Answering Natural Language Questions on RDF Knowledge Bases in French

ESWC-QALD 2017 Challenge, Task 1

Mathieu Tremblay\textsuperscript{1}
Nikolay Radoev\textsuperscript{1}
Michel Gagnon\textsuperscript{1}
Amal Zouaqa\textsuperscript{2}

\textsuperscript{1} Polytechnique Montréal \textsuperscript{2} University of Ottawa
Ask Me in Any* Language (AMAL) system

*Soon
Ask Me in Any* Language (AMAL) system

*Soon

AMAL
Ask Me in Any Language (AMAL) is a system developed for the QALD-7 competition (Question Answering over Linked Data 2017). It is currently only available in French, but will be extended to support multiple languages in the near future.

Question:
Qui est le créateur de Batman?

Results:

© 2017 - Mathieu Tremblay, Nikolay Radiev
System description
System description

• Focuses on the French language (for now).
System description

• Focuses on the French language (for now).

• Does not rely on given keywords.
System description

- Focuses on the French language (for now).
- Does not rely on given keywords.
- No modifications required to the question format.
System description

• Focuses on the French language (for now).
• Does not rely on given keywords.
• No modifications required to the question format.
• Primary focus on Simple Questions.
System description

- Focuses on the French language (for now).
- Does not rely on given keywords.
- No modifications required to the question format.
- Primary focus on Simple Questions.

Analyses given questions and builds custom SPARQL queries to find answers in the DBpedia dataset.
Simple Questions
Simple Questions

• Single entity-property relationship:
  • *Qui est l’épouse de Barack Obama?* (Who is Barack Obama’s wife?)
Simple Questions

• Single entity-property relationship:
  • *Qui est l’épouse de Barack Obama?* (Who is Barack Obama’s wife?)

• Entity-entity relationship:
  • *Michelle Obama est-elle l’épouse de Barack Obama?* (Is Michelle Obama the wife of Barack Obama?)
Simple Questions

• Single entity-property relationship:
  • *Qui est l’épouse de Barack Obama?* (Who is Barack Obama’s wife?)

• Entity-entity relationship:
  • *Michelle Obama est-elle l’épouse de Barack Obama?* (Is Michelle Obama the wife of Barack Obama?)

• Limited support for more **Complex** questions (still a work in progress).
System overview process
System overview process

QUESTION

Pattern Matching
Question Type Analyzer
Translator
Entity Extractor
Custom Rules
DBpedia Properties
Property Extractor

SPARQL Query Builder

ANSWER
Question Type Analyzer

• Question Types
  • Date
  • Boolean
  • Number
  • Resource
Question Type Analyzer

• Question Types
  • Date
  • Boolean
  • Number
  • Resource

• Question Subtypes
  • List
  • Aggregation
Question Type Analyzer

• Question Types
  • Date
  • Boolean
  • Number
  • Resource

• Question Subtypes
  • List
  • Aggregation

• Pattern matching (both lexical and positional):
  • VERB-t-il(elle) = Boolean Question
Question Type Analyzer

• Question Types
  • Date
  • Boolean
  • Number
  • Resource

• Question Subtypes
  • List
  • Aggregation

• Pattern matching (both lexical and positional):
  • VERB-t-il(elle) = Boolean Question

• RESOURCE as generic type
Question Type Analyzer

• Question Types
  • Date
  • Boolean
  • Number
  • Resource

• Question Subtypes
  • List
  • Aggregation

• Pattern matching (both lexical and positional):
  • VERB-t-il(elle) = Boolean Question

• RESOURCE as generic type

• Aggregation categories:
  • Specific ordering (Ascending vs Descending)
  • Counting (all of, how many, number of, etc)
Question Solver
Question Solver

Entity Extractor

• Strip unnecessary words from the question.
• Find the noun groups and match them to DBpedia entities.
• Add manipulations with a distance based search priority.
• Use sameAs or a Translator module to convert to English entities.
Question Solver

**Entity Extractor**

- Strip unnecessary words from the question.
- Find the noun groups and match them to DBpedia entities.
- Add manipulations with a distance based search priority.
- Use `sameAs` or a Translator module to convert to English entities.

**Property Extractor**

- Remove the target Entity from the question string.
- Attempt to find the remaining tokens.
- Use DBpedia disambiguation links for more accurate results.
- Use custom lexicon mapping common properties to French expressions.
Question Solver
Question Solver

Translator

• Use the sameAs links available in DBpedia to translate to English.
• Use Google Translate API if no sameAs link is available.
• Some manual translations were required
Question Solver

Translator

• Use the sameAs links available in DBpedia to translate to English.
• Use Google Translate API if no sameAs link is available.
• Some manual translations were required

Custom rules

• Every Question Solver matches a particular Question Type.
• Solvers written as a separate and replaceable modules.
• Solvers implement their own custom rules for answering.
Aggregation Custom Rules
Aggregation Custom Rules

• Distinguish between requiring **ORDERING** and **COUNTING**
  • Pattern Matching
  • Keywords
    Ex: DET-ADVERB-ADJECTIVE for **ORDERING**
    Le nombre de ( number of), Combien de ( How many ), etc for **COUNTING**
**Aggregation Custom Rules**

• Distinguish between requiring **ORDERING** and **COUNTING**
  
  • Pattern Matching
  
  • Keywords
    
    Ex: DET-ADVERB-ADJECTIVE for **ORDERING**
    
    Le nombre de ( number of), Combien de ( How many ), etc for **COUNTING**

• Ordering questions :

  • Build a list of keywords to distinguish **ASCENDING** and **DESCENDING** order
    
    Ex: tallest, highest = **DESCENDING**
    
    smallest = **ASCENDING**

  • Exceptions : Dates
    
    Ex : Youngest = most recent( highest) date.
SPARQL Query Builder
SPARQL Query Builder

• Supports basic ASK and SELECT queries
SPARQL Query Builder

• Supports basic ASK and SELECT queries
  Ex : Quel est le budget de Pulp Fiction? (How much did Pulp Fiction cost?)
  Query : SELECT DISTINCT ?n WHERE {
    <http://dbpedia.org/resource/Pulp_Fiction>
    <http://dbpedia.org/ontology/budget> ?n . }

• Supports COUNT and ORDER BY queries for AGGREGATION Questions
  Ex : Combien de langues sont parlés au Turkmenistán? (How many languages are spoken in Turkmenistan?)
  Query: SELECT (COUNT(DISTINCT ?x) as ?c) WHERE {
    <http://dbpedia.org/resource/Turkmenistan>
    <http://dbpedia.org/ontology/language> ?x . }
SPARQL Query Builder

• Supports COUNT and ORDER BY queries for AGGREGATION Questions
  Ex: Quel est l’enfant de Meryl Streep? (Who is the oldest child of Meryl Streep?)
  Query: SELECT DISTINCT ?uri
  WHERE {
    ?uri dbo:birthDate ?d .
  }
  ORDER BY ASC(?d)
  OFFSET 0 LIMIT 1"
Simple Example

- *Michelle Obama est-elle l’épouse de Barack Obama ?* (Is Michelle Obama the wife of Barack Obama?)
- 1. Question Type: Boolean
- 2. Stripped question: Michelle Obama [] l’épouse de Barack Obama.
- 4. Entity relationship: *épouse de* (spouse)
- 5. SPARQL Query: `ASK WHERE {res:Barack_Obama dbo:spouse res:Michelle_Obama .}`
- 6. Result: TRUE
Aggregation Example

- *Quel est le plus grand joueur de basket?* (Who is the tallest basketball player?)
- 1. Question Type: Resource, Question Subtype: Aggregation
- 2. Stripped question: [] joueur de basket.
- 3. Entity extractor: joueur de basket (BasketballPlayer).
- 4. Aggregation keyword: le plus grand (tallest)
### Evaluation and Results
2017 train dataset

<table>
<thead>
<tr>
<th>Total Questions</th>
<th>211</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Questions</td>
<td>165</td>
</tr>
<tr>
<td>Correct Simple</td>
<td>150</td>
</tr>
<tr>
<td>Correct Total</td>
<td>164</td>
</tr>
<tr>
<td>Simple Average</td>
<td>0.909</td>
</tr>
<tr>
<td>Total Average</td>
<td>0.777</td>
</tr>
</tbody>
</table>
Future improvements

• Support more complex questions.
• Support multiple languages.
• Support more generic rules.
• Use multiple versions of Dbpedia.
Questions?

• Comments ?

• Insults ?