



# QuestIO: a Question-based Interface to Ontologies

Danica Damljanović

Valentin Tablan

Kalina Boncheva

University of Sheffield

d.damljanovic@dcs.shef.ac.uk

Funded by: European Commission – 6th  
Framework

Project Reference: IST-2004-026460



# Content

- Objective and Motivation  
Problems and challenges  
Our Approach  
Achievements  
Evaluation  
What next?  
Questions?

# Objective

Developing a tool for querying the knowledge store using text-based Natural Language (NL) queries.



# Motivation

- Downsides of existing query languages (e.g., SeRQL, SPARQL):
  - complex syntax,  
not easy to learn,  
writing queries is error-prone task,  
requires understanding of Semantic Web technologies.

# Does it make sense?

“Java Class for parameters for processing resources in ANNIC?”



```
select c0,"[inverseProperty]", p1, c2,"[inverseProperty]", p3,  
c4,"[inverseProperty]", p5, i6  
from {c0} rdf:type {<http://gate.ac.uk/ns/gate-  
ontology#JavaClass>}, {c2} p1 {c0}, {c2} rdf:type  
{<http://gate.ac.uk/ns/gate-  
ontology#ResourceParameter>}, {c4} p3 {c2}, {c4}  
rdf:type {<http://gate.ac.uk/ns/gate-  
ontology#ProcessingResource>}, {i6} p5 {c4}, {i6}  
rdf:type {<http://gate.ac.uk/ns/gate-  
ontology#GATEPlugin>}  
where p1=  
http://gate.ac.uk/ns/gate-ontology#parameterHasType and  
p3=  
http://gate.ac.uk/ns/gate-ontology#hasRunTimeParameter  
and p5=  
http://gate.ac.uk/ns/gate-ontology#containsResource and  
i6=<http://gate.ac.uk/ns/gate-ontology#annic>
```

# First prototype...

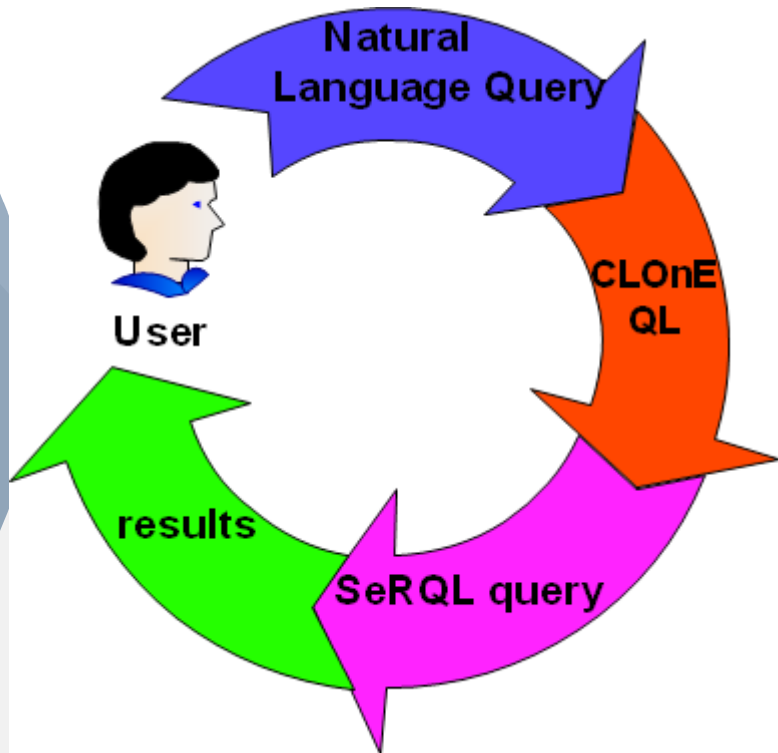
## A **Controlled** Language for Ontology Querying:

- recognizing patterns in a text-based query and creating SeRQL queries accordingly;

### Limitations:

- requires syntactically correct sentences;
- cannot process concept-based queries such as 'accommodation Rome';
- can process a limited set of queries.

# Challenges



to enhance robustness;

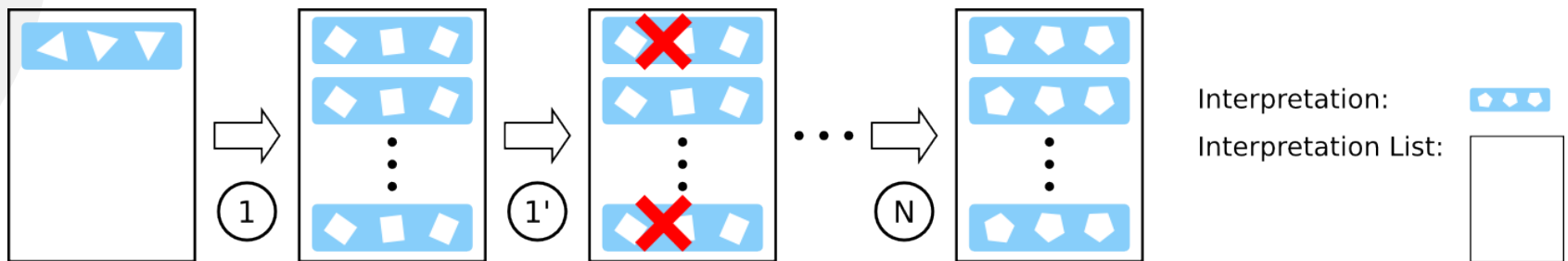
to accept queries of any length and form;

to be portable and domain independent.

# From questions to answers

The text query is transformed into a SeRQL query using a set of *Transformers*. The input and an output for a *Transformer* is an *Interpretation*: **Interpretations** are used as a container for information.

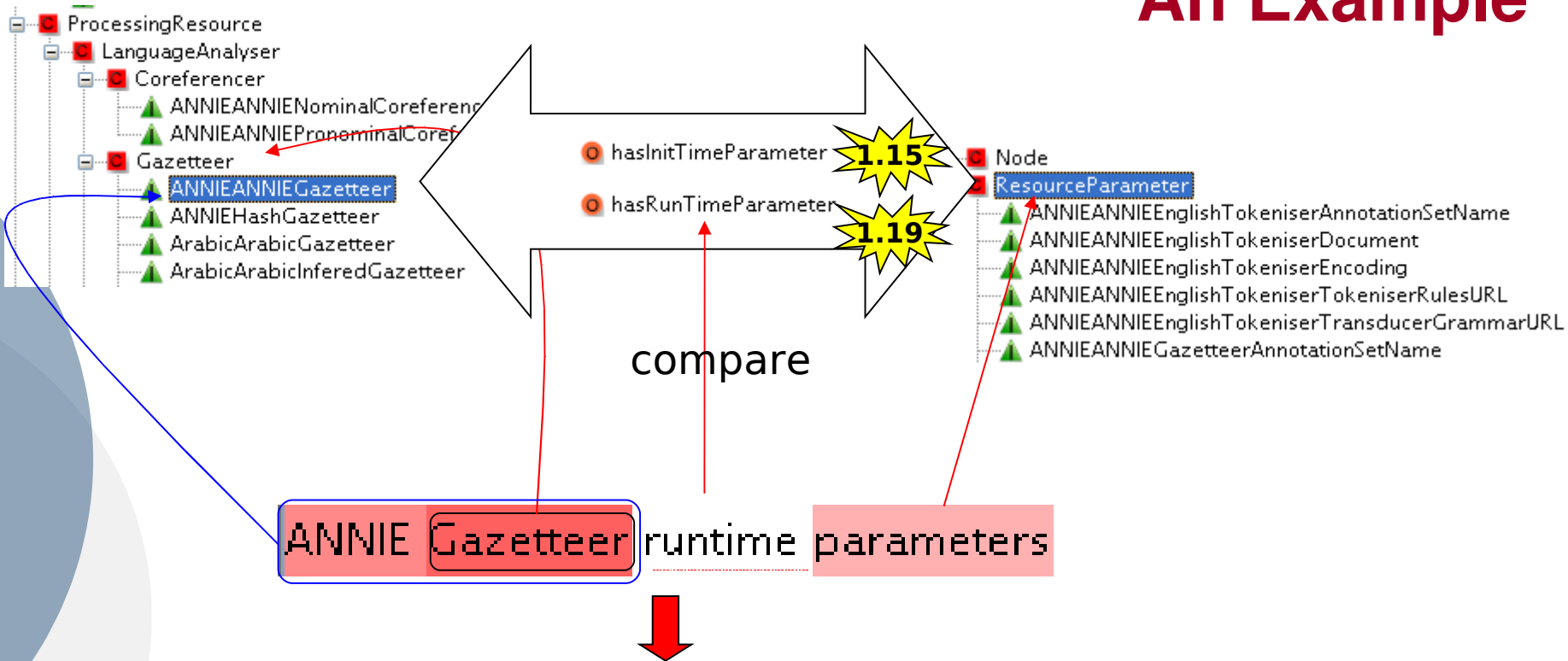
**Transformer** represents an algorithm for converting a type of interpretation into another.



# From questions to answers

- Producing ontology-aware annotations
- Filtering annotations
- Identifying relations between annotated concepts
- Scoring relations
- Creating SeRQL queries and showing results

# An Example



Result:

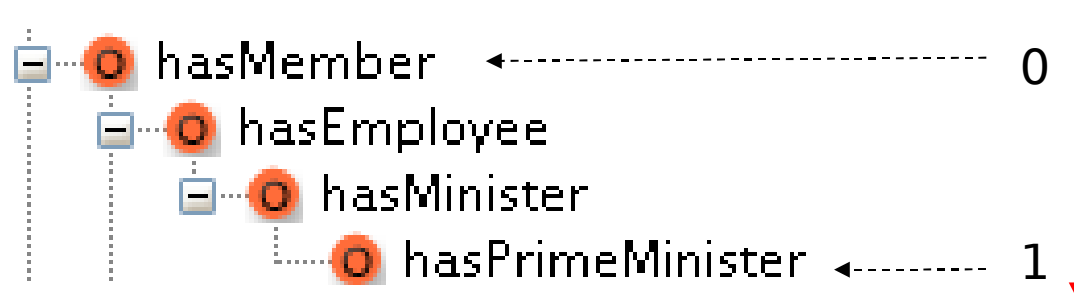
```
ANNIE Gazetteer --> hasRunTimeParameter --> document
ANNIE Gazetteer --> hasRunTimeParameter --> ANNIEANNIEGazetteerAnnotationSetName
ANNIE Gazetteer --> hasRunTimeParameter --> ANNIEANNIEGazetteerWholeWordsOnly
ANNIE Gazetteer --> hasRunTimeParameter --> longestMatchOnly
```

# Scoring relations

We combine three types of scores:

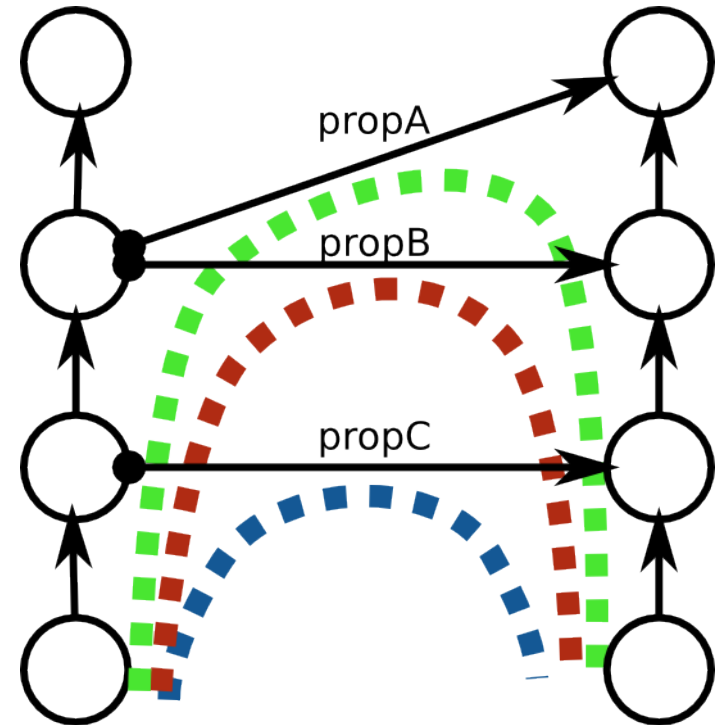
*similarity score* - using Levenshtein similarity metrics we compare input string from the user with the relevant ontology resource

*specificity score* is based on the subproperty relation in the ontology definition.



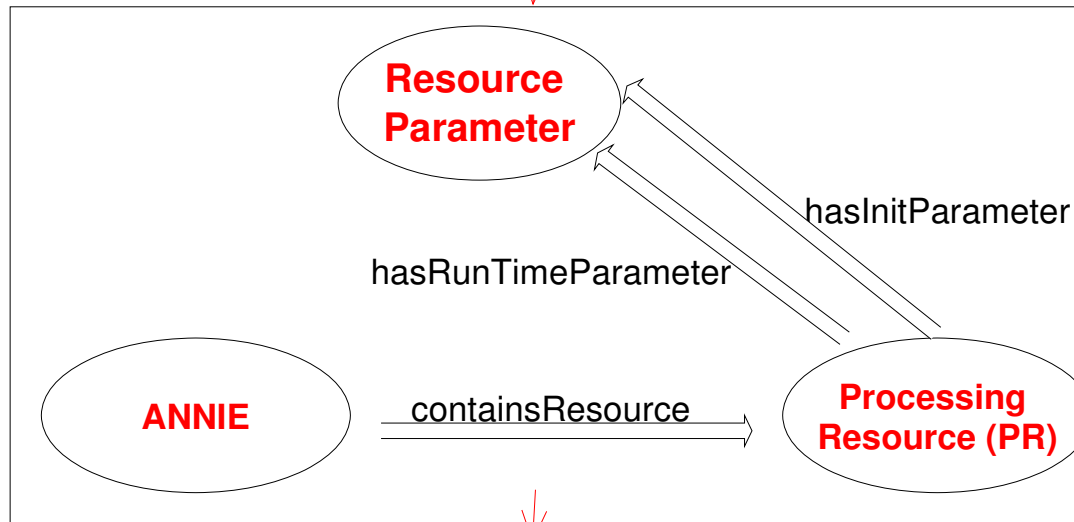
## Scoring relations (II)

*distance score is inferring an implicit specificity of a property based on the level of the classes that are used as its domain and range.*



# Relative clauses

What are the **parameters** of **PR** that is included in **ANNIE**?

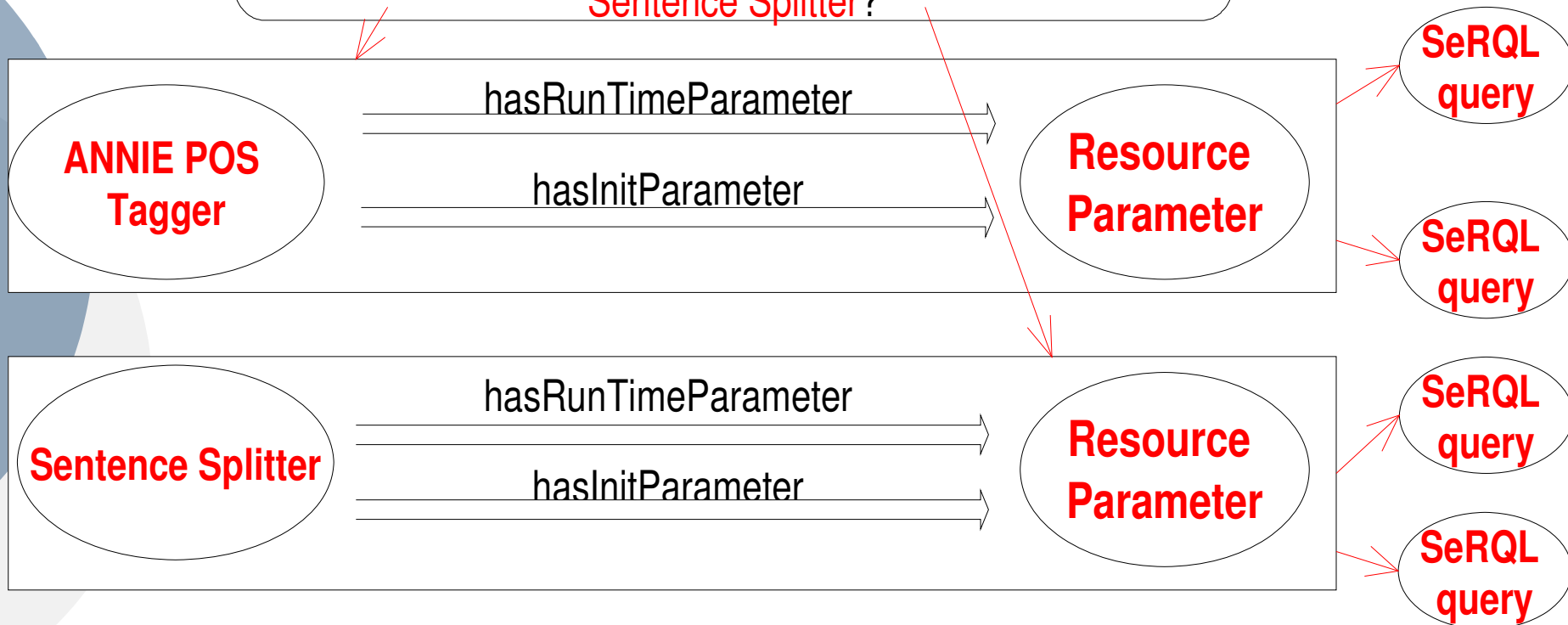


```
select y,p,x from {x} rdf:type {<http://gate.ac.uk/ns/gate-ontology#ResourceParameter>}, {y}
  rdf:type {<http://gate.ac.uk/ns/gate-ontology#ProcessingResource>},{y}
  <http://gate.ac.uk/ns/gate-ontology#hasInitTimeParameter> {x},{y} p {x},{z}
<http://gate.ac.uk/ns/gate-ontology#containsResource> {y} where z=<http://gate.ac.uk/ns/gate-ontology#ANNIE>
```

```
select y,p,x from {x} rdf:type {<http://gate.ac.uk/ns/gate-ontology#ResourceParameter>}, {y}
  rdf:type {<http://gate.ac.uk/ns/gate-ontology#ProcessingResource>},{y}
  <http://gate.ac.uk/ns/gate-ontology#hasRunTimeParameter> {x},{y} p {x},{z}
<http://gate.ac.uk/ns/gate-ontology#containsResource> {y} where z=<http://gate.ac.uk/ns/gate-ontology#ANNIE>
```

# Grouping of elements

What are the **parameters** of **ANNIE POS Tagger** OR **Sentence Splitter**?



# Our achievements

- **Dynamically** generating SeRQL queries.
- **Unlimited number of concepts in a query.**  
Partially supporting **relative clauses**:
  - What are the parameters of the PR that is included in ANNIE plug-in?

**Grouping identified concepts** to support more complex queries:

Which PRs are included in annic AND annie?

What are the parameters of POS Tagger OR Sentence Splitter?

**Setting the environment for implementing user interaction:**

Tracking transformations from text to the SeRQL query so that user can be easily returned to the stage where he can change/refine his query.

# Evaluation

We evaluated:

- coverage and correctness
- scalability and portability

# Evaluation on coverage and correctness

We manually collected 36 questions posted by GATE users to the project's mailing list in the past year, for example:

- Which PRs take ontologies as a parameter?

Which plugin is the VP Chunker in?

What is a processing resource?

## Evaluation on coverage and correctness (2)

22 out of 36 questions were *answerable*  
(the answer was in the knowledge base):

12 *correctly answered* (54.5%)

6 *with partially corrected answer*  
(27.3%)

system *failed* to create a SeRQL query or  
created a wrong one for 4 questions  
(18.2%)

Total score:

**68%** correctly answered

32% did not answer at all or did not  
answer correctly

# Evaluation on scalability and portability

Sizes of the knowledge bases created based on:

GATE ontology: <http://gate.ac.uk/ns/gate-ontology>

Travel ontology:

<http://goodoldai.org.yu/ns/tgproton.owl>

GATE Knowledge Base	
Classes	42
Object Properties	23
Instances	594
Total size (C + P + I)	659
Initialisation time	19 seconds

Travel Knowledge Base	
Classes	318
Object Properties	86
Instances	2790
Total size (C + P + I)	3194
Initialisation time	109 seconds

# Evaluation on scalability and portability

Query execution times:

Query size (number of properties)	Execution time (seconds)	Number of results	Actual query
<b>GATE Knowledge Base</b>			
1	0.148	15	<i>“processing resources in ANNIE?”</i>
2	0.234	37	<i>“parameters for processing resources in ANNIE?”</i>
3	0.298	37	<i>“Java Class for parameters for processing resources in ANNIE?”</i>
<b>Travel Knowledge Base</b>			
1	1.013	52	<i>“countries in asia”</i>
2	2.030	52	<i>“capitals of countries in asia”</i>
3	3.307	52	<i>“capitals of countries in global regions in asia”</i>

# What next?

- Using implemented transformations to employ user interaction:
  - When the system is not able to make decisions autonomously it will require additional input from the user.

Improving the algorithms for generating SeRQL queries.

Optimization of the tool initialization (scalability issues).

More evaluation on scalability (with KIM).

Evaluate its expressivity against that of SeRQL.

Try technologies for soft matching and synonym retrieval, e.g., between hotel and accommodation.

## Online demo

URL will be provided to all participants to try during the tutorial from their web browser



# Questions?

Thank you!