UNIVERSITÄT PADERBORN

DATA SCIENCE RESEARCH GROUP

RECENT ADVANCES IN NATURAL LANGUAGE PROCESSING

TOPIC: ENTITY ENABLED RELATION LINKING

UNIVERSITÄT PADERBORN

Introduction

- Relation Linking

Natural Langauge Question + Knowledge Graph $\xrightarrow{\text{Task}}$ Relation Identification

- Popular Knowledge Graphs :

Source: https://bit.ly/2L5wlBB

Source: https://bit.ly/3apSska

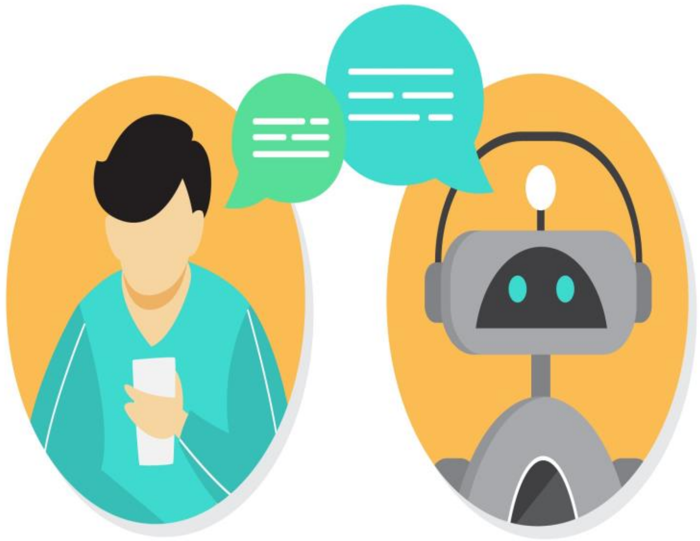
Source: https://en.wikipedia.org/wiki/YAGO_(database)

Source: https://www.wikidata.org/wiki/Wikidata:Main_Page

UNIVERSITÄT PADERBORN

Existing Approaches (a)

- Development of **end-to-end Question Answering Systems (QA)**
- Mapping / Linking entites and relations
- Freebase Knowledge Graph
- Extract correct answers

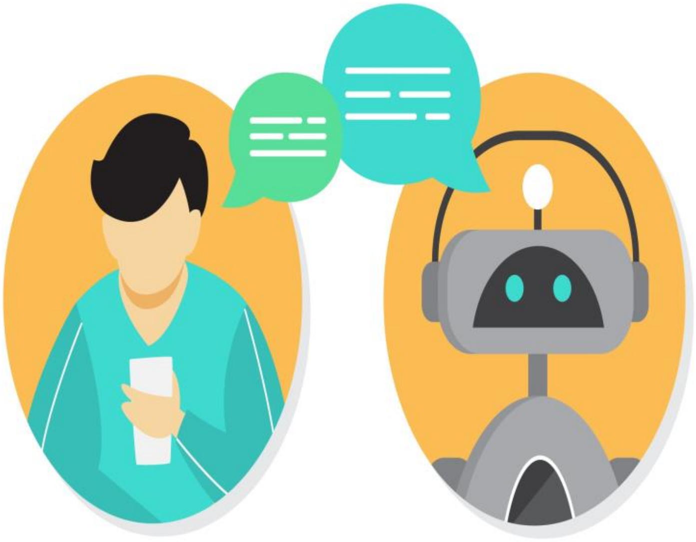


- **Why** Freebase Knowledge Graph?
  - Availabilty of large training Data in benchmarks.

UNIVERSITÄT PADERBORN

Existing Approaches (b)

- Development of **end-to-end Question Answering Systems** focused on **Semantic Parsing (SQA)**
- Semantics associated with Input Question
- QA pipeline
- Extract correct answers



UNIVERSITÄT PADERBORN

Existing Challenges :

- **HOW?**
  - extraction of entity and relation candidates in given input question?
  - link the relation and entity candidates to knowledge graphs?

UNIVERSITÄT PADERBORN

Novel Approach

- **Collaborative QA systems**
- Main idea: Much focus on relation linking, than entity linking.
  - Existing Frameworks for QA systems: Qanary, OKBQA, Frankenstein
- Reuse QA components.
- Embedding in Frankenstein frameworks.
- Compare state-of-art approaches : failure in precision and runtime.
- Because, ignoring Context of entities.

UNIVERSITÄT PADERBORN

Novel Approach (continued..)

- Hence, Candidate Property List is used to support relation linking over Dbpedia knowledge graph.
- This property list is used in constructing SPARQL Queries.
- Improved precision and runtime.

UNIVERSITÄT PADERBORN

Background :

- **Knowledge Graph:**
  - Given data sub-graph A and schema sub-graph T
  - we define knowledge Graph, $\mathcal{G} = T\ U\ A$
  - Facts in data sub-graph A is represented as triples :
    1. Relation Insertion (h,r,t), where h(t) is head(tail) entity and r is relation
    2. Type Assertion (e, rdf:type, C), where e is entity , rdf:type is relation instance and C is type.
  - Schema sub-graph τ incudes Type Inclusion Axioms and Relation Axioms.

UNIVERSITÄT PADERBORN

Background (Contnued..)

- **Problem Statement**
  - Given Knowledge Graph $\mathcal{G}$, an input question q, set of entities Eq.
  - Identify : set of Relations Rq.

UNIVERSITÄT PADERBORN

Related Work

- Resoursces and Systems for relation linking over Dbpedia

1. **PATTY** : Textual patterns denoting binary relations.
2. **BOA** : Predicate representations.
3. **SIBKB** : Searching mechanisms for linking the natural language to knowledge graphs.
4. **EARL** : Joint entity and Relation linking.
5. **Rematch** : Dependency parsing.

UNIVERSITÄT PADERBORN

Approach

- Analysis of 100 randomly chosen QA pairs.
- **Preliminary Observations**
  - Most predicates SPARQL queries associated with input question are properties of entities in natural question.
  - No natural language label of relation in the question given.

    Which comic characters are painted by Bill Finger?
  - Hidden Relations in question given.

    How many shows does HBO have?

UNIVERSITÄT PADERBORN

Approach (continued..)

Hypothesis: "The relations in questions are properties of the entities occurring in the question or properties of the types of these entities."

Developing Hypotheses on EERL Framework:

1. **Relation Keyword Extraction** – extract relation keywords.

Example: $q = \text{"}Which\ comic\ characters\ are\ painted\ by\ Bill\ Finger?\text{"}$

$$\downarrow$$

$$relation\ phrase = \text{"}painted\ by\text{"}$$

2. **Keyword based Relation Expansion** – expand extracted relation keywords using background knowledge

$$\begin{array}{c} relation\ phrase = \text{"painted by"} \\ + \\ \text{background knowldge} = \text{"PATTY"} \end{array} \rightarrow \text{"painter "}$$ (most suitable relation pharse)

UNIVERSITÄT PADERBORN

Approach (continued..)

3. **Entity linking –** link entities to DBpedia IRIs.

- Given Knowledge Graph, $\mathrm{G} = T\ UA$
- 2 types :
- Explicit Relation $\rightarrow$ (A)
- q = *"The spouse of Barack Obama is Michelle Obama"*
- Represented as Triplet in RDF

(dbr: Barack_Obama, dbo:spouse, dbr:Michelle_Obama)

- Implicit Relation $\rightarrow$ (T)
- q = *"Barack Obama is born in Honolulu"*
- *dbo:birthPlace* (explicit relation)
- $\mathrm{dbo{:}\ Agent} \rightarrow \mathrm{dbo{:}\ HomeTown}$
- *dbr:Barack: (dbr:Barack_Obama,rdfs:type,dbo:Agent)*

and

(dbo:HomeTown,rdfs:domain, dbo:Agent).

UNIVERSITÄT PADERBORN

Approach (continued..)

4.Entity-Based relation Expansion – to form candidate property list (by 2 Expansions)

- <u>**Expansion**</u> 1 - Explicit Property List (EPL)
- Fetch property set from the instance triples A.
- Only ontologies with respect to entity.
- *q = "Which comic characters are painted by Bill Finger?"*
- *dbo:painter* (i.e most suitable relation phrase)
- Expansion 1 on dbo

  EPL – *dbr:Bill_Finger*

  result – dbo:creator

- <u>**Expansion 2**</u> – Implicit Property List (IPL)
- Iteration based on reasoning from T
- Get domain and ranges from schema T. **Global** domain and ranges + **Local** Domain and ranges
- *q = "How many shows does HBO have?"*
- *dbr:HBO* (EPL)
- *dbo:producer* ((i.e most suitable relation phrase)
- *on dbr:HBO : (local)*

  rdf:type (dbr:HBO) → dbo:Broadcaster

- *dbo:Broadcaster → dbo:channel (global)*

UNIVERSITÄT PADERBORN

Approach (continued..)

5. **Relation linking –** To get best relation Rq (SIBKB approach).
   - Re-rank candidates from EPL or IPL list.
   - Three approaches
     1. **Existence** Re-ranking and Extending - existence of relations (SIBKB)
     2. **LD** Re-ranking and Extending – calculate Levenshtein Distance
     3. **Synonym** Re-ranking and Extending – calculate distance between Synonyms set and Property Candidate list.

UNIVERSITÄT PADERBORN

Experimental Setup

- **Data Sets**
  - Question Answering over Linked Datta challenge (QALD) : 58% of simple questions.
    - QALD-5 has 340 questions and QALD-7 has 215 questions
  - LC-QuAd : 5000 questions, with 20% simple questions
- **SOTA systems**
  - SIBKB
  - Rematch
  - EARL
- **Experimental Settings**
  - 1 virtual server, with 8 cores and 32 GB RAM running on $Ubuntu$ 16.04.3 operating system
  - Frankenstein Resource Platform

UNIVERSITÄT PADERBORN

Results

- Proposed Novel Approach out - performs baseline systems in all 3 datasets.
- Does not have major performance drop for complex questions.

Table 1: Performance of EERL Framework compared to various Relation Linking tools

QA Component	Dataset	Precision	Recall	F-score
SIBKB	QALD-5	0.27	0.34	0.29
ReMatch	QALD-5	0.36	0.39	0.37
EARL	QALD-5	0.17	0.21	0.19
EERL	QALD-5	**0.43**	**0.49**	**0.45**
SIBKB	QALD-7	0.33	0.35	0.34
ReMatch	QALD-7	0.35	0.38	0.37
EARL	QALD-7	0.30	0.31	0.30
EERL	QALD-7	**0.42**	**0.46**	**0.43**
SIBKB	LC-QuAD	0.15	0.18	0.16
ReMatch	LC-QuAD	0.18	0.20	0.19
EARL	LC-QuAD	0.20	0.25	0.21
EERL	LC-QuAD	**0.53**	**0.58**	**0.55**

Source: https://bit.ly/2MCiU1Y

UNIVERSITÄT PADERBORN

Discussion

- **Pro's**
  - Much focus on Contextual properties about entities – increased performance.
  - Using Property Candidate list : speeds up retrieving relations.
  - Also, prevent filtering of candidates in Re-ranking step.
- **Con's**
  - limited performance when, Ontology is not defined.
  - KGs with no clear and correct definition.

Table 2 : Run Time (average_seconds/question)

system	QALD-7	LC-QuAD
SIBKB	1.1	2.2
ReMatch	110	130
EERL	**1.3**	**1.8**

Source : https://bit.ly/3cC8kmH

UNIVERSITÄT PADERBORN

Thank you ☺