms specified by the user.



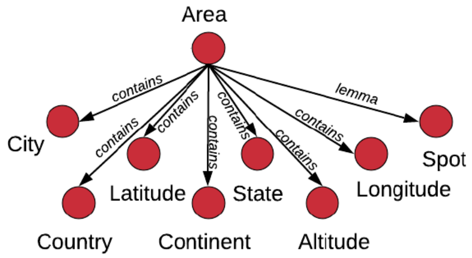


Figure 7: Ego networks corresponding to *Ocean* and *Area*.

Our approach proceeds by searching for synonymies between the nodes of E_{Ocean} and E_{Area}. Since

[12] In this figure, for layout reasons, we do not show the arc labels because they are the same as the ones of the corresponding arcs of Figures 3, 4 and 5.

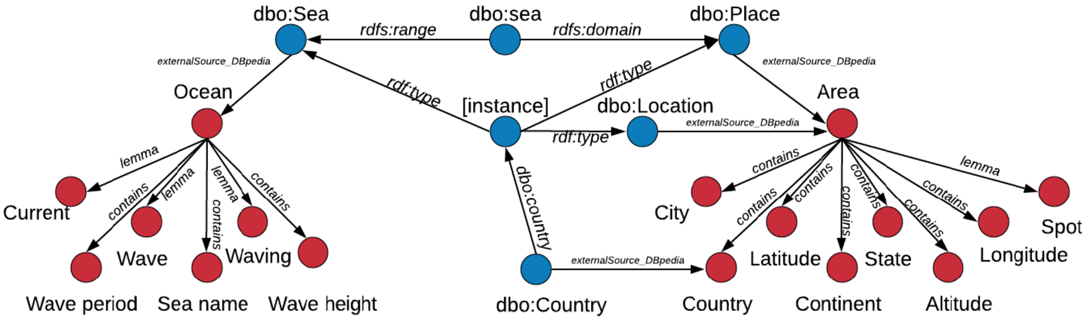


Figure 8: The integrated thematic view.

it does not find them, it returns the set $V = \{V_{Ocean}, V_{Area}\}$, where V_{Ocean} (resp., V_{Area}) coincides with E_{Ocean} (resp., E_{Area}).

At this point, the semantic enrichment procedure (see Section 4.3) is executed. As shown in Figure 8, first each term is semantically aligned to the corresponding DBpedia entry (e.g., *Ocean* is linked to *dbo:Sea*, *Area* is linked to *dbo:Location* and *dbo:Place*, whereas *Country* is linked to *dbo:Country*[13], respectively). After a single iteration, the triples ⟨*dbo:sea rdfs:range dbo:Sea*⟩ and ⟨*dbo:sea rdfs:domain dbo:Place*⟩ are retrieved; they mean that a property *dbo:sea* is defined in DBpedia, stating that a *dbo:Place* is lapped by a *dbo:Sea* (e.g, Italy by the Mediterranean sea). Other connections can be found by moving to specific instances of the mentioned resources. Indeed, the triples ⟨*instance rdf:type dbo:Sea*⟩, ⟨*instance rdf:type dbo:Location*⟩ and ⟨*instance rdf:type dbo:Place*⟩ are retrieved. Finally, a triple ⟨*instance dbo:country dbo:Country*⟩ is also determined. As a result, the semantic enrichment procedure succeeded in merging the two views, which were still separated after the view extraction step.

In order to validate the results, we also performed an evaluation done by a panel of three experts. The experts were given the graphs in Figures 3, 4, 5 and were asked to manually extract the thematic views starting from the two topics "Ocean" and "Area". Each expert manually produced the views including the nodes that, according to them, were relevant to each topic. We compared the views produced by the experts to those generated by our approach, and considered as valid those nodes on which the majority (at least 2 out of 3 experts) agreed. Finally, we calculated precision and recall, that are reported in Table 2. Experts mostly agreed with each other, with some exceptions. For instance, two of them recognized the node *seaquake* as relevant for "Ocean" and *environment* as relevant for "Area", which were not included in our views. On the other hand, the majority did not agree on *current* being relevant for "Ocean". As also shown by the results, apart from these cases, their views overlap with those produced by our approach.

[13] Prefixes *dbo* and *dbr* stand for http://dbpedia.org/ontology/ and http://dbpedia.org/resource/

	Precision	Recall	F-Measure
Ocean	0.86 (6/7)	0.86 (6/7)	0.86
Area	0.90 (9/10)	0.83 (10/12)	0.87
Average value	0.88	0.85	0.86

Table 2: Results of the expert validation for the example case.

6 Experiments

In this section, we present the experiments that we carried out to evaluate the performance of our approach from several viewpoints. Specifically, we describe the testbed in next subsection and the experiments on *view extraction*, along with the underlying motivations and the obtained results, in Subsections 6.2, 6.3 and 6.4. Then, we present the experiments regarding *semantic enrichment* in Subsection 6.5, whereas we provide details on computation time of both view extraction and semantic enrichment in Subsection 6.6.

6.1 Adopted Testbed

To perform our experimental campaign, we built six data lakes $DL_1, \ldots, DL_6$ with an increasing number of metadata. Each data lake consisted of 20 sources, with heterogeneous formats. For each data lake DL_k, by following the methodology described in Section 3.1.2, we built a graph G_k^{BT} representing its technical and business metadata. Hereafter, we use the notation G_k^E to represent the corresponding external graph, and the notation G_k^{Ext} to represent the union of G_k^{BT} and G_k^E.

The number of nodes of G_k^{BT} for the six data lakes $DL_1, \ldots, DL_6$ were 208, 356, 572, 928, 1482 and 2392, respectively. The interested reader can find these data lakes in CSV format at the address http://daisy.dii.univpm.it/dl/datasets/thematicViews.

We carried out all the tests presented in this section on a server equipped with an Intel I7 Quad Core 7700 HQ processor and 16 GB of RAM with Ubuntu 16.04 operating system. To implement our approaches we adopted Python, powered with the NetworkX library, as programming language, and Neo4J (Version 3.4.5) as underlying DBMS.

6.2 Cohesion

The approach proposed in Section 4 is aimed to extract thematic views from graphs representing data sources. These views should have both structural and semantic cohesion higher than the ones measured for original data sources. The verification of this assumption is the goal of the first experiment.

We considered two well known structural cohesion measures used in network analysis literature, namely *clustering coefficient* and *density* (M.Tsvetovat and Kouznetsov 2011). The clustering coefficient of a node is defined as the probability that two randomly chosen (but distinct) neighbors of it are connected. The clustering coefficient of a network is the average of the clustering coefficients of its nodes. The density of a network is given by the ratio of the real arcs of the network to the maximum number of arcs that could be present in it. Both clustering coefficient and density range in the real interval $[0, 1]$; the higher their value the higher the corresponding network cohesion.

For each data lake, we considered 4 groups of topic sets with 1, 2, 4 and 8 topics, respectively. We chose this range by considering that the vast majority of search queries is composed of less than 3 words, according to the statistics published by various search engines (e.g., in Yahoo the average length is 2.35 and approximately 85% of queries include less than 4 words (Yi et al. 2008) while queries with 8 or fewer words account for more than 99% of searches. Similar trends are reported by other search engines (Spink et al. 2001)). The size of the set, that is the number of its topics, is denoted by $|T|$. For each size, we randomly generated 10 different sets of topics and we used them as input to our approach. For each topic set, our approach returned a thematic view for which we computed the corresponding clustering coefficient and density. Finally, for each data lake, we averaged the obtained results and compared them with the average clustering coefficient and the average density of the corresponding original data sources. The obtained results are reported in Tables 3 and 4.

G_k^{BT} (size)	Average clustering coefficient (real sources)	Average clustering coefficient (thematic views)			
		$\|T\| = 1$	$\|T\| = 2$	$\|T\| = 4$	$\|T\| = 8$
G_1^{BT} (208)	0.242	0.328	0.379	0.387	0.412
G_2^{BT} (356)	0.280	0.353	0.397	0.422	0.444
G_3^{BT} (572)	0.294	0.402	0.442	0.485	0.492
G_4^{BT} (928)	0.358	0.451	0.483	0.505	0.516
G_5^{BT} (1482)	0.394	0.491	0.515	0.532	0.536
G_6^{BT} (2392)	0.396	0.523	0.537	0.546	0.548

Table 3: Values of the clustering coefficient of the data sources and the thematic views against the size of G_k^{BT} and the size of the topic set.

G_k^{BT} (size)	Average density (real sources)	Average density (thematic views)			
		$\|T\| = 1$	$\|T\| = 2$	$\|T\| = 4$	$\|T\| = 8$
G_1^{BT} (208)	0.255	0.265	0.270	0.284	0.301
G_2^{BT} (356)	0.268	0.296	0.308	0.315	0.324
G_3^{BT} (572)	0.279	0.396	0.399	0.405	0.411
G_4^{BT} (928)	0.273	0.481	0.489	0.494	0.514
G_5^{BT} (1482)	0.290	0.551	0.561	0.573	0.580
G_6^{BT} (2392)	0.278	0.615	0.615	0.626	0.634

Table 4: Values of the density for the data sources and the thematic views against the size of G_k^{BT} and the size of the topic set.

.

From the analysis of these tables, we can observe that, in almost all cases, the values of both clustering coefficient and density are higher or much higher for thematic views than for the original data sources. This is clearly a confirmation of the goodness of our approach, which returns thematic views more cohesive than the original sources. Conversely, if views were selected randomly, they would have had a distribution of arcs similar to the full data lake, and therefore a comparable value for cohesion. We also observe that when $|T|$ increases, the values of both clustering coefficient and density increase. This can be explained by observing that, in processing T, our approach selects the portions of networks containing at least one topic of T. When $|T|$ increases, the portion of networks selected by our approach increases too, and the probability of selecting nodes that are synonyms and, hence, will be merged, increases as well; this leads to a higher cohesion value.

6.3 Connecting capability and node distribution in thematic views

Another quality parameter for thematic views is their connecting capability, that is the capability of a view to connect more data sources. This capability depends on the number of synonym relationships added to the model by the proposed approach, and, in turn, on the number of merged nodes. Indeed, we merge all the pairs of synonym nodes coming from different data sources, since "de-facto" they refer to the same concepts (see functions *graft()* and *mergeNodes()* in Algorithm 1). Hence, the number of merged nodes can be used to count the synonymy relationships in the model. The higher the number of merged nodes in a thematic view, the higher its capability of connecting data sources.

For each data lake and thematic view used in the previous subsections, we computed the ratio of the number of merged nodes to the number of nodes of the view (MN_{view}). Furthermore, we compute the ratio of the number of different data sources, which the merged nodes belong to, to the total number of data sources (MN_{source}). Clearly, the higher the two ratios, the higher the connecting capability of thematic views. Results averaged for $|T|$ are reported in Table 5.

$\mathcal{G}_k^{BT}$ (size)	Average MN_{view}				Average MN_{source}			
	$\|T\| = 1$	$\|T\| = 2$	$\|T\| = 4$	$\|T\| = 8$	$\|T\| = 1$	$\|T\| = 2$	$\|T\| = 4$	$\|T\| = 8$
$\mathcal{G}_1^{BT}$ (208)	0.308	0.460	0.523	0.558	0.378	0.473	0.492	0.457
$\mathcal{G}_2^{BT}$ (356)	0.386	0.519	0.613	0.658	0.369	0.474	0.531	0.492
$\mathcal{G}_3^{BT}$ (572)	0.544	0.667	0.786	0.818	0.488	0.481	0.453	0.444
$\mathcal{G}_4^{BT}$ (928)	0.694	0.791	0.866	0.883	0.457	0.432	0.422	0.418
$\mathcal{G}_5^{BT}$ (1482)	0.814	0.887	0.944	0.950	0.457	0.519	0.811	0.921
$\mathcal{G}_6^{BT}$ (2392)	0.913	0.965	0.978	0.981	0.520	0.676	0.838	0.923

Table 5: Average MN_{view} and average MN_{source} against the size of G_k^{BT} and the size of the topic set.

From the analysis of these tables, we observe that our approach returns satisfying results. Indeed, note that MN_{view} increases when the size of the data lake increases. Furthermore, MN_{view} slightly increases when $|T|$ increases. Similar trends can be observed for the average MN_{source}. In this case, the value increases more as $|T|$ increases. Conversely, if there were no semantic relations (i.e. synonymy) it would not have been possible to merge sources, whatever the size of the network.

In order to deepen this investigation, for each thematic view, we compared the distribution of its nodes against the data sources they belong to. Indeed, if almost all the nodes of a thematic view derive from only one data source, the information contribution provided by the view itself would be very small because it would be analogous to the one provided by the corresponding source. On the contrary, if the nodes of a thematic view derive from several data sources, then the view provides new and valuable knowledge. Based on this reasoning, we evaluated the heterogeneity of the provenance of each node in a thematic view. For this purpose, we adapted the Herfindahl Index (Hirschman 1964) to our context. This index is very used in several research fields of Economics from several decades; for instance, it is exploited to evaluate the concentration degree in an industry.

In order to adapt the Herfindahl Index to our scenario, consider a data lake DL consisting of m data sources $\{D_1, D_2, \ldots, D_m\}$. Consider, also, a thematic view V_j derived by our approach. Let n_j be the number of nodes of V_j and let n_{j_k}, $1 \leq k \leq m$, be the fraction of the nodes of V_j belonging to D_k. The Herfindahl Index H_j of V_j is defined as $\sum_{k=1}^{m} \left(\frac{n_{j_k}}{n_j}\right)^2$. The Herfindahl Index generally used in Economics ranges in the real interval $\left[\frac{1}{m}, 1\right]$. In our case, since a node can derive from the merge of more synonymous nodes (specifically, it could represent at most m nodes), H_j can range in the real interval $\left[\frac{1}{m}, m\right]$. The higher the value of H_j, the higher the concentration degree of the nodes of data sources in V_j. As previously pointed out, a desired property is the ability to build thematic views connecting nodes that belong to different sources. In terms of the Herfindahl Index, this means to have values of the index as lower as possible[14].

Table 6 reports the average values of the Herfindahl Index computed for each data lake.

$\mathcal{G}_k^{BT}$ (size)	Average Herfindhal Index			
	$\lvert T\rvert = 1$	$\lvert T\rvert = 2$	$\lvert T\rvert = 4$	$\lvert T\rvert = 8$
$\mathcal{G}_1^{BT}$ (208)	0.322	0.312	0.308	0.296
$\mathcal{G}_2^{BT}$ (356)	0.284	0.277	0.272	0.264
$\mathcal{G}_3^{BT}$ (572)	0.242	0.236	0.225	0.217
$\mathcal{G}_4^{BT}$ (928)	0.163	0.156	0.147	0.143
$\mathcal{G}_5^{BT}$ (1482)	0.121	0.118	0.114	0.108
$\mathcal{G}_6^{BT}$ (2392)	0.112	0.110	0.087	0.068

Table 6: Average values of the Herfindahl Index of thematic views against the size of $\mathcal{G}_k^{BT}$ and the size of the topic set.

Results show that the nodes of a thematic view are distributed among several sources. Indeed, on average, the Herfindahl Index is 0.196 and the maximum value is 0.322. As expected, the values of H_j improve with the increase of both the size of the data lake and $|T|$, that is with the increase of the probability of having synonymy relationships in a thematic view. Table 6 confirms and strengthens the results obtained by computing the number of merged nodes in a thematic view.

6.4 Efficiency obtained thanks to thematic views

This subsection is devoted to measure the efficiency guaranteed by the presence of thematic views in a data lake. In order to perform this experiment we randomly selected a set $PSet$ of 52348 pairs of nodes (n_s, n_t) such that: *(i)* n_s and n_t belong to at least one thematic view; *(ii)* there exists at least one path from n_s to n_t.

In order to measure efficiency, for each pair (n_s, n_t), we conducted both a Breadth-First Search and a Depth-First Search in such a way as to reach n_t starting from n_s. Each of these searches was performed two times; during the former one we assumed the existence of no thematic views; instead, during the latter one, we assumed their presence. We computed the number $f_{st}^{BFS} = \frac{\widehat{num_{st}^{BFS}}}{num_{st}^{BFS}}$. Here, the numerator $\widehat{num_{st}^{BFS}}$ denotes the number of nodes involved in the search in presence of thematic views, whereas the denominator num_{st}^{BFS} indicates the number of nodes involved in the search but in absence of thematic views. In an analogous fashion, we computed num_{st}^{DFS}, $\widehat{num_{st}^{DFS}}$ and f_{st}^{DFS}, which correspond to the previous parameters but for Depth-First Search. Clearly, the lower f_{st}^{BFS} and f_{st}^{DFS}, the higher the contribution of the thematic views to reduce the number of nodes necessary to reach n_t from n_s and, consequently, the higher the efficiency that our thematic view detection approach can provide.

We averaged the values of f_{st}^{BFS} and f_{st}^{DFS} on all the pairs of $PSet$ and we obtained f^{BFS} and f^{DFS}. We performed this task for all the six data lakes introduced in Section 6.1. The obtained

[14] Consider that, since we have 20 real sources in the data lakes adopted in our experimental campaign, the value of H_j can range in the real interval $[0.05, 20]$.

results are reported in Table 7.

$\mathcal{G}_k^{BT}$ (size)	f^{BFS}	f^{DFS}
$\mathcal{G}_1^{BT}$ (208)	0.888	0.867
$\mathcal{G}_2^{BT}$ (356)	0.711	0.691
$\mathcal{G}_3^{BT}$ (572)	0.624	0.603
$\mathcal{G}_4^{BT}$ (928)	0.564	0.542
$\mathcal{G}_5^{BT}$ (1482)	0.556	0.553
$\mathcal{G}_6^{BT}$ (2392)	0.421	0.388

Table 7: Values of f^{BFS} and f^{DFS} against the size of $\mathcal{G}_k^{BT}$.

From the analysis of this table we can observe that our approach can contribute to decrease the number of the nodes of a data lake involved in the visit of n_t starting from n_s and, therefore, to increase the efficiency of this task. Observe that the decrease of the number of involved nodes becomes very high as the data lake size increases.

This result could lead one to conclude that it is appropriate to create a huge number of thematic views in a data lake. Actually, this is not the case. Indeed, the creation and the maintenance of thematic views is expensive and, therefore, it is necessary a right tradeoff between the benefits they cause and the costs they require.

6.5 Effectiveness of semantic enrichment

This experiment aimed at evaluating the effectiveness of the semantic enrichment step discussed in Section 4.3 in terms of its capability of integrating two separate views.

In order to perform the experiments with a large number and variety of views to integrate, we proceeded as follows:

- *Alignment*: starting from the views obtained in the previous step, we firstly aligned each node to the corresponding DBpedia URI.
- *Configuration*: for each view, we randomly selected a subset of nodes (i.e. URIs), according to a parameter (*size_of_view*) ranging from 3 to 10. Nodes to expand are selected randomly because the size of each view may vary. Hence, this parameter helps to keep the dimension under control in order to experimentally evaluate how the size affects effectiveness.
- *Extraction of triples and extension of views*: according to the approach described in Subsection 4.3, we extracted from DBpedia triples related to the randomly selected nodes. The triples are then used to extend the views. This step is executed a number of times (*num_extensions*), ranging from 1 to 2.
- *Verification*: we analyzed the views to verify whether there are paths linking them (i.e., if they have been merged).

For each specific combination of *size_of_view* and *num_extensions*, we repeated the whole procedure 100 times with a different set of URIs for each execution. At each iteration, we computed the following measures:

- percentage of cases in which views were merged;
- average number of nodes of the final merged view (for the cases in which it was obtained);

The obtained average results are shown in Table 8. From the analysis of this table, we can see that the chance to obtain a merged view increases with the size of the views and with the number of extensions, going from 2% to 46%.

We also computed the average length of a path. This parameter does not depend on the size of view. We obtained that it is equal to 1.5 when *num_of_extensions* is equal to 1, and to 2.8 when it is equal to 2.

num_extensions=1	size_of_view			
	3	5	7	10
%	0.02	0.03	0.08	0.14
avg size	120	196	271	382

num_extensions=2	size_of_view			
	3	5	7	10
%	0.06	0.1	0.15	0.46
avg size	790	1144	1810	2493

Table 8: Performance of the semantic enrichment step.

6.6 Computation time

In this experiment, we aimed at evaluating the computation time of our approach. As for view extraction, in Figure 9, we report the execution time against the size |T| of the topic set for the six data lakes that we considered in our experimental campaign.

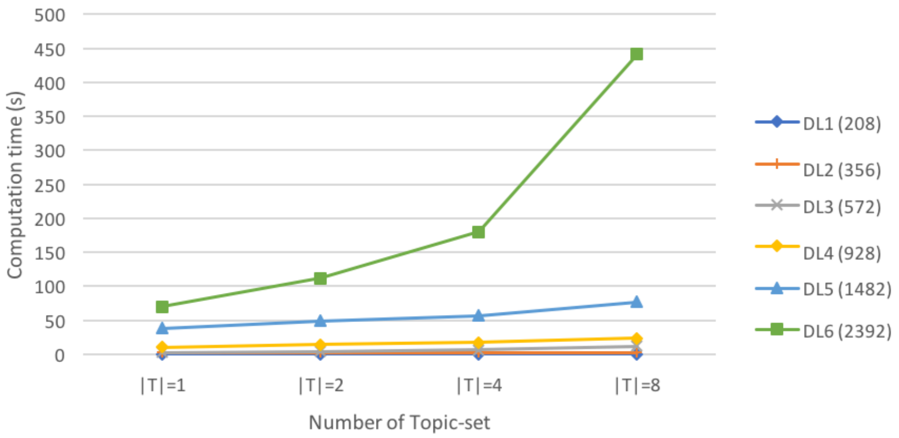


Figure 9: Average computation time of the view extraction task against the size of the data lake and the size of the topic set.

From the analysis of this figure, we can observe that our approach obtains satisfying results. Specifically, the computation time is always very low for data lakes having at most 1482 nodes. Instead, for data lakes with more than 2392 nodes, the computation time is low for $|T| = 1$ or $|T| = 2$. Then, it increases, even if it remains acceptable for $|T| = 4$, whereas it becomes excessive for $|T| = 8$.

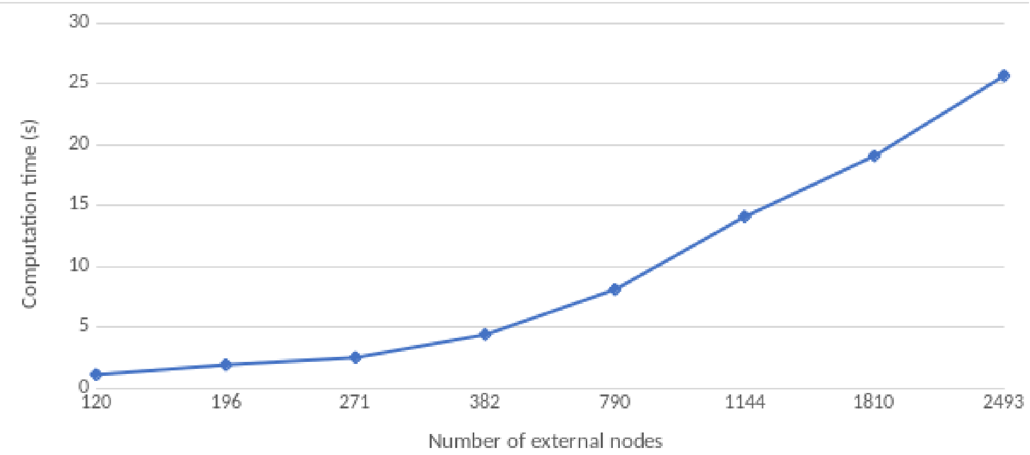


Figure 10: Computation time (in seconds) of the semantic enrichment task against the number of external nodes taken into consideration.

However, with regard to this fact, we must point out that topic sets consisting of 8 keywords are very uncommon[15].

As for the semantic enrichment step, in Figure 10, we report the computation time of a single iteration of the algorithm, against the number of external nodes taken into consideration. As we can see, even in presence of a very large number of nodes (i.e., registering about 2500 different types of object or attribute), the time required for this step is low (it does not exceed 30 seconds) and almost negligible w.r.t. the computation time of the view extraction activity. This time can be further reduced by introducing suitable stop conditions, such as stopping the procedure as soon as a first path has been retrieved.

7 Conclusion

In this paper, we have proposed a new network-based model to uniformly represent the structured, semi-structured and unstructured sources of a data lake. Then, we have illustrated a new approach to "structuring" unstructured sources. Finally, based on these two tools, we have defined a new approach to extracting topic-guided views from the sources of a data lake. This last approach consists of two steps; the former is based on ego networks, whereas the latter leverages semantic relationships.

This paper is not to be intended as an ending point. Instead, we think that it should be the starting point of a new family of approaches aiming at handling information systems in the new big data oriented scenario. By proceeding in this direction, first we plan to define an *unsupervised* approach to extracting topic-guided views from a data lake. This could be extremely useful in presence of a huge number of sources composing the data lake, or if we want to preliminarily construct a set of semantically homogeneous views to "offer" to a user (think, for instance, of a big data analytics scenario).

[15] As a matter of fact, a topic set with 8 keywords would encompass a great number of different concepts and, as such, it would not be generally able to capture a clear and specific desire of a user.

We also plan to define new approaches to: *(i)* supporting a flexible and lightweight querying of the sources of a data lake; *(ii)* extracting complex knowledge patterns; *(iii)* performing schema matching and schema mapping; *(iv)* carrying out data reconciliation and integration. Differently from the already existing approaches, these new generation-approaches should be strongly oriented to data lakes and should be specifically conceived to effectively and efficiently managing unstructured data sources.

References

Abiteboul, S. and Duschka, O. (1998). Complexity of answering queries using materialized views. In *Proc. of the International Symposium on Principles of database systems (SIGMOD/PODS'98)*, pages 254–263, Seattle, WA, USA. ACM.

Aversano, L., Intonti, R., Quattrocchi, C., and Tortorella, M. (2010). Building a virtual view of heterogeneous data source views. In *Proc. of the International Conference on Software and Data Technologies (ICSOFT'10)*, pages 266–275, Athens, Greece. INSTICC Press.

Bachtarzi, C. and Bachtarzi, F. (2015). A model-driven approach for materialized views definition over heterogeneous databases. In *Proc. of the International Conference on New Technologies of Information and Communication (NTIC'15)*, pages 1–5, Mila, Algeria. IEEE.

Bergamaschi, S., Castano, S., Vincini, M., and Beneventano, D. (2001). Semantic integration and query of heterogeneous information sources. *Data & Knowledge Engineering*, 36(3):215–249.

Bidoit, N., Colazzo, D., Malla, N., and Sartiani, C. (2018). Evaluating queries and updates on big xml documents. *Information Systems Frontiers*, 20(1):63–90.

Bilalli, B., Abelló, A., Aluja-Banet, T., and Wrembel, R. (2016). Towards intelligent data analysis: the metadata challenge. In *Proc. of the International Conference on Internet of Things and Big Data (IoTBD'16)*, pages 331–338, Rome, Italy.

Biskup, J. and Embley, D. (2003). Extracting information from heterogeneous information sources using ontologically specified target views. *Information Systems*, 28(3):169–212. Elsevier.

Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022. Microtone Publishing.

Bouadjenek, M. R., Hacid, H., and Bouzeghoub, M. (2016). Social networks and information retrieval, how are they converging? A survey, a taxonomy and an analysis of social information retrieval approaches and platforms. *Information Systems*, 56:1–18.

Bougouin, A., Boudin, F., and Daille, B. (2013). Topicrank: Graph-based topic ranking for keyphrase extraction. In *Proc. of the International Joint Conference on Natural Language Processing (IJCNLP'13)*, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.

Brackenbury, W., Liu, R., Mondal, M., Elmore, A., Ur, B., Chard, K., and Franklin, M. (2018). Draining the Data Swamp: A Similarity-based Approach. In *Proc. of the International Workshop on Human-In-the-Loop Data Analytics (HILDA'18)*, page 13, Houston, Texas, USA. ACM.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., and Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289. Elsevier.

Castano, S. and Antonellis, V. D. (1999). Building views over semistructured data sources. In *Proc. of the International Conference on Conceptual Modeling (ER'99)*, pages 146–160, Paris, France. Springer.

Chen, C., Shyu, M.-L., and Chen, S.-C. (2016). Weighted subspace modeling for semantic concept retrieval using gaussian mixture models. *Information Systems Frontiers*, 18(5):877–889.

Corbellini, A., Mateos, C., Zunino, A., Godoy, D., and Schiaffino, S. (2017). Persisting big-data: The NoSQL landscape. *Information Systems*, 63:1–23. Elsevier.

De Meo, P., Quattrone, G., Terracina, G., and Ursino, D. (2006). Integration of XML Schemas at various "severity" levels. *Information Systems*, 31(6):397–434.

Debattista, J., Lange, C., and Auer, S. (2014). Representing dataset quality metadata using multi-dimensional views. In *Proc. of the International Conference on Semantic Systems (SEM'14)*, pages 92–99, Leipzig, Germany. ACM.

Dessi, A. and Atzori, M. (2016). A machine-learning approach to ranking rdf properties. *Future Generation Computer Systems*, 54:366 – 377.

Dublin Core Metadata Initiative (2012). DCMI Metadata Terms. Technical report.

Fan, W., Wang, X., and Wu, Y. (2016). Answering pattern queries using views. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):326–341. IEEE.

Fang, H. (2015). Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem. In *Proc. of the International Conference on Cyber Technology in Automation (CYBER'15)*, pages 820–824, Shenyang, China. IEEE.

Farid, M., Roatis, A., Ilyas, I., Hoffmann, H., and Chu, X. (2016). CLAMS: bringing quality to Data Lakes. In *Proc. of the International Conference on Management of Data (SIGMOD/PODS'16)*, pages 2089–2092, San Francisco, CA, USA. ACM.

García-Moya, L., Kudama, S., Aramburu, M., and Berlanga, R. (2013). Storing and analysing voice of the market data in the corporate data warehouse. *Information Systems Frontiers*, 15(3):331–349.

Hai, R., Geisler, S., and Quix, C. (2016). Constance: An intelligent data lake system. In *Proc. of the International Conference on Management of Data (SIGMOD 2016)*, pages 2097–2100, San Francisco, CA, USA. ACM.

Hai, R., Quix, C., and Zhou, C. (2018). Query Rewriting for Heterogeneous Data Lakes. In *Proc. of the International Conference on European Conference on Advances in Databases and Information Systems(ADBIS'18)*, pages 35–49, Budapest, Hungary. Springer.

Halevy, A. (2001). Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294. Springer.

Hamadou, H. and Ghozzi, F. (2018). Querying Heterogeneous Document Stores. In *Proc. of the International Conference on Enterprise Information Systems (ICEIS'18)*, pages 58–68, Madeira, Portugal.

Heath, T. and Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136.

Hirschman, A. (1964). The paternity of an index. *The American Economic Review*, 54(5):761–762.

Hitzler, P. and Janowicz, K. (2013). Linked Data, Big Data, and the 4th Paradigm. *Semantic Web*, 4(3):233–235.

Janjua, N., Hussain, F., and Hussain, O. (2013). Semantic information and knowledge integration through argumentative reasoning to support intelligent decision making. *Information Systems Frontiers*, 15(2):167–192.

Keith, A., Cyganiak, R., Hausenblas, M., and Zhao, J. (2011). Describing linked datasets with the void vocabulary. Technical report.

Klettke, M., Awolin, H., Storl, U., Muller, D., and Scherzinger, S. (2017). Uncovering the evolution history of data lakes. In *Proc. of the International Conference on Big Data (IEEE BigData 2017)*, pages 2462–2471, Boston, MA, USA. IEEE.

Kondrak, G. (2005). N-gram similarity and distance. In *String processing and information retrieval*, pages 115–126. Springer.

Konstantinou, N., Koehler, M., Abel, E., Civili, C., Neumayr, B., Sallinger, E., Fernandes, A., Gottlob, G., Keane, J., and Libkin, L. (2017). The VADA architecture for cost-effective data wrangling. In *Proc. of the International Conference on Management of Data (SIGMOD'17)*, pages 1599–1602, Chicago, IL, USA. ACM.

Lassila, O., Swick, R. R., et al. (1998). Resource description framework (rdf) model and syntax specification.

Maccioni, A. and Torlone, R. (2018). KAYAK: A Framework for Just-in-Time Data Preparation in a Data Lake. In *Proc. of the International Conference on Advanced Information Systems Engineering (CAiSE'18)*, pages 474–489, Tallinn, Estonia. Springer.

Madhavan, J., Bernstein, P., and Rahm, E. (2001). Generic schema matching with Cupid. In *Proc. of the International Conference on Very Large Data Bases (VLDB 2001)*, pages 49–58, Rome, Italy. Morgan Kaufmann.

McPherson, M., Smith-Lovin, L., and Cook, J. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444. JSTOR.

Mouttham, A., Kuziemsky, C., Langayan, D., Peyton, L., and Pereira, J. (2012). Interoperable support for collaborative, mobile, and accessible health care. *Information Systems Frontiers*, 14(1):73–85.

Mouzakitis, S., Papaspyros, D., Petychakis, M., Koussouris, S., Zafeiropoulos, A., Fotopoulou, E., Farid, L., Orlandi, F., Attard, J., and Psarras, J. (2017). Challenges and opportunities in renovating public sector information by enabling linked data and analytics. *Information Systems Frontiers*, 19(2):321–336.

M.Tsvetovat and Kouznetsov, A. (2011). *Social Network Analysis for Startups: Finding connections on the social web.* O'Reilly Media, Inc.

Navigli, R. and Ponzetto, S. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250. Elsevier.

Oram, A. (2015). *Managing the Data Lake.* Sebastopol, CA, USA. O'Reilly.

Palopoli, L., Pontieri, L., Terracina, G., and Ursino, D. (2000). Intensional and extensional integration and abstraction of heterogeneous databases. *Data & Knowledge Engineering*, 35(3):201–237.

Palopoli, L., Saccà, D., Terracina, G., and Ursino, D. (2003a). Uniform techniques for deriving similarities of objects and subschemes in heterogeneous databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):271–294.

Palopoli, L., Terracina, G., and Ursino, D. (2001). A graph-based approach for extracting terminological properties of elements of XML documents. In *Proc. of the International Conference on Data Engineering (ICDE 2001)*, pages 330–337, Heidelberg, Germany. IEEE Computer Society.

Palopoli, L., Terracina, G., and Ursino, D. (2003b). DIKE: a system supporting the semi-automatic construction of Cooperative Information Systems from heterogeneous databases. *Software Practice & Experience*, 33(9):847–884.

Palopoli, L., Terracina, G., and Ursino, D. (2003c). Experiences using DIKE, a system for supporting cooperative information system and data warehouse design. *Information Systems*, 28(7):835–865.

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20. Wiley, New York.

Singh, K. and Singh, V. (2016). Answering graph pattern query using incremental views. In *Proc. of the International Conference on Computing (ICCCA'16)*, pages 54–59, Greater Noida, India. IEEE.

Spink, A., Wolfram, D., Jansen, M. B. J., and Saracevic, T. (2001). Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3):226–234.

Wang, J., Li, J., and Yu, J. (2011). Answering tree pattern queries using views: a revisit. In *Proc. of the International Conference on Extending Database Technology (EDBT/ICDT'11)*, pages 153–164, Uppsala, Sweden. ACM.

Wang, J. and Yu, J. (2012). Revisiting answering tree pattern queries using views. *ACM Transactions on Database Systems*, 37(3):18. ACM.

Wu, X., Theodoratos, D., and Wang, W. (2009). Answering XML queries using materialized views revisited. In *Proc. of the International Conference on Information and Knowledge Management (CIKM '09)*, pages 475–484, Hong Kong, China. ACM.

Yi, J., Maghoul, F., and Pedersen, J. (2008). Deciphering mobile search patterns: A study of yahoo! mobile search queries. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 257–266, New York, NY, USA. ACM.

Claudia Diamantini PhD, is Associate Professor at the Department of Information Engineering, Polytechnic University of Marche, where she also holds the role of Vice Dean of the Faculty of Engineering. Her research interests include umbalanced learning, data mining and knowledge discovery, data semantics and knowledge graphs in data mining and advanced analytics settings. On these topics she has worked within national international projects, also with coordination responsibilities. She is author of more than 150 publications in peer-reviewed journals, books and conferences, and is involved in the organization of conferences and workshops.

Paolo Lo Giudice received the MSc Degree in ICT Engineering from the University Mediterranea of Reggio Calabria in October 2016. He is currently a PhD Student in ICT Engineering at the same University. His research interests include Social Network Analysis, Social Internetworking, Source and Data Integration, Ecosystems consisting of Internet of Things, Innovation Management, Knowledge Extraction and Representation, Biomedical Applications, Data Lakes. He is an author of 12 papers.

Domenico Potena is Associate Professor at the Department of Information Engineering, Polytechnic University of Marche. He received the MSc degree in Electronic Engineering from the University of Ancona, Italy, in 2001, and the Ph.D. in Information Systems Engineering from the Polytechnic University of Marche, Italy, in 2004. From June 2005 to October 2008, he was post-doctoral fellow and then from November 2008 to June 2019 researcher at the Department of Computer Science, Management and Automation Engineering of Polytechnic University of Marche. His research interests include knowledge discovery in databases, data mining, big data, process mining, data warehousing and information systems. He authored more than 130 papers.

Emanuele Storti received the Ph.D. degree in Computer Engineering from the Polytechnic University of Marche in 2012 and is currently working as a post-doctoral fellow and an adjunct professor at the Department of Information Engineering. His research interests include Semantic Technologies, Knowledge Management, Open Data, Business Intelligence. As a Eurodoc officer, he is collaborating with several EOSC (European Open Science Cloud) projects.

Domenico Ursino received the MSc Degree in Computer Engineering from the University of Calabria in July 1995. He received the PhD in System Engineering and Computer Science from the University of Calabria in January 2000. From January 2005 to December 2017, he was an Associate Professor at the University Mediterranea of Reggio Calabria. From January 2018, he is a Full Professor at the Polytechnic University of Marche. His research interests include Source and Data Integration, Data Lakes, Social Network Analysis, Social Internetworking, Ecosystems consisting of Internet of Things, Innovation Management, Knowledge Extraction and Representation, Biomedical Applications, Recommender Systems. In these research fields, he published more than 190 papers