CubeQA

Question Answering on RDF Data Cubes

Konrad Höffner, Jens Lehmann, Ricardo Usbeck

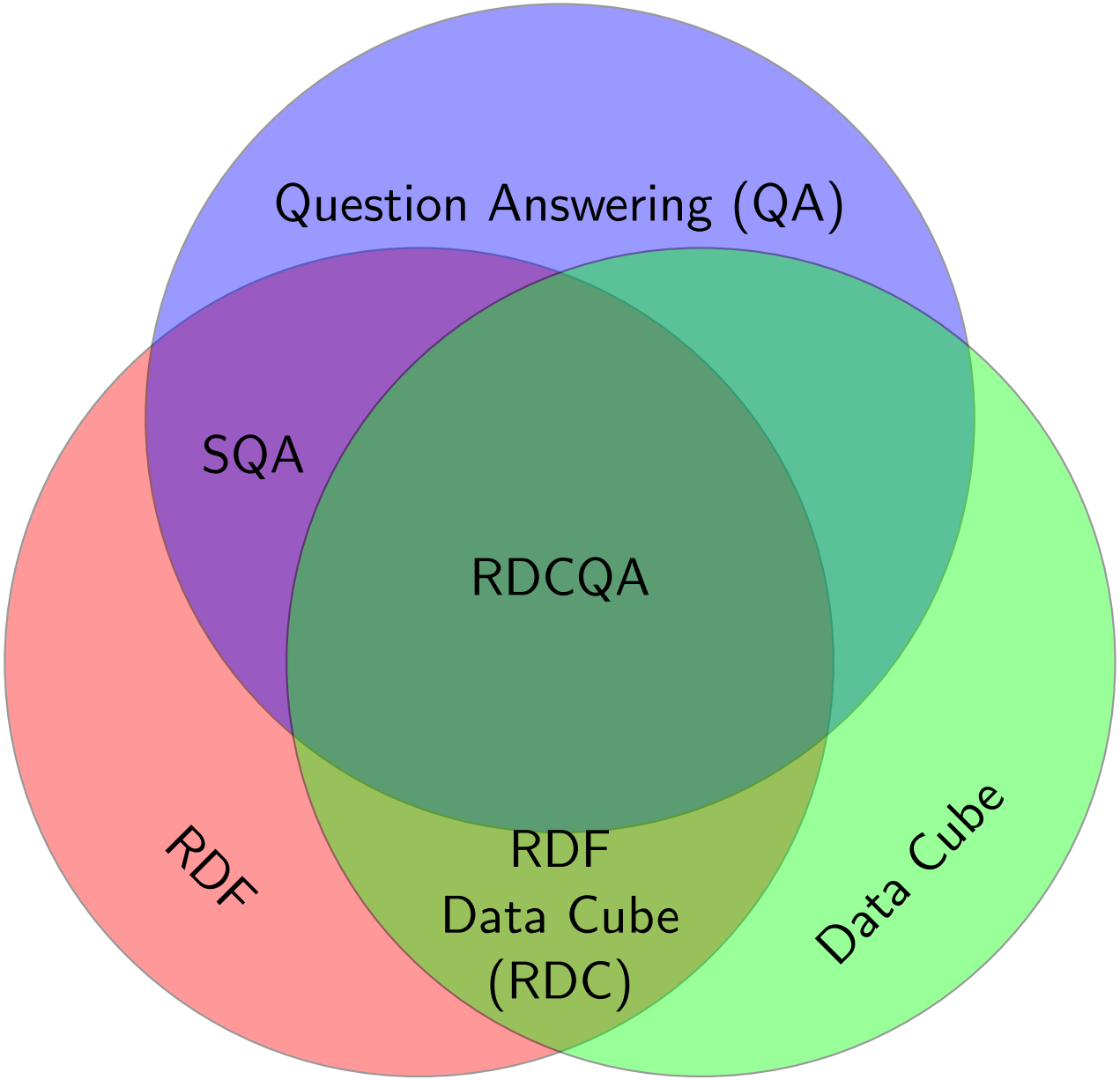
University of Leipzig, AKSW/MOLE, PhD Student

2016-10-20

Motivation

- large amounts of RDF Data Cubes (RDCs)
- domains i.e. finance, medicine, demographics
- multidimensional data opaque to the end user
- reliance on predefined visualizations
  - bias
  - coverage
  - less new insights

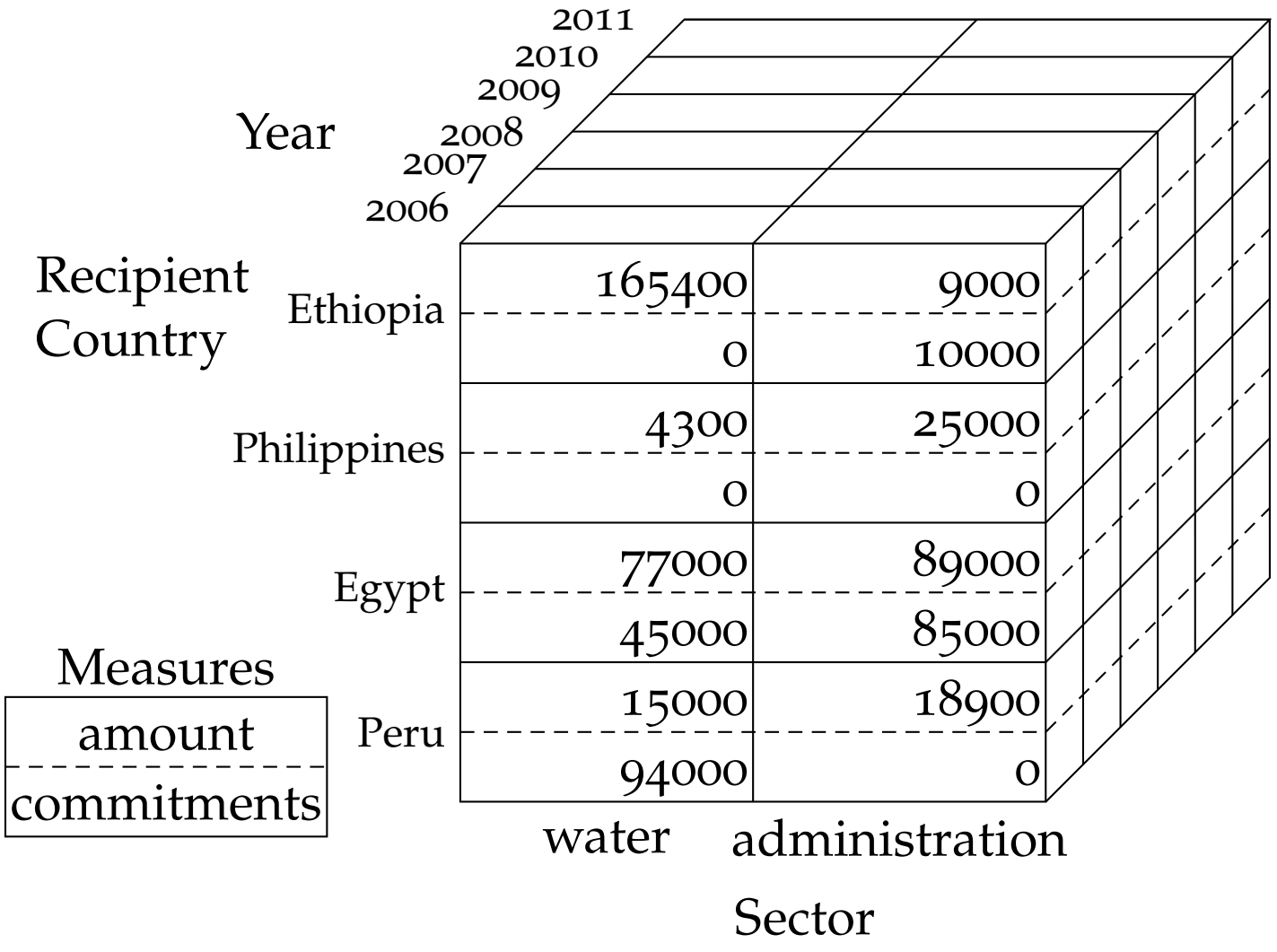
CubeQA—The first RDCQA system



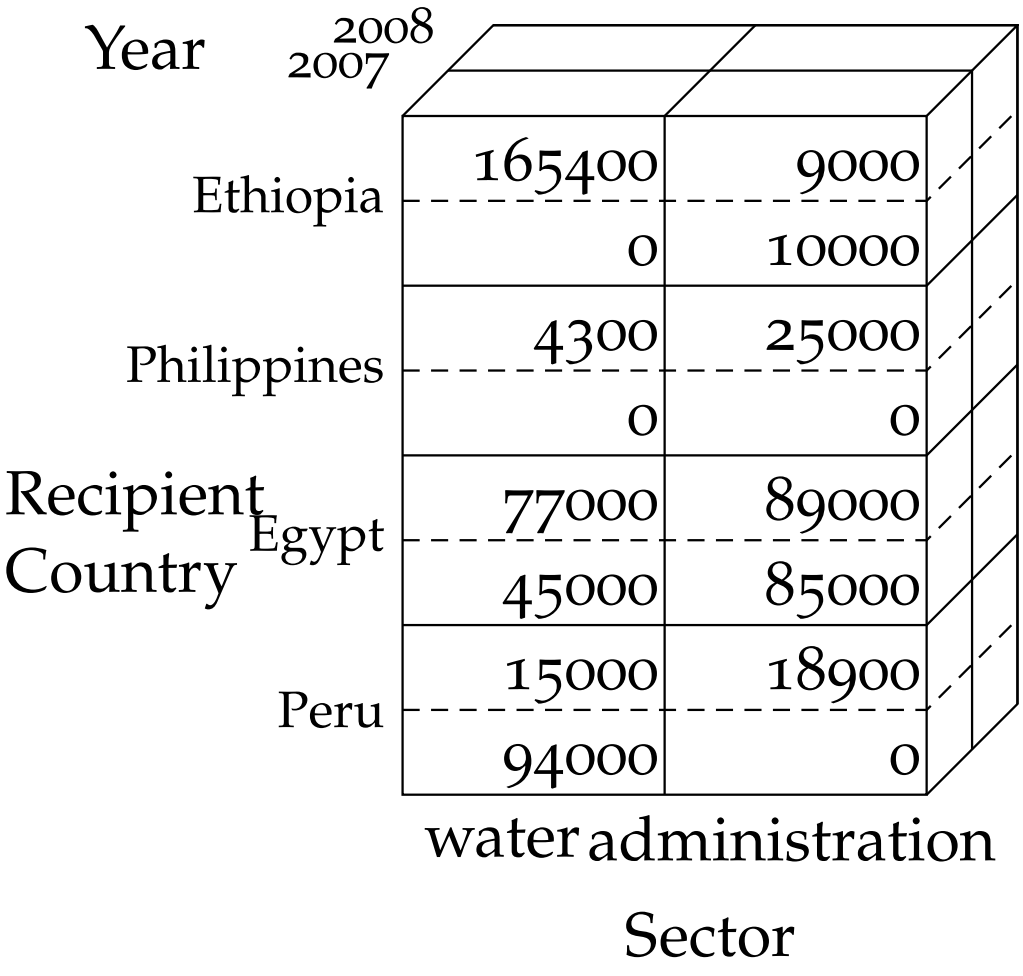
Example Question

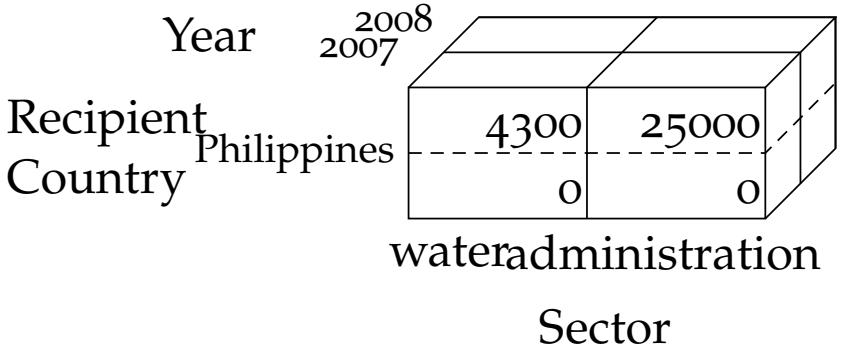
How much did the Philippines receive in the years of 2007 to 2008?

Data Cube Model



Datacube Operations: Dice and Slice





Corpus

- Motivation: optimize for common questions
- participants provided questions with multidimensional information needs
- 50 questions with no domain restriction

Excerpt of Corpus

- How much money, does Leipzig and Dresden spend on child care in relation to the birth rate in comparison to the average in Saxony.
- What is the average monthly income of a German citizen?
- How much money was invested to fight bicycle thefts in Leipzig?
- How many citizens live in a certain area?
- How much does Germany spend on research a year?

Corpus Properties

restriction	dimension value	29
	dimension value range	5
	measure value range	2
	top k measure value	5
	top k dimension value	1
expected answer type	measure value	14
	measure value aggregate	10
	dimension count	2
	dimension value	7
referenced	measure name	30
	measure unit	2
	dimension name	3

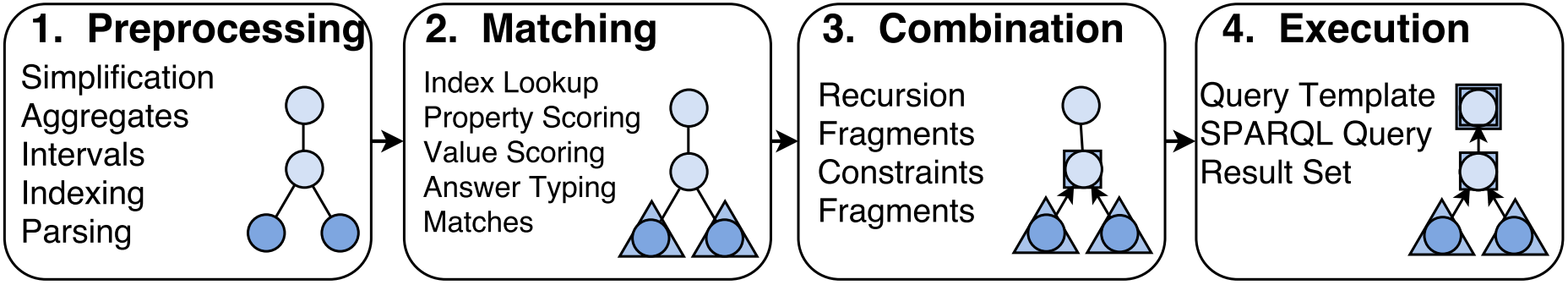
QALD6T3 Benchmark

- evaluate algorithm
- promote RDCQA
- clean corpus
- rewrite to target datasets
- train and test set

Observations and Assumptions

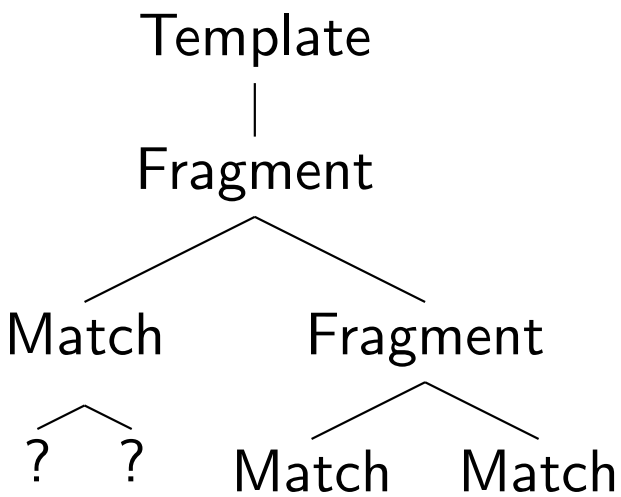
- structurally complex but semantically simple questions
- information need is subset of a Data Cube
- query model as conjunction
  - empty question selects everything
  - phrases are restrictions on dimension values

Pipeline Structure



TCQA—Tree Based Question Answering

- Recursive visit of parse tree
- Stanford statistical english parser resulting in phrase structure
- adaptable to other languages
- Top-down matching, bottom-up combining



Match

$m = (\rho, \gamma)$, where

- ρ partial *property scoring function*, $\rho : P \rightarrow (0, 1]$ and
- γ partial *value scoring function*, $\gamma : P \rightarrow (L \cup U) \times (0, 1]$.
  - P—component properties
  - L—literals
  - U—uris

Example Match

How much did the **Philippines** receive in the years of 2007 to 2008?

$$m = (\rho, \gamma)$$
$$\rho : \varnothing \rightarrow (0, 1]$$
$$\gamma : \{\texttt{:recipient-country} \mapsto (\texttt{:ph}, 1)\}$$

Constraint, Fragment

Definition (Constraint)

$c = (G, \omega, \lambda)$, where:

- G is a set of SPARQL *triple patterns* and *filters*
- ω is an optional *order by* modifier, $\omega \in (\{\text{ASC}, \text{DESC}\} \times P) \cup \{\text{null}\}$
- λ is an optional *limit* modifier, $\lambda \in \mathbb{N}^+ \cup \{\text{null}\}$

Definition (Fragment)

$f = (M, C)$, where M set of matches, C set of constraints.

Interval Constraint

How much did the Philippines receive in the **years** of **2007 to 2008**?

$c_i = (\{\texttt{?o } p \texttt{ ?x, filter(?x} > x_1\texttt{) AND (?x} < x_2\texttt{)}\}, \text{null}, \text{null})$[1]

({?o :refYear ?y, filter(year(?y)>=2007 AND year(?y)<=2008)}, null, null).

[1] $p \in P$, limits x_1 and x_2

Converting Fragment to Template, Execution

- leftover property value references of unmatched properties over a threshold $\rightarrow$ value constraint
- all other references are discarded
- constraints in template $\rightarrow$ SPARQL query
- query execution $\rightarrow$ result set $\rightarrow$ answer

Evaluation

- algorithm output O
- correct answers C of QALD6T3
- evaluation metrics:
  - precision $p = \frac{|C \cap O|}{|O|}$
  - recall $r = \frac{|C \cap O|}{|C|}$
  - F_1 score $F_1 = 2\frac{pr}{p+r}$
- average over each benchmark question
- define $p = 0$ for empty answers for average global F_1

Evaluation

Algorithm	Benchmark	$\varnothing p$	$\varnothing r$	$\varnothing F_1$
TCQA	train	0.40	0.32	0.32
TCQA	test	0.49	0.41	0.44

Error causes (train set)

error cause	*n*
ambiguity	30
lexical gap	18
query structure	17
unknown	1
no error	34
total errors	66

Ambiguity

- RDCQA: property values without property names
- "2007": year, dollar amount, number of people, aggregated value?
- domain-independent keyphrase preprocessing
- match combination
- target dataset ambiguity minimized through whole-query scoring

Lexical Gap

- difference in surface forms between question and knowledge base
- quantity reference ("amount") vs. unit ("dollars") vs. type ("foreign aid")
- TCQA matches both property ranges and labels
- fallback to default measure
- future RDC vocabulary may contain multiple unit measures

Query Structure

- misidentification of query structure
- query structure not supported yet
  - SPARQL subqueries
  - disjunctions (or)

QALD6T3-participants

Algorithm	Benchmark	$\varnothing p$	$\varnothing r$	$\varnothing F_1$
TCQA	train	0.40	0.32	0.32
QA[3]	test	0.59	0.62	0.53
TCQA	test	0.49	0.41	0.44
Sparklis[1]	test	0.96	0.94	0.95

[1] Sparqlis is a query builder, not a QA system.

Future Work

- extended corpus
- yearly evaluation challenge
- implied information needs
- SQA-RDCQA hybrid

Thank You! Questions?

Konrad Höffner
AKSW Research Group
Augustusplatz 10, Room P905
04109 Leipzig, Germany
konrad.hoeffner@uni-leipzig.de
http://aksw.org/KonradHoeffner.html

https://github.com/AKSW/cubeqa