An approach to extracting thematic views from highly heterogeneous sources of a data lake

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

[1] DII, Polytechnic University of Marche
[2] DIIES, University "Mediterranea" of Reggio Calabria

Abstract. In the last years, data lakes are emerging as an effective and 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. One of the main issues to address in this scenario consists in the extraction of thematic views from the (very heterogeneous and generally unstructured) data 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, "structure" unstructured data. Finally, we define a technique to extract thematic views from the sources of a data lake, based on similarity and other semantic relations among the metadata of data sources.

1 Introduction

In the last years, data lakes are emerging 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 [13]. 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. One of the main issues to address in a scenario comprising many data sources extremely heterogeneous in their format, structure and semantics, consists in the extraction of thematic views from data sources [3], i.e., the construction of views concerning one or more topics of interest for the user, obtained by extracting and merging data coming from different sources. This problem has been largely investigated in the past for structured and semi-structured data sources stored in a data warehouse [26, 8, 22], and this witnesses its extreme relevance. However, it is esteemed that, currently, more than 80% of data sources are unstructured

SEBD 2018, June 24-27, 2018, Castellaneta Marina, Italy.

[9]. 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. Indeed, networks are very flexible structures that allow the modeling of almost all phenomena that researchers aim at investigating [7]. 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 of topics $T = \{T_1, T_2, \ldots, T_n\}$ 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 notion of ego network, which actually represents the core of the proposed approach. The latter 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.

This paper is organized as follows: Section 2 illustrates related literature. In Section 3, we present the proposed approach. In particular, first we describe a unifying model for data lake representation; then, we present our approach to partially structuring unstructured sources; finally, we discuss the two steps of our approach for thematic view extraction. In Section 4, we present our example case, whereas, in Section 5, we draw our conclusions and discuss future work.

2 Related Literature

The new data lake scenario is characterized by several peculiarities that make it very different from the data warehouse paradigm. Hence, it is necessary to adapt (if possible) old algorithms conceived for data warehouses or to define new approaches capable of handling and taking advantage of the specificities of this new paradigm. However, most approaches proposed in the literature for data integration, query answering and view extraction do not completely fit the data lake paradigm. For instance, [8] proposes some techniques for building views on semi-structured data sources based on some expected queries. Other researchers focus on materialized views and, specifically, on throughput and execution time; therefore, they a-priori define a set of well-known views and, then, materialize them. Two surveys on this issue can be found in [16, 1]. The authors of [26] investigate the same problem but they focus on XML sources. The approach of [25] addresses the same issue by means of query rewriting; specifically, it transforms a query Q into a set of new queries, evaluates them, and, then, merges the corresponding answers to construct the materialized answer to Q. [4] proposes an approach to constructing materialized views for heterogeneous

DBpedia: `http://dbpedia.org`

databases; it requires the presence of a static context and the pre-computation of some queries.

Another family of approaches exploits materialized views to perform tree pattern querying [24] and graph pattern queries [12]. Unfortunately, all these approaches are well-suited for structured and semi-structured data, whereas they are not scalable and lightweight enough to be used in a dynamic context or with unstructured data. An interesting advance in this area can be found in [23]. Here, the authors propose an incremental approach to address the graph pattern query problem on both static and dynamic real-life data graphs. Other kinds of views are investigated in [6] and [3]. In particular, this last paper uses virtual views to access heterogeneous data sources without knowing many details of them. For this purpose, it creates virtual views of the data sources.

Finally, semantic-based approaches have long been used to drive data integration in databases and data warehouses. More recently, in the context of big data, 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 [17]. In the data lake scenario, semantic techniques have been successfully applied to more efficiently integrate and handle both structured and unstructured data sources by aligning data silos and better managing evolving data models. For instance, in [15], the authors discuss a data lake system with a semantic metadata matching component for ontology modeling, attribute annotation, record linkage, and semantic enrichment. Furthermore, [14] presents a system to discover and enforce expressive integrity constraints from data lakes. Similarly to what happens in our approach, knowledge graphs in RDF are used to drive integration. To reach their objectives, these techniques usually rely on information extraction tools (e.g., Open Calais) that may assist in linking metadata to uniform vocabularies (e.g., ontologies or knowledge repositories, such as DBpedia).

3 Description of the proposed approach

3.1 A unifying model for data lake representation

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 [21], 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: *(i)* $\mathcal{M}_k^T$ denotes *technical metadata*. It represents the type, the format, the structure and the schema of the corresponding data. It

RDF Concepts and abstract Syntax: `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`

`http://www.opencalais.com`

is commonly provided by the source catalogue. *(ii)* $\mathcal{M}_k^O$ represents *operational metadata*. It includes the source and target locations of the corresponding data, the associated file size, the number of their records, and so on. Usually, it is automatically generated by the technical framework handling the data lake. *(iii)* $\mathcal{M}_k^B$ indicates *business metadata*. It comprises the business names and descriptions assigned to data fields. It also covers business rules, which can become integrity constraints for the corresponding data source.

In this paper, we consider only business metadata. Indeed, they denote, at the intensional level, the information content stored in the data lake sources and are those of interest for supporting the extraction of thematic views from a data lake, which is our ultimate goal. In order to represent $\mathcal{M}_k^B$, our model adopts a notation typical of XML, JSON and many other semi-structured models. 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: *(i)* Att_k indicates the set of the attributes of $\mathcal{M}_k^B$; *(ii)* Smp_k represents the set of the simple elements of $\mathcal{M}_k^B$; *(iii)* Cmp_k denotes the set of the complex elements of $\mathcal{M}_k^B$. In this context, 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 one-to-one 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. Finally, let y be a simple element of $\mathcal{M}_k^B$. Att_{k_y} represents the set of the attributes of y, whereas $N_{k_y}^{Att}$ denotes the set of the corresponding nodes.

A_k indicates the set of the arcs of $\mathcal{M}_k^B$. It consists of two subsets: $A_k = A'_k \cup A''_k$. Here: *(i)* $A'_k = \{(n_x, n_y) | n_x \in N_k^{Cmp}, n_y \in N_{n_x}^{Obj}\}$, i.e., there is an arc from a complex element of $\mathcal{M}_k^B$ to each object directly contained in it. *(ii)* $A''_k = \{(n_x, n_y) | n_x \in N_k^{Smp}, n_y \in N_{n_x}^{Att}\}$, i.e., there is an arc from a simple element of $\mathcal{M}_k^B$ to each of its attributes.

3.2 An approach to partially structuring unstructured sources

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 consists of four phases, namely: *(i)* creation of nodes; *(ii)* derivation and management of part-of relationships; *(iii)* derivation of lexical and string similarities; *(iv)* management of lexical and string similarities.

Phase 1. During this phase, our approach creates a complex element 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, the next phases are necessary.

Phase 2. During this phase, our approach adds 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, if k_2 is registered as a lemma of k_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 [20]. When this arc has been added, n_{k_1} must be considered a complex element, instead of a simple one.

Phase 3. During this phase, our approach starts by deriving lexical similarities. In particular, it states that there exists a similarity between the nodes n_{k_1}, corresponding to the keyword k_1, and n_{k_2}, corresponding to the keyword k_2, if k_1 and k_2 have at least one common lemma in a suitable thesaurus. Also in this case, we have adopted BabelNet. After having found lexical similarities, our approach derives string similarities and states that there exists a similarity between n_{k_1} and n_{k_2} 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 [18] 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 $KeySim$ 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 it concludes that there exists a similarity between n_{k_1} and n_{k_2}. We have experimentally set $th_k = 0.70$ and $th_{kmin} = 0.50$. At the end of this phase, our approach has found some (lexical and/or string) similarities, each stating that a node n_{k_i} is similar to a node n_{k_j}.

Phase 4. During this phase, our approach aims at managing the similarities found during Phase 3. In particular, if there exists a (lexical and/or string) sim-

In this paper, we use the term "lemma" according to the meaning it has in BabelNet [20]. Here, given a term, its lemmas are other objects (terms, emoticons, etc.) contributing to specify its meaning.

ilarity between two nodes n_{k_i} and n_{k_j}, it merges them into one node $n_{k_{ij}}$, which inherits all the incoming and outgoing arcs of n_{k_i} and n_{k_j}. After all similarities have been considered, it could happen that there exist two or more arcs from a node n_{k_i} to a node n_{k_j}. In this case, our approach merges them into one arc.

3.3 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 is called "supervised" because it requires the user to specify the set of topics $T = \{T_1, T_2, \cdots, T_l\}$, which should be present in the thematic view(s) to extract. It consists of two steps, the former mainly based on the structure of the sources at hand, the latter mainly focusing on the corresponding semantics. We describe these two steps in the next subsections.

Step 1. The first step of our approach receives a data lake DL, a set of topics $T = \{T_1, T_2, \cdots, T_l\}$, representing the themes of interest for the user, and a dictionary Syn of synonymies involving the objects stored in the sources of DL. This dictionary could be a generic thesaurus, such as BabelNet [20], 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 [10], MOMIS [5] or Cupid [19]).

In this step, the concept of ego network [2, 11] plays a key role. We recall that an ego network consists of a focal node (the ego) and the nodes it is directly connected to (the "alters"), plus the ties, if any, between the alters.

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 synonymous of T_i in DL. Let $N_i = \{n_{i_1}, n_{i_2}, \cdots, n_{i_q}\}$ be the corresponding nodes. First, Step 1 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. Then, it merges all the egos into a unique node n_i. In this way, it obtains a unique ego network E_i from $E_{i_1}, E_{i_2}, \cdots, E_{i_q}$. If a synonymy exists 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. At the end of this task, we have a unique ego network E_i corresponding to T_i.

After having performed the previous task for each topic of T, we have a set $E = \{E_1, E_2, \cdots, E_l\}$ of l ego networks. At this point, Step 1 finds all the synonymies of Syn involving objects of the ego networks of E and merges the corresponding nodes. After all the possible synonymies involving objects of the ego network of E have been considered and the corresponding nodes have been merged, a set $V = \{V_1, \cdots, V_g\}$, $1 \leq g \leq l$, of networks representing potential views is obtained.

If $g = 1$, then it is possible to conclude that Step 1 has been capable of extracting a unique thematic view comprising all the topics required by the user. Otherwise, there exist more views each comprising some (but not all) of the topics of interest for the user. If $g = 1$, Step 2 is performed to make more

precise and complete the unique view representing all the topics of T. If $g > 1$, Step 2 aims at finding other relationships, different from synonymies, among the objects of the views of V in such a way as to try to obtain a unique view embracing all the topics of interest for the user.

Step 2. This step starts by enriching each view $V_i \in V$. For this purpose, it connects each of its elements to all the semantically related concepts taken from a reference knowledge repository.

In this work, we rely on DBpedia, one of the largest knowledge graphs in the Linked Data context, including more than 4.58 million entities in RDF. To this aim, first each element of V_i (including its synonyms) is mapped to the corresponding entry in DBpedia. In many cases, such a mapping is already provided by BabelNet. Then, for each DBpedia entry, all the related concepts are retrieved. In DBpedia, knowledge is structured according to the Linked Data principles, 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. Both subjects and properties are resources (i.e., nodes in DBpedia's knowledge graph), whereas objects may be either resources or literals (i.e., values of some primitive data types, such as strings or numbers). Each triple represents the minimal component of the knowledge graph. This last is built by merging triples together. Therefore, retrieving the related concepts for a given element x implies finding all the triples where x is either the subject or the object.

For each view $V_i \in V$, the procedure to extend it consists of the following three substeps:

1. *Mapping*: for each node $n \in V_i$, its corresponding DBpedia entry d is found.
2. *Triple extraction*: all the related triples $\langle d, p, o\rangle$ and $\langle s, p, d\rangle$, i.e., all the triples in which d is either the subject or the object, are retrieved.
3. *View extension*: for each retrieved triple $\langle d, p, o\rangle$ (resp., $\langle s, p, d\rangle$), V_i is extended by defining a node for the object o (resp., s), if not already existing, linked to n through an arc labeled as p.

These three tasks are repeated for all the views of V. As previously pointed out, this enrichment procedure is particularly important if $|V| > 1$ because the new derived relationships could help to merge the thematic views that was not possible to merge during Step 1. In particular, let $V_i \in V$ and $V_j \in V$ be two views of V, and let V_i' and V_j' be the extended views corresponding to them. If there exist two nodes $n_{i_h} \in V_i'$ ad $n_{j_k} \in V_j'$ such that $n_{i_h} = n_{j_k}$, then they can be merged in one node; if this happens, V_i' and V_j' 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 views comprising

Whenever this does not happen, the mapping can be automatically provided by the DBpedia Lookup Service (`http://wiki.dbpedia.org/projects/dbpedia-lookup`). Here, two nodes are equal if the corresponding name coincide.

all the topics specified by the user can be found. In this last case, our approach still returns the enriched views of V and leaves the user the choice to accept of reject them.

4 An example case

In this section, we present an example case aiming at illustrating the various tasks of our approach. Here, we consider: *(i)* a structured source, called *Weather Conditions* (W, in short), whose corresponding E/R schema is not reported for space limitations; *(ii)* two semi-structured sources, called *Climate* (C, in short) and *Environment* (E, in short), whose corresponding XML Schemas are not reported for space limitations; *(iii)* an unstructured source, called *Environment Video* (V, in short), consisting of a YouTube video and whose corresponding keywords 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 approaches mentioned in Section 3.2, we obtain the corresponding representations in our network-based model, shown in Figure 1.

Assume, now, that a user specifies the following set T of topics of her interest: $T = \{Ocean, Area\}$. First, our approach determines the terms (and, then, the objects) in the five 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 ($V.Ocean$) 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 $Obj_2 = \{W.Place, C.Place, V.Region, E.Location\}$.

Step 1 of our approach proceeds by constructing the ego networks corresponding to the objects of Obj_1 and Obj_2. They are reported in Figure 2.

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 in Figure 3(a). The same task is performed to 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 $C.Altitude$ 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 $C.Altitude$ in another arc. In this way, the ego network shown in Figure 3(b) 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.

Here and in the following, we use the notation $S.o$ to indicate the object o of the source S.

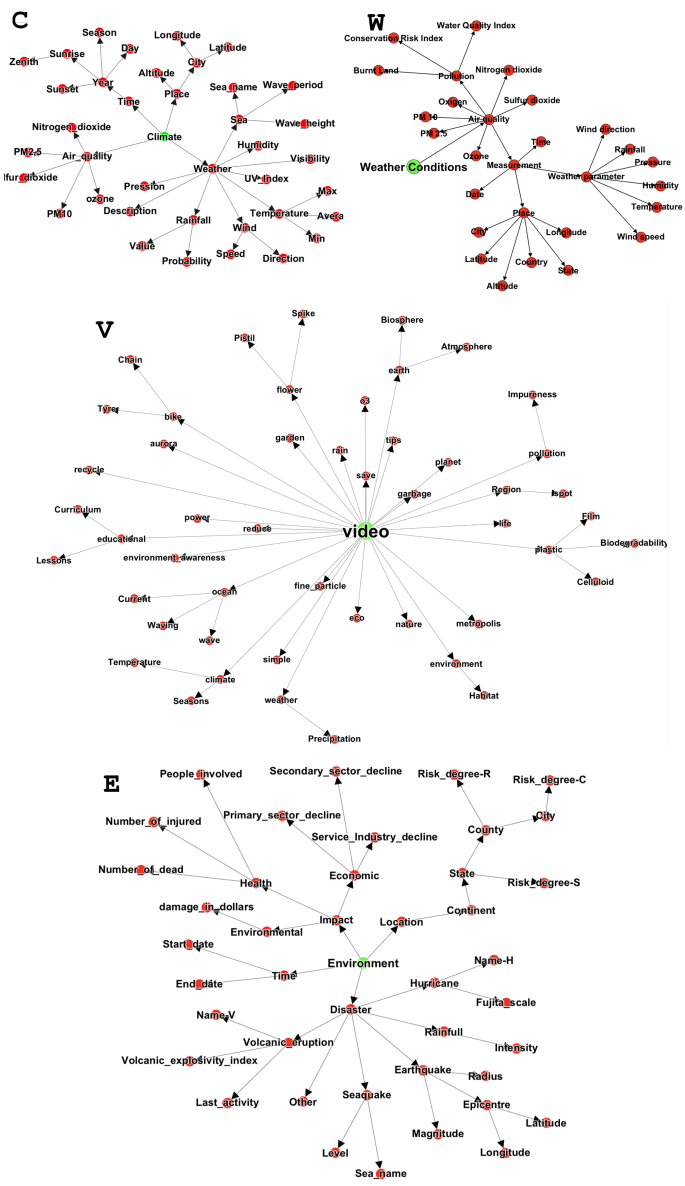

Fig. 1. Network-based representations of the four sources into consideration.

Now, Step 1 verifies if there are any synonyms between a node of E_{Ocean} and a node of E_{Area}. Since this does not happen, it terminates and returns the set $V = \{V_{Ocean}, V_{Area}\}$, where V_{Ocean} (resp., V_{Area}) coincides with E_{Ocean} (resp., E_{Area}).

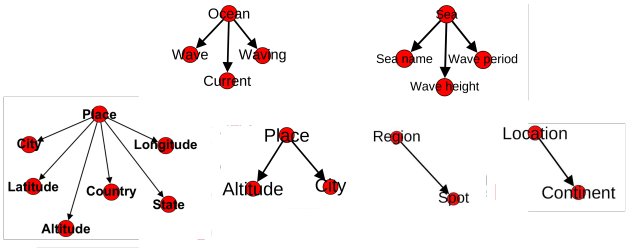

Fig. 2. Ego networks corresponding to *V.Ocean*, *C.Sea*, *W.Place*, *C.Place*, *V.Region* and *E.Location*.

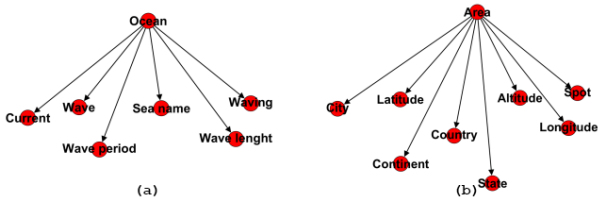

Fig. 3. Ego networks corresponding to *Ocean* and *Area*.

At this point, Step 2 is executed. As shown in Figure 4, first each term (synonyms included) 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*, while *Country* to *dbo:Country*, respectively). After a single iteration, the triples ⟨*dbo:sea rdfs:range dbo:Sea*⟩ and ⟨*dbo:sea rdfs:domain dbo:Place*⟩ are retrieved. They correspond to the DBpedia property *sea*, which relates a country to the sea from which it is lapped. Other connections can be found by moving to specific instances of the mentioned resources. In this way, the following triples are retrieved: ⟨*instance rdf:type dbo:Sea*⟩, ⟨*instance rdf:type dbo:Location*⟩, ⟨*instance rdf:type dbo:Place*⟩, meaning that there are resource instances having types *Sea*, *Location* and *Place* simultaneously (e.g., *dbr:Mediterranean_Sea*). Furthermore, a triple ⟨*instance dbo:country dbo:Country*⟩ can be retrieved, meaning that those instances being a *Sea*, a *Location* or a *Place* specify in which *dbo:Country* they are located through the *dbo:country* property. In this example case, Step 2 succeeded in merging the two views that Step 1 had maintained separated.

5 Conclusion

In this paper, we have presented a new network-based model to uniformly represent the structured, semi-structured and unstructured sources of a data lake. Then, we have proposed a new approach to, at least partially, "structuring"

Prefixes *dbo* and *dbr* stand for `http://dbpedia.org/ontology/` and `http://dbpedia.org/resource/`

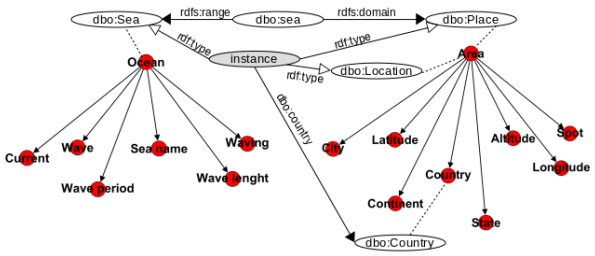

Fig. 4. The integrated thematic view.

unstructured data. Finally, based on these two tools, we have defined a new approach to extracting thematic views from the sources of a data lake consisting of two steps, based on ego networks (Step 1) and semantic relationships (Step 2). This paper is not to be intended as an ending point. Actually, it could 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 thematic views from a data lake. Then, we plan to define new approaches to supporting a flexible and lightweight querying of the sources of a data lake, as well as approaches to schema matching, schema mapping, data reconciliation and integration strongly oriented to data lakes based mainly on unstructured data sources.

References

1. S. Abiteboul and O.M. Duschka. 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, 1998. ACM.
2. V. Arnaboldi, M. Conti, A. Passarella, and F. Pezzoni. Analysis of ego network structure in online social networks. In *Proc. of the International Conference on Privacy (PASSAT'12)*, pages 31–40, Amsterdam, Netherlands, 2012. IEEE.
3. L. Aversano, R. Intonti, C. Quattrocchi, and M. Tortorella. 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, 2010. INSTICC Press.
4. C. Bachtarzi and F. Bachtarzi. 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, 2015. IEEE.
5. S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. Semantic integration and query of heterogeneous information sources. *Data & Knowledge Engineering*, 36(3):215–249, 2001.
6. J. Biskup and D. Embley. Extracting information from heterogeneous information sources using ontologically specified target views. *Information Systems*, 28(3):169–212, 2003. Elsevier.
7. M. R. Bouadjenek, H. Hacid, and M. Bouzeghoub. 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, 2016.
8. S. Castano and V. De Antonellis. Building views over semistructured data sources. In *Proc. of the International Conference on Conceptual Modeling (ER'99)*, pages 146–160, Paris, France, 1999. Springer.
9. A. Corbellini, C. Mateos, A. Zunino, D. Godoy, and S.N. Schiaffino. Persisting big-data: The NoSQL landscape. *Information Systems*, 63:1–23, 2017. Elsevier.
10. P. De Meo, G. Quattrone, G. Terracina, and D. Ursino. Integration of XML Schemas at various "severity" levels. *Information Systems*, 31(6):397–434, 2006.
11. R. DeJordy and D. Halgin. Introduction to ego network analysis. *Boston MA: Boston College and the Winston Center for Leadership & Ethics*, 2008.
12. W. Fan, X. Wang, and Y. Wu. Answering pattern queries using views. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):326–341, 2016. IEEE.
13. H. Fang. 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, 2015. IEEE.
14. M. Farid, A. Roatis, I.F. Ilyas, H. Hoffmann, and X. Chu. 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, 2016. ACM.
15. R. Hai, S. Geisler, and C. Quix. Constance: An intelligent data lake system. In *Proc. of the International Conference on Management of Data (SIGMOD/PODS'16)*, pages 2097–2100, San Francisco, CA, USA, 2016. ACM.
16. A. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, 2001. Springer.
17. P. Hitzler and K. Janowicz. Linked Data, Big Data, and the 4th Paradigm. *Semantic Web*, 4(3):233–235, 2013.
18. G. Kondrak. N-gram similarity and distance. In *String processing and information retrieval*, pages 115–126, 2005. Springer.
19. J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In *Proc. of the International Conference on Very Large Data Bases (VLDB 2001)*, pages 49–58, Rome, Italy, 2001. Morgan Kaufmann.
20. R. Navigli and S.P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012. Elsevier.
21. A. Oram. *Managing the Data Lake*. Sebastopol, CA, USA, 2015. O'Reilly.
22. L. Palopoli, L. Pontieri, G. Terracina, and D. Ursino. Intensional and extensional integration and abstraction of heterogeneous databases. *Data & Knowledge Engineering*, 35(3):201–237, 2000.
23. K. Singh and V. Singh. Answering graph pattern query using incremental views. In *Proc. of the International Conference on Computing (ICCCA'16)*, pages 54–59, Greater Noida, India, 2016. IEEE.
24. J. Wang, J. Li, and J.X. Yu. 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, 2011. ACM.
25. J. Wang and J.X. Yu. Revisiting answering tree pattern queries using views. *ACM Transactions on Database Systems*, 37(3):18, 2012. ACM.
26. X. Wu, D. Theodoratos, and W.H. Wang. 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, 2009. ACM.