Towards a Benchmark for Knowledge Base Exchange

Bahar Ghadiri Bashardoost[1], Renée J. Miller[2], and Kelly Lyons[1]

[1] University of Toronto
{ghadiri,klyons}@cs.toronto.edu
[2] Northeastern University
miller@northeastern.edu

Abstract. As interest in supporting data exchange between heterogeneous knowledge bases (KBs) has increased, so has interest in benchmarking KB exchange systems. One important benchmark, LODIB (Linked Open Data Integration Benchmark), has been proposed that reflects the real and deep heterogeneity of knowledge bases in the Linked Open Data Cloud. In this position paper, we reflect on requirements for benchmarks of KB exchange systems and bring to bear important lessons learned from other data exchange systems. Specifically, we consider the importance, in data exchange, of explicitly modeling incompleteness, something that is at the heart of relational data exchange and that is solved using principled value invention methods. We also consider the incompleteness that arises naturally within knowledge bases and how that influences KB exchange. Instances of a single class in a KB may exhibit heterogeneity in their structure and modeling this is important in a KB exchange benchmark. Using these insights, we propose a set of new requirements for a KB exchange benchmark.

Keywords: Data Exchange · Knowledge bases · Benchmarking.

1 Introduction

Machines cannot comprehend a large portion of data produced by humans. The grand vision of the Semantic Web is to be able to create a web-scale decentralized corpus of information and knowledge that can be processed automatically by machines [11]. Large-scale machine-readable KBs are one of the main enablers of the Semantic Web vision. These KBs contain rich information about domain-specific and general concepts (also known as classes or types), the relationships among them (also known as properties), and their instances (also known as individuals). KBs provide a means for machines to be able to make inferences

DI2KG 2019, August 5, 2019, Anchorage, Alaska.

and answer questions about the knowledge they store. To this end, it is very important that a KB is populated with relevant facts that model the domain that the KB tries to represent. However, populating an existing KB is not a trivial task. One powerful method for KB expansion is to share or exchange information between different, heterogeneous KBs.

Researchers have begun to formalize the problem of exchanging knowledge between a source and a target KB given a set of mapping rules [7, 4, 5, 3]. In addition, several tools have emerged which aim to automatically generate mapping rules between two KBs [30, 32, 29, 33], making KB exchange[1] more achievable than ever. Naturally, benchmarks are also beginning to be proposed to evaluate solutions to challenges that arise in practical KB exchange such as expressiveness of mapping languages used [34], or the efficiency and the quality of automatic mapping rule generation [31, 34]. Although these benchmarks propose an important set of criteria for evaluating various aspects of practical KB exchange, we believe that their evaluation criteria can be enriched to raise the bar for evaluating KB exchange.

Notably, our work considers the materialized exchange of data which can be distinguished from OBDA and OBDI. In Ontology-based Data Access (OBDA) the goal is typically to provide a virtual ontology (or KB) interface over databases expressed in different data models (also known as federation or mediation) [38]. In Ontology-based Data Integration (OBDI), also known as ontology merging, the goal is to create a single ontology (KB) that represents all information in a set of source KBs or DBs [37, 35, 22]. Solutions for OBDI mostly assume the heterogeneity is limited and can be reconciled with simple mappings like `same-as`, `subclass-of`, or `equivalent-class`. In contrast, both OBDA and KB exchange use much richer mapping rules that need to be represented using more expressive mapping languages. The benchmarking of KB exchange can inform the benchmarking of both OBDA and OBDI. In this position paper, we look at lessons learned from benchmarks for data exchange and mapping generation systems and from how KBs are being used in practice. We propose a new set of requirements for KB mapping generation tools that can be incorporated into benchmarks to broaden and deepen the tool evaluations. Having such benchmarks can attract more attention to this important area by highlighting the challenges that remain and improving the quality of research by pointing out new issues that are unique to KB exchange.

1.1 Benchmarking Data Exchange

Populating a structured information resource (target) using data translated from another structured resource (source) is one of the oldest problems in data management. Data exchange [19] is a prominent approach for solving this problem when both source and target adhere to a relational or nested relational data

[1] The term *KB Exchange* was introduced by Arenas et al. [3] and we use it in this work to distinguish a specific setting of the data exchange problem in which the source and target are both KBs.

model. The theory of data exchange was introduced by Fagin et al. [19]. This work laid the theoretical foundation of data exchange and identified important tasks in data exchange, including materializing a good target instance and answering queries. In data exchange, a set of mapping rules specify the relationship between a source and target schema. As defined by Fagin et al. [19], given a source schema, a target schema, an instance of the source schema, and a set of mapping rules, *data exchange* is the problem of creating an instance of the target that reflects the source instance as closely as possible. Fagin et al. defined when a target instance is a *solution* for data exchange and defined a class of good solutions (called universal solutions). They also defined a declarative mapping language (source-to-target tuple-generating-dependencies) for relational schemas that has a good balance between expressiveness and algorithmic properties and has since been widely adopted in tools and generalized to other data models [6, 2, 4]. An important issue in data exchange is *value invention* [21] which is required when the target models data not present in the source that plays a structural role in connecting other data [9].

The problem of data exchange assumes that mapping rules are given. However, in practice one important challenge that needs to be addressed is how to obtain these mapping rules. These rules can be defined manually, however the manual process is burdensome [15]. As a result, there is a large body of literature on systems that reduce the burden of creating these rules manually, by making the rule creation process as automatic as possible. Clio was an early mapping generation system for relational and XML schemas [27, 26], and this has been followed by dozens of other research and industrial mapping systems [13]. Most systems use a set of correspondences between the source and target, and enrich them using implicit information that lies within the schemas (or instances) of the source and target to generate semantically correct mapping rules. Systems that use declarative mapping rules also translate them into executable programs (such as queries or scripts) that produce a single data exchange solution.

Over time, benchmarks have been developed to evaluate these systems. One of the first was STBenchmark [1] that proposed the use of mapping scenarios (or mapping patterns) that represent a set of transformations that should be supported by any mapping tool. More recently, iBench [8] permits the efficient creation of benchmarks with large and complex schemas and mappings. iBench provides control over a large set of mapping characteristics including the amount and complexity of the value invention required in a correct mapping solution. Bellahsene et al. [10] provide a survey of work on evaluating both schema matching and mapping discovery between schemas in different data models. An important (and computationally hard) issue in benchmarking data exchange is understanding if the solutions produced by mapping rules are always universal solutions [8].

1.2 Benchmarking Knowledge Base Exchange

Arenas et al. [7] were one of the first to investigate data exchange among KBs. They identified the potential incompleteness of a source instance as one of the

main challenges that needs to be addressed in KB exchange [7, 4, 5, 3]. For an incomplete source instance, decisions on how to exchange the unknown entities of the source are not trivial. Another important challenge which arises in KB exchange is that both atomic values (RDF literals) and individuals (RDF IRIs) need to be exchanged.

Unlike in data exchange, there has been less work focused on automated generation of mapping rules for KB exchange [30, 32, 29, 33]. Additionally, a single declarative mapping constraint language has not been widely adopted, so most systems generate executable mappings directly which produce a single KB exchange instance. Most of these tools generate mapping rules by enriching a set of given correspondences among two KBs, using clues that lie within the constraints of the KBs [30, 32, 29]. Buhmann et al. [14] stated that, in practice, there is a lack of KBs that contain high quality schema axioms and sufficient instance data adhering to the schema. Thus, it is important for KB mapping generation tools to rely on a small set of axioms which are commonly used in a large number of KBs to create mappings. Relying on other less used axioms (or axioms not part of a widely used schema language like RDFS) limits the applicability of a tool and the ability to compare it with others. Like many KB integration or alignment approaches [37, 36], we think it is reasonable for a mapping generation tool to require at most a small set of commonly used constructs, (for example, `rdfs:subClassOf`, `rdfs:domain`, `rdfs:range`, and `rdf:type`). Such constructs are present in most KBs (even those that are automatically created from relational or semi-structured data sets) and are part of the W3C standard OWL Web Ontology Language. We refer to tools that aid in the generation of KB mapping rules for KB exchange between two KBs as *KMGT*.

Benchmarks such as STBenchmark and metadata generation tools like iBench evaluate tools using a set of parameterized mapping scenarios (or mapping patterns). To evaluate *KMGTs*, DTSBenchmark [31], proposed a set of patterns when the source and target are both KBs. These patterns were later extended in LODIB (linked open data integration benchmark) [34]. Since a single mapping language for KB exchange is not yet widely adopted, LODIB is mainly designed to benchmark the expressive power of mapping languages. This benchmark has been also used to evaluate *KMGTs*. The scenarios proposed in DTSBenchmark and LODIB represent an important set of transformations that should be supported by *KMGTs*. In this paper, we identify additional scenarios that we think should be supported by any mapping generation tool and any language that aims to express KB mapping rules. Our goal is to push the research on KB exchange to the next level.

2 Requirements for *KMGTs*

We now describe some challenges that every *KMGT* should address in order for it to be of use in practical applications of KB exchange. Any benchmark that aims to evaluate *KMGTs* should include scenarios that can be used to test *KMGTs*' output in the presence of these challenges. Thus, in addition to describ-

ing these challenges, we include example scenarios which can be incorporated in benchmarks. Note that the challenges discussed in our paper augment and do not replace other benchmarks.

2.1 Dealing with Incomplete Correspondences

In data exchange, mapping rules are created by enriching a given set of correspondences with implicit information which lies within the schemas, the data, or which is provided by users (through crowd-sourcing or visual interfaces). The algorithms proposed for generating these rules often assume that the set of given correspondences is *incomplete* [23, 24, 26, 28, 12]. This is a realistic assumption since these correspondences are usually the output of an ontology alignment (a.k.a., schema matching) process. Assuming that an alignment process can automatically produce *all* correspondences between two KBs with high precision is usually unrealistic [17]. When the set of correspondences is incomplete, a mapping generation tool aims to produce mapping rules based on understanding the different ways that corresponding schema elements can be associated with each other. To the best of our knowledge, current *KMGTs* assume the input correspondences are complete. If there is no correspondence to a target element (for example a property P), current *KMGTs* do not populate the element with any data. We argue instead that if there is source data that could be used to populate the element (for example, a source property path between source elements that are matched by correspondences to the target domain and range of P), then a useful function of a *KMGT* is to suggest a (perhaps ranked) list of alternative ways to populate the unmatched target element.

Unfortunately, the LODIB benchmark does not consider this case since as mentioned, it is designed to benchmark the expressive power of languages that represent the mapping rules. DTSBenchmark is designed to benchmark the *KMGTs*, however the scenarios proposed in this benchmark always contain complete sets of correspondences. An example of a scenario that tests the ability of a *KMGT* to handle incomplete correspondences is given below.

Scenario 1. Missing Correspondence to a Target Property:
Figure 1.a represents the RDFS layer of a source and target KB and a single correspondence which has been identified between them. We argue that a *KMGT* should create mapping rules that not only translate the source `Person` instances into the target `Person`, but also suggest possible ways of populating the unmatched `related` property in the target. For example, a *KMGT* should suggest a set of mapping rules that express that if a `Person` is `supervised` by another `Person` in the source, these two `Person`s are `related` in the target. Or if a `Person` P_1 `hasWorkedOn` `Project` J, and a `Person` P_2 is a `contributor` to `Project` J in the source, then P_1 should be `related` to P_2 in the target. □

A *KMGT* should be able to suggest a set of mapping rules that are consistent with the schemas and correspondences (and perhaps rank them if additional information like data examples are available [15]). Indeed, a lesson learned from

data exchange is that an important role of *KMGTs* is in systematically enumerating possible mappings, something humans do poorly [18].

2.2 Knowledge Base Value Invention

Of course, sometimes an unmatched target element cannot be populated with source data. This occurs often in data exchange. In order to materialize a target instance, sometimes values need to be filled in for the undetermined elements. For instance, Clio [27] creates `oids` (using Skolem functions), which are unique identifiers (also called a *labeled nulls*), when source data is matched to multiple relations connected by unmatched target foreign keys. These labeled nulls help capture some of the structural characteristics of the target data. Similarly, when dealing with KBs, sometimes a value needs to be created in order to preserve the associations between the resources of the target KB. As discussed by Arenas et al. [4], *blank nodes* in knowledge graphs can play the same role that *labeled nulls* play in relational and nested relational data exchange (see [20] for more information on blank nodes). Thus similar to relational data exchange, blank nodes should be used if an association between two transferred values must be represented and the mapping rules must have enough information to guide the generation of these blank nodes. Unfortunately, in the benchmarks proposed so far, there is no scenario that clearly evaluates value invention in $KMGT$s. We propose two scenarios in this direction.

Scenario 2. Value Invention 1:
Consider Figure 1.b. An `office` is not a `contact` so the IRIs of `offices` are not exchanged into `contacts`. However, the $KMGT$ must be able to create a blank node which associates `address` and `phone` properly. That is, in this scenario a $KMGT$ should be able to capture a source-to-target dependency which expresses that for each `office` in the source that has an `address` a, and a `phone` p, a and p are associated with each other in the target. □

The following example illustrates why supporting the scenario above is not trivial for $KMGT$s. Imagine a $KMGT$ produces the following queries as mapping rules to transfer data from a source to a target. The `where` clauses of these queries select data from the source and the `construct` clauses create the facts in the target KB using the target structure and the values selected from the source. These two queries should not be considered correct in a benchmark evaluation since they do not guarantee that for a single `office` in the source with `phone` *416-345* and `address` *Toronto* that a single blank node is created in the target associated with these same values.

```
construct{
  ?-:a a trgt:Contact.
  ?-:a trgt:phone ?trgtPhone}
where{
  ?o a src:Office.
  ?o src:phone ?srcPhone.
  bind(?srcPhone as ?trgtPhone)}
```

&

```
construct{
  ?-:b a trgt:Contact.
  ?-:b trgt:address ?trgtAddress}
where{
  ?o a src:Office.
  ?o src:Address ?srcAddress.
  bind(?srcAddress as ?trgtAddress)}
```

Scenario 3. Value Invention 2:
Consider Figure 1.c. The difference between this scenario and Scenario 2 is that the blank nodes which need to be created are for a concept (`Address`) which does not have any data property which participates in a correspondence. However, appropriate blank nodes need to be created to correctly associate individual people and their countries in the target. □

2.3 Dealing with an Open-World Assumption

Traditionally, the data exchange problem is defined over a setting where the source instance is assumed to be complete and has a single interpretation. However, KBs follow an *open-world assumption* and are by nature incomplete. The mapping rules which are generated automatically must be able to handle *unknown* values in the source KB properly. Consider Figure 1.b and imagine that a $KMGT$ produced the following query as a mapping rule.

```
1   construct{
2     ?-:p a trgt:Contact.
3     ?-:p trgt:phone ?trgtPhone.
4     ?-:p trgt:address ?trgtAddress}
5   where{
6     ?o a src:Office.
7     ?o src:Address ?srcAddress.
8     ?o src:phone ?srcPhone.
9     bind(?srcAddress as ?trgtAddress)
10    bind(?srcPhone as ?trgtPhone)}
```

It is straightforward to see that the mapping rule expressed by the SPARQL query above is not enough to transfer all instances of `src:Office`. For example the query above can not be used to transfer information of an `Office` which does not have a `Phone`. One way to fix the problem above is to break the above query into two queries (see Scenario 2). However, as we have seen in Section 2.2, breaking queries into separate smaller queries is not trivial and can result in other problems. Another approach when dealing with missing values of KBs is to use SPARQL's `optional` keyword [39] and this approach can be adapted in KB exchange. Whatever the approach, it is important that a $KMGT$ create a correct target instance even when the source is incomplete and that a benchmark include scenarios to test this. An example scenario for testing the ability of a $KMGT$ to deal with an incomplete source instance is given below.

Scenario 4. Missing Values in the Source:
Consider Figure 1.b. Also, assume that the source instance contains the following facts: `Office(office1), Office(office2), phone(office1, '416-345'), phone(office2, '416-456'), address(office1, 'Toronto')`. In this case, it is expected that the mapping rule can materialize a correct instance of the target such as the following (where c1 and c2 are blank nodes) `Contact(-:c1), Contact(-:c2), phone(-:c1, '416-345'), phone(-:c2, '416-456'), address(-:c1, 'Toronto')`. It is important that all data from the source including `office2` with an unknown `address` be mapped to the target. □

2.4 Dealing with Cycles

A cycle can be used in a KB to model the relationships between individuals of the same type. For instance, Figure 1.a depicts many cycles each modeling a relationship between two people. This type of cycle is very common. For instance, any KB created from a social network contains a large number of these cycles, since social networks model various relationships between instances of a specific type. We believe that an algorithm that generates mapping rules should be able to handle this type of cycle. Scenario 5 can be used to test whether a $KMGT$ can correctly map a cycle. We have said that an important role of *KMGTs* is in systematically enumerating possible mappings. Cycles pose a challenge as the possible mappings become infinite.

Scenario 5. Property path with same domain and range:
Consider Figure 1.a. It is expected that a $KMGT$ can generate mapping rules based on cycles of source and target KBs. For instance a mapping rule for the setting introduced in this scenario might express something like [2]:

$\forall(x, y), x \neq y,$
where $(x, y) \in [[((hasSupervisor^* + (hasWorkedOn.contributor)^*)$
$.(hasSupervisor^* + (hasWorkedOn.contributor)^* + hasSupervisor^{-*}))^*]]_{src}$
then $(x, y) \in [[related]]_{trgt}$

What is important is for a $KMGT$ to consider alternatives to help a mapping designer to arrive at a semantically correct mapping. □

Not all mapping languages will contain recursion so a $KMGT$ may create mappings that only traverse a cycle a fix number of times. Nonetheless, a benchmark should include scenarios containing cycles and where different mappings are desired, some that include a single traversal (e.g., only an immediate supervisor) and some that include more (e.g., all supervisors and their supervisors).

3 Vision for Knowledge Base Exchange Benchmarks

Data exchange between heterogeneous data sources is important for sharing data among organizations but, until recently, much of the data exchange research has focused on relational and nested relational data models. Initiatives such as ontology based data access (OBDA) [38] aim to facilitate the exchange of data between a relational source and a target KB. New exciting work is emerging on the theory and practice of KB exchange, where a KB can be used to expand the knowledge contained in another KB. We believe that expanding the current set of benchmarks in a principled way can enhance research in both mapping generation and KB exchange.

Our position paper enumerates a few important benchmarking issues, but of course is not complete. One issue that we did not include is dealing with instances with multiple most-specific types. Suppose that in a source KB two concepts `Employee` and `Student` are sub-classes of another concept `Person`. Now

[2] Notation borrowed from Kostylev et al. [25].

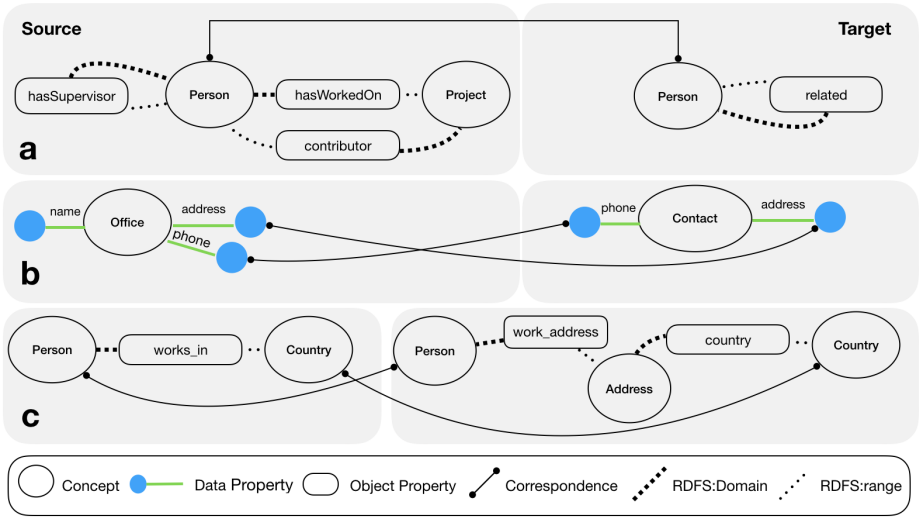

Fig. 1. RDFS layer of Source and target KBs and the correspondences between them

assume that instance i is both an `Employee` and a `Student` and instance j is only an `Employee`. Then the mapping rules generated should be able to transfer i and j to the target while respecting the fact that i and j share some properties while differing on other properties.

Benchmarking KB exchange presents many important new research challenges. Testing whether a set of mappings is equivalent can already be undecidable for data exchange and is more complex for KB exchange [4]. For a benchmark, we must develop new scalable ways of testing if mappings (or KB exchange solutions) for a benchmark are correct and sufficient. Duo et al. [16] argue that when it comes to the problem of exchanging data between two KBs, even checking that a given set of mapping rules is consistent with each other and with the axioms of the source and the target KBs is not a trivial task. The consistency of a set of mapping rules generated by a $KMGT$ can be a powerful measure of quality that benchmarks can report. Thus, we also suggest new approaches need to be proposed to be able to efficiently evaluate the consistency of the mapping rules generated by a $KMGT$.

Acknowledgement

This research was supported in part by a Natural Sciences and Engineering Research Council (NSERC) Strategic Partnership Grant.

References

1. Alexe, B., Tan, W.C., Velegrakis, Y.: Stbenchmark: towards a benchmark for mapping systems. PVLDB **1**(1), 230–244 (2008)

2. Amano, S., Libkin, L., Murlak, F.: Xml schema mappings. In: ACM PODS. pp. 33–42 (2009)
3. Arenas, M., Botoeva, E., Calvanese, D.: Knowledge base exchange. In: International Workshop on Description Logics. p. 4 (2011)
4. Arenas, M., Botoeva, E., Calvanese, D., Ryzhikov, V.: Knowledge base exchange: The case of owl 2 ql. Artificial Intelligence **238**, 11–62 (2016)
5. Arenas, M., Botoeva, E., Calvanese, D., Ryzhikov, V., Sherkhonov, E.: Exchanging description logic knowledge bases. In: KR. AAAI Press (2012)
6. Arenas, M., Libkin, L.: Xml data exchange: consistency and query answering. JACM **55**(2), 7 (2008)
7. Arenas, M., Pérez, J., Reutter, J.: Data exchange beyond complete data. JACM **60**(4), 28 (2013)
8. Arocena, P.C., Glavic, B., Ciucanu, R., Miller, R.J.: The ibench integration metadata generator. PVLDB **9**(3), 108–119 (2015)
9. Arocena, P.C., Glavic, B., Miller, R.J.: Value invention in data exchange. In: SIGMOD. pp. 157–168 (2013)
10. Bellahsene, Z., Bonifati, A., Duchateau, F., Velegrakis, Y.: On evaluating schema matching and mapping. In: Schema matching and mapping, pp. 253–291. Springer (2011)
11. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific american **284**(5), 28–37 (2001)
12. Bonifati, A., Mecca, G., Pappalardo, A., Raunich, S., Summa, G.: Schema mapping verification: The spicy way. In: EDBT. pp. 85–96. ACM (2008)
13. Bonifati, A., Mecca, G., Papotti, P., Velegrakis, Y.: Discovery and correctness of schema mapping transformations. In: Schema matching and mapping, pp. 111–147. Springer (2011)
14. Bühmann, L., Lehmann, J.: Universal owl axiom enrichment for large knowledge bases. In: EKAW. pp. 57–71. Springer (2012)
15. ten Cate, B., Kolaitis, P.G., Tan, W.C.: Schema mappings and data examples. In: EDBT. pp. 777–780. ACM (2013)
16. Dou, D., McDermott, D., Qi, P.: Ontology translation on the semantic web. In: Journal on data semantics II, pp. 35–57. Springer (2005)
17. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, 2nd edn. (2013)
18. Fagin, R., Haas, L., Hernández, M., Miller, R., Popa, L., Velegrakis, Y.: Clio: Schema mapping creation and data exchange. Conceptual modeling: foundations and applications pp. 198–236 (2009)
19. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theoretical Computer Science **336**(1), 89–124 (2005)
20. Hogan, A., Arenas, M., Mallea, A., Polleres, A.: Everything you always wanted to know about blank nodes. Journal of Web Semantics **27**, 42–69 (2014)
21. Hull, R., Yoshikawa, M.: ILOG: declarative creation and manipulation of object identifiers. In: VLDB. pp. 455–468 (1990)
22. Jiménez-Ruiz, E., Grau, B.C.: Logmap: Logic-based and scalable ontology matching. In: ISWC. pp. 273–288. Springer (2011)
23. Kimmig, A., Memory, A., Miller, R.J., Getoor, L.: A collective, probabilistic approach to schema mapping. In: ICDE. pp. 921–932 (2017)
24. Kimmig, A., Memory, A., Miller, R.J., Getoor, L.: A collective, probabilistic approach to schema mapping using diverse noisy evidence. TKDE (2018)
25. Kostylev, E.V., Reutter, J.L., Romero, M., Vrgoč, D.: Sparql with property paths. In: ISWC. pp. 3–18. Springer (2015)

26. Miller, R.J., Haas, L.M., Hernández, M.A.: Schema mapping as query discovery. In: VLDB. vol. 2000, pp. 77–88 (2000)
27. Popa, L., Velegrakis, Y., Hernández, M.A., Miller, R.J., Fagin, R.: Translating web data. In: VLDB. pp. 598–609 (2002)
28. Qian, L., Cafarella, M.J., Jagadish, H.: Sample-driven schema mapping. In: SIGMOD. pp. 73–84. ACM (2012)
29. Qin, H., Dou, D., LePendu, P.: Discovering executable semantic mappings between ontologies. OTM pp. 832–849 (2007)
30. Rivero, C.R., Hernández, I., Ruiz, D., Corchuelo, R.: Generating sparql executable mappings to integrate ontologies. In: ER. pp. 118–131. Springer (2011)
31. Rivero, C.R., Hernández, I., Ruiz, D., Corchuelo, R.: On benchmarking data translation systems for semantic-web ontologies. In: CIKM. pp. 1613–1618. ACM (2011)
32. Rivero, C.R., Hernández, I., Ruiz, D., Corchuelo, R.: Exchanging data amongst linked data applications. Knowl Inf Syst **37**(3), 693–729 (2013)
33. Rivero, C.R., Hernández, I., Ruiz, D., Corchuelo, R.: Mapping rdf knowledge bases using exchange samples. Knowledge-Based Systems **93**, 47–66 (2016)
34. Rivero, C.R., Schultz, A., Bizer, C., Ruiz Cortés, D.: Benchmarking the performance of linked data translation systems. In: LDOW. CEUR-WS (2012)
35. Stumme, G., Maedche, A.: Fca-merge: Bottom-up merging of ontologies. In: IJCAI. vol. 1, pp. 225–230 (2001)
36. Suchanek, F.M., Abiteboul, S., Senellart, P.: Paris: Probabilistic alignment of relations, instances, and schema. PVLDB **5**(3), 157–168 (2011)
37. Udrea, O., Getoor, L., Miller, R.J.: Leveraging data and structure in ontology integration. In: SIGMOD. pp. 449–460. ACM (2007)
38. Xiao, G., Calvanese, D., Kontchakov, R., Lembo, D., Poggi, A., Rosati, R., Zakharyaschev, M.: Ontology-based data access: A survey. IJCAI (2018)
39. Xiao, G., Kontchakov, R., Cogrel, B., Calvanese, D., Botoeva, E.: Efficient handling of sparql optional for obda. In: ISWC. pp. 354–373. Springer (2018)