A Natural Language Query Interface to Structured Information

(proceedings page 361)

Valentin Tablan
Danica Damljanovic
Kalina Bontcheva
University of Sheffield

Information access

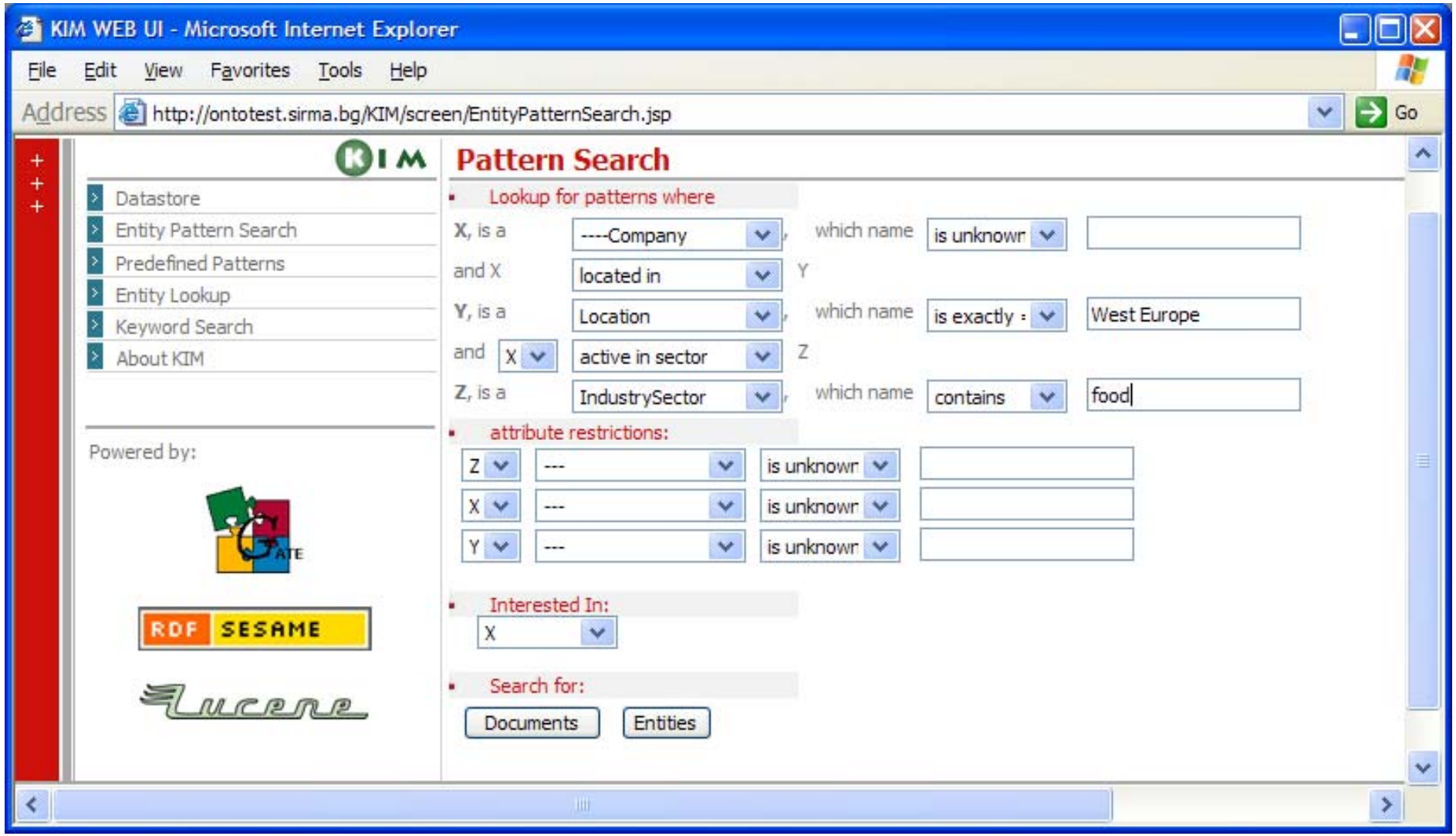
"Capitals of countries in Asia"

- Full text search
  - Several iterations.
  - A lot of work.
- Conceptual search:
  - Can make use of abstractions and generalisations powered by ontology back-end.
  - With the right ontology/knowledge base, it's easy!

Just type this in the query field:


```
select c0, c3
from
  {c0} rdf:type {<pupp#Capital>},
  {c3} p1 {c0},
  {c3} rdf:type {<pupp#Country>},
  {c3} p4 {i6},
  {i6} rdf:type {<pupp#Continent>}
where
  p1=<pupp#hasCapital> and
  p4=<pupp#locatedIn> and
  i6=<wkb#Continent_T.2>
```

...or fill in this form

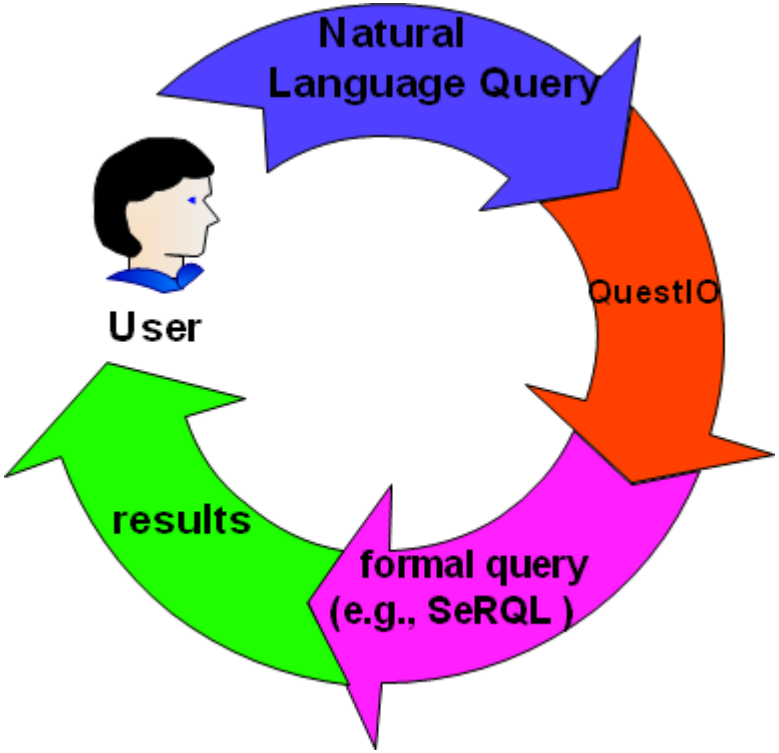


QuestIO: Question-based Interface to Ontologies

Natural Language interface for querying knowledge bases.

Aims to:

- Be domain independent.
- Be easy to use – require no training.
- Work with short, agrammatical queries (Google-like).



QuestIO: Domain Independent

Easy to change between ontologies with little or no effort.

- Build vocabulary directly from ontology:
  - Ontologies contain lots of text entries (resource names, labels, comments, string property values).
  - Normalise for morphology, capitalisation, segmentation, CamelCaseWords:

```
CapitalCity, capitalCity, capital_city
  → Capital City
```

- Then put everything in a large gazetteer (FST lookup).

Query Construction

- Formal query (SeRQL, SPARQL):
  - A list of objects or variables chained by predicates.

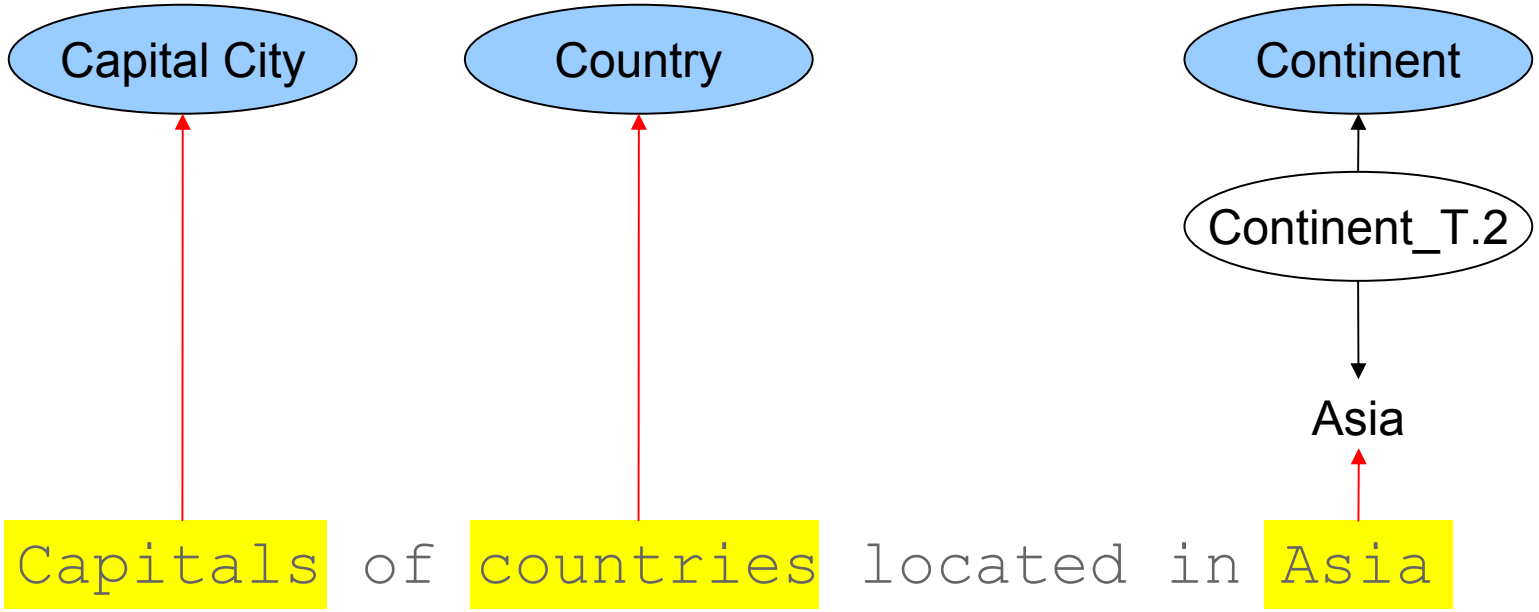
- Natural Language query:
  - A list of interrogative pronouns and known objects linked by [implied] predicates.

What countries are in Asia?

Is London capital of any country?

Query Construction: Step 1 – find objects

- Identify known objects in the NL query
  - Normalise the query for morphology, etc.
  - Find matching lexicalisations from the gazetteer.
  - Identify corresponding classes.



Query Construction: Step 2 – find predicates

Construct a formal query by finding appropriate properties to link the concepts found.

- Build a list of candidate properties based on ontology schema (using domain and range constraints).
- Rank the properties to find the most appropriate ones.
  - Use several techniques, to cover for most cases.

Property Ranking: String Similarity

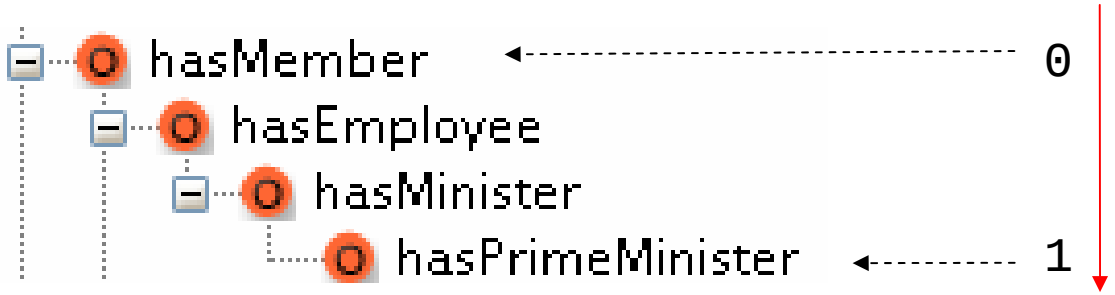
- Compare query fragments with candidate property names using Levenshtein[1] string similarity metric.

- "of" → ?
- "located in" → locatedIn

[1] Using Sam Chapman's simmetrics implementation.

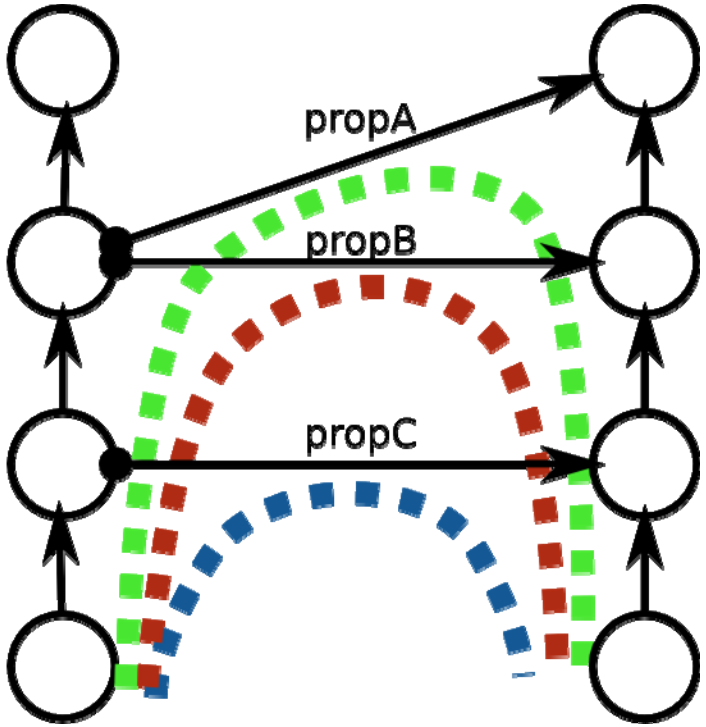
Property Ranking: Ontology Structure

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



Property Ranking: Ontology Structure (II)

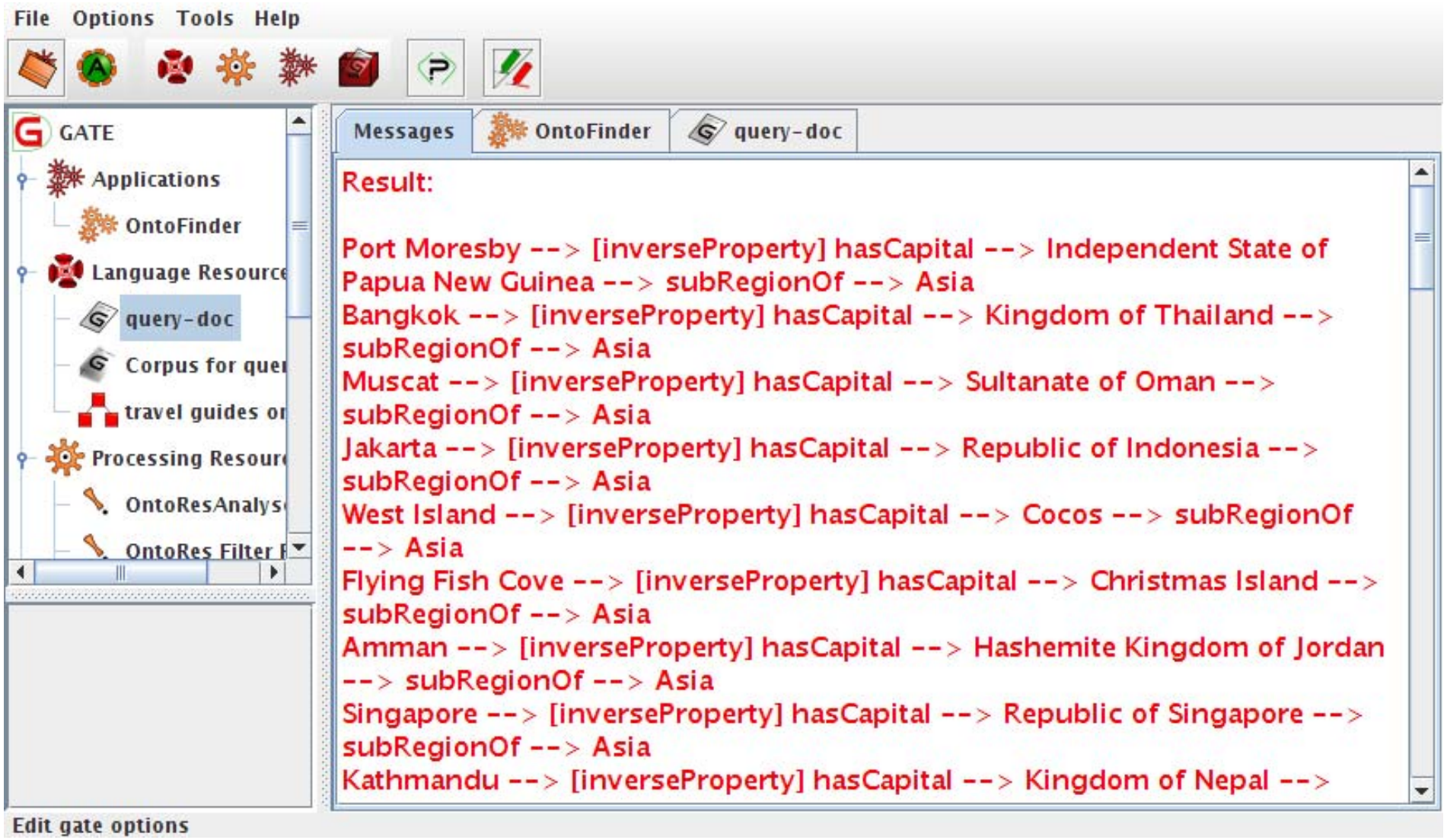
- distance from concepts: inferring an implicit specificity of a property based on the level of the classes that are used as its domain and range.



Query Execution

- Build formal queries, using identified objects and candidate predicates.
- Execute queries sorted by:
  - Object preference level (e.g. instance names are preferred to associated property values).
  - Property ranking order.
- ...until [some] results are found.
- Note that predicates may be reversed!

Results!


File Options Tools Help
GATE
Applications
OntoFinder
Language Resource
query-doc
Corpus for quer
travel guides or
Processing Resour
OntoResAnalys
OntoRes Filter F
Messages
OntoFinder
query-doc
Result:
Port Moresby --> [inverseProperty] hasCapital --> Independent State of Papua New Guinea --> subRegionOf --> Asia
Bangkok --> [inverseProperty] hasCapital --> Kingdom of Thailand --> subRegionOf --> Asia
Muscat --> [inverseProperty] hasCapital --> Sultanate of Oman --> subRegionOf --> Asia
Jakarta --> [inverseProperty] hasCapital --> Republic of Indonesia --> subRegionOf --> Asia
West Island --> [inverseProperty] hasCapital --> Cocos --> subRegionOf --> Asia
Flying Fish Cove --> [inverseProperty] hasCapital --> Christmas Island --> subRegionOf --> Asia
Amman --> [inverseProperty] hasCapital --> Hashemite Kingdom of Jordan --> subRegionOf --> Asia
Singapore --> [inverseProperty] hasCapital --> Republic of Singapore --> subRegionOf --> Asia
Kathmandu --> [inverseProperty] hasCapital --> Kingdom of Nepal -->
Edit gate options

Evaluation – datasets

- Travel guides ontology:
  - Uses a section of PROTON[1], relevant to geography concepts.
  - Populated with the relevant instances from the KIM[2] knowledge base.
- GATE Ontology:
  - A semi-automatically derived ontology/knowledge base describing the GATE[3] text mining platform.

1 http://proton.semanticweb.org

2 http://www.ontotext.com/kim/

3 http://gate.ac.uk.

Evaluation: scalability (init time[1])

- Ontologies have not been customised or changed prior using with QuestIO!

	GATE kb	Travel kb
Classes	42	318
Object Properties	23	86
Instances	594	2790
Total size	659	3194
Initialisation time	16 seconds	22 seconds

[1] Times are lower than reported in the paper due to ongoing optimisation work.

Evaluation: scalability (run time)

Query	Objects	Time (ms)
Countries	1	70
Countries in Asia	2	108
Capitals of countries in Asia	3	135
Capitals of countries in global regions part of Asia	4	240

Evaluation: Coverage and correctness

- 36 questions extracted from GATE list
  - 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

Demo

http://www.gate.ac.uk/questio-client-app/search.js

- Travel guides ontology:
  - Continents, countries, cities (capitals only).
- Example questions:
  - Countries in Europe or North America
  - Asia's global regions
  - Capitals of countries (located) in Africa
  - ...

Future Work

Move toward a session-based approach

- Don't just say “Nothing found”;
- Use session history to guide the search (affect ranking);
- Keep user profiles with custom lexicalisations (e.g. *“works for”* vs. `isEmployedBy`).

Thanks

- ...to you, for your attention!
- ...to the EC, for funding the TAO project!

  (http://www.tao-project.eu)
- ...to Vanessa Lopez (KMI, Open University, UK), for letting us play with the Aqualog system!

- **Questions?**