A New Metadata Model to Uniformly Handle Heterogeneous Data Lake Sources

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, Ancona, Italy
d.ursino@univpm.it
[2] DIIES, University "Mediterranea" of Reggio Calabria, Reggio Calabria, Italy

Abstract. Metadata have always played a key role in favoring the cooperation of heterogeneous data sources. This role has become much more crucial with the advent of data lakes, in which case metadata represent the only possibility to guarantee an effective and efficient management of data source interoperability. For this reason, the necessity to define new models and paradigms for metadata representation and management appears crucial in the data lake scenario. In this paper, we aim at addressing this issue by proposing a new metadata model well suited for data lakes. Furthermore, to give an idea of its capabilities, we present an approach that leverages it to "structure" unstructured sources and to extract thematic views from heterogeneous data lake sources.

1 Introduction

Metadata have always played a key role in favoring the cooperation of heterogeneous data sources [3,6,19,20]. This role was already relevant in the past architectures (e.g., Cooperative Information Systems and Data Warehouses) but has become much more crucial with the advent of data lakes [8]. Indeed, in this new architecture, metadata represent the only possibility 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 [18]). 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.

In this paper, we aim at providing a contribution in this setting and we propose a new metadata model well suited for data lakes. 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 [18]). However, it complements them with new ideas and, in particular, with the power guaranteed by a network-based and semantics-driven representation of metadata. Thanks to this choice, our model can benefit from all the results already found

A. Benczúr et al. (Eds.): ADBIS 2018, CCIS 909, pp. 165–177, 2018.
https://doi.org/10.1007/978-3-030-00063-9_17

in network theory and semantics-driven approaches. As a consequence, it can allow a large variety of sophisticated tasks that the metadata models currently adopted do not guarantee. For instance, it allows the definition of a structure for unstructured data, which currently represent more than 80% of available data sources. Furthermore, it allows the extraction of thematic views from data sources [2], 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, and this witnesses its extreme relevance. These are only two of the tasks that can benefit from our model and, in this paper, we illustrate them. Actually, many other ones could be thought and investigated, and they will represent the subject of our future research efforts.

This paper is structured as follows: Sect. 2 illustrates related literature. In Sect. 3, we propose our metadata model. Section 4 presents the application of this model to the problems of structuring unstructured data and of extracting thematic views from heterogeneous data lake sources. In Sect. 5, we present our example case, whereas, in Sect. 6, we draw our conclusions and discuss future work.

2 Related Literature

In the literature, several metadata classifications have been proposed in the past. For instance, the authors of [4] propose a tree-based classification. They split metadata into several categories, propose a conceptual schema of the metadata repository and use RDF for metadata modeling. 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 in proposed in [18]. This is also the model adopted by Zaloni. It divides metadata based on their generation time or on the meaning and information they bring. In this latter case, metadata can be divided 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, it assumes 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 tasks, but, at the same time, not excessively complex in such a way as to prevent a slow processing.

Several metadata models and frameworks are widely adopted by the Linked Data community (e.g., DCMI Metadata Terms and VoID). DCMI Metadata Terms [13] 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) [14]

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*, describing the structure and the schema of datasets, mostly used for supporting querying and data integration.

As for the applications of our metadata model proposed in this paper (i.e., structuring of unstructured data and thematic view extraction), most approaches proposed in the literature to carry out this task do not completely fit the data lake paradigm. Two surveys on this issue can be found in [1,11].

Another family of approaches leverages materialized views to perform tree pattern querying [22] and graph pattern queries [7]. 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. Interesting advances in this area can be found in [2,5,21].

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 [12]. 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 model (see, for instance, [9,10]). 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 tools assisting users in linking metadata to uniform vocabularies (e.g., ontologies or knowledge repositories, such as DBpedia).

3 A Unifying Model for Representing the Metadata of Data Lake Sources

In this section, 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: $\mathcal{M}_{DL} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m\}$.

3.1 Typologies of Metadata

Following what it is said in [18], metadata can be divided into three categories, namely: *(i) Business metadata*, which include business rules (e.g., the upper and lower limit of a particular field, integrity constraints, etc.); *(ii) Operational metadata*, which include information generated automatically during data processing (e.g., data quality, data provenance, executed jobs); *(iii) Technical metadata*,

which include information about data format and schema. Based on this reasoning, $\mathcal{M}_k$ can be represented as the union of three sets $\mathcal{M}_k^B \cup \mathcal{M}_k^O \cup \mathcal{M}_k^T$.

As an advancement of the model of [18], we observe that these three subsets are intersected with each other (as shown in Fig. 1). For instance, since business metadata contain all business rules and information allowing to better understand 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).

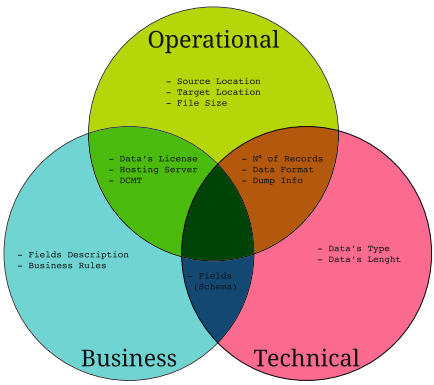

Fig. 1. The three kinds of metadata proposed by our model.

In this paper, we focus on business metadata and on the intersection between them and the technical ones. This intersection 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 one, 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 adopted to give an idea of the source content.

3.2 A Network-Based Model for Business and Technical Metadata

As already mentioned, in this paper we focus especially on the business and technical metadata and on their intersection. 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 in this paper.

We indicate by $\mathcal{M}_k^{BT}$ the intersection between $\mathcal{M}_k^B$ and $\mathcal{M}_k^T$. We denote by Obj_k the set of all the objects stored in $\mathcal{M}_k^{BT}$. The concept of "object" depends on 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 $G_k^{BT} = \langle N_k, A_k \rangle$. For each object $o_{k_j} \in Obj_k$ 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 Obj_k, in the following, we will use the two terms interchangeably.

On the other hand, each $a_{k_i} = \langle (n_s, n_t), l_{k_i} \rangle \in A_k$ is an arc; here, n_s is the source node, n_t is the target one, whereas l_{k_i} is a label representing the kind of relationhip between n_s and n_t. Some possible relationships are: *(i) Structural relationship*: it is represented by the label "contains" and is used to represent the relationhip between a relational table and its attributes, a complex object and its simple ones, or between a simple object and its attributes. *(ii) Similarity relationship*: it is represented by the label "similarTo" and denotes a form of similarity between two objects. We will see an example of its semantics and usage in Sect. 4.1. *(iii) Lemma relationship*: it is represented by the label "lemma" and denotes that the target node is a lemma of the source one. Again, its usage will be clear in Sect. 4.1.

Our model enables a scalable and flexible approach in the representation and management of 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[1]). In the following, we refer to an extension of G_k^{BT} as G_k^{Ext}. It consists of $G_k^{Ext} = G_k^{BT} \cup G^E$, where G^E is an external knowledge graph. An arc from a node of G_k^{BT} and its corresponding node in G^E will be labeled as "externalSource_X", where X is the name of the external knowledge graph at hand.

4 Examples of Applications of Our Metadata Model

As pointed out in the Introduction, in order to give an idea of the expressiveness and the power of our data model, in this section, we will exploit it in two application tasks, namely "structuring" unstructured data sources and extracting thematic views from heterogeneous data lake sources.

4.1 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

[1] http://wiki.dbpedia.org.

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.

During the first phase, it creates a node representing the source as a whole and a node for each keyword. Then, it links the former to the latter through arcs with label "contains". During the second phase, it adds an arc with label "lemma" 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[2] of k_1 in a suitable thesaurus (we adopted BabelNet [17] for this purpose). During the third phase, our approach derives 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, it 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 metric on k_1 and k_2, is higher than a suitable threshold th_k. After several experiments, we have chosen N-Grams [15] as string similarity metric. In both these cases, if there exist a similarity between n_{k_1} and n_{k_2}, our approach adds an arc with label "similarTo" from n_{k_1} to n_{k_2}, and vice versa. During the fourth phase, if there exists a pair of arcs with label "similarTo" between two nodes n_{k_i} and n_{k_j}, our approach merges them into one node $n_{k_{ij}}$, which inherits all the incoming and outgoing edges of n_{k_i} and n_{k_j}. Finally, if there exist two or more arcs from a node n_{k_i} to a node n_{k_j} with the same label, our approach merges them into one node[3].

4.2 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 Sect. 3. 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.

Step 1 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 [17], a domain-specific thesaurus, or a dictionary obtained by taking into account the structure

[2] In this paper, we use the term "lemma" according to the meaning it has in BabelNet [17]. Here, given a term, its lemmas are other objects (terms, emoticons, etc.) contributing to specify its meaning.

[3] Please note that Phases 3 and 4 could be merged in a unique one, avoiding to define arcs with label "similarTo". Here, we maintain these arcs and both phases to keep the information about similarity between nodes for future use.

and the semantics of the sources, which the corresponding objects refer to (such as the dictionaries produced by XIKE [6], MOMIS [3] or Cupid [16]). 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, 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. 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 there exists 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.

Step 2 starts by constructing the graph G_k^{Ext} obtained by extending G_k^{BT} with an external knowledge graph G^E (in this work, we rely on DBpedia). For this purpose, first it links each node n_{i_j} of V_i to the corresponding entry $n_{e_{ij}} \in G^E$ through an arc with label "externalSource_DBpedia". In our scenario, such a DBpedia node $n_{e_{ij}}$ is already specified in the BabelNet entry corresponding to n_{i_j} (or to any of its synonyms in Syn)[4]. Then, for each $n_{e_{ij}}$ considered above, 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. 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 $n_{i_j} \in V_i$, its corresponding DBpedia entry $n_{e_{ij}}$ is found. *(2) Triple extraction*: all the related triples $\langle n_{e_{ij}}, p, o\rangle$ and $\langle s, p, n_{e_{ij}}\rangle$, i.e., all the triples in which $n_{e_{ij}}$ is either the subject or the object, are retrieved. *(3) View extension*: for each retrieved triple $\langle n_{e_{ij}}, p, o\rangle$ (resp., $\langle s, p, n_{e_{ij}}\rangle$), V_i is extended by defining a node for the object o (resp., s), if not already existing, linked to n_{i_j} through an edge labeled as p. Substeps 2 and 3 are recursively repeated for each new added node. The procedure stops after a given number of iterations, limiting the length of external incoming and outcoming paths of nodes in V_i. The longer the path, the weaker the semantic link between nodes.

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

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 was not possible to merge during the Step 1. In particular, let $V_i \in V$ and $V_l \in V$ be two views of V, and let V_i' and V_l' be the extended views corresponding to them. If there exist two nodes $n_{i_h} \in V_i'$ ad $n_{l_k} \in V_l'$ such that $n_{i_h} = n_{l_k}$[5], then they can be merged in one node; in this way, V_i' and V_l' 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 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.

5 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 approach mentioned in Sect. 4.2, we obtain the corresponding representations in our network-based model, shown in Fig. 2[6].

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 ($V.Ocean$[7]) 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 Fig. 3[8].

[5] Here, two nodes are equal if the corresponding name coincide.

[6] In this figure, 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.

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

[8] In this figure, for layout reasons, we do not show the arc labels because they are the same as the corresponding arcs of Fig. 2.

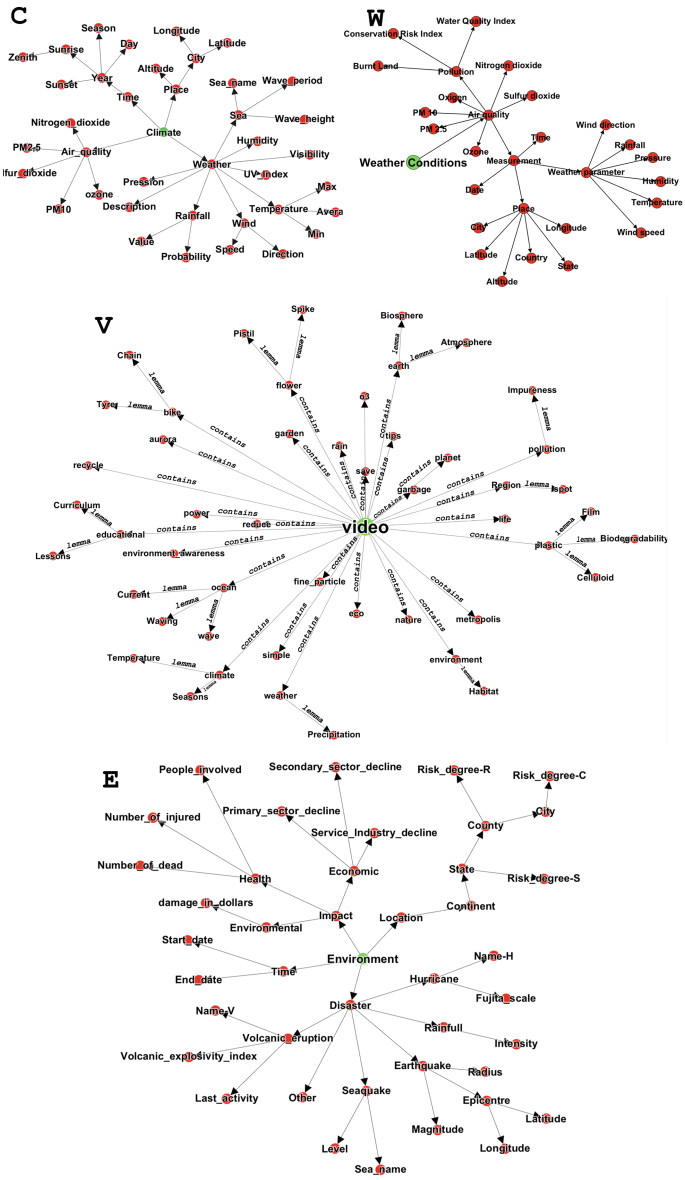

Fig. 2. Network-based representations of the four sources into consideration.

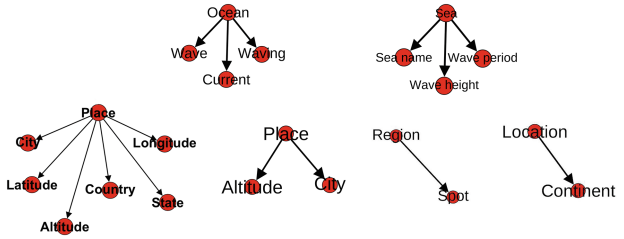

Fig. 3. Ego networks corresponding to *V.Ocean*, *C.Sea*, *W.Place*, *C.Place*, *V.Region* and *E.Location*.

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 Fig. 4(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 Fig. 4(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.

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 returns the set $V = \{V_{Ocean}, V_{Area}\}$, where V_{Ocean} (resp., V_{Area}) coincides with E_{Ocean} (resp., E_{Area}).

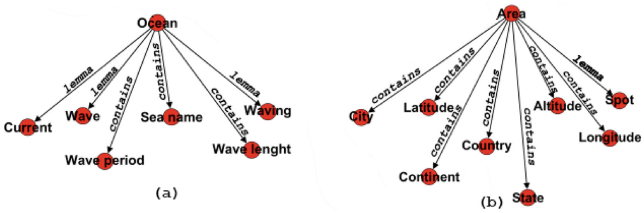

Fig. 4. Ego networks corresponding to *Ocean* and *Area*.

At this point, Step 2 is executed. As shown in Fig. 5, 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*[9], respectively). After a single iteration, the following triples are retrieved: ⟨*dbo*:*sea rdfs*:*range dbo*:*Sea*⟩ and ⟨*dbo*:*sea rdfs*:*domain dbo*:*Place*⟩. Other connections can be found by moving to specific instances of the mentioned resources. Indeed, the following triples are retrieved: ⟨*instance rdf*:*type dbo*:*Sea*⟩, ⟨*instance rdf*:*type dbo*:*Location*⟩, ⟨*instance rdf*:*type dbo*:*Place*⟩. Furthermore, a triple ⟨*instance dbo*:*country dbo*:*Country*⟩ can be retrieved. As a result, Step 2 succeeded in merging the two views that were separated after Step 1.

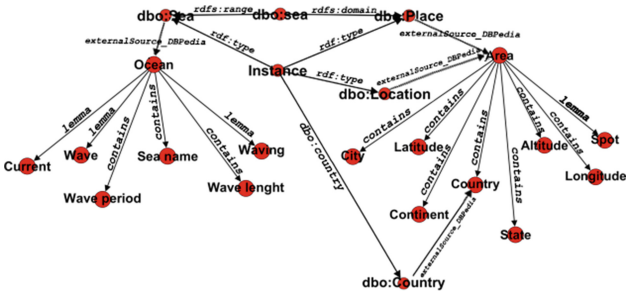

Fig. 5. The integrated thematic view.

6 Conclusion

In this paper, we have proposed a new metadata model well suited for representing and handling data lake sources. We have seen that our model starts from the ones generally used by data lake companies (in particolar, it starts from the model of Zaloni), but complements them with new ideas and, in particular, with the power guaranteed by a network-based and semantics-driven representation of available data. We have also seen that our model can allow a large variety of sophisticated tasks that the current metadata models cannot guarantee. This paper is not to be intended as an ending point. Actually, it could be the starting point of a new family of approaches that leverage our metadata model to address several open issues in data lake research; think, for instance, of approaches to supporting a flexible and lightweight querying of the sources of a data lake, as well as of approaches to schema matching, schema mapping, data reconciliation and integration strongly oriented to data lakes based mainly on unstructured data sources.

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

References

1. Abiteboul, S., Duschka, O.M.: Complexity of answering queries using materialized views. In: Proceedings of the International Symposium on Principles of Database Systems (SIGMOD/PODS 1998), Seattle, WA, USA, 1998, pp. 254–263. ACM (1998)
2. Aversano, L., Intonti, R., Quattrocchi, C., Tortorella, M.: Building a virtual view of heterogeneous data source views. In: Proceedings of the International Conference on Software and Data Technologies (ICSOFT 2010), Athens, Greece, 2010, pp. 266–275. INSTICC Pressd (2010)
3. Bergamaschi, S., Castano, S., Vincini, M., Beneventano, D.: Semantic integration and query of heterogeneous information sources. Data Knowl. Eng. **36**(3), 215–249 (2001)
4. Bilalli, B., Abelló, A., Aluja-Banet, T., Wrembel, R.: Towards intelligent data analysis: the metadata challenge. In: Proceedings of the International Conference on Internet of Things and Big Data (IoTBD 2016), Roma, Italy, 2016, pp. 331–338 (2016)
5. Biskup, J., Embley, D.: Extracting information from heterogeneous information sources using ontologically specified target views. Inf. Syst. **28**(3), 169–212 (2003). Elsevier
6. De Meo, P., Quattrone, G., Terracina, G., Ursino, D.: Integration of XML schemas at various "severity" levels. Inf. Syst. **31**(6), 397–434 (2006)
7. Fan, W., Wang, X., Wu, Y.: Answering pattern queries using views. IEEE Trans. Knowl. Data Eng. **28**(2), 326–341 (2016). IEEE
8. Fang, H.: Managing data lakes in big data era: what's a data lake and why has it became popular in data management ecosystem. In: Proceedings of the International Conference on Cyber Technology in Automation (CYBER 2015), Shenyang, China, 2015, pp. 820–824. IEEE (2015)
9. Farid, M., Roatis, A., Ilyas, I.F., Hoffmann, H., Chu, X.: CLAMS: bringing quality to Data Lakes. In: Proceedings of the International Conference on Management of Data (SIGMOD/PODS 2016), San Francisco, CA, USA, 2016, pp. 2089–2092. ACM (2016)
10. Hai, R., Geisler, S., Quix C.: Constance: an intelligent data lake system. In: Proceedings of the International Conference on Management of Data (SIGMOD/PODS 2016), San Francisco, CA, USA, 2016, pp. 2097–2100. ACM (2016)
11. Halevy, A.: Answering queries using views: a survey. VLDB J. **10**(4), 270–294 (2001). Springer
12. Hitzler, P., Janowicz, K.: Linked data, big data, and the 4th paradigm. Semant. Web **4**(3), 233–235 (2013)
13. Dublin Core Metadata Initiative. DCMI metadata terms. Technical report (2012)
14. Keith, A., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets with the void vocabulary. Technical report (2011)
15. Kondrak, G.: *N*-gram similarity and distance. In: Consens, M., Navarro, G. (eds.) SPIRE 2005. LNCS, vol. 3772, pp. 115–126. Springer, Heidelberg (2005). https://doi.org/10.1007/11575832_13
16. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with Cupid. In: Proceedings of the International Conference on Very Large Data Bases (VLDB 2001), Rome, Italy, 2001, pp. 49–58. Morgan Kaufmann (2001)
17. Navigli, R., Ponzetto, S.P.: BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (2012). Elsevier

18. Oram, A.: Managing the Data Lake. O'Reilly, Sebastopol (2015)
19. Palopoli, L., Pontieri, L., Terracina, G., Ursino, D.: Intensional and extensional integration and abstraction of heterogeneous databases. Data Knowl. Eng. **35**(3), 201–237 (2000)
20. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**(4), 334–350 (2001)
21. Singh, K., Singh, V.: Answering graph pattern query using incremental views. In: Proceedings of the International Conference on Computing (ICCCA 2016), Greater Noida, India, 2016, pp. 54–59. IEEE (2016)
22. Wang, J., Li, J., Yu, J.X.: Answering tree pattern queries using views: a revisit. In: Proceedings of the International Conference on Extending Database Technology (EDBT/ICDT 2011), Uppsala, Sweden, 2011, pp. 153–164. ACM (2011)