Evaluation of query transformations without data

Short paper

Jérôme David
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG
Grenoble, France
Jerome.David@univ-grenoble-alpes.fr

Jérôme Euzenat
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG
Grenoble, France
Jerome.Euzenat@inria.fr

Pierre Genevès
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG
Grenoble, France
Pierre.Geneves@cnrs.fr

Nabil Layaïda
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG
Grenoble, France
Nabil.Layaida@inria.fr

ABSTRACT

Query transformations are ubiquitous in semantic web query processing. For any situation in which transformations are not proved correct by construction, the quality of these transformations has to be evaluated. Usual evaluation measures are either overly syntactic and not very informative —the result being: correct or incorrect— or dependent from the evaluation sources. Moreover, both approaches do not necessarily yield the same result. We suggest that grounding the evaluation on query containment allows for a data-independent evaluation that is more informative than the usual syntactic evaluation. In addition, such evaluation modalities may take into account ontologies, alignments or different query languages as soon as they are relevant to query evaluation.

CCS CONCEPTS

• **Information systems → Semantic web description languages**; • **Theory of computation → Logic and verification**; **Database query processing and optimization (theory)**;

KEYWORDS

SPARQL Query transformation; Query containment; Transformation evaluation

ACM Reference Format:
Jérôme David, Jérôme Euzenat, Pierre Genevès, and Nabil Layaïda. 2018. Evaluation of query transformations without data: Short paper. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3184558.3191617

With the availability of standard web knowledge representation languages such as RDF and OWL, SPARQL querying is becoming ubiquitous to access data. As for SQL before it, the language allows for its manipulation before being evaluated. SPARQL queries may be transformed to optimise their evaluation, to deal with heterogeneous vocabularies or to use more restricted query languages.

Proc. WWW RoD workshop, April 24th, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3191617

$t_1(q) =$

```
SELECT ?x ?y
WHERE {
 ?x p ?y .
 ?z q ?y .
 ?z r ?x .
}
```

$t_2(q) =$

```
SELECT ?v ?w
WHERE {
 ?b q ?w .
 ?v p ?w .
}
```

$t_3(q) =$

```
SELECT ?x ?y
WHERE {
 ?x p ?y .
}
```

$t_4(q) =$

```
SELECT ?x ?y
WHERE {
 ?x p ?y .
 ?y q ?z .
}
```

$q_R =$

```
SELECT ?x ?y
WHERE {
 ?x p ?y .
 ?z q ?y .
}
```

Figure 1: Sample queries resulting from transformations (the FROM clause is omitted; DISTINCT could be added for the evaluation since we use the set semantics).

In particular, query transformation is at the heart of ontology-based data access (OBDA) [10] and federated querying [15, 12, 14]. Some transformations are theoretically proved correct by construction (this is the case for some OBDA) [3, 1], however some others, especially when they connect data sources expressed in different ontologies, may be designed by hand [9] or resorting to ad hoc rules [5, 12, 7, 17, 16]. In such cases, the quality of these transformations has to be assessed.

1 EVALUATING TRANSFORMED QUERIES

Query transformation methods transform a query q into a replacement query $t(q)$. They may be used to evaluate the query against the same data source, i.e., database or RDF graph, or against a different data source using a different data model.

The evaluation of the quality of such transformations is measured by computing a value that we will consider in the [0 1] interval. It is usually performed along two different modalities.

In the first modality [7], an initial query q and the query it is expected to be transformed in q_R (reference query) are given and

query	G_1	G_2
q_R	$\langle a,c\rangle\ \langle f,d\rangle\ \langle c,e\rangle$	$\langle a,c\rangle\ \langle b,c\rangle\ \langle c,e\rangle\ \langle g,e\rangle$
$t_1(q)$	$\langle a,c\rangle\ \langle f,d\rangle\ \langle c,e\rangle$	$\langle a,c\rangle\ \langle c,e\rangle$
$t_2(q)$	$\langle a,c\rangle\ \langle f,d\rangle\ \langle c,e\rangle$	$\langle a,c\rangle\ \langle b,c\rangle\ \langle c,e\rangle\ \langle g,e\rangle$
$t_3(q)$	$\langle a,c\rangle\ \langle f,d\rangle\ \langle c,e\rangle$	$\langle a,c\rangle\ \langle b,c\rangle\ \langle c,e\rangle\ \langle f,d\rangle\ \langle g,f\rangle\ \langle g,d\rangle\ \langle g,e\rangle$
$t_4(q)$	$\langle f,d\rangle$	$\langle f,d\rangle\ \langle g,d\rangle\ \langle g,f\rangle$

Table 1: Results of query evaluation against data sets G_1 and G_2 of Figure 2 and 3 (queries are evaluated with the set semantics instead of the standard bag semantics).

t	q_R	$m(t)$	$\tilde{p}(t)$	$\tilde{r}(t)$	$p(t)$	$f(t)$	$r(t)$	D
t_1	q_R	0	1	0	1.	1.	1.	G_1
					1.	.67	.5	G_2
t_2	q_R	1	1	1	1.	1.	1.	G_1
					1.	1.	1.	G_2
t_3	q_R	0	0	1	1.	1.	1.	G_1
					.57	.73	1.	G_2
t_4	q_R	0	0	0	1.	.50	.33	G_1
					0.	0.	0.	G_2

Table 2: Transformation evaluation measures (gray cells illustrate the implications of Section 3).

$m(t)$ is computed as:

$$m(t) = \begin{cases} 1 & \text{if } t(q) \cong q_R \\ 0 & \text{otherwise} \end{cases}$$

The equality predicate ($\cong$) is usually not strict syntactic equality but may be equality modulo commutativity and variable renaming —this is the case when comparing $t_2(q)$ and q_R in Figure 1— or equality with respect to queries in a normal form.

In the second modality [9, 17, 16], an initial query q and the expected evaluation results R_D against a data source D are given, very often such that $R_D = eval(q_R,D)$ for some reference query q_R. R_D is then compared to the result of $eval(t(q), D)$. It is then possible to define classical measures such as precision ($p(t)$) and recall ($r(t)$) on this set of answers as:

$$p(t) = \frac{|eval(t(q), D) \cap R_D|}{|eval(t(q), D)|} \quad r(t) = \frac{|eval(t(q), D) \cap R_D|}{|R_D|}$$

the F-measure is computed in the usual way as the harmonic mean between precision and recall.

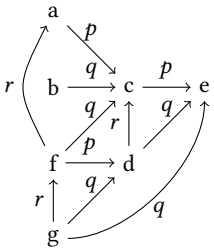
Usually evaluation is performed against a benchmark involving various such tests whose results are aggregated or averaged. Table 1 shows the results of evaluating queries q_R, t_1, t_2, t_3 and t_4 of Figure 1 against the data sets G_1 and G_2 of Figure 2 and 3. From this, the measures m, p, f and r reported on Table 2 are computed.

When the measure $m(t)$ is at 1, both precision and recall are at 100%. However, they can be at 100% with $m(t) = 0$.

2 PROBLEMS

Three problems may be identified with such evaluation measures.

(1) There is a gap between the two measures. In particular, it is possible to have a 100% correct result for precision and recall



$\langle a,p,c\rangle\ \langle b,q,c\rangle$
$\langle f,r,a\rangle\ \langle f,q,c\rangle$
$\langle c,p,e\rangle\ \langle f,p,d\rangle$
$\langle g,r,f\rangle\ \langle g,q,d\rangle$
$\langle g,q,e\rangle\ \langle d,q,e\rangle$
$\langle d,r,c\rangle$

Figure 2: Data set G_1.

with a query which is not the expected one. Table 2 shows that, though the syntactic comparison determines that t_1 and t_3 do not yield the reference query, the use of the data set G_1 does not allow to discriminate them. Conversely, though the use of precision and recall with respect to G_2 correctly determines that t_1 returns incomplete results and t_3 returns incorrect results, this information is not available by using the syntactic measure which grants them the same value, 0, as t_4.

(2) Precision and recall are highly dependent on the selected data set. This is obvious from Table 2 as G_1 finds the result of transformation t_1, t_2 and t_3 equally perfect, though G_2 identifies true negatives in t_1 and false positives in t_3. In addition, obtaining R_D, on large data sources may be a resource-consuming task, so benchmarks are not easy to build.

(3) The syntactic measure is very rough as it only tells if the query is the expected one or not. This does not allow to discriminate queries, though precision, recall and F-measure may permit to rank them on a more precise scale. The information that t_1 only provides correct answers and that t_3 always provides all answers, is not available by using the syntactic measure which grants them 0.

The apparent added-value of precision and recall is, in fact, very dependent on the data set. Indeed, as soon as its value is not *necessarily* 1. it is always possible to tune the data set to obtain a different value —through adding and suppressing triples that will generate more true positive or more false positive. Hence, one may argue that the finer grain provided by these measures is misleading and that there is actually only three values, for either precision and recall: it is necessarily 1. or not (necessarily 0. is only obtained by the empty query).

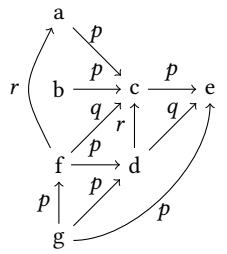
Checking what is necessary does not depend on the data set and is prone to static analysis. Hence, we suggest here that by using query containment instead of syntactic equality, it is possible to improve such methods without resorting to data sets.

3 CONTAINMENT-BASED TESTS

In order to better qualify the quality of transformations, a containment test can replace the equality test. A query q is contained in another q', noted $q \sqsubseteq q'$ if, for any RDF graph G, $eval(q, D) \subseteq eval(q', D)$ [2, 11]. The evaluation measures can then be defined as:

$$\tilde{p}(t) = \begin{cases} 1 & \text{if } t(q) \sqsubseteq q_R \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \tilde{r}(t) = \begin{cases} 1 & \text{if } t(q) \sqsupseteq q_R \\ 0 & \text{otherwise} \end{cases}$$

this has the advantage of benefiting from well-understood definitions that go beyond the implementation of the equality predicate. It also splits the measure in two different meaningful ways in the



$\langle a, p, c\rangle$ $\langle b, p, c\rangle$
$\langle f, r, a\rangle$ $\langle f, q, c\rangle$
$\langle c, p, e\rangle$ $\langle f, p, d\rangle$
$\langle g, p, f\rangle$ $\langle g, p, d\rangle$
$\langle g, p, e\rangle$ $\langle d, q, e\rangle$
$\langle d, r, c\rangle$

Figure 3: Data set G_2.

sense that it may be used to determine if one can expect fully correct or fully complete results. Hence, this is strictly more informative than the measure m.

In particular, if $R_D = eval(q_R, D)$, we know that:

$$t(q) \sqsubseteq q_R \Rightarrow eval(t(q), D) \subseteq eval(q_R, D)$$

or $\tilde{p}(t) = 1 \Rightarrow p(t) = 1.$, and

$$t(q) \sqsupseteq q_R \Rightarrow eval(t(q), D) \supseteq eval(q_R, D)$$

or $\tilde{r}(t) = 1 \Rightarrow r(t) = 1.$, and this for any D. Hence, this is true for all data sets.

It is still possible to compute a F-measure $\tilde{f}$ in the usual way following the intuition that precision denotes correction and recall denotes completeness [6]. However, this measure loses information as it will only have two values, 0 and 1 (actually, $\tilde{f}$ would be $\tilde{m}$ if $\tilde{m}$ were m computed with semantic equivalence).

This may also be combined as:

$$t(q) \equiv q_R \Rightarrow eval(t(q), D) = eval(q_R, D)$$

or $\tilde{m}(t) = 1 \Rightarrow f(t) = 1.$.

Such containment-based tests do not provide a fine-grained measure of the proportion of results that are missed or wrong. However, they can tell if none is missed and none is wrong and are valid for all data sources. This could be an important information if someone is interested in a transformation that does not miss answers (choose t_3) or does not return irrelevant answers (take t_1).

Hence, containment-based measures address Problem (1) by providing two boolean values instead of only one which are intermediate between the two types of measures. The approach retrieves $\tilde{m}$ (arguably better than m) by simply taking the conjunction of the two values. It also approximates precision and recall by being able to guarantee that precision or recall must be 100%.

The approach deals with Problem (2) by being independent from data sources, and Problem (3) by providing the more precise information about the query behaviour, preserving the link with precision and recall indicating that some answers may be incorrect or missing respectively.

It may also be used on a whole test bench instead of on a single test. In such a case the results can be averaged (which proportion of the containment tests are successful).

4 TRANSFORMATION COMPARISON

Although, the results may still seem rough by returning a pair of boolean, they may be used to compare the merits of different transformations together. Indeed, containment is a partial order relation that may be assessed to compare several transformations. Hence it is possible to position all the queries obtained by the transformations with respect to each others and to observe that a query is closer to the reference query than another (although this is not the case in the example given in Figure 4).

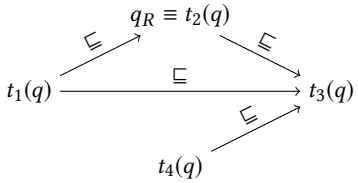


Figure 4: Transformation order induced by query containment on the queries of Figure 1.

This applies for one query. It is possible to account for a set of queries by replacing queries by transformations in each graph, preserving a single q_R node, replacing $\equiv$ by two $\sqsubseteq$ and $\sqsupseteq$ edges, and finally intersecting these graphs, i.e., their set of edges. If a transformation is always returning less answers than another or the reference query, they should be related by a $\sqsubseteq$ edge. It is also possible to take the union of these graphs and weight the edges depending of the number of graphs in which they appear.

5 ONTOLOGIES AND QUERY LANGUAGES

If the data sources are expressed in a particular schema or ontology O, it is possible to use containment modulo schema $\sqsubseteq_O$ [4, 3] in order to perform the test. This can still be achieved independently from any data set.

In addition, if the query evaluation mechanism can take such ontologies into account under a particular entailment regime reg, then the previous inequalities may be rendered as:

$$t(q) \sqsubseteq_O^{reg} q_R \Rightarrow eval^{reg}(t(q), D \cup O) \subseteq eval^{reg}(q_R, D \cup O)$$

Figure 5 shows a genuine SPARQL query $t_5(q)$ using the OWL property `rdfs:subPropertyOf`. If considered as simple SPARQL queries, there is no containment relation between $t_5(q)$ and q_R. However, if the OWL-entailment regime is used, $t_5(q) \equiv q_R$ because the triple pattern `?z q ?y` in q_R will match all triples which entail it, including those involving a subproperty of `q`.

This also applies for queries in SPARQL variants: the containment test must be defined with respect to the specific variant to

$t_5(q) =$

```
SELECT ?x ?y
WHERE {
  ?x p ?y .
  ?z ?r ?y .
  ?r sPO q .
}
```

$t_6(q) =$

```
SELECT ?x ?y
WHERE {
  ?x p ?y .
  ?z ?r ?y .
  ?r sPO* q .
}
```

Figure 5: Queries expressed with respect to ontological expressions (`sPO` stands for `rdfs:subPropertyOf`).

match the corresponding evaluation operation. This has already been done for PSPARQL [2], a variant of SPARQL 1.0 extended with property path expressions, now included in SPARQL 1.1.

Figure 5 shows a PSPARQL query $t_6(q)$ using a property path of an undefined number of `rdfs:subPropertyOf` properties (aiming at implementing the transitivity of this relation). If q_R is evaluated as a PSPARQL query, then $q_R \sqsubseteq t_6(q)$ because `?z q ?y` in q_R only matches those triples with the `q` property, but $t_6(q)$ will match these and in addition all those properties related to it by a chain of `rdfs:subPropertyOf` in the data.

6 ALIGNMENTS

When a query expressed in a vocabulary O is transformed in a query expressed in a vocabulary O', this way of performing evaluation is very convenient because it does not rely on any alignment: the transformation t may involve an alignment or not, but the queries $t(q)$ and q_R are both expressed in the vocabulary O' and thus can be tested for containment, modulo O' or not.

We had already defined measures for evaluating ontology alignments taking into account the idea that users may want precise answers to queries or on the contrary complete answers to queries [6]. This led to precision-oriented and recall-oriented evaluation measures depending on the direction of the transformation and the property (correctness or completeness) expected by the user. The measures were relaxing syntactic precision and recall by tolerating that they return more precise or more general classes than the corresponding ones. This does not directly translate to better precision and recall with respect to specific SPARQL queries due to the use of operations, such as MINUS, which would require inverting the orientation.

Containment-based transformation evaluation actually provides a new way to compare ontology alignments in the context of query evaluation. Indeed, if one considers that the transformation t is parameterised by an alignment A [5], then the evaluation of t_A is an evaluation of A. This procedure has been used in the ontology alignment evaluation campaigns using specific data sets on which the reference results were available [8]. Using containment-based evaluation, the same can be obtained from a set of reference queries without relying on data sets: several alignments, provided by different matchers, may be compared on this benchmark in their capability to transform queries. This also provides an alternative way to evaluate alignments in an oriented way with respect to their use in query processing since the result will be related to the necessary precision and recall.

7 CONCLUSION

Query transformation quality is usually evaluated through two types of measures with their qualities (easy to define/fine grained) and pitfalls (rough/data set dependent+difficult to define). We proposed an intermediate type of measures, grounded on query containment, that is as easy to define as the first, but more precise in its outcome and whose result is valid for any data set.

Such measures have the advantage that they allow to order different transformations on a solid basis instead of on a scale tightly depending on the chosen data set. They can also be defined using query subsumption [11] instead of containment. This would result in valid measures but the relations with the initial measures may not directly hold.

The approach can be defined for any type of containment relation relying on different query languages, ontologies or inference regimes. It can also be thought of as an alignment evaluation measure.

Query containment may be computationally expensive (well-defined patterns with OPTIONAL are Π_2^p-complete [11], many known procedures are in EXPTIME [4]) and in some cases undecidable (bag semantics or full SPARQL including SELECT and OPTIONAL [13]). However, when it is practicable it does only depend on the size of queries and not that of usually larger data sets. Hence we expect this to be acceptable for an evaluation task.

REFERENCES

[1] Meghyn Bienvenu, Stanislav Kikot, Roman Kontchakov, Vladimir Podolskii, Vladislav Ryzhikov, and Michael Zakharyaschev. 2017. The complexity of ontology-based data access with OWL2 QL and bounded treewidth queries. In *Proc. 36th Symposium on Principles of Database Systems (PODS), Chicago (US)*, 201–216.

[2] Melisachew Wudage Chekol, Jérôme Euzenat, Pierre Genevès, and Nabil Layaïda. 2011. PSPARQL query containment. In *Proc. 13th International Symposium on Database Programming Languages (DBPL), Seattle (US)*.

[3] Melisachew Wudage Chekol, Jérôme Euzenat, Pierre Genevès, and Nabil Layaïda. 2012. SPARQL query containment under RDFS entailment regime. In *Proc. 6th International Joint Conference on Automated Reasoning (IJCAR), Manchester (UK)*, 134–148.

[4] Melisachew Wudage Chekol, Jérôme Euzenat, Pierre Genevès, and Nabil Layaïda. 2012. SPARQL query containment under SHI axioms. In *Proc. 26th AAAI Conference on Artificial Intelligence, Toronto (CA)*, 10–16.

[5] Gianluca Correndo, Manuel Salvadores, Ian Millard, Hugh Glaser, and Nigel Shadbolt. 2010. SPARQL query rewriting for implementing data integration over linked data. In *Proc. EDBT/ICDT Workshops, Lausanne (CH)*.

[6] Marc Ehrig and Jérôme Euzenat. 2005. Relaxed precision and recall for ontology matching. In *Proc. K-CAP Workshop on Integrating Ontologies, Banff (CA)*, 25–32.

[7] Pascal Gillet, Cássia Trojahn dos Santos, Ollivier Haemmerlé, and Camille Pradel. 2013. Complex correspondences for query patterns rewriting. In *Proc. 8th International Workshop on Ontology Matching (OM), Sydney (AU)*, 49–60.

[8] Antoine Isaac, Shenghui Wang, Claus Zinn, Henk Matthezing, Lourens van der Meij, and Stefan Schlobach. 2009. Evaluating thesaurus alignments for semantic interoperability in the library domain. *IEEE Intelligent Systems*, 24, 2, 76–86.

[9] Prateek Jain, Peter Yeh, Kunal Verma, Cory Henson, and Amit Sheth. 2009. SPARQL query re-writing using partonomy based transformation rules. In *Proc. 3rd International Conference on Geospatial semantics (GeoS), Mexico (MX)*, 140–158.

[10] Roman Kontchakov, Mariano Rodriguez-Muro, and Michael Zakharyaschev. 2013. Ontology-based data access with databases: A short course. In *Proc. 9th International Reasoning Web Summer School, Mannheim (DE)*, 194–229.

[11] Andrés Letelier, Jorge Pérez, Reinhard Pichler, and Sebastian Skritek. 2013. Static analysis and optimization of semantic web queries. *ACM Transactions on Database Systems*, 38, 4, 25:1–25:45.

[12] Konstantinos Makris, Nektarios Gioldasis, Nikos Bikakis, and Stavros Christodoulakis. 2010. Ontology mapping and SPARQL rewriting for querying federated RDF data sources. In *Proc. On the Move to Meaningful Internet Systems (OTM), Hersonissos (GR)*, 1108–1117.

[13] Reinhard Pichler and Sebastian Skritek. 2014. Containment and equivalence of well-designed SPARQL. In *Proc. 33rd Symposium on Principles of Database Systems (PODS), Snowbird (UT US)*, 39–50.

[14] Eric Prud'hommeaux and Carlos Buil-Aranda. 2013. SPARQL 1.1 Federated Query. Recommendation. W3C.

[15] Bastian Quilitz and Ulf Leser. 2008. Querying distributed RDF data sources with SPARQL. In *Proc. 5th European Semantic Web Conference (ESWC), Tenerife (ES)*, 524–538.

[16] Élodie Thiéblin, Fabien Amarger, Ollivier Haemmerlé, Nathalie Hernandez, and Cássia Trojahn dos Santos. 2016. Rewriting SELECT SPARQL queries from 1: n complex correspondences. In *Proc. 11th International Workshop on Ontology Matching (OM), Kobe (JP)*, 49–60.

[17] Ana Isabel Torre Bastida, Jesús Bermúdez, and Arantza Illarramendi. 2015. Query approximation in the case of incompletely aligned datasets. In *Actas XX Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Santander (ES)*.