Towards Question Answering on Statistical Linked Data

Konrad Höffner
University of Leipzig
Department of Computer Science
Augustusplatz 10, 04103 Leipzig
hoeffner@informatik.uni-leipzig.de

Jens Lehmann
University of Leipzig
Department of Computer Science
Augustusplatz 10, 04103 Leipzig
lehmann@informatik.uni-leipzig.de

ABSTRACT

As an increasing amount of statistical data is published as linked data, intuitive ways of satisfying information needs and getting new insights out of the data become more and more important. Question answering systems provide such an intuitive interface by translating natural language queries into SPARQL, which is the native query language of RDF knowledge bases. Statistical data, however, is structurally very different from other data and cannot be queried using existing approaches. We analyze the particularities of statistical data represented in the RDF Data Cube Vocabulary in relation to question answering and sketch a new question answering algorithm on statistical data. In order to estimate typical user questions, a statistical question corpus is compiled and its elements are categorized.

1. INTRODUCTION

As an increasing amount of statistical data is published as linked data, intuitive ways of satisfying information needs and obtaining new insights out of statistical data become increasingly important. Currently, aggregates and visualizations usually have to be manually configured and are thus not available for all datasets. Additionally, they represent only one view of the data which can be intentionally selected to support a certain agenda or political view. Systems for Semantic Question Answering (SQA) provide an intuitive interface to linked data by translating natural language queries into SPARQL, which is the native query language of RDF knowledge bases. This empowers non-expert users to draw their own, unbiased conclusions. Statistical data is, however, queried differently and thus needs adapted vocabularies and querying methods. For example, in traditional SQA, users typically ask about entities with certain properties, such as "Who is the wife of Barack Obama?" On statistical data, users typically ask about measurement values such as budgets for a certain purpose or about entities with certain values or value ranges, such as "Which were the top 10 funded research institutions in Europe in 2013?" This motivates our contribution of compiling statistical user questions to generate a domain independent statistical vocabulary which we analyze to extract commonly used phrases and the information need they represent. The structure of statistical data is also very different from other data and can not be readily queried using existing SQA algorithms. One of the reasons is that statistical observations can have many dimensions and their values alone are meaningless without the proper context and further processing. For instance, "What is the 2014 public transportation budget of Frankfurt?" is a typical question that is directly expressed in the following RDF triple pattern: :Frankfurt :publicTransportBudget2014Eur "5.6E7"^^xsd:decimal. Modelling statistical data in this way is not practical, as it does not expose the full structure of the data, requires an immense amount of manual modelling and makes it nearly impossible to select certain facts based on restrictions, such as "all spendings in 2014". In contrast to a general knowledge base such as DBpedia, statistical knowledge bases usually adhere to a strict meta model.

Section 2 presents this meta model, compares it with typical RDF modelling and shows how this results in a different type and dimensionality of answers. Section 3 shows existing SQA approaches and their workflow and differentiates, which of their parts can be reused and which ones have to be adapted for statistical data. Section 4 analyses the question corpus, especially in regards to expected answer types and direct and indirect references to aggregate functions. Section 5 proposes an algorithm that extends existing work with handling of RDF Data Cubes (RDCs) by adapting named-entity recognition (NER), query formulation and answer formulation. Section 6 concludes with our plan to extend the corpus and implement as well as evaluate the algorithm presented here.

2. THE RDF DATA CUBE FORMAT

Statistical data can be expressed using a *data cube* (also *OLAP cube* or *hypercube*) which is a multi-dimensional dataset in which statistical observations are central. Each cell corresponds to an observation that contains measurements. The RDF Data Cube Vocabulary [1] allows expressing data cubes in linked data. The principal unit is the dataset. Each dataset consists of a model and observations. The model contains component properties, which are either dimensions, measures or attributes, whose range is defined either using data types (e.g. `xsd:dateTime`) or code lists. An observations contains exactly one value for each dimension and measure. Listing 1 shows, how the information satisfying the transport example can be modelled using an RDC.

SEM '14 September 04 - 05 2014, Leipzig, AA, Germany

ACM 978-1-4503-2927-9/14/09 ...$15.00.
http://dx.doi.org/10.1145/2660517.2660521.

Listing 1: an RDC observation

```
:obs  rdf:type       qb:Observation;
      qb:DataSet     :CityBudget;
      :city          :Frankfurt;
      :refYear       "2014"^^xsd:gYear;
      :category      :publicTransport.
      :subCategory   :Bus.
      :amount        "3.6E7";
      :currency  dbpedia:Euro.
```

This observation contains four dimensions: *city*, *refYear*, *category* and *subCategory* as well as one measure, the *amount*. Three of the dimension values are fixed by the question, while one, *subCategory*, is unspecified.

Listing 2: another RDC observation

```
:obs  rdf:type       qb:Observation;
      qb:DataSet     :CityBudget;
      :city          :Frankfurt;
      :refYear       "2014"^^xsd:gYear;
      :category      :publicTransport.
      :subCategory   :Tram.
      :amount        "2.2E7";
      :currency  dbpedia:Euro.
```

Listing 2 shows another observation that satisfies the same question, as it contains the same value for all fixed dimensions and only varies in the value of *subCategory*. Because no two observations in a data cube may have the same value for all dimensions, only questions with unspecified (free) dimensions can have multiple answers. Traditional SQA approaches, such as AutoSPARQL TBSL [12] mostly treat and display multiple answers as sets or lists. There are also facetted browsing approaches such as Broccoli[1] [3] and Facete [10] which use property values to restrict and navigate sets of instances and to find similar ones based on commons values, for example other US presidents when selecting Barack Obama. While the navigation in facetted browsing is multidimensional, the visualization is still a list. Statistical data offers this possiblity as well but it also allows for more elaborate visualizations, depending on the dimensionality d of the data selected by the question, where $d \leq |\text{free dimensions}| + |\text{measures}| - 1$.[2] For example, Listings 1 and 2 contain one measure and one free dimension with different values, $d = 1$, so a pie chart may be used, while with $d = 2$ e.g. a scatter chart is possible. In addition to the traditional list-display, the measurement values of a set of observations can also be aggregated into single value, such as the total amount or arithmetic mean.

3. RELATED WORK

To the best of our knowledge, there is no existing work using question answering on statistical linked data.[3] Question answering on linked data in general however is an active area of research with several different benchmark competitions which help evaluate and compare the multitude of QA approaches such as the general QALD [4] challenges or the specialized BioASQ [11] competition for biomedical data.

IBM Watson [8] is a massively parallel question answering system that integrates its responses among many different sources, including DBpedia [2], Wikipedia and WordNet. Instead of the standard approach, candidates are generated first using multiple interpretations and are then selected based on a combination of scores.

TBSL [12] combines a domain independent and a domain dependent lexicon, which already exists for knowledge bases such as DBpedia and can be adapted to others. Figure 1 shows a typical SQA pipeline with extensions as used by TBSL, which we plan to adopt as a base. First, the users supplies a natural language question or statement. Next, a tagger identifies parts of speech such as nouns and verbs and two lexicons are used to parse the question. The parse structure along with the identified entities is used to contruct a semantic representation, which is then transformed to an incomplete SPARQL query with placeholders for the entities. Next, entities are identified. For resources and classes, this is done using a Apache Solr index, which is much faster than doing a reverse label lookup on a SPARQL endpoint and allows fuzzy matching to bridge the lexical gap. Expressions of properties vary wildly, however, and are thus not matched well enough using the index alone. The BOA [6] pattern library tackles this problem by providing various phrases commonly used to refer to a certain property. The entities are then entered in the placeholders of templates, forming full SPARQL queries that are scored before the one with the highest score is executed. The answer is presented to the user as a list whose items can be marked as correct or incorrect in order to improve the SPARQL query using the AutoSPARQL [7] algorithm. Treo [5] is a different approach that performs entity recognition and disambiguation using Wikipedia based semantic relatednes and spreading activation. It takes advantage of the context of a word in a sentence and the assumption, that all entities in a sentence are somehow related and thus similar concepts have a higher probability of being correctly identified.

Other approaches for querying linked data include facetted browsing approaches such as Broccoli [3] and Facete [10], which allow intuitive navigation from a certain starting resource of list of resources using property values.

4. QUESTION CORPUS

1	What was the average student grade per semester in year 2010?
2	How much money, does Leipzig and Dresden spend on child care in relation to the birth rate in comparison to the average in Saxony.
3	What is the average monthly income of a German citizen?
4	How much money was invested to fight bicycle thefts in Leipzig?

Table 1: The first questions of the first five survey participants.

The corpus consists of 50 questions that are individually provided by six researchers in the field who were asked to provide questions that were typical for their statistical information needs with a focus on governement financial spending and budget data. Table 2 presents the different ques-

[1] `http://broccoli.informatik.uni-freiburg.de`

[2] The $\leq$ sign occurs instead of $=$ because all values in a set of observations for a free dimension may be equal.

[3] Which is different from statistical question answering, which uses statistical methods for problems such as named-entity recognition and word-sense disambiguation.

tion words which give information about the expected answer type (EAT). Some SQA approaches, such as IBM Watson [8], use the EAT to filter out wrong answer candidates and thus improve the precision of the answer. On the corpus however, the information gained using EATs is small, as 47[4] of the 50 questions can refer to measurement values (see Table 2). Additionally, the dimensionality of the answer needs to be taken into account which can only be known after executing the query in full or, to get an upper bound, determining the dataset and relating its model with the dimensions which are fixed in the query. As such, additional steps need to be taken in order to determine the expected presentation type (EPT) of the answer. The most common one is that of a single (0-dimensional) observation's value for a certain measurement, which can be presented as a simple text sentence, such as "The Frankfurt city budget of bus transportation in 2014 is 36 million euro". When the result contains multiple answers, a listing of the values is the traditional solution but in data cubes it is not an intuitive answer for the user, as the number of results can be very large and the results need be to either aggregated or visualized. Table 3 shows the frequencies of the EPTs in the corpus. While visualisation is the most common type, it is explicitly mentioned only in two of the 19 cases ("display it on a map" and "what does [...] look like"). In 9 of the 12 questions where single values are expected, an aggregate is necessary in order to generate this value (see Table 4). In 5 cases, the aggregate type is explicitly mentioned ("average", "total", "the biggest") while in 4 questions it has to be inferred ("How many kids are born in Berlin on a single day?"). If multiple aggregations are possible (e.g. arithmetic mean or the median), they can all be presented to the user at the same time. Users often don't mention the name of a measure but instead its unit, e.g. "How much money was invested to fight bicycle thefts in Leipzig?". In this case, the attributes describing the unit (see Listing 1) will be used to select the correct measure.

question word	expected answer type	f
how much	quantity (uncountable)	19
what	any	12
how many	quantity (countable)	11
which	equivalent to "what"	3
where	location or purpose	2
how is	any	1
relate	comparison or visualization	1
none (statement)	any	1
total		50

Table 2: Frequency of question words in the corpus

5. ALGORITHM OUTLINE

SQA is typically done sequentially in a pipeline architecture. In order to reduce the complexity of the task, we plan to reuse the TBSL algorithm and modify the following parts:

Preprocessing.

[4]"relate" and "where" are the only question words which cannot relate to measurement values

expected presentation type	f
visualization	19
single measurement value	12
percentage value	4
entity or set of entities	4
correlation statement	1
unknown	10
total	50

Table 3: Expected presentation types

phrase	aggregate	f
average	arith. mean	3
total	sum	1
(on) a <timespan>	arith. mean	2
how much does ... a <class> <measure>	arith. mean	2
the biggest	max	1
total		9

Table 4: References to aggregates in the corpus

While the NLP parsing can be kept, explicit references to aggregates are detected using a manually created mapping based on Table 4, removed from the sentence and later used in the answer presentation step.

Domain Dependent Lexicon.

Using the extended corpus, we plan to add common statistical question patterns to the domain independent lexicon.

SPARQL Template Generation.

TBSL creates SPARQL templates such as the following:

```
SELECT ?y WHERE {<RESOURCE> <PROPERTY> ?y.}
```

To conform with the RDC meta model, the template generation needs to generate templates such as the following:

```
SELECT ?o WHERE
{?o a qb:Observation.
 ?o qb:dataSet ?d.
```

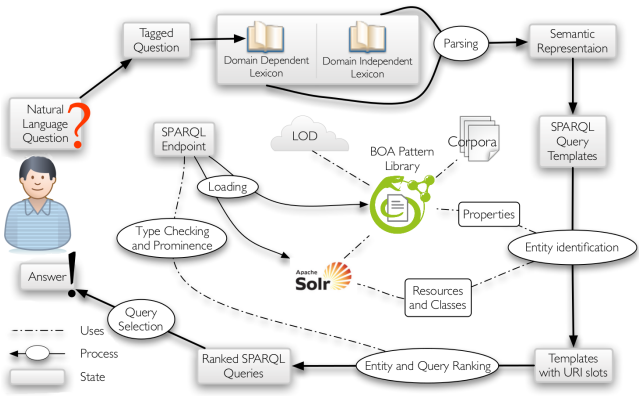

Figure 1: TBSL query generator overview (source: [12])

```
?d qb:structure ?model.
?model qb:component <DIMENSION>.}
```

Entity and Query Ranking.

In the existing algorithm, queries are ranked using similarity and prominence scores before execution. RDCs offer a greater homogenity inside a dataset, as all observations provide exactly one value for each dimension and measure. As such, there is a smaller amount of potential dimensions and measures and thus less expected errors because of misleading textual similarities. Because RDCs are organized in datasets, those datasets that contain the queried information need to be identified. In order to prevent multiple aggregation of the same datum, only one dataset is chosen for each query. In the simplest case, a question explicitly mentions a dataset. The dataset can then be found by matching the refering phrase, for example "hospitals in 2014" to the dataset label and description "Hospital spending in 2014". Alternatively, all dataset models need to be searched for the dimension entries, first the code lists of the coded properties, for "hospital" in the example, and then the values of date-time properties for the datum of "2014". However, references to dimensions can often not be differentiated from references to datasets before execution. Because of the sparsity of multi-dimensional datasets, only a small part of the possible slot assignements is expected to be non-empty. As such, we plan to split the ranking into two steps, where several of the highest ranking queries will be executed and the final ranking is based on the returned answers.

Answer Presentation.

Instead of displaying lists, results are displayed using either an answer sentence in case of single value, as visualizations using the CubeViz [9] tool or as an aggregate.

6. FUTURE WORK

This article lays the basis for our proposed system, which we plan to implement and evaluate in later work.

Extending the corpus.

While the 50-question corpus allows to draw basic conclusions about typical questions, a larger corpus provides more fine-grained and confident information, more rarely used patterns and allows comparisons by different user groups.

Implementation and Evaluation of the Algorithm.

We plan to integrate the proposed algorithm in the existing TBSL algorithm because it uses a domain independent lexicon which can be adapted to RDCs as well as domain dependent additions, which can be reused. Its property matching has a high recall through the usage of the BOA [6] framework, which is important as the phrases which reference dimensions and measures in RDCs are different to their text-based descriptions in the corpus in many cases.

Implementing a benchmark.

In order to evaluate algorithms on statistical data, we plan to adapt the extended corpus to fit the government budget and spending datasets of LinkedSpending[5].

[5] `http://linkedspending.aksw.org`

References

[1] The RDF data cube vocabulary. Technical report, W3C, 2013.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *Proc. of the 6th International Semantic Web Conference (ISWC)*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2008.

[3] H. Bast, F. Bäurle, B. Buchhold, and E. Haussmann. Broccoli: Semantic full-text search at your fingertips. *arXiv preprint arXiv:1207.2615*, 2012.

[4] P. Cimiano, V. Lopez, C. Unger, E. Cabrio, A.-C. Ngonga Ngomo, and S. Walter. Multilingual question answering over linked data (qald-3): Lab overview. In P. Forner, H. Müller, R. Paredes, P. Rosso, and B. Stein, editors, *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, volume 8138 of *Lecture Notes in Computer Science*, pages 321–332. Springer Berlin Heidelberg, 2013.

[5] A. Freitas, J. G. Oliveira, E. Curry, S. O'Riain, and J. C. P. da Silva. Treo: Combining entity-search, spreading activation and semantic relatedness for querying linked data. In *Proc. of QALD-1 at ESWC 2011*, 2011.

[6] D. Gerber and A.-C. N. Ngomo. Extracting multilingual natural-language patterns for rdf predicates. In *Knowledge Engineering and Knowledge Management*, pages 87–96. Springer, 2012.

[7] J. Lehmann and L. Bühmann. Autosparql: Let users query your knowledge base. In *Proc. of ESWC 2011*, 2011.

[8] J. W. Murdock, A. Kalyanpur, C. Welty, J. Fan, D. A. Ferrucci, D. Gondek, L. Zhang, and H. Kanayama. Typing candidate answers using type coercion. *IBM Journal of Research and Development*, 56(3.4):7–1, 2012.

[9] P. E. Salas, F. M. D. Mota, K. Breitman, M. A. Casanova, M. Martin, and S. Auer. Publishing statistical data on the web. *International Journal of Semantic Computing*, 06(04):373–388, 2012.

[10] C. Stadler, M. Martin, and S. Auer. Exploring the web of spatial data with facete. In *Proc. of WWW*, pages 175–178. Int. WWW Conf. Steering Committee, 2014.

[11] G. Tsatsaronis, M. Schroeder, G. Paliouras, Y. Almirantis, I. Androutsopoulos, E. Gaussier, P. Gallinari, T. Artieres, M. R. Alvers, M. Zschunke, et al. BioASQ: A challenge on large-scale biomedical semantic indexing and question answering. In *2012 AAAI Fall Symposium Series*, 2012.

[12] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over RDF data. In *Proc. of WWW*, pages 639–648, 2012.