Decentralized Semantic Learning Infrastructure for Lifelong Learning

Sara El Hassad, Hala Skaf-Molli, Patricia Serrano-Alvarado, Pascal Molli, Emmanuel Desmontils

Decentralized Semantic Learning Infrastructure for Lifelong Learning

Sara El hassad, Hala Skaf-Molli, Patricia Serrano-Alvarado, Pascal Molli, and Emmanuel Desmontils

LS2N, University of Nantes, Nantes, France
(1) Hala.Skaf@univ-nantes.fr, (2) firstname.lastname@univ-nantes.fr

1 Introduction

Lifelong learning plays a fundamental role for the professional development of learners. It starts at the initial training, from school to university with all degrees and diplomas obtained, and continues throughout his career, with the different jobs occupied. In fact, personal learning data will be maintained for a long time, therefore they must be under the control of the learner.

Nowadays, personal learning data is dispersed across different and heterogeneous data sources: linkedIn, viadeo, e-portfelio, university web site, etc. None of these infrastructures provid a trusted environment for long term learning, data capitalization and personal learning management [16].

We envision a *Decentralized Semantic Learning Infrastructure for lifelong learning* based on semantic web technologies which offers seamless personal learning data integration from multiples and heterogeneous data sources, thanks to learner ontologies [13,2]. We propose two data integration services, the first service stores learner data in a private datastore, and the second one collect data during learning processes which allows to get fresh data, thanks to virtual data integration [18]. learner needs to interact with different communities in order to develop his autonomy which is an important point for lifelong learning. Decentralization of our infrastructure provides full usage control to the data owner, each learner can decide who can access what type of data and how they can reuse data.

In this paper, we study the state of the art of the two main points of our infrastructure for lifelong learning: semantic personal data integration of each learner in section 2 and trusted data sharing between learners 3

2 Semantic Personal Data Integration

Querying autonomous and heterogeneous data sources is challenging. There are two main approaches to resolve this problem of data integration: Data Warehousing [21], and Mediators and Wrappers [22]. A data integration system is defined using two schema: the global schema and the source schema, and a mapping between these two schemas.

Definition 1 (Data Integration System [3]). *A data integration system $\mathcal{I}$ is a triple $<\mathcal{G}, \mathcal{S}, \mathcal{M}>$ where*

- *$\mathcal{G}$ is the global schema, expressed in the relational model, possibly with constraints.*
- *$\mathcal{S}$ is the source schema, also expressed in the relational model.*
- *$\mathcal{M}$ is the mapping between $\mathcal{G}$ and $\mathcal{S}$, constituted by a set of assertions of the form $q_{\mathcal{S}} \subseteq q_{\mathcal{G}}$, where $q_{\mathcal{S}}$ and $q_{\mathcal{G}}$ are two queries of the same arity, respectively over the source schema $\mathcal{S}$ and over the global schema $\mathcal{G}$.*

2.1 Data Warehousing

The principe of data warehousing is to transform all data into the same form and to store it in a private space. Materialized views can be used in order to optimize the query execution. However materialized view selection is a non-trivial problem that has been deeply studied in the literature [7,10,5,12,9].

Data Warehousing approach is limited by the problem of freshness[1], i.e. in the case of data updates, querying local data may produce stale answers and materialized views must be maintained which can have a very high cost.

2.2 Virtual Integration using Mediators and Wrappers

In this approach, queries are posed on an intermediate layer called global schema $\mathcal{G}$ (see definition 1), without having to move data from sources, which will be transformed over global schema by wrappers and will be interrogated by mediators. Views represent the relation between data and the global schema. This approach allow querying fresh data where data transformation is performed in dynamic fashion during the query time. There is four approaches for quering heterogeneous data sources: Global-As-View (GAV) [14], Local-As-View (LAV) [14], Global-Local-As-View (GLAV) [8] and semLAV [18] which is an extention of LAV for SPARQL queries.

Example 1. Lets consider the global schema composed by the following relations: name(person, name), hasDiploma(person, diploma), university(diploma, university), year(diploma, year), hasProgram(diploma, university, year, program) and hasCourses(program, cours). Listing 2.2 presents the conjonctive query q written according to the global schema, that asks for courses taken by a person.

```
q(C) :- name(M,N), hasDiploma(M,D), university(D,U),
year(D,Y), hasProgram(D,U,Y,P), hasCourses(P,C)
```

Listing 1.1: Conjunctive query q

GAV mediators

Definition 2 (GAV approach [14]). *In the GAV approach, for each relation R in the mediated schema, we write a query over the source relations specifying how to obtain R's tuples from the sources.*

For instance, each relation of the global schema from example 1 can be written as views over sources s1, s2 and s3 as shown in Listing 1.1. Including new sources and updating data sources may require the addition of new views or the modification of several already defined views. For instance, if a new data source s4 that provides detailed course programme is included, two views have been added, one with a join with existing source s3 as shown in Listing 1.2. The query is simply rewritten by substituting global schema relations in the query by their correspond source views using query unfolding [6]. Listing 1.3 shows the query q and his rewrtiting r by remplacing global schema relations in q using views defined in Listing 1.1. In GAV approach, queries are naturaly rewritten but including or updating data sources is not obvious.

```
name(M,N) ⊇ s1(M,N)
hasDiploma(M,D) ⊇ s1(M,N), s2(N,D,V)
university(D,U) ⊇ s1(M,N), s3(N,D,U)
year(D,Y) ⊇ s1(M,N), s3(N,D,Y)
```

Listing 1.2: GAV mapping with three sources s1, s2, s3

```
name(M,N) ⊇ s1(M,N)
hasDiploma(M,D) ⊇ s1(M,N), s2(N,D,V)
university(D,U) ⊇ s1(M,N), s3(N,D,U)
year(D,Y) ⊇ s1(M,N), s3(N,D,Y)
hasProgram(D,U,Y,P) ⊇ s3(D,U,Y), s4(D,P)
hasCourses(P,C) ⊇ s4(P,C)
```

Listing 1.3: GAV mapping after including s4

```
q(C) :- name(M,N), hasDiploma(M,D), university(D,U),
year(D,Y), hasProgram(D,U,Y,P), hasCourses(P,C)
r(A1) :- s1(M,N), s2(N,D,V), s3(N,D,U), s3(N,D,Y),
s4(D,P), s4(P,C)
```

Listing 1.4: Query q and its rewriting r using assertions from Listing 1.1

LAV mediators

Definition 3 (LAV approach [14]). *In the LAV approach, the contents of a data source are described as a query over the mediated schema relation.*

Listing 1.4 provides an example of mapping between sources s1, s2, s3 and s4 and global schema. For example, from source s3 we can get informations about university where the learner obtained his diploma and the year of graduation. Since sources are defined as views over global schema, including or updating data sources do not affect existing mappings. Answering queries in LAV approach is not as easy as in GAV approach, we cannot just unfolding the query using GAV mappings, but we need to define rewritings of the query using notions of query containment and equivalence [6,11], containment mapping [6], containment [6], equivalent rewriting [11] and maximally-contained rewriting [11]. However, from one conjunctive query, a very high number of rewritings may be generated. Theorem 1 provides the number of candidate rewritings in the worst case.

Theorem 1 (Number of Candidate Rewritings [1]). *Let N, O and M be the number of query subgoals, the maximal number of views subgoals, and the set of views, respectively. The number of candidate rewritings in the worst case is:* $(O\times |M|)^N$.

—-Calculer number of candidate rew ... ?

```
s1(N,D,Y) ⊆ name(M,N), hasDiploma(M,D), year(D,Y)
s2(D,U) ⊆ Diploma(D), university(D,U)
s3(D,U,Y) ⊆ hasDiploma(M,D), university(D,U), year(D,Y)
s4(D,P,C) ⊆ Diploma(D), hasProgram(D,U,Y,P), hasCourses(P,C)
```

Listing 1.5: LAV mapping with four sources s1, s2, s3, s4

```
q(C) :- name(M,N), hasDiploma(M,D), university(D,U),
year(D,Y), hasProgram(D,U,Y,P), hasCourses(P,C)
r1(C) :- s1(N,D,Y), s2(D,U), s4(D,P,C)
r2(C) :- s1(N,D,Y), s3(D,U,Y), s4(D,P,C)
```

Listing 1.6: Query q and its two contained rewritings r1 and r2 using assertions from Listing 1.4

GLAV mediators

GLAV approach is a generalization of both GAV and LAV approaches [8]. Mappings in GLAV are defined between views of sources schema. and views of global schema as shown in Listing 1.6. GAV and LAV mappings are also GLAV mappings when only one relation of source schema or global schema is defined in GLAV mappings, however the converse is not true, i.e. GLAV mappings are neither LAV nor GAV mappings.
Query processing tasks in GLAV, i.e. query answering, query rewriting, perfectness of rewritings and relative query containment to a mapping, are at least as complex as LAV tasks for conjonctives queries [4].

```
s1(N,D,Y), s2(D,U) ⊆ name(M,N), hasDiploma(M,D),
  year(D,Y), Diploma(D), university(D,U)
s1(N,D,Y), s3(D,U,Y) ⊆ name(M,N), hasDiploma(M,D),
  year(D,Y), university(D,U)
```

Listing 1.7: GLAV sample mappings

semLAV mediators

Definition 4.

semLAV is a LAV-based approach with smart materialization of relevant views. It selects relevant views for a query Q and ranks them in order to maximize query results. Only top k ranked views are materialized. Contrary to traditional LAV that depends on the number of rewrited conjunctive queries, SemLav avoids generating rewritings and depends on the number and the size of relevant views. Relevant views of query Q is computed using algorithm 1 [17] that correspond to the first part of the bucket algorithm [15,11]. Algorithm 1 create a bucket for each subgoal of query Q where a bucket is a set of its relevant views, then bucket views are sorted according to the number of covered subgoals.
The global schema instance in semLAV is constructed using algorithm 2. One view is selected from each bucket in an iterative fashion, and its data is loaded into the instance.
Query can be executed after the addition of a new view to the instance which produces partial results or it can be executed after including all k views to the instance.
semLAV is a smart adaptation of LAV mediatoe for SPARQL queries, that allows the execution of the SPARQL query by voiding the generation of the exponential number of query rewriting, as with the traditional LAV mediator.

3 Trusted Data Sharing

They exists two kind of infrastructure in education domain: centralized infrastructures, for instance MAhara, LinkedIn, viadeo and decentralized infrastructures (peer-to-peer) sush as Elena and Edutella (see table **??**.

Tools	Context	Centralized/P2P	Access Control
Mahara[1]	e-portfolio	Centralized	Views
Elena [20]	educational services	P2P	-
Edutella [19]	educational services	P2P	-
LinkedIn[2]	professional social network	Centralized	-
Viadeo[3]	professional social network	Centralized	-

In our infrastructure we promote the peer-to-peer architecture that allows the total control of private data. Each data owner can associate different policies for reusing their data, depending on their privacy preferences.

Algorithm 1: CreateBuckets Algorithm [17]

Input: V: set of View; m: integer; Q: ConjunctiveQuery (with m subgoals)

```
1  Procedure createBuckets(V, Q)
2      forall i ∈ 1 ≤ i ≤ m do
3          Bucket_i ⟵ ∅
4      end
5      forall q ∈ body(Q) do
6          forall v ∈ V do
7              forall w ∈ body(v) do
8                  if predicate(q) = predicate(w) then
9                      if y = argument(k, w) ∧ distinguishable(y, v) then
10                         ψ(y) = argument(k, q)
11                     else
12                         ψ(y) = newVariable(Q, V)
13                     end
14                 end
15                 if satisfiable(body(Q) ∧ (∀p : p ∈ body(v) : ψ(p))) then
16                     if (∀a, i : distinguishable(a, Q) ∧ a = argument(i, q) :
                           distinguishable(argument(i, w), v)) then
17                         Bucket_i ⟵ Bucket_i ∪ ψ(head(v))
18                     end
19                 end
20             end
21         end
22     end
23 end
24 end procedure
```

Algorithm 2: The Global Schema Instance Construction and Query Execution [18]

Input: Q: Query
Input: $Buckets$: Predicate $\rightarrow$ $List < View >$
Input: k: Int
Output: A: Set<Answer>
$Stacks$: Predicate $\rightarrow$ Stack<View> V_k : Set<View> G: RDFGraph
forall $p \in domain(Buckets)$ **do**
  $Stacks(p) \leftarrow toStack(Buckets(p))$
end
while $(\exists p| : \neg empty(Stacks(p))) \wedge |V_k| < k$ **do**
  forall $p \in domain(Stacks) \wedge \neg empty(Stacks(p))$ **do**
    $v \leftarrow pop(Stack(p))$
    if $v \notin V_k$ **then**
      load v into G only if is not redundant
      $A \leftarrow A \cup exec(Q, G)$ {Option 1: Execute Q after each successful load}
      $V_k \leftarrow V_k \cup \{v\}$
    end
  end
end
$A \leftarrow exec(Q, G)$ {Option 2: execute before exit}

Only Mahara provide a limited access control of personal data by defining data views.

4 Conclusion

In this paper we discute the state of the art of the two main issues to construct the decentralized and semantic learning Infrastructure for lifelong learning, i.e. semantic personal data integration and trusted data sharing.

Acknowledgement

This work is part of the multidisciplinary project SEDELA, funded by CominLabs, that brings together three laboratories: LS2N - *University of Nantes*, CREAD - *University of Rennes 2*, and Lab-STICC - *IMT-Atlantique*.

References

1. Abiteboul, S., Manolescu, I., Rigaux, P., Rousset, M.C., Senellart, P.: Web data management. Cambridge University Press (2011)
2. Ameen, A., Khan, K.U.R., Rani, B.P.: Creation of ontology in education domain. In: T4E. pp. 237–238. IEEE Computer Society (2012)

3. Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M.: On the expressive power of data integration systems. In: International Conference on Conceptual Modeling. pp. 338–350. Springer (2002)
4. Calvanese, D., De Giacomo, G., Lenzerini, M., Vardi, M.Y.: Query processing under glav mappings for relational and graph databases. In: Proceedings of the VLDB Endowment. vol. 6, pp. 61–72. VLDB Endowment (2012)
5. Chirkova, R., Halevy, A.Y., Suciu, D.: A formal perspective on the view selection problem. The VLDB Journal—The International Journal on Very Large Data Bases 11(3), 216–237 (2002)
6. Doan, A., Halevy, A., Ives, Z.: Principles of data integration. Elsevier (2012)
7. Espinola, R.H.C.: Indexing rdf data using materialized sparql queries (2012)
8. Friedman, M., Levy, A.Y., Millstein, T.D., et al.: Navigational plans for data integration. AAAI/IAAI 1999, 67–73 (1999)
9. Goasdoué, F., Karanasos, K., Leblay, J., Manolescu, I.: View selection in semantic web databases. Proceedings of the VLDB Endowment 5(2), 97–108 (2011)
10. Gupta, H.: Selection of views to materialize in a data warehouse. In: International Conference on Database Theory. pp. 98–112. Springer (1997)
11. Halevy, A.Y.: Answering queries using views: A survey. The VLDB Journal 10(4), 270–294 (2001)
12. Karloff, H., Mihail, M.: On the complexity of the view-selection problem. In: Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 167–173. ACM (1999)
13. Katis, E., Kondylakis, H., Agathangelos, G., Vassilakis, K.: Developing an ontology for curriculum and syllabus. In: ESWC (Satellite Events). Lecture Notes in Computer Science, vol. 11155, pp. 55–59. Springer (2018)
14. Levy, A.Y.: Logic-based techniques in data integration. In: Logic-based artificial intelligence, pp. 575–595. Springer (2000)
15. Levy, A.Y., Rajaraman, A., Ordille, J.J.: Querying heterogeneous information sources using source descriptions. In: VLDB. pp. 251–262. Morgan Kaufmann (1996)
16. Mawas, N.E., Gilliot, J., Garlatti, S., Serrano-Alvarado, P., Skaf-Molli, H., Eneau, J., Lameul, G., Marchandise, J., Pentecouteau, H.: Towards a self-regulated learning in a lifelong learning perspective. In: CSEDU (1). pp. 661–670. SciTePress (2017)
17. Montoya, G.: Answering SPARQL Queries using Views. (Répondre aux Requêtes SPARQL grâce aux Vues). Ph.D. thesis, University of Nantes, France (2016)
18. Montoya, G., Ibáñez, L.D., Skaf-Molli, H., Molli, P., Vidal, M.E.: Semlav: Local-as-view mediation for sparql queries. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XIII, pp. 33–58. Springer (2014)
19. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: Edutella: a p2p networking infrastructure based on rdf. In: Proceedings of the 11th international conference on World Wide Web. pp. 604–615. ACM (2002)
20. Simon, B., Miklos, Z., Nejdl, W., Sintek, M., Salvachua, J.: Elena: A mediation infrastructure for educational services. In: WWW (Alternate Paper Tracks) (2003)
21. Theodoratos, D., Sellis, T., et al.: Data warehouse configuration. In: VLDB. vol. 97, pp. 126–135 (1997)
22. Wiederhold, G.: Mediators in the architecture of future information systems. Computer 25(3), 38–49 (1992)