REPRESENTATION AND QUERIES

ANDRIANA GKANIATSOU

1. Dataset Translation

The datasets that we have are represented in various formats: RDF, excel(in terms of relational schema) and pdf with tables. We need to translate them into a format that will allow us to express the mismatches. Two options are (i) translate the datasets into RDF or RDFS, (ii) translate the datasets into predicate logic.

RDF/RDFS One advantage of this approach is that we already have the SEPA datasets in RDF. Also, it is a widely used representation, so, it would make sense assume that the datasets were originally represented in triples. This is shown better with an example. Consider the data entry shown in Table 1 and Table 2 (both come form the same dataset, but they were too big to fit in a single table). If we translate it into RDF then we have the following representation:

```
<flood = 3845>
<dfo:country> India </dfo:country>
<dfo:other> Nepal </dfo:other>
<dfo:detailedLocations> Uttar Pradesh low lying areas of Gonda, Balrampur,
Faizabad, Barabanki,Rajasthan </dfo:detailedLocations>
<dfo:began> 23-Jul-11 </dfo:began>
<dfo:ended>9-Aug-11 </dfo:ended>
```

Now, consider an incoming query asking for the date the flood ended, by providing the location and the date the flood started:

```
select ?x
where
?a dfo:country India
?a dfo: location Gonda
?a dfo: dateBegan 23-Jul-11
?a dfo: dateEnded  ?x
```

The above query, will not return any results because of the `dfo:location`, `dfo:dateBegan`, `dfo:dateEnded` and `Gonda` mismatches. This query can be more complicated, if for example the date is not presented in the Day-Month-Year format, but in the Month-Day-Year format. We can treat all these mismatches only as semantic mismatches *e.g.* the `Gonda` and `Gonda, Balrampur, Faizabad, Barabanki,Rajasthan`. To deal with structural mismatches, one solution would be to define different heuristics that can indicate for example, how many words an entity consists of *e.g.*, if there exists a capital letter within the word,

and this letter is not at the beginning of the word, then we can split it into further words, and reason about the mismatch.

RDFS is expressive enough to define classes, sub-classes, properties, sub-properties, range and domain. However, RDFS (and RDF) support only binary predicates which might be a drawback.

Register	Country	Other	Detailed Locations
3845	India	Nepal	Uttar Pradesh,low lying areas of Gonda,Balrampur,Faizabad,Barabanki,Rajasthan

TABLE 1. Some of the fields of the table from the Global Active Archive of Large Flood Events

Began	Ended
23-Jul-11	9-Aug-11

TABLE 2. Some of the fields of the table from the Global Active Archive of Large Flood Events

First Order Predicate Logic The relational model for database is based on first-order predicate logic, so translating the datasets into that format, especially those that are presented in excel (table) seems natural. Let us illustrate an example. Consider the Table 1. If we translate it in predicate logic, then we will have:

`dfo(Register, Country, Other, DetailedLocations, Began, Ended)`

and an instance of that relation would be:

`dfo(3845,india,nepal,[pradesh,low lying areas of Gonda, Balrampur, Faizabad, Barabanki,Rajasthan], 23-Jul-11, 9-Aug-11)`. Now, consider the previous query, in predicate logic:

`dfo(Id, india, Other, gonda, 23-Jul-11, End)`.

The only mismatch that we have is `gonda`. In this case, we can treat the list as a predicate and add the missing arguments.

2. QUERIES

We know present sets of possible queries we might accept, depending on the dataset. Global Active Archive of Large Flood Events. The dataset contains information about floods. Some possible queries are:

- Ask for a specific location.
- Ask for the main cause of the disaster.
- Ask the for the country of the disaster.
- Ask for the number of affected people.
- Ask for the severity of the disaster.
- Ask for coordinates of the affected area.
- Ask whether the disaster has ended, or it is still going on.

All the previous queries can be combined with additional info that the query provides. For example, a query might ask for a specific location providing the country and the start date of the disaster: flood(Id, UK, DetailedLocation, 13-August-2012, 14-August-2012).
SEPA. Incoming queries are concerned with finding water bodies that are possibly at risk of flooding. Some possible queries are:

- Ask for the water body description, *eg.*, river.
- Ask for the water body name.
- Ask for the water body coordinates.
- Ask for the water body degree of danger.
- Ask for the local authority that is responsible for that water body. item Ask for the catchment of the water body.

All these queries are based on some predefined information, for example the the location that we are interested in, or the coordinates *etc.* The same category of queries applies to the SPA (Special Protection Areas) dataset.
NHS. Incoming queries are concerned with the identification of emergency departments, close to the affected area. Some possible queries are:

- Ask for the name of a department.
- Ask for a 24h open department, and, or
- ask for an emergency department.
- Ask the area of the department.
- Ask the address of the department.

NAPTAN (Trasportation). . Incoming queries are concerned with the identification of public nodes of transportation in the affected by the disaster area. Some possible queries are:

- Ask for a train station/airport/coach station *etc.*
- Ask for the coordinates of the transportation node.
- Ask for the exact location of a transportation node.

2.1. **Query Structure.** An incoming query might:(i) Have less arguments than our representation. (ii) Have more arguments than our representation. (iii) Have arguments in different order. (iv) Have lists as arguments whereas, we have single arguments and *vice versa*. (v) Have a different predicate name. (vi) Have different constants than the ones we do.