Evaluation of query transformations without data

Jérôme David  Jérôme Euzenat  Pierre Genevès  Nabil Layaïda

Laboratoire d’Informatique de Grenoble
Montbonnot, France
Jerome.Euzenat@inria.fr
May 29, 2018

Problem

- Query transformation \( t \), transforms a query \( q \) into another query \( t(q) \)
- Many applications: optimisation, mediation, federation...
- How to **experimentally** evaluate query transformations?
- Quality measures normalised in the \([0, 1]\) interval.

Example (Queries)

\[
q_R =
\begin{align*}
&\text{SELECT } ?x ?y \\
&\text{WHERE } \\
&\quad ?x \ p \ ?y . \\
&\quad ?z \ q \ ?y . \\
&\end{align*}
\]

\[
t_1(q) =
\begin{align*}
&\text{SELECT } ?x ?y \\
&\text{WHERE } \\
&\quad ?x \ p \ ?y . \\
&\quad ?z \ r \ ?x . \\
&\end{align*}
\]

\[
t_2(q) =
\begin{align*}
&\text{SELECT } ?v ?w \\
&\text{WHERE } \\
&\quad ?b \ q \ ?w . \\
&\quad ?v \ p \ ?w . \\
&\end{align*}
\]

\[
t_3(q) =
\begin{align*}
&\text{SELECT } ?x ?y \\
&\text{WHERE } \\
&\quad ?x \ p \ ?y . \\
&\quad ?y \ q \ ?z . \\
&\end{align*}
\]

\[
t_4(q) =
\begin{align*}
&\text{SELECT } ?x ?y \\
&\text{WHERE } \\
&\quad ?x \ p \ ?y . \\
&\end{align*}
\]

Current approaches: query equality

Given a reference query \( q_R \):

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

\( \cong \) is not strict syntactic equality.

- variables may be renamed;
- clauses may be reordered;
- may be computed through isomorphism;
- ...
Current approaches: precision and recall

Given a reference answer $R_D$ to query $q$ in dataset $D$:

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

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

If $q_R$ is available, then take $R_D = \text{eval}(q_R, D)$

Example (Data sets $G_1$)

Example (Data sets $G_2$)

Example (Results)
Limitations

1. There is a gap between the two measures.
2. Precision and recall are highly dependent on the selected data set.
3. The syntactic measure is very rough as it only tells if the query is the expected one or not.

Using query containment

Query containment: \( q \subseteq q' \iff \forall D, \text{eval}(q, D) \subseteq \text{eval}(q', D) \)

Given a reference query \( q_R \):

\[
\tilde{p}(t) = \begin{cases} 
1 & \text{if } t(q) \subseteq q_R \\
0 & \text{otherwise}
\end{cases}
\] \( \text{and } \tilde{r}(t) = \begin{cases} 
1 & \text{if } t(q) \supseteq q_R \\
0 & \text{otherwise}
\end{cases} \)

Consequences

If \( R_D = \text{eval}(q_R, D) \):

\[
t(q) \subseteq q_R \Rightarrow \text{eval}(t(q), D) \subseteq \text{eval}(q_R, D) \\
\text{or } \tilde{p}(t) = 1 \Rightarrow p(t) = 100\%
\]

\[
t(q) \supseteq q_R \Rightarrow \text{eval}(t(q), D) \supseteq \text{eval}(q_R, D) \\
\text{or } \tilde{r}(t) = 1 \Rightarrow r(t) = 100\%
\]

This, for any \( D \).

Example (Results)

<table>
<thead>
<tr>
<th>t</th>
<th>( q_R )</th>
<th>( m(t) )</th>
<th>( \tilde{p}(t) )</th>
<th>( p(t) )</th>
<th>( f(t) )</th>
<th>( r(t) )</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>( q_R )</td>
<td>0</td>
<td>1</td>
<td>1.</td>
<td>.57</td>
<td>.50</td>
<td>( G_1 )</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>( q_R )</td>
<td>1</td>
<td>1</td>
<td>1.</td>
<td>1.</td>
<td>1.</td>
<td>( G_1 )</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>( q_R )</td>
<td>0</td>
<td>0</td>
<td>.57</td>
<td>.73</td>
<td>.33</td>
<td>( G_1 )</td>
</tr>
<tr>
<td>( t_4 )</td>
<td>( q_R )</td>
<td>0</td>
<td>0</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>( G_2 )</td>
</tr>
</tbody>
</table>
Benefits

- Intermediate between query equality and precision/recall;
- Independent from evaluation data set;
- Slightly less rough than initial query equality;
- Provides indication about the loss of completeness/correctness;
- Well-grounded.

Transformation comparison

Query subsumption is an order relation: may be used for ordering transformations.

Ontologies 1

Containment modulo ontology:

\[ q \equiv_0 q' \iff \forall D, \text{eval}(q, D \cup O) \subseteq \text{eval}(q', D \cup O) \]
Ontologies 2 (entailment regime)

Containment under entailment regime:
\[ q \subseteq^{reg} q' \iff \forall D, \text{eval}^{reg}(q, D) \subseteq \text{eval}^{reg}(q', D) \]

\[ q = \begin{align*}
SELECT \ ?x \ ?y \\
WHERE \{ \\
\ ?x \ p \ ?y \\
\ ?z \ q \ ?y \\
\} 
\end{align*} \]

\[ t_5(q) = \begin{align*}
SELECT \ ?x \ ?y \\
WHERE \{ \\
\ ?x \ p \ ?y \\
\ ?z \ q \ ?y \\
\ ?r \ sP0 \ q \\
\} 
\end{align*} \]

Alignment

- Query subsumption compare queries expressed w.r.t. the same ontology.
- An ontology alignment \( A \) may be used, under certain condition, as a query transformation \( t_A \).
- Hence the same measures may be used for evaluating different alignments
- with respect to the query transformation task.

PSPARQL containment:
\[ q \subseteq^\text{PSPARQL} q' \iff \forall D, \text{eval}^\text{PSPARQL}(q, D) \subseteq \text{eval}^\text{PSPARQL}(q', D) \]

\[ q = \begin{align*}
SELECT \ ?x \ ?y \\
WHERE \{ \\
\ ?x \ p \ ?y \\
\ ?z \ q \ ?y \\
\} 
\end{align*} \]

\[ t_6(q) = \begin{align*}
SELECT \ ?x \ ?y \\
WHERE \{ \\
\ ?x \ p \ ?y \\
\ ?z \ r \ ?y \\
\ ?r \ sP0^* \ q \\
\} 
\end{align*} \]

Comparison

<table>
<thead>
<tr>
<th>reference measure</th>
<th>alignment</th>
<th>query containment</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec./Rec.</td>
<td>Semantic Prec./Rec.</td>
<td>Syntactic Containment based</td>
</tr>
<tr>
<td>transformation</td>
<td>not use driven</td>
<td>use driven</td>
<td>data dependent use driven</td>
</tr>
<tr>
<td></td>
<td>×</td>
<td>data independent use driven</td>
<td>data dependent use driven</td>
</tr>
</tbody>
</table>

David, Euzenat, Genevès, Layaïda: Evaluation of query transformations without data
Conclusion

- Well-founded approach to evaluate transformations;
- Adaptable to various contexts (languages, ontologies, alignments);
- Requires solid implementations of query containment solvers.

http://moex.inria.fr
http://tyrex.inria.fr

Firstname . Lastname @ inria . fr