Formal Query Generation for Question Answering over Knowledge Bases

Hamid Zafar, Giulio Napolitano, Jens Lehmann
Agenda

• Introduction
• SPARQL Query Generator (SQG)
  • Capture subgraph
  • Find valid walks
  • Rank queries
• Empirical results
• Summary
Introduction

Question answering over Knowledge graphs
What are some artists on the show whose opening theme is Send It On?

```
SELECT DISTINCT ?uri WHERE {
  ?x <https://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/TelevisionShow>}
```
# Common Architectures

## End-to-End
- Single process
  + No error propagation
  - Limited support for complex questions

## Pipeline
- Consists of multiple components including
  - Named Entity Disambiguation
  - Relation Extraction
  - Query Generation (QG)
  + Reusable components
  + Limited focus
  - Propagate the error along the pipeline
Pipeline Architecture

Query Generation Component

- The **Query Generation** is a common component in QA systems
- Error analysis from [4] showed the importance of the **Query Generation** and its effect on the overall performance of the QA pipeline

Requirements of Query Generation

- Cope with large-scale KGs
- Ability to manage noisy input to handle error propagation
- Question type identification
- Support for composite question
- Syntactic ambiguity of the input question
SPARQL Query Generation (SQG)

- **Hypothesis:** The formal interpretation of the question is a walk in the KG which contains the target entities and relations of the input questions plus the answer node.

- **Inputs:** Question along with the linked entities and relations
What are some artists on the show whose opening theme is "Send It On"?

Linked entities/relation from KG

- dbo:Artist
- dbo:artists
- dbo:artist/coCreator
- dbo:storyBoardArtists

- dbo:show
  - dbo:show"name

- dbo:openingTheme
  - dbo:openingIn
  - dbo:reOpening
  - dbo:openingThemeSong

- dbr:Send_It_On_(D’Angelo_song)
- dbr:Send_It_On_(Disney’s_Friends_for_Change_song)
- dbr:Bring_It_On...Bring_It_On
Architecture

- Question
- +Annotations
- Feature Identif.
- Query Generation
- D. Parse Tree
- Ranking Model
- Ordered Queries

features

candidate queries

tree rep.

of question
Feature identification

• SVM model on tf-idf representation of input questions
  • Establish the type of the question (e.g. boolean, count or list)
    • Affects the query formation process
  • Hidden relation identification (e.g. what is the birthplace of X and Y)
Query Generation- Capturing subgraph

- Capture the connected subgraph which contains the linked entities/relation and arbitrary unbounded nodes.
- Limited to one and two hop distance
Valid walks

• Walk: A walk in a knowledge graph is a sequence of edges along the nodes they connect.

• Valid Walk: A walk is valid w.r.t a set of entities and relations, if and only if it contains all of them.
Extract valid walks from the subgraph
Extract valid walks from the subgraph
Ranking Model

- **Goal**: Rank the valid walks w.r.t. the semantic of the input question

- **Hypothesis**: the structure of the walks is a distinctive feature to distinguish the similarity between the candidate walks and the input question
LSTM
Tree-LSTM

Dependency Parsing Tree
Tree-Rep. of Candidate Queries

(a) What are some artists on the show whose opening theme is Send It On?

(b) What TV shows with Send It On as their opening theme are the artists of Send it On?

(c) TV shows with Send It On as their opening theme are the artists of what?

(d) Which TV shows has a opening them which is among the artists of Send it On?
Tree-LSTM as Ranking model

- Candidate walks
- Query Tree-LSTM
- Latent repr. of query
- Similarity Function
- Sim. Score
- Question
- Question Tree-LSTM
- Latent repr. of question
Evaluation Setup

• Dataset: LC-QuAD
  • 5000 Q/A pairs with different complexity and types of questions

• Baseline QA systems:
  • Sina Shekarpour et al. "SINA: semantic interpretation of user queries for question answering on interlinked data. Web Semant”
  • NLIWOD https://github.com/dice-group/NLIWOD

• Baseline for the Ranking Model:
  • LSTM
Evaluation - Scenarios

- Top-1 correct: Questions annotated w. correct entities/relations
- Top-5 EARL+correct: Questions annotated w. list of candidate entities/relations (correct ones forcefully injected if not exists)
- Top-5 EARL: Questions annotated w. list of candidate entities/relations

- EARL: an entity/relation linking tool, Dubey et al. “EARL: joint entity and relation linking for question answering over knowledge graphs”
Evaluation - Ranking model

Considering the tree-representation significantly improves the performance.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>LSTM F1-measure</th>
<th>Tree-LSTM F1-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-1 correct</td>
<td>0.54</td>
<td>0.75</td>
</tr>
<tr>
<td>Top-1 EARL+correct</td>
<td>0.41</td>
<td>0.84</td>
</tr>
<tr>
<td>Top-1 EARL</td>
<td>0.32</td>
<td>0.74</td>
</tr>
</tbody>
</table>

- Micro F1-measure

Better generalization
## Evaluation- vs. Baselines

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sina*</td>
<td>0.23</td>
<td>0.25</td>
<td>0.24</td>
</tr>
<tr>
<td>NLIWOD*</td>
<td>0.48</td>
<td>0.49</td>
<td>0.48</td>
</tr>
<tr>
<td>SQG</td>
<td>0.76</td>
<td>0.74</td>
<td>0.75</td>
</tr>
</tbody>
</table>

* Sina and NLIWOD results are taken from Singh et al. "Why reinvent the wheel—let's build question answering systems together”
- on a subset of LC-QuAD containing 3,200 questions
Summary

- Reusable and Scalable approach
- Managing noisy annotations
- Exploit structural similarity of input question and candidate queries

Thanks you for you attention.

Questions?

Code is available at:
https://github.com/AskNowQA/SQG