LC-QuAD
Large-scale Complex Question Answering Dataset

Priyansh Trivedi¹, Gaurav Maheshwari¹, Mohnish Dubey¹, Jens Lehmann¹,²

¹ University of Bonn, Bonn, Germany
² Fraunhofer IAIS, St. Augustin, Germany

ISWC 2017, Vienna
Outline

Question Answering
Motivation
Dataset(s)
LC-QuAD
Process of Creating LC-QuAD
Takeaways
Future Directions
Question Answering over KG

Understand the intent of a factual question, and return the implicit KB resource.

Typically treated as a translation problem from natural language to formal language.

Seen major advancement in the past five years.
Motivation
Challenges when set, can then be overcome.

In other words ...
Datasets precede Research

In 2013, Berant et al. released WebQuestions. Current State of the Art: 69% (Liang et al., 2016)

Over the past 8 years, Question Answering over Linked Data (QALD) challenge is being held. Over 38 submissions.
Other Incentives

Dearth of large QA dataset over DBpedia.

Traditional dataset generation methods are time consuming, do not scale.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Logical Forms</th>
<th>Complex Questions</th>
<th>Target KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free917 (Cai et al., 2013)</td>
<td>917</td>
<td>Yes</td>
<td>Yes</td>
<td>Freebase</td>
</tr>
<tr>
<td>WebQuestions (Berant et al., 2013)</td>
<td>5810</td>
<td>No</td>
<td>Yes</td>
<td>Freebase</td>
</tr>
<tr>
<td>SimpleQuestions (Bordes et al., 2015)</td>
<td>108442</td>
<td>No</td>
<td>No</td>
<td>Freebase</td>
</tr>
<tr>
<td>30M Factoid (Serban et al., 2016)</td>
<td>30000000</td>
<td>No</td>
<td>No</td>
<td>Freebase</td>
</tr>
<tr>
<td>QALD (Unger et al., 2016)</td>
<td>450</td>
<td>Yes</td>
<td>Yes</td>
<td>DBpedia</td>
</tr>
</tbody>
</table>

*unless unintentionally made complex
## Dataset(s)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Logical Forms</th>
<th>Complex Questions</th>
<th>Target KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free917 (Cai et al., 2013)</td>
<td>917</td>
<td>Yes</td>
<td>Yes</td>
<td>Freebase</td>
</tr>
<tr>
<td>WebQuestions (Berant et al., 2013)</td>
<td>5810</td>
<td>No</td>
<td>Yes</td>
<td>Freebase</td>
</tr>
<tr>
<td>SimpleQuestions (Bordes et al., 2015)</td>
<td>108,442</td>
<td>No</td>
<td>No</td>
<td>Freebase</td>
</tr>
<tr>
<td>30M Factoid (Serban et al., 2016)</td>
<td>30,000,000</td>
<td>No</td>
<td>No</td>
<td>Freebase</td>
</tr>
<tr>
<td>QALD (Unger et al., 2016)</td>
<td>450</td>
<td>Yes</td>
<td>Yes</td>
<td>DBpedia</td>
</tr>
<tr>
<td><strong>LC-QuAD</strong></td>
<td><strong>5000</strong></td>
<td><strong>Yes</strong></td>
<td><strong>Yes</strong></td>
<td><strong>DBpedia</strong></td>
</tr>
</tbody>
</table>
LC-QuAD
LC-QuAD

✔ has complex questions
LC-QuAD

✔ has complex questions
✔ has SPARQL queries
LC-QuAD

✔ has complex questions
✔ has SPARQL queries
✔ has boolean and aggregate based queries
LC-QuAD

✔ has complex questions
✔ has **SPARQL** queries
✔ has boolean and aggregate based queries
✔ is supervised *(gold standard)*
LC-QuAD

✔ has complex questions
✔ has **SPARQL** queries
✔ has boolean and aggregate based queries
✔ is supervised *(gold standard)*
✔ is extensible
LC-QuAD

✔ has complex questions
✔ has **SPARQL** queries
✔ has boolean and aggregate based queries
✔ is supervised *(gold standard)*
✔ is extensible
✔ is awesome 🍩
Dataset Creation Process
Traditionally

Natural Language Questions are collected/created.

Thereafter, their logical form is **manually** created.
Traditionally

Name someone influenced by J. R. R. Tolkien, who won the Hugo Award?

SELECT DISTINCT ?uri WHERE {
  ?uri dbpedia-owl:award dbr:Hugo_Award
}
Traditionally

Name someone influenced by J. R. R. Tolkien, who won the Hugo Award?

Prerequisites
✔ understanding KG
✔ understanding Schema
✔ understanding target formal language (SPARQL)
✔ no room for errors
✔ understand NL
Inverting the Process

SELECT DISTINCT ?uri WHERE {
  ?uri dbo:award dbr:Hugo_Award
}

Name someone influenced by J. R. R. Tolkien, who won the Hugo Award?
Inverting the Process

SELECT DISTINCT ?uri WHERE {
    ?uri dbo:award dbr:Hugo_Award
}

Name someone influenced by J. R. R. Tolkien, who won the Hugo Award?
Upon Further Simplification

SELECT DISTINCT ?uri WHERE {
  ?uri dbo:award dbr:Hugo_Award
}

What is person whose influenced by is J. R. R. Tolkien, and award is the Hugo Award.

Name someone influenced by J. R. R. Tolkien, who won the Hugo Award?
Upon Further Simplification

SELECT DISTINCT ?uri WHERE {
  ?uri dbo:award dbr:Hugo_Award }
Upon Further Simplification

Increases the speed of creating questions.

Reduces domain expertise required.

Can afford slight errors.

Allowing us to scale up!

Prerequisites

✔ understanding KG Schema
✔ understanding target formal language (SPARQL)
✔ no room for errors.
✔ understand NL
Remaining Challenges

Automatically create SPARQL queries.

Convert SPARQL queries to intermediary NLQs.
Automatically create SPARQL Queries

Create SPARQL Templates.

```
SELECT DISTINCT ?uri WHERE {
  ?uri e_to_e_out1 e_out1.
  ?uri e_to_e_out2 e_out2
}
```
Automatically create SPARQL Queries

Create SPARQL Templates.
Automatically create SPARQL Queries

Manually select entities as answers to our queries.

Stephen King
Automatically create SPARQL Queries

Collect the 2-hop subgraph around these entities.
Automatically create SPARQL Queries

Juxtapose the SPARQL triple pattern on this subgraph.
SPARQL Queries Generated

SELECT DISTINCT ?uri WHERE {
    ?uri dbo:award dbr:Hugo_Award
}

33
Remaining Challenges

Automatically create SPARQL queries.

Convert SPARQL queries to intermediary NLQs.
Creating intermediary NLQs

SELECT DISTINCT ?uri WHERE {
  ?uri dbo:award dbr:Hugo_Award
}

Whose influenced by is J. R. R. Tolkien, and award is the Hugo Award.
Question Templates (NNQT)

SELECT DISTINCT ?uri WHERE {
  ?uri e_to_e_out1 e_out1. ?uri e_to_e_out2 e_out2
}

Whose e_to_e_out1 is e_out1, and e_to_e_out2 is the e_out2.
Template Instances

SELECT DISTINCT ?uri WHERE {
    ?uri dbo:award dbr:Hugo_Award
}

Whose influenced by is J. R. R. Tolkien, and award is the Hugo Award.
SELECT DISTINCT ?uri WHERE {
  ?uri dbo:award dbr:Hugo_Award
}

Whose influenced by is J. R. R. Tolkien, and award is the Hugo Award.

Name someone influenced by J. R. R. Tolkien, who won the Hugo Award?
SELECT DISTINCT ?uri WHERE {
  ?uri dbo:award dbr:Hugo_Award
}

Whose influenced by is J. R. R. Tolkien, and award is the Hugo Award.

Name someone influenced by J. R. R. Tolkien, who won the Hugo Award?
Name someone influenced by J. R. R. Tolkien, who won the Hugo Award?

SELECT DISTINCT ?uri WHERE {
    ?uri dbo:award dbr:Hugo_Award
}

Whose influenced by is J. R. R. Tolkien, and award is the Hugo Award.
Manual Work

Two Step Process:

1. First Human Intervention corrects the grammar of NNQTs, and sometimes paraphrases the questions.
2. Verifies the Question, and in case of missing space, typo etc, fixes them.
What is the <Band> whose <current members> are <Joe Hahn> and <Chester Bennington>?

SELECT DISTINCT ?uri WHERE {
  ?uri dbp:currentMembers dbr:Chester_Bennington }
Discussion
Dataset Characteristics

5,000 Questions
33 SPARQL Templates
18% Simple Questions

12.29\textsubscript{w} Avg. Question Size
04 ‘16 DBpedia Version
150+ Downloads*

*as of 16th October, 2017
Controlling Size and Variety

Too many queries generated per subgraph.
Predicate links disproportionate (eg. dbp:birthplace).
Metadata triples.

Filters based on predicate whitelist.
Stochastically prune the subgraph.
Limitations

No literals are included in the questions.

No UNION, OPTIONAL queries.

No conditional aggregates.

No out-of-scope questions.
Future Directions

Creating baselines.

Automatic grammar correction.

Complex SPARQL templates.

Keeping up with DBpedia versions.
References, Citations


This presentation uses licensed works by generous individuals/organizations, namely:

- Harald Nezbeda (Vienna, Slide 2)
- Icons8.com (multiple slides)
- Pinguino’s Flickr Account (Slide 29, 30)
Questions?

See what I did there?
See for yourself.

lc-quad.sda.tech