COMP624121 Querying Data on the Web

Lab Work SP

In relational databases, the content of every table must conform to an explicitly declared schema. In native XML databases, likewise, every document may be compelled to conform to an explicitly declared schema. In RDF stores, however, this is not the case.

The overall, high-level goal of this lab work is for you to gain an insight into some of the consequences ensuing from the lack of schema constraints over RDF stores. In particular, if one is asked to write a set of SPARQL queries to achieve a certain goal, some questions arise, such as:

- How does one go about understanding what data is there, and what can we ask of it?
- How much time is spent finding out?
- Does this make the task of writing queries harder?
- For example, is it more likely that you will spent longer debugging the queries before they work?

So, there needs to be an exploratory stage in which you, the query writer, must familiarize yourself with the opportunities (i.e., the subjects, predicates and objects) that are available in one or more RDF datasets. You can do this using SPARQL itself, of course, but is this effective? Is it efficient?

In this lab work, we will fix a target information content for you and you will aim to create the queries that aim to derive that content from one SPARQL endpoint.

Note that some of the desired information content will not be available (you will need to convince yourself of that and then explain why in your report). Note too that some of the desired information content may well be there but you cannot find (perhaps in the time available) a way of reaching it (e.g., some predicate may be missing and therefore linking/navigating to the desired content proves impossible or exceedingly hard).

In terms of information content your goals is to create a set of SPARQL queries over the **DBpedia** that, for a country in the relational version of the **Mondial** database, generates the same information (excluding the geographical aspects such as lakes, mountains, seas, rivers, etc., but including, if possible, provinces, countries, population, etc.) as is available in **Mondial**.

In other words, you will

1. start by studying again the relational version of Mondial
2. focussing on the non-geographical information that you can retrieve (basically using selection-projection-join-aggregation, as shown in the **mondial-abh.pdf** referential dependency diagram, then
3. setting yourself the target of obtaining that information from **DBpedia** using SPARQL queries.

Note that we will not be strict about minor variations (e.g., the names given to countries, the populations of a city, and similar cases), minor inconsistencies (e.g., the province to which a city belongs, and similar cases), and minor incompletenesses (e.g, one city in one is missing in the other, and similar cases).

Note also that there is a version of **Mondial** in RDF, but you will **not** use it. You will try and create, from **DBpedia** and using SPARQL, the information content of the relational **Mondial** restricted to what you can obtain by querying country and ignoring geographical information (i.e., information about geographical features and concepts).

However, creating the set of queries is not your main task. Moreover, the task for which most marks are available) is **not** to get the set of SPARQL queries absolutely right.

Your main task is to write a report about your experience of writing the set of SPARQL queries. In other words, the main goal, really, is for you to record, reflect and report on your experience of writing the SPARQL queries over **DBpedia**.

Here. what we mean to learn is, given the need for exploration of possibilities and opportunities (which, as we said, is one consequence of the schema-less nature of SPARQL/RDF) *how effective and how efficient you managed to be*.

By *how effective* we mean an answer to the question: "*To what extent did you succeed in writing a set of SPARQL queries that gathers from **DBpedia** the same information that is available through relational **Mondial**?*"

Note that we are not asking you to be 100% effective: you may not able to retrieve all the information, your set of SPARQL queries may be incomplete. You don't lose marks for that. You gain marks for explaining what the problem was that prevented you, be it a problem with **DBpedia**, be it a problem of complexity (e.g., you ran out of time), be it a problem of inadequacy of SPARQL, be it the lack of a schema, or yet something else.

Note also that, here, you don't need to worry about single tuples, single values, single pieces of information. We're only interested in questions such as: "Can we find the provinces of a country using SPARQL against **DBpedia**?" Note that we don't mean *all* the provinces of *all* the countries.

By *how efficient* we mean an answer to the following questions:

- *How much time did it take you to explore of the information content of the **DBpedia** using SPARQL queries that try and gauge what is available in that resource?*
- *How much time did it take you to discover how to write the queries in the set of SPARQL queries you have succeeded in writing?*

Again, note that we are not asking you to be extremely efficient: it may take you quite long to even begin to write the first SPARQL query (most probably, one that tells you which are the countries in **DBpedia**). You don't lose marks if it takes you very long. You gain marks for explaining what the problem was that prevented you, be it a problem with **DBpedia**, be it a problem of complexity (e.g., you ran out of time before the submission deadline), be it a problem of inadequacy of SPARQL, be it the lack of a schema, or yet something else.

Note also that, here, you don't need to worry about very detailed timekeeping. Just keep a timesheet of the amount of hours and minutes (down to 5 minutes accuracy) that you spend in this lab.

For example, you may have several rows saying something (slightly more specific perhaps than)

1) Exploring which predicates may be relevant: Monday 13:00-13:30
2) …
3) …
4) Exploring which predicates may be relevant: Monday 18:00-19:30

and then do some aggregation and commenting, e.g., something like

TASK	*TIME SPENT*	*COMMENTS*
…		
Exploring which predicates may be relevant	*2 hours*	*(a) DBpedia was down often, (b) It took me long to find a path from country to province*
…		

Please, don't take the hypothetical numbers above as any kind of hint: they're invented numbers for the sake of an example. Such tasks may take you much less time or much more time. Finding out the (rough, approximate) time is one of the goals in this lab work.

Task 1: Write a set of SPARQL queries as described above, compiling a timesheet as you do, and writing comments on the efficiency and effectiveness of your work on the task.

The queries will be typed, tested and executed in the landing page for the DBpedia SPARQL endpoint:

http://dbpedia.org/sparql

You will find it useful to also consult the DBpedia Ontology directly:

http://mappings.dbpedia.org/server/ontology/classes/

Task 2: Write a report that prints the set of queries produced in Task 1, and, using the timesheet and comments compiled during Task 1, assesses the impact that the schema-less nature of SPARQL/RDF had on the effectiveness and efficiency of your work on Task 1.

Marking

- Only Task 2 is awarded marks, but note that the outcome of Task 1 is crucial for Task 2. You should interpret this as, once more, we don't expect the set of SPARQL queries to be perfect: we expect you to describe the process by which you created that set.
- Task 2, as the whole of the lab, is worth 60 marks and contributes up to 6 marks to the final mark for the course unit.
- The marks will be awarded for the quality of the report on the effectiveness and efficiency of the task as described in the preamble above.

Tips

Using exploratory queries

By *exploratory query* is meant one that essentially is trying to explore (a sample of) what subjects, predicates and objects there are. For example, the query

```
SELECT DISTINCT ?Concept
WHERE {[] a ?Concept}
LIMIT 100
```

gives you a glimpse as to what URIs or literals occur in object position when the predicate is `rdf:type` (abbreviated as `a`). The `[]` denotes a blank node, and essentially says that you don't care about the instances of the concepts. (If the tick in 'Strict checking of void variables' is not immediately clear to you, explore and you will learn what its function is.)

One query that might get you started is the following:

```
SELECT DISTINCT ?C
WHERE {?C a dbo:Country}
```

Try it. You would expect this to be a list of countries, right? But while indeed resources representing countries are returned, e.g.

http://dbpedia.org/page/Brazil

we also get information that is not about a country per se, e.g.

http://dbpedia.org/page/Captaincies_of_Brazil

(These are just illustrations! You don't need to stop and understand the resources the links above point you to. It is just so that you begin to get in thinking mode!)

We expect that a significant part of your time (but proves us wrong!) will be spent exploring the content of the **DBpedia** this way.

Being aware of the namespace prefixes you can use

In the landing page, the top right hand corner has a link to the predefined Namespace Prefixes .You should explore this.

(You may also prefer to use, perhaps complementarily, the iSPARQL interface that is reachable by a link in the top right hand corner of the landing page too. You should explore it.)

Using results format to help exploration

Remember that if you use HTML as 'Results Format', you are given links that you can clink on and explore more. For example, the query above returns a table with HTML links. If you click on

http://dbpedia.org/ontology/Place

you will be taken to a description of the concept *Place* in the **DPpedia Ontology**. And this may tell you more about the data and how to use it to achieve your goal. There are other options than simple HTML. You should explore your options.

Using LIMIT

Be careful to use LIMIT otherwise you'll get a flood of results. On the other hand, you may need to increase LIMIT or remove it altogether in order to find what you're looking for.

Using execution timeout

Be careful to use a good value for 'Execution timeout'. Perhaps the default needs to be replaced by a larger value (e.g., 10 times larger, even 100 times larger).

About the timesheet and comments

The timesheet and comments (see above) are important but, please, do not go into a panic if the times are not are not extremely accurate, or if you forget to make a note of one among ten things you did, etc. It's crucial that your report reflects your own thoughts on whether the schema-less nature of SPARQL was (or was not) an impediment to your being effective (i.e., getting the queries right) or efficient (i.e., getting (some of) the queries in not overly long times).

About the open-ended nature of this lab work

Finally, keep in mind that the goal of this lab is not to get the queries right at all costs: the goal is for you to to record, reflect and report on your experience of writing SPARQL queries over **DBpedia** that aim to meet certain data requirements (viz., recreating the non-geographical information content about countries that you can obtain from the relational version of Mondial).