Towards Ontology-Based Question Answering in Vague Domains

Panos Alexopoulos, Andrew Walker, Jose Manuel Gomez-Perez, Manolis Wallace

9th International Workshop on Semantic Media and Adaptation

Corfu, Greece, November 6-7, 2014

Question Answering Systems

- Question Answering systems (QA) enable users to ask questions in Natural Language (NL) and receive concise answers.
  - E.g., “Which country has won the most World Cups?”

- With the rapid growth of the Semantic Web there has been a growing interest in developing QA systems that can answer questions over ontologies and Linked Data.

  - **Systems:** PowerAqua, Pythia, Freya etc.

  - **QALD Challenges:** A series of evaluation campaigns on (multilingual) question answering over Linked Data (4 campaigns so far).

- Nevertheless, the issue of **vagueness in QA systems** has not been adequately considered yet.

- Vagueness is a semantic phenomenon where predicates admit borderline cases, i.e. cases where it is not determinately true that the predicate applies or not (Shapiro 2006).

- This happens when predicates have blurred boundaries:
  - What's the threshold number of years separating old and not old films?
  - What are the exact criteria that distinguish modern restaurants from non-modern?

Vague Questions against a Non-Vague Knowledge Base

- *The question is "**I want an expensive restaurant**" but in the knowledge base the attribute price has only numerical values (e.g. 20 euros).*

Non-Vague Questions against a Vague Knowledge Base

- *The question is "**I want a restaurant at 20 euros**" but in the knowledge base the attribute price has only vague values (e.g. cheap, moderate, expensive).*

Vague Questions against a Vague Knowledge Base

- *The question is "**I want a relatively cheap restaurant**" but the knowledge base has only "**cheap**" restaurants.*

- **Problem definition:** *Given an ontology-based question answering scenario where **both ontology elements and user queries may be expressed with vague terms**, how can we effectively interpret the users' queries and provide them with accurate answers?*

- *This has two main subproblems:*

  1. *How do we **formally represent the meaning** of the domain's **vague terms** and concepts within the system's ontology*

  2. *How do we **use** this **formalization** for **interpreting** and **answering vague questions**?*

Fuzzy Concepts

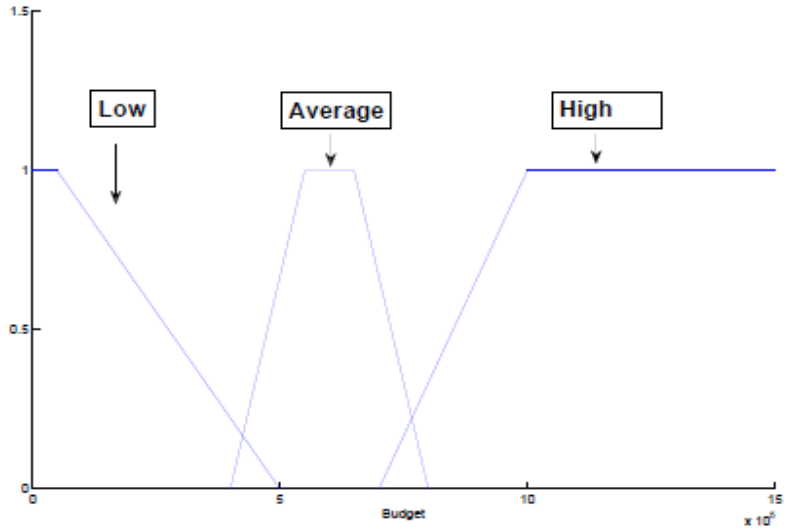
- *A fuzzy ontology concept may have instances that belong to it at certain degrees.*
- *E.g. John is a TallPerson to a degree of 0.5.*

Fuzzy Relations and Attributes

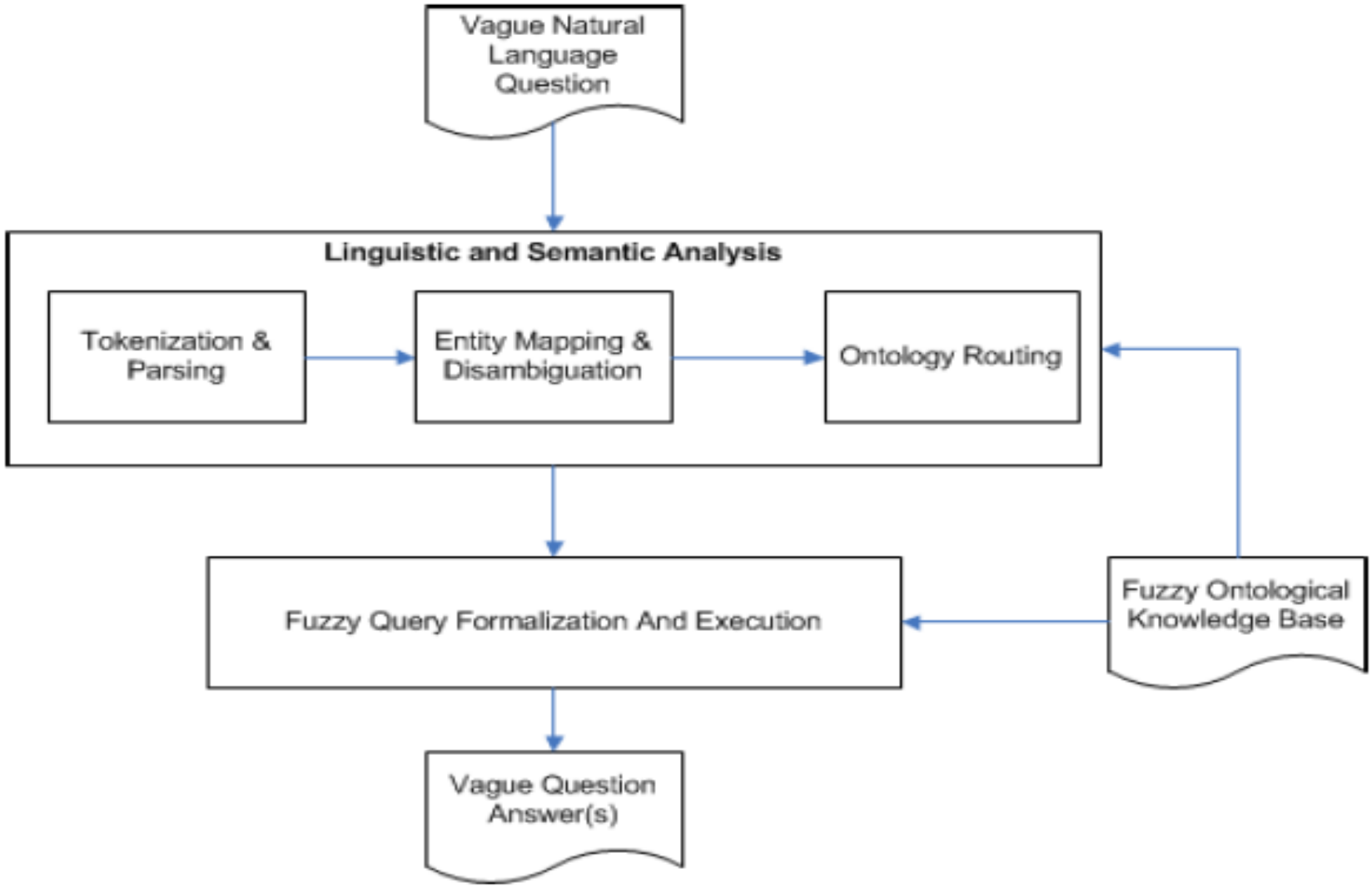
- *A fuzzy ontology relation links concept instances at certain degrees.*
- *Π.χ. John is expert at Machine Learning to a degree of 0.9.*
- *Similarly, a fuzzy attribute assigns literal values to concept instances at certain degrees.*

Fuzzy Datatypes

- *A fuzzy datatype consists of a set of vague terms which may be used within the ontology as attribute values.*
  - *Π.χ. Low, Average, High for the attribute Project Budget.*
- *Each term is mapped to a fuzzy set that defines the term's meaning.*



Vague Question Answering Pipeline



Fuzzy Ontology Construction

- Any process/methodology can be used but with 3 additional steps:
  - We define fuzzy datatypes even for attributes that take non-vague values in the ontology as long as these may be vaguely expressed by the users.
  - For such attributes we **"rewrite"** their relevant value axioms.
    - E.g. "***A has price 20 Euros***" becomes "***A has price 'cheap" to degree d1***" and ***"A has price 'moderate' to degree d2***"
    - Degrees d1 and d2 are derived from the fuzzy datatype.
  - For attributes that take vague values we **"expand"** their relevant value axioms:
    - E.g. "***A has price 'cheap'*** becomes "***A has price 'moderate" to degree d1***" and ***"A has price 'expensive' to degree d2***"
    - Degrees d1 and d2 are derived from the fuzzy subsumption between the fuzzy terms.

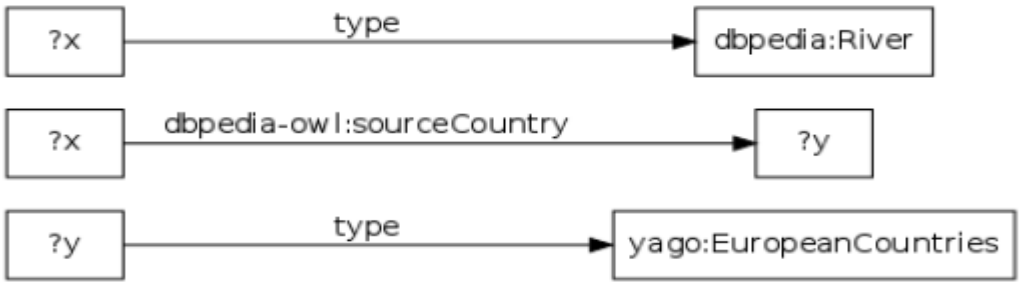
Question Linguistic and Semantic Analysis

- This stage involves the transformation of a natural language question into a set of ontological statements that express the required information.

- This typically includes the following steps:

  - **Tokenization and parsing** of the question's text.
  - **Entity mapping**, namely the mapping of the question's tokens to the ontological entities they refer to.
  - **Entity routing**, namely the identification of possible relation paths that link the identified entities of the question within the knowledge base

- E.g., the question "Which rivers start from a European Country" becomes:



Query Fuzzification and Execution

- The fuzzification stage is the conversion of the derived triples to a formal fuzzy query language.
- In this paper we consider the f-SPARQL query language which allows to query ontologies that are formalized in the f-DL-Lite language.
- f-DL-Lite allows the definition of fuzzy assertions like :
  - Elvis is a FamousPerson to degree at-least 0.9
  - John is expert in Ontologies to degree at-least 0.7.
- f-SPARQL supports two types of queries:
  - **Threshold queries:** Ask for entities that satisfy fuzzy axioms to certain degrees.
  - **General Fuzzy Queries**: Ask for the degrees to which given axioms are true.
- *The conversion of the query triples to f-SPARQL depends on the query and knowledge base content.*

Query Fuzzification – Case 1

- The user's query involves a fuzzy concept or fuzzy relation (e.g. "Give me all our competitors")
- Then we form the query as a General Fuzzy Query:

```
#GFCQ:SEM=FUZZYTHRESHOLD# SELECT ?x WHERE
{ ?x rdf:type Competitor. #DG# 1.0 }
```

- The execution of this query against the fuzzy ontology will return all the competitors with their actual degrees.
- These degrees make the answer more complete as they reflect the extent to which each result entity is actually considered a competitor in the knowledge base.

Query Fuzzification – Case 2

- The user's query involves an attribute associated to a fuzzy datatype (e.g., price)
- The attribute's **value is non-vague in the knowledge base but vague in the user's query**
- Then, we form again a General Fuzzy Query like the following:

```
#GFCQ:SEM=FUZZYTHRESHOLD#
SELECT ?x
WHERE {?x price ~ Expensive . #DG# 1.0}
```

- This query is executed on the a priori fuzzified facts and retrieves the relevant entities along with their degrees.

Query Fuzzification – Case 3

- The user's query involves an attribute associated to a fuzzy datatype (e.g., price)
- The attribute's **value** is **vague in both the knowledge base and the user's query**.
- Then, we form the same f-SPARQL query as in case 2

```
#GFCQ:SEM=FUZZYTHRESHOLD#
SELECT ?x
WHERE {?x price  Expensive . #DG# 1.0}
```

- This time, however, the query will be executed on the expanded fuzzified vague terms of the knowledge base.
- This ensures that different vague terms will match (even if only to a certain degree) as long as they have some overlap in their meanings, thus increasing the recall of the system.

Query Fuzzification – Case 4

- The user's query involves an attribute associated to a fuzzy datatype (e.g., price)
- The attribute's **value in the knowledge base is vague but non-vague in the user's query**.
- Then we use the fuzzy datatype to express the query's crisp value by means of its corresponding vague terms.
- E.g., the query "Give me all our restaurants with a price of 20 euros" will be transformed into a set of threshold queries as follows:

```
#GFCQ:SEM=FUZZYTHRESHOLD# SELECT ?x WHERE
  ?x price  cheap . #DG# 0.6
}

#GFCQ:SEM=FUZZYTHRESHOLD# SELECT ?x WHERE
  ?x price  moderate . #DG# 0.4
}
```

- Executing these two queries will retrieve all restaurants that are moderate to a degree of 0.4 or cheap to a degree of 0.6.

Concluding Remarks

- We proposed a novel fuzzy-ontology based architecture for performing ontology-based question answering in vague domains.
- Future work includes mainly the assessment of the QA pipeline's efficiency:
  - Execution of the system against fuzzy knowledge bases of increasing size and complexity.
  - Measurement of the average time it takes to interpret and answer a question.

iSOCO
enabling the networked economy

Thank you!

Questions?

Dr. Panos Alexopoulos

Semantic Applications Research Manager

palexopoulos@isoco.com
(t) +34 913 349 797

iSOCO Barcelona
Av. Torre Blanca, 57
Edificio ESADE CREAPOLIS
Oficina 3C 15
08172 Sant Cugat del Vallès
Barcelona, España
(t) +34 935 677 200

iSOCO Madrid
Av. del Partenón, 16-18, 1º7ª
Campo de las Naciones
28042 Madrid
España
(t) +34 913 349 797

iSOCO Pamplona
Parque Tomás
Caballero, 2, 6º4ª
31006 Pamplona
España
(t) +34 948 102 408

iSOCO Valencia
C/ Prof. Beltrán Báguena, 4
Oficina 107
46009 Valencia
España
(t) +34 963 467 143

iSOCO Colombia
Complejo Ruta N
Calle 67, 52-20
Piso 3, Torre A
Medellín
Colombia
(t) +57 516 7770 ext. 1132

Gartner Key Vendor
Virtual Assistant 2013