Improving the Quality of the User Experience by Query Answer Modification

João Pedro V. Pinheiro[1], Marco A. Casanova[1], Elisa S. Menendez[2]

[1]Department of Informatics, PUC-Rio, Rio de Janeiro, RJ, Brazil

[2]Federal Institute of Education, Science and Technology of Sergipe, Aracaju, SE, Brazil

{jpinheiro,casanova}@inf.puc-rio.br, elisa.menendez@academico.ifs.edu.br

Introduction

- Question Answering (QA) systems combine techniques from multiple fields of computer science, among which:
  - Natural Language Processing (NLP)
  - Information Retrieval
  - Machine Learning (ML)
  - Semantic Web
- A QA system may be split into two parts: *Question*, which receives a user's input in natural language, transforms it into a SPARQL query and searches the RDF knowledge base; and *Answer*, which displays consistent results in a human-readable format to the user

Introduction

- This paper addresses the problem of *query answer modification* to improve the quality of the user's experience
- The study proposes a process that reorganizes the original query answer by applying heuristics to summarize the results and to create a user dialog that guides the presentation of the results
- Heuristics decide:
  - which properties are interesting to apply aggregations (group by operations)
  - which template questions best fit each case
  - if the answer is ready to be displayed to the user, or if the answer must be improved

Motivation

- *"Which artists were born on May 30th?"*

Artist	Genre	Birth Date	Death Date	Gender	Nationality
Goodman, Benny	Jazz	1909-05-30	1986-06-13	Male	American
Leonhardt, Gustav	Classical	1928-05-30	2012-01-16	Male	Dutch
Green, CeeLo	Pop	1974-05-30	-	Male	American
Biosphere	Electronic	1962-05-30	-	Male	Norwegian
Fredriksson, Marie	Pop	1958-05-30	2019-12-09	Female	Swedish
…	…	…	…	…	…
Banhart, Devendra	Folk	1981-05-30	-	Male	American

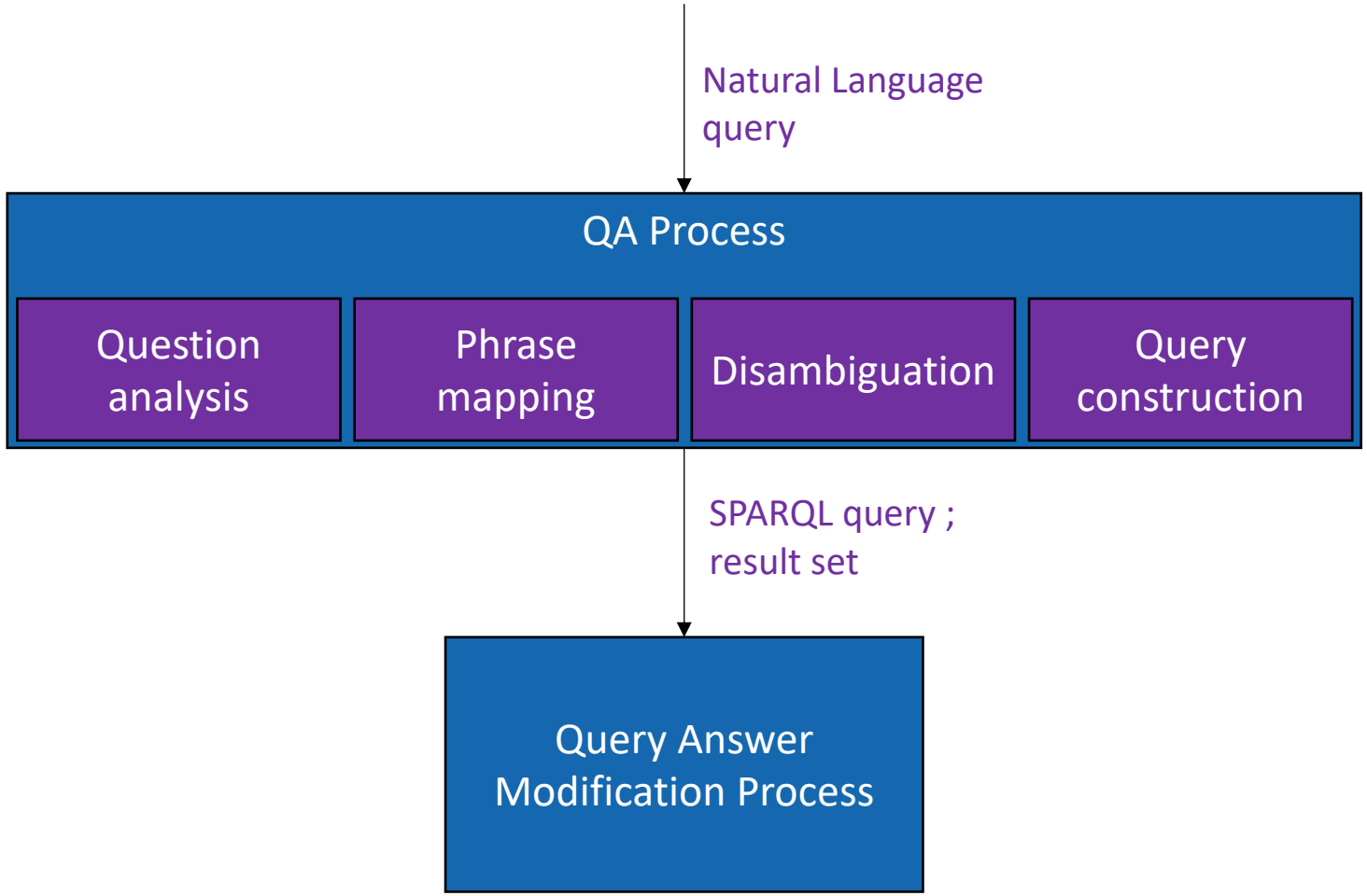
Motivation

- Instead of listing the results, the system may formulate questions to the user based on the prior result set, such as:
  - *“Do you want to list American or European artists?”*
  - *“Do you prefer Jazz, Pop, or Classical music?”*
  - *“Do you want to filter by active artists?”*

The query answer modification process

- Several studies addressed the problem of creating a question-answering (QA) interface to databases
- Usually, the proposed QA process has four steps:
  - Question Analysis
  - Phrase Mapping
  - Disambiguation
  - Query Construction
- The *query answer modification* process we propose starts after the query is executed

The query answer modification process



The Query Answer Modification Process

Transforming single-column into three-column result sets

- We noticed that most queries listed in the Question Answering over Linked Data (QALD[1]) challenges had single-column answers
- A simple approach for enriching the answers is to add to the instances returned their property values

```
prefix mo: <http://purl.org/ontology/mo/>
prefix foaf: <http://xmlns.com/foaf/0.1/>

select distinct ?artist
from <http://musicbrainz.org>
where {
      ?artist a mo:MusicArtist .
      ?artist dbo:birthDate ?date .
      filter(regex(?date, "5-30$", "i")) .
}
```


artist
mo:MusicArtist/1
mo:MusicArtist/2
mo:MusicArtist/3
mo:MusicArtist/4
mo:MusicArtist/5
mo:MusicArtist/6
mo:MusicArtist/7
mo:MusicArtist/8
mo:MusicArtist/9
mo:MusicArtist/10

[1]http://qald.aksw.org

The Query Answer Modification Process

Transforming single-column into three-column result sets

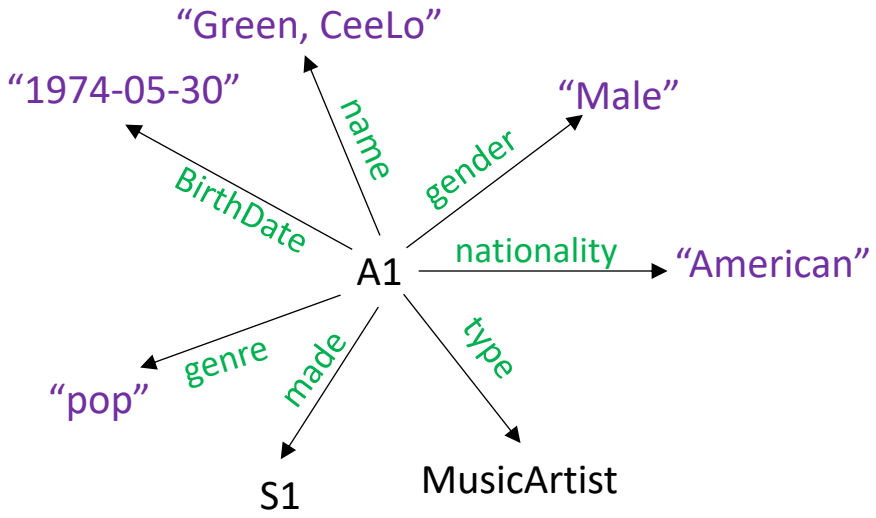
```
prefix mo: <http://purl.org/ontology/mo/>
prefix foaf: <http://xmlns.com/foaf/0.1/>

select distinct ?artist ?predicate ?object
where {
    {
        select distinct ?artist
        from <http://musicbrainz.org>
        where {
            ?artist a mo:MusicArtist .
            ?artist dbo:birthDate ?date .
            filter(regex(?date, "5-30$", "i")) .
        }
    } # prior query as subquery
    ?artist ?predicate ?object .
    filter(isLiteral(?object)) .
}
```

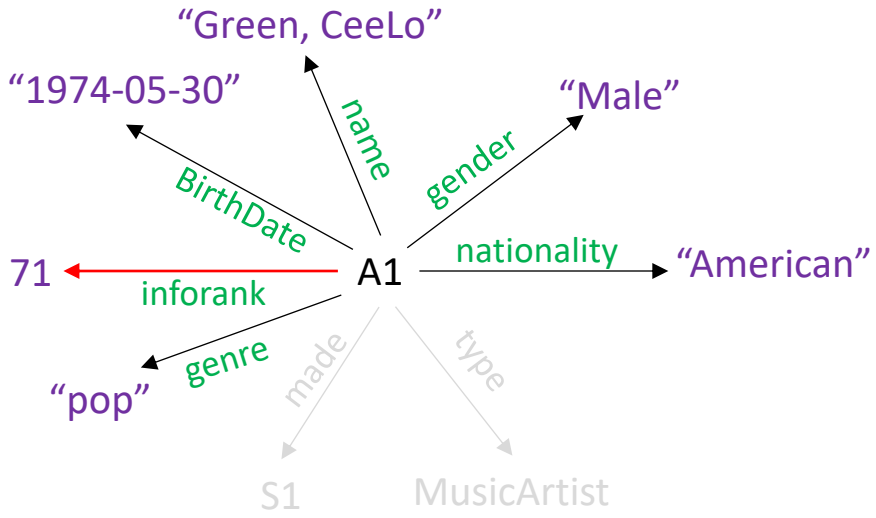
artist	predicate	object
mo:MusicArtist/1	foaf:name	"Green, CeeLo"
mo:MusicArtist/1	mo:genre	"pop"
mo:MusicArtist/1	dbo:BirthDate	"1974-05-30"
mo:MusicArtist/1	dbo:DeathDate	""
mo:MusicArtist/1	foaf:gender	"Male"
mo:MusicArtist/1	dbp:nationality	"American"
mo:MusicArtist/2	foaf:name	"Leonhardt, Gustav"
mo:MusicArtist/2	mo:genre	"Classical"
mo:MusicArtist/2	dbo:BirthDate	"1928-05-30"
mo:MusicArtist/2	dbo:DeathDate	"2012-01-16"
mo:MusicArtist/2	foaf:gender	"Male"
mo:MusicArtist/2	dbp:nationality	"Dutch"
mo:MusicArtist/3	foaf:name	"Goodman, Benny"
mo:MusicArtist/3	mo:genre	"Jazz"
mo:MusicArtist/3	dbo:BirthDate	"1909-05-30"
mo:MusicArtist/3	dbo:DeathDate	"1986-06-13"
mo:MusicArtist/3	foaf:gender	"Male"
mo:MusicArtist/3	dbp:nationality	"American"

The Query Answer Modification Process

Frequency analysis based on computed metadata

- There are two types of frequencies used in the system
  - *global frequency*: defined over full graph and computed once
  - *local frequency*: defined over sub-graph and computed at run time
  - both *frequencies* are computed over predicates pointing to literals only
- A parameterized threshold is used to filter predicates that are candidates to be used in a group by operation (default >= 0.4)





The Query Answer Modification Process

Computing the user dialog

- As mentioned in the previous step, an aggregation process is applied over the filtered predicates
  - these predicates are evaluated sorted by its *local frequency*
- A parameterized threshold related to the # aggregated values is used to decide if a template question must be used (default <= 5)
- These aggregated values are used as alternatives to fill the template question chosen
- Template questions are choice questions formulated in natural language and returned to the user

The Query Answer Modification Process

Computing the user dialog

- Using the same example as before, the filtered predicates are:
  - *foaf:gender*
    - aggregate values: *female* and *male*
  - *dbo:background*
    - aggregate values: *non_performing_personnel*, *non_vocal_instrumentalist*, and *solo_singer*
- Examples of template questions and their usage:
  - Do you prefer {value1}, or {value2} {subject}?
    - *Do you prefer female, or* male *artists?*
  - From {range1} to {range2}, how much/many do you want to {subject}?
  - Between {value1}, {value2}, and {value3}, which {subject} do you prefer?
    - *Between non_performing_personnel, non_vocal_instrumentalist, and solo_singer, which artists do you prefer?*
  - …

The Query Answer Modification Process

Ranking the final results

- This last step is only triggered when the process detects that there are no other candidate predicates for summarization
  - in other words, all predicates left have # aggregated values > 5
- The *inforank* metric is used to sort and filter relevant information in the final answer
  - a family of importance measures proposed in [Menendez et al. 2019]
  - helps our process prioritize the most relevant triples of the result set (subject)

Menendez, E. S., Casanova, M. A., Leme, L. A. P., and Boughanem, M. (2019). Novel node importance measures to improve keyword search over rdf graphs. In International Conference on Database and Expert Systems Applications, pages 143–158. Springer. DOI:https://doi.org/10.1007/978-3-030-27618-8_11

Expected Contributions

- Process based on heuristics that reorganizes a query answer to improve the user's experience
- Parameterized thresholds allow customized behavior for each scenario which
  - different knowledge bases may require different needs
- These thresholds are used to:
  - filter interesting properties to apply aggregations
  - select template questions that best fit in each case
- The proposed approach is easily pluggable to any QA system

Improving the Quality of the User Experience by Query Answer Modification

João Pedro V. Pinheiro[1], Marco A. Casanova[1], Elisa S. Menendez[2]

[1]Department of Informatics, PUC-Rio, Rio de Janeiro, RJ, Brazil

[2]Federal Institute of Education, Science and Technology of Sergipe, Aracaju, SE, Brazil

{jpinheiro,casanova}@inf.puc-rio.br, elisa.menendez@academico.ifs.edu.br