Implementation So Far

Andriana Gkaniatsou

1 Implementation: New Functionalities

The system has two new functionalities :

SQL Compatible The system translates the incoming predicate into a SQL query, connects to the database and sends the query. The results of the query are written into an external file. If the query fails then the table schema is extracted.

SPARQL Diagnosis This approach is based on the RDF schema that has been previously extracted (in case the query fails). Main assumption is that we have full access to the dataset and we have the appropriate permissions to extract the schema.

2 SQL Functionality

The SQL translation subsystem consists of the translation system (same for SPARQL queries) and a bash file which is responsible for querying the database. The main steps that we followed are the following;

Data: predicateName(Arg$_1$, Arg$_2$, ..., Arg$_n$)
Result: SQL query, extract schema query
write SELECT;
while *ArgumentList not empty* **do**
    find all Variables;
    find all Constants;
end
write SELECT ;
while *Variables not empty* **do**
    write Variable[i];
    remove Variable[i] from Variables;
end
write FROM PredicateName;
if *Constants not empty* **then**
    while *Constants not empty* **do**
        find Type[i] for Constant[i];
        write Type[i] = Constant[i];
        remove Constant[i] from Constants;
    end
end

Algorithm 1: Translation Process

Data: SQL query, schema retrieval query
Result: answers; table schema
connect to database;
send SQL query;
if *SQL query succeeds* **then**
    do nothing;
else
    retrieve schema;
end

Algorithm 2: Run Query Process

If the querry fails, then the database server points out the reason that the query failed. An example output is the following:
`ERROR: column city does not exist`. However, if the query contains many errors, the server will produce an error message only for one of the errors that might exists. This can be overcome, by repairing each time the error the database points, until no error message it produced. This approach might need some natural language analysis (?) or some user input on the error that the database indicated (message is difficult to parse automatically unless we have a standard pattern).

3 Diagnosis

The following approach is entirely based on the dataset schema[1] We assume that we have permission rights to the datasets, thus, we extract the schema of the dataset. Then we search for all the terms of our query that do not exist within the dataset schema. We consider these terms as the ones that need repair. We then split all terms in out query into multiple words (if possible). We apply the same procedure for all terms in the schema. Then we search for terms in the schema that contain word(s) of our query. In more detail the steps that we followed are the following:

```
Data: incoming query list, schema
Result: possible matches
foreach  query term j QueryTerm do
  | split term into words;
  | store them into QueryWords[j] list
end
foreach schema term i do
  | split term into words;
  | store them into SchemaWords[i] list;
end
foreach query term j do
  | foreach schema term i do
  |   | if j == i then
  |   |   | add j to NoErrorTerm list;
  |   | end
  | end
end
ErrorTerm is QueryTerm - NoErrorTerm;
foreach query term j in QueryTerm do
  | foreach word k in QueryWord do
  |   | find all query words such that;
  |   | foreach schema term i do
  |   |   | foreach schema word m in SchemaWord do
  |   |   |   | k==m;
  |   |   |   | add i to PossibleMatches;
  |   |   | end
  |   | end
  | end
end
```

Algorithm 3: Schema Based Diagnosis

The outcome of this procedure is lists of possible matches. For example, if the error

[1]It has only been tested for RDF datasets

term is riverArea, then such lists will be [sub, basin, district] [2] and [river, name]. A possible approach would be to use o vocabulary, such as Wordnet, to find the best candidate for the repair. For instance, in the previous example if we consider area as synonym of district, then the best candidate is [sub, basin, district].

[2]Original term was subBasinDistrict