GLAMORISE (GeneraL Aggregation MOdule for RelatIonal databaSEs)

A Novel Solution for the Aggregation Problem in a Natural Language Interface to Databases (NLIDB)

Alexandre F. Novello, Marco A. Casanova

Department of Informatics, PUC-Rio, Rio de Janeiro, RJ – Brazil

{anovello, casanova}@inf.puc-rio.br

Introduction

- **Question Answering (QA)** - is a field of study dedicated to building systems that automatically answer questions asked in natural language.
- **Natural Language Interface to Database (NLIDB)** - The translation of a question asked in natural language into a structured query (SQL or SPARQL) in a database.

Contribution

- NLIDB systems usually do not deal with the treatment of aggregations, but they produce good results for normal queries.
- The contribution of this paper is the creation of a generic module to be used in NLIDB systems.
- This module allows them to perform queries with aggregations, on condition that the result of the NLIDB is, or can be transformed into, a result set in the form of a table.
- Hence, it can even be used with Triplestore (RDF Store) NLIDBs with the proviso that the result is presented as a table.

The VLDB Journal (2019) 28:793–819
https://doi.org/10.1007/s00778-019-00567-8

REGULAR PAPER

A comparative survey of recent natural language interfaces for databases

Katrin Affolter[1] · Kurt Stockinger[1] · Abraham Bernstein[2]

#	Natural language question	Challenges
Q1	Who is the director of 'Inglourious Basterds'?	J, F(s)
Q2	All movies with a rating higher than 9.	J, F(r)
Q3	All movies starring Brad Pitt from 2000 until 2010.	J, F(d)
Q4	Which movie has grossed most?	J, O
Q5	Show me all drama and comedy movies.	J, U
Q6	List all great movies.	C
Q7	What was the best movie of each genre?	J, A
Q8	List all non-Japanese horror movies.	J, F(n)
Q9	All movies with rating higher than the rating of 'Sin City'.	J, S
Q10	All movies with the same genres as 'Sin City'.	J, 2xS

Table 1 Ten sample input questions based on SQL/SPARQL operators that are answerable on the sample world. (**J**oin; **F**ilter (**s**tring, **r**ange, **d**ate or **n**egation); **A**ggregation; **O**rdering; **U**nion; **S**ubquery; **C**oncept)

- NLIDB systems usually do not deal with the treatment of aggregations, but they produce good results for normal queries.

✗ ? 19 / 26 (73,08%)
△ 5 / 26 (19,23%)
✓ 2 / 26 (7,69%)

			Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SQL	SPARQL
Keyword	SODA	2012	✓	△	△	✗	△	✓	✗	✗	✗	✗	✓	✗
	NLP-Reduce	2007	✓	✗	?	✗	?	✓	✗	?	✗	✗	✗	✓
	Précis	2008	△	✗	✗	✗	△	✗	✗	△	✗	✗	✓	✗
	QUICK	2009	✓	✗	✗	✗	✗	?	✗	?	✗	✗	✗	✓
	QUEST	2013	✓	?	?	?	?	✗	?	?	✗	✗	✓	✗
	SINA	2015	✓	?	?	✗	△	✗	✗	?	✗	✗	✗	✓
	Aqqu	2015	✓	?	?	?	?	?	✗	?	✗	✗	✗	✓
Pattern	NLQ/A	2017	✓	✓	?	✓	✓	✓	✓	?	?	?	✗	✓
	QuestIO	2008	✓	✗	✗	✗	✓	?	✗	?	✗	✗	✗	✓
Parsing	ATHENA	2016	✓	✓	✓	✓	✓	✓	△	✗	✓	✗	✓	✗
	Querix	2006	✓	?	?	✓	✓	?	?	?	?	?	✗	✓
	FREyA	2010	✓	?	?	✓	?	?	?	✗	?	?	✗	✓
	BELA	2012	✓	?	?	?	?	?	?	?	?	?	✗	✓
	USI Answers	2013	✓	✓	✓	?	✓	✓	△	✓	?	?	✓	✓
	NaLIR (NaLIX)	2014	✓	✓	?	✓	✓	✗	✓	✓	✓	✓	✓	✗
	BioSmart	2017	✓	?	?	?	✓	✓	?	?	?	?	✓	✗
Grammar	TR Discover	2015	✓	?	?	✗	△	?	✗	△	?	?	✓	✓
	Ginseng	2005	✓	?	?	?	✓	?	?	?	?	?	✗	✓
	SQUALL	2014	△	△	△	△	△	△	△	△	△	△	✗	✓
	MEANS	2015	✓	✗	?	✗	?	✓	✗	✗	✗	✗	✗	✓
	AskNow	2016	✓	?	✓	?	✓	△	?	?	✓	?	✗	✓
	SPARKLIS	2017	✓	✓	✓	△	✓	?	△	✓	✓	✓	✗	✓
	GFMed	2017	✓	✗	✗	✓	?	?	?	✓	?	?	✗	✓
Commercial	Google		✓	△	✗	✓	✓	△	△	✓	✓	✓	?	?
	Siri		✓	✗	△	✗	✗	✗	✗	✗	✗	✗	?	?
	IMDb		△	✓	✓	✗	✓	✗	✗	✗	✗	✗	?	?

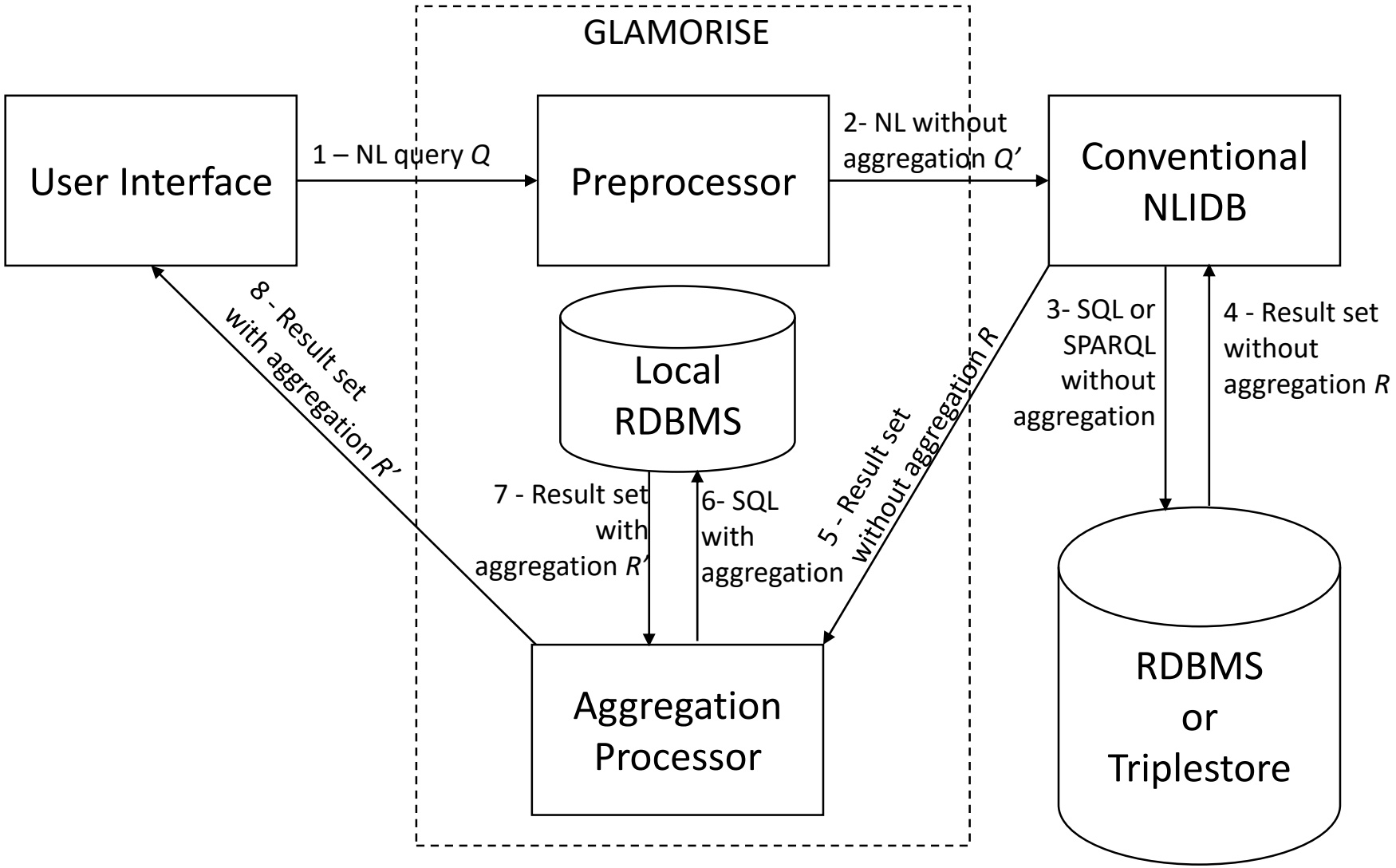
✓, can answer; ▲, strict syntax or partly answerable; ✗, cannot answer; ?, not documented

Contribution

This work covers aggregations with specificities such as:

- Ambiguities;
- Time-scale differences;
- Aggregations in multiple attributes;
- The use of superlative adjectives;
- Basic unit measure recognition;
- Aggregations in attributes with compound names.

GLAMORISE Architecture

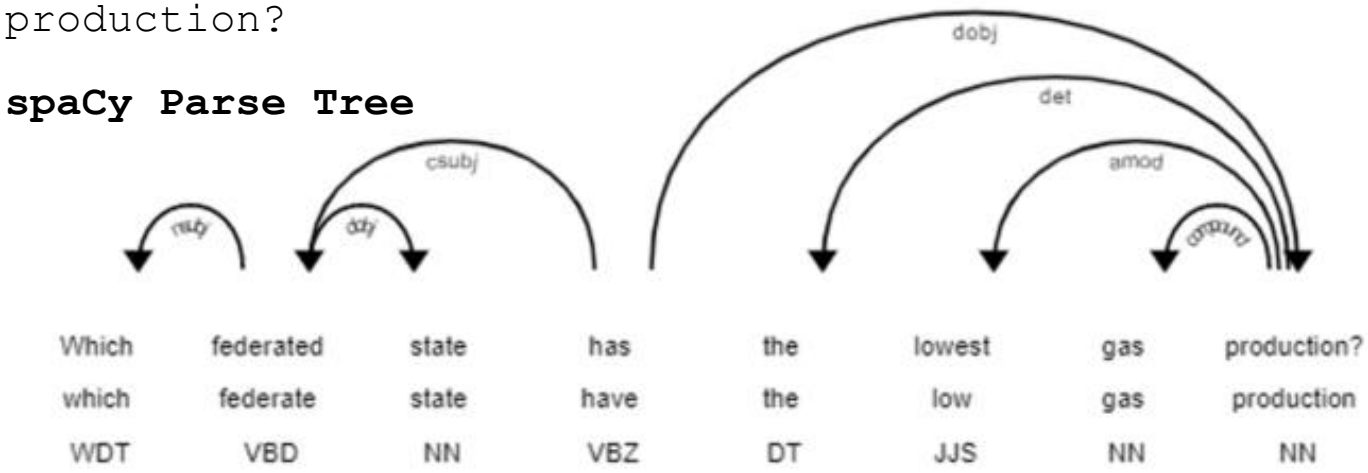


Superlative Adjectives

The superlative adjectives are suppressed and depending on the type of the superlative a *min* or *max* function is added to the aggregation functions metadata, also the respective aggregation field is added.

Natural Language Query: Which federated state has the lowest gas production?

spaCy Parse Tree



GLAMORISE Internal Properties

Preprocessor

```
aggregation_fields = ['gas_production']
aggregation_functions = ['min']
cut_text = ['lowest']
prepared_query = 'Which federated state has the gas production?'
```

Aggregation Processor

```
post_processing_group_by_fields = ['state']
sql = 'SELECT state, min(gas_production) as min_gas_production FROM
NLIDB_result_set GROUP BY state ORDER BY state‘
```

Aggregations with time-scale differences

What was the average yearly production of oil in the state of Alagoas?

- The problem would arise if the data stored in the table is on a monthly basis.
- Two attributes, one for the year and another for the month.
- The most attentive readers will notice that this query is different from the previous examples. Namely, there are two aggregation functions. The first performs the sum of the grouped attribute, “year”, and then the average of all years is calculated.

Name	Type
FIELD	TEXT
BASIN	TEXT
STATE	TEXT
OPERATOR	TEXT
CONTRACT_NUMBER	TEXT
OIL_PRODUCTION	REAL
GAS_PRODUCTION	REAL
MONTH	INTEGER
YEAR	INTEGER

SQL

```
SELECT AVG(SUM(oil_production)) as avg_sum_oil_production
FROM nlidb_result_set
WHERE state = 'Alagoas'
GROUP BY year
```

Aggregations with time-scale differences

What was the average yearly production of oil in the state of Alagoas?

- The problem would arise if the data stored in the table is on a monthly basis.
- Two attributes, one for the year and another for the month.
- The most attentive readers will notice that this query is different from the previous examples. Namely, there are two aggregation functions. The first performs the sum of the grouped attribute, “year”, and then the average of all years is calculated.

Name	Type
FIELD	TEXT
BASIN	TEXT
STATE	TEXT
OPERATOR	TEXT
CONTRACT_NUMBER	TEXT
OIL_PRODUCTION	REAL
GAS_PRODUCTION	REAL
MONTH	INTEGER
YEAR	INTEGER

SQLite

```
SELECT AVG(sum_oil_production) as avg_sum_oil_production
FROM   (SELECT SUM(oil_production) as sum_oil_production
        FROM nlidb_result_set
        WHERE state = 'Alagoas'
        GROUP BY year)
```

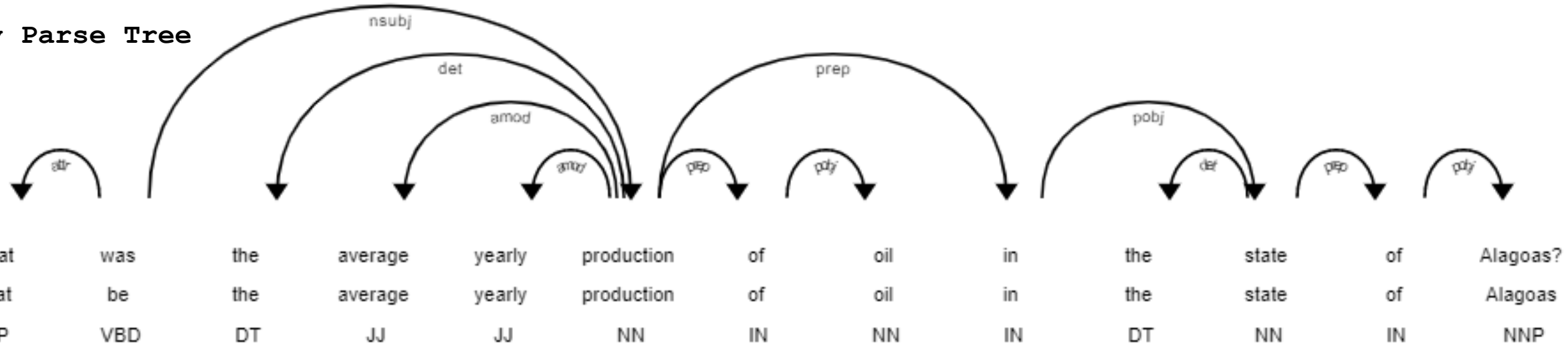
Aggregations with time-scale differences

- The **Preprocessor** converts the adjective , in the case of the example "yearly", to its corresponding noun, in this case "year".
- When we receive the NLIDB result set for this type of question, we also receive a metadata result set with information regarding the timescale in which the data is stored (daily, monthly, yearly, etc.).
- If the question was asked in a different scale, the **Aggregation Processor** does the aggregation accordingly (SUM(*field*) and GROUP BY.
- In the figure we can see how the **Preprocessor** recognizes the sentences and separates the "average" interpretation, which is the normal aggregation that will be made, from the "yearly" interpretation which is the time-scale aggregation.

Aggregations with time-scale differences

Natural Language Query: What was the average yearly production of oil in the state of Alagoas?

spaCy Parse Tree



GLAMORISE Internal Properties

Preprocessor

```
aggregation_fields = ['oil_production']
aggregation_functions = ['avg']
cut_text = ['average']
replaced_text = {'yearly': 'year'}
time_scale_aggregation_fields = ['oil_production']
time_scale_aggregation_functions = ['sum']
time_scale_group_by_fields = ['year']
prepared_query = 'What was the year production of oil in the state of Alagoas?‘
```

Aggregation Processor

```
sql = 'SELECT avg(oil_production) as avg_oil_production FROM (SELECT sum(oil_production) as oil_production
FROM NLIDB_result_set GROUP BY year)'
```

Proof-of-concept

- To test the performance of GLAMORISE, we implemented a mock NLIDB prepared to receive the set of testing questions (next slide).
- Note that there are questions with completely different phrasings.
- First, we confirmed that the questions were preprocessed correctly, removing or substituting words (aggregation elements) as necessary.
- Second, we confirmed that the generated SQL queries with aggregation were correct.
- Taking into account the 21 questions that were asked, all of them were answered correctly.

Proof-of-concept

ID	NLQ
Q1	What was the production of oil in the state of Rio de Janeiro?
Q2	What was the average monthly production of oil in the state of Rio de Janeiro?
Q3	What was the average yearly production of oil in the state of Alagoas?
Q4	How many fields are there in Paraná?
Q5	What was the maximum production of oil in the state of Ceará per field?
Q6	What was the minimum gas production in the state of São Paulo per basin?
Q7	What was the average monthly oil production by the operator Petrobrás?
Q8	What was the mean yearly gas production per field?
Q9	What was the mean gas production per month per field?
Q10	What was the per month mean gas production per field?
Q11	What was the per field mean gas production per month?
Q12	What was the mean monthly petroleum production by field in the state of Rio de Janeiro?
Q13	What was the mean yearly petroleum production by field by Rio de Janeiro?
Q14	What was the average monthly production of oil per field in the state of Rio de Janeiro and year 2015?
Q15	What was the average yearly production of oil per field and state in the year in 2015?
Q16	What was the mean gas production per field with production greater than 100 cubic meters?
Q17	What was the mean gas production per basin with production less than 1000 cubic meters?
Q18	Which field produces the most oil per month?
Q19	Which basin has the highest yearly oil production?
Q20	Which federated state has the lowest gas production?
Q21	Which state of the federation has the lowest gas production?

Future Work

- Refine and expand aggregations treatments
- ~~Mock NLIDB~~ Real NLIDB
- Integration via Web Service (JSON)
- Web Interface
- Other datasets (QALD)
- More advanced unit-measures recognition
- Elliptical aggregations.
  - E.g.: *“What was the mean yearly gas production per field before ~~the year~~ 2015?”*
- Subqueries (other than the time-scale problem stated and resolved).
  - E.g.: *“Which fields produced more oil than the average for all fields?”*

More About NLIDBs in SBBD

- ***Tutorial 1:***

  Palavras, apenas: Métodos e Técnicas para Interfaces de Linguagem Natural em Bancos de Dados – Altigran,* Brandell, Lucas e Paulo (UFAM)
- ***Short Papers (October 2nd, Friday):***
  - ***11:40*** - Improving the Quality of the User Experience by Query Answer Modification – João Pedro e Casanova (PUC-Rio) e Elisa (IFS)
  - ***12:00*** - *Keyword Search over COVID-19 Data – Yenier, Grettel, Melissa, Novello, Bruno, Cleber e Casanova (PUC-Rio) e Luiz André (UFF)*

GLAMORISE (GeneraL Aggregation MOdule for RelatIonal databaSEs)

A Novel Solution for the Aggregation Problem in a Natural Language Interface to Databases (NLIDB)

Alexandre F. Novello, Marco A. Casanova

Department of Informatics, PUC-Rio, Rio de Janeiro, RJ – Brazil

{anovello, casanova}@inf.puc-rio.br