



Build a **Question Answering** system overnight

Jens Lehmann, Gaurav Maheshwari, Priyansh Trivedi, Mohnish Dubey, Denis Lukovnikov,

ESWC 2019, Portoroz

Outline

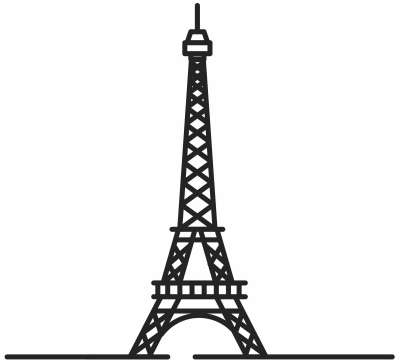
Introduction to QA

QA Approaches Overview

Neural Ranking Approach

Hands on

Introduction

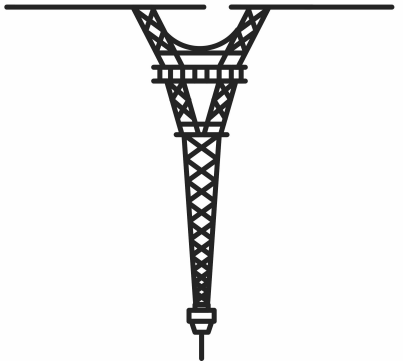


Factoid Questions

How high is the Eiffel Tower?

When was LOTR released?

Where was ESWC held in 2018?



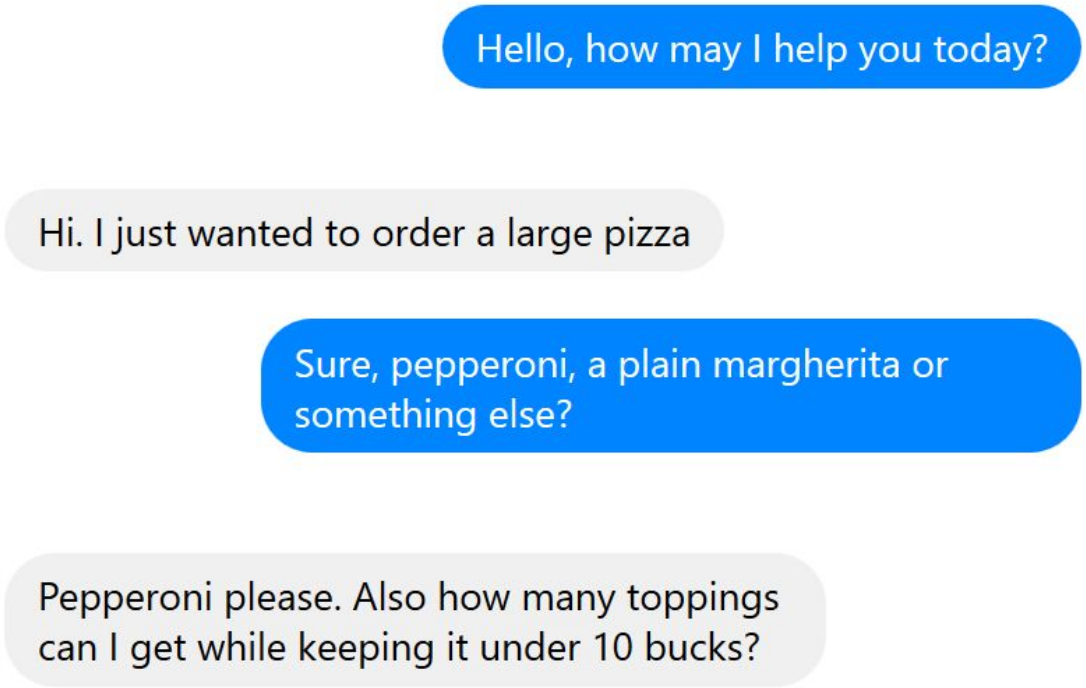
Non Factoid Questions

Why is the Eiffel tower in Paris?

Why is LOTR sooooo good?

Where should ESWC 2020 be held?

Factoid?


Hello, how may I help you today?
Hi. I just wanted to order a large pizza
Sure, pepperoni, a plain margherita or something else?
Pepperoni please. Also how many toppings can I get while keeping it under 10 bucks?

Domain Specific QA

Q: How do I get from Venice to Portoroz?

Q: Why is my car making this weird noise?

Q: How much did the consumer price index differ b/w greece and EU average in 2008?

... a festival called **Wianki** (Polish for **Wreaths**) have become a tradition and a yearly event in the programme of cultural events in Warsaw. The festival traces its roots to a peaceful pagan ritual where maidens would float their **wreaths** of herbs on the water to predict when they would be married, and to whom ...

Reading Comprehension QA

What is the polish word for wreaths?

Visual Question Answering

Who is wearing glasses?

man

woman

Where is the child sitting?

fridge

arms

Is the umbrella upside down?

yes

no

How many children are in the bed?

2

1

Images from http://www.visualqa.org Goyal et al. '17

QA over Knowledge Graphs

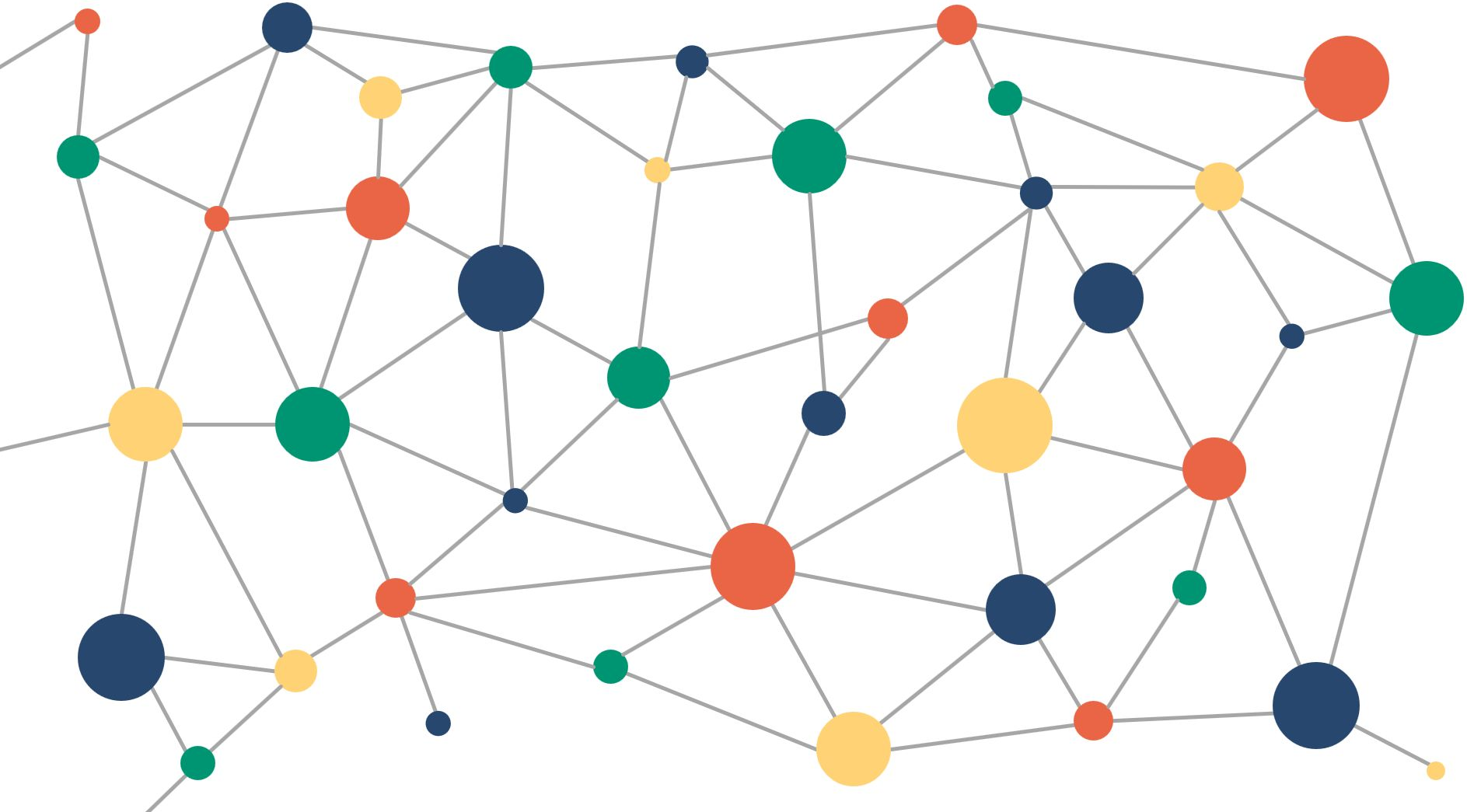
Given:

- a natural language question
- a source knowledge graph

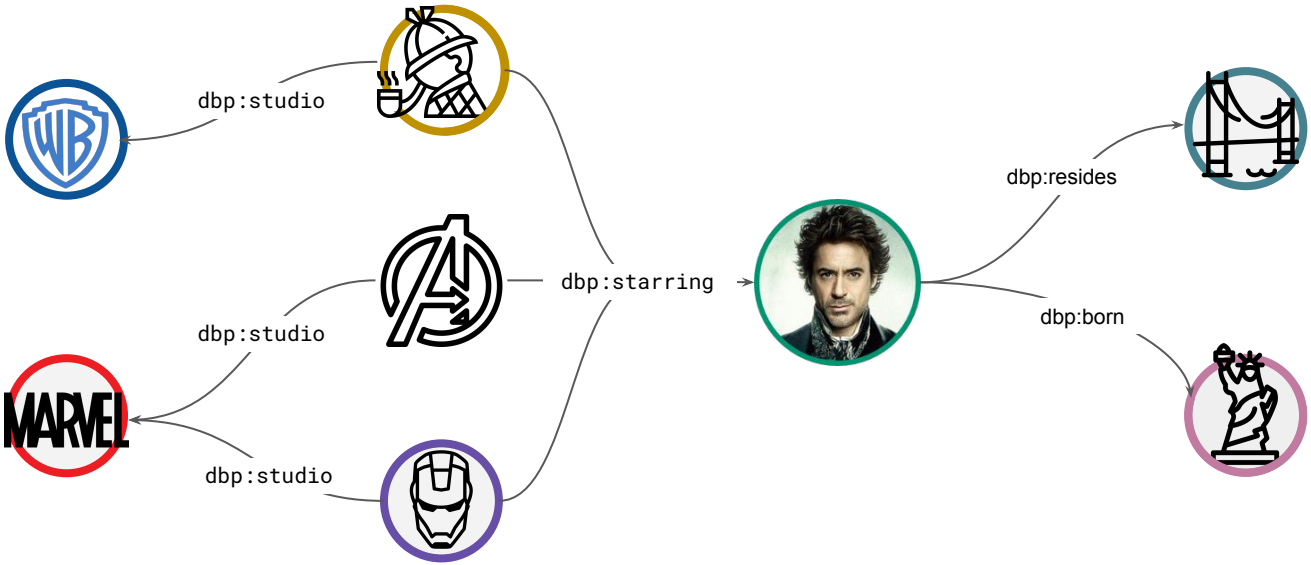
Find the subset of the *knowledge graph* intended to be the answer of the question.

Knowledge Graphs

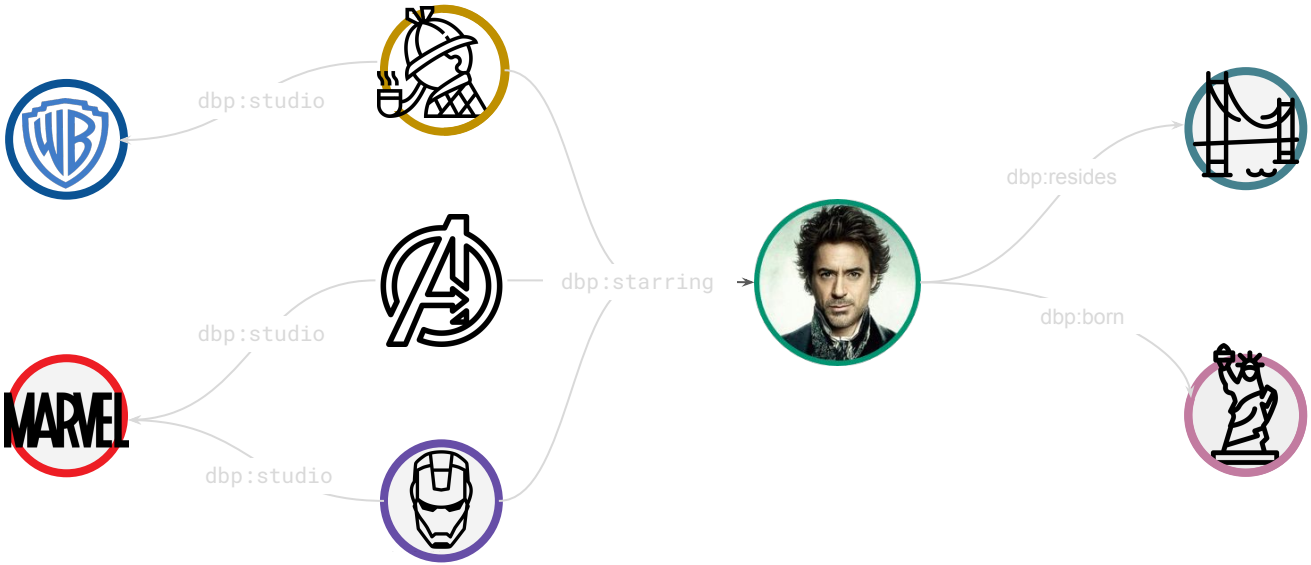
"*Things instead of strings*"



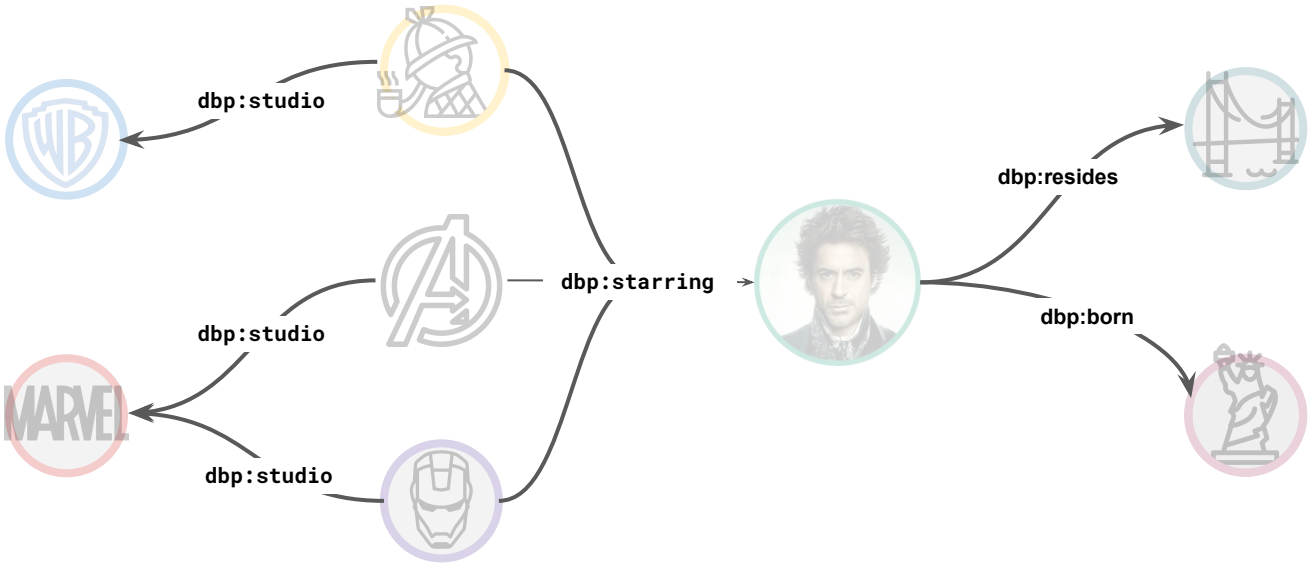
Knowledge Graph (KG)



KG Entities


dbp:studio
dbp:studio
dbp:studio
dbp:starring
dbp:resides
dbp:born
MARVEL

KG Predicates



KGs are large

DBpedia[1]: < 6.0 million entities (2016-04 edition).

Freebase[2]: 1.9B triples. Depreciated.

Wikidata[3]: 47M entities.

KGs are large **and expressive**

Need **formal query language** to manipulate or extract information from KG.

Examples SQL, Datalog

Formal Query Languages

Formal Query Languages

Structure (Grammar)

Semantics

Example:

Natural Language
(not formal)

Who made Iron man?

Example:

Natural Language
(not formal)

λ-DCS

```
dbp:creators(dbr:Iron_Man, ?answer)
```

Example:

Natural Language
(not formal)

λ-DCS

SPARQL

```
SELECT ?uri WHERE {
    dbr:Iron_Man dbp:creators ?uri.
}
```

SPARQL Example

How many Marvel movies was Robert Downey Jr. casted in?

SPARQL Example

How many **Marvel** **movies** was **Robert Downey Jr.** **casted** in?

```
SELECT COUNT(?uri) WHERE {
    ?uri dbp:studio dbr:Marvel_Studios.
    ?uri dbo:starring dbr:Robert_Downey_Jr
}
```

Example

How many Marvel movies was Robert Downey Jr. casted in?

```
SELECT COUNT(?uri) WHERE {
    ?uri dbp:studio dbr:Marvel_Studios.
    ?uri dbo:starring dbr:Robert_Downey_Jr
}
```

Example

How many Marvel movies was Robert Downey Jr. casted in?

```
SELECT COUNT(?uri) WHERE {
    ?uri dbp:studio dbr:Marvel_Studios.
    ?uri dbo:starring dbr:Robert_Downey_Jr
}
```

Example

How many Marvel movies was Robert Downey Jr. casted in?

```
SELECT COUNT(?uri) WHERE {
    ?uri dbp:studio dbr:Marvel_Studios.
    ?uri dbo:starring dbr:Robert_Downey_Jr
}
```

Example: Simple Questions

Who made Iron man?

```
SELECT ?uri WHERE {
    dbr:Iron_Man dbp:creators ?uri.
}
```

Example: Complex Questions

Name all **Warner Brothers** movies.

```
SELECT ?uri WHERE {
    ?uri dbp:studio dbr:Warner_Bros .
}
```

Example: Complex Questions

Name all **Warner Brothers** **movies**?

```
SELECT ?uri WHERE {
    ?uri dbp:studio dbr:Warner_Bros .
    ?uri rdf:type dbo:Film .
}
```

Example: Complex Questions

Name all Warner Brothers movies released post 1990?

```
SELECT ?uri WHERE {
    ?uri dbp:studio dbr:Warner_Bros .
    ?uri rdf:type dbo:Film .
    ?uri dbo:releaseDate ?date .
    FILTER (?date >= xsd:date("1990-01-01"))
}
```

Example: Boolean Queries

Did Robert Downey Junior act in Iron Man?

```
ASK WHERE {
    dbr:Iron_Man_(2008_film)
    dbo:starring
    dbr:Robert_Downey_Jr
}
```

QA as NL to SPARQL

QA can be seen as a task of converting natural language question to SPARQL (formal) queries.

Semantic parsing?

Question Answering

Natural Language Variations

Name all the movies in which Robert Downey Jr Acted?

Which movies have RDJ?

Flicks where I can see Robert DJ?

Find me all the films casting Rober Downey Jr ?

List all the movies starring Robert Downey Junior?

RDJ has acted in which movies?

Entity Linking (EL)

Name all the movies in which **Robert Downey Jr** Acted?

Which movies have **RDJ**?

Flicks where I can see **Robert DJ**?

Find me all the films casting **Rober Downey Jr** ?

List all the movies starring **Robert Downey Junior**?

RDJ has acted in which movies?

EL - **Disambiguating Entities**

Who is the CEO of **Apple**?

 **Apple** belongs to which genus?

movie character

 **Downey** played **Iron Man** in which year?

Who is the alter ego of **Iron man**?

comic character

Relation Linking (RL)

dbo:starring

Name all the movies in which **Robert Downey Jr** **Acted**?

Which movies have **RDJ**?

Flicks where I can see **Robert DJ**?

Find me all the films **casting** **Rober Downey Jr** ?

List all the movies **starring** **Robert Downey Junior**?

RDJ **has acted** in which movies?

RL - **Implicit Predicates**

Name all the movies in which **Robert Downey Jr** **Acted**?

Which movies have **RDJ**?

Flicks where I can see **Robert DJ**?

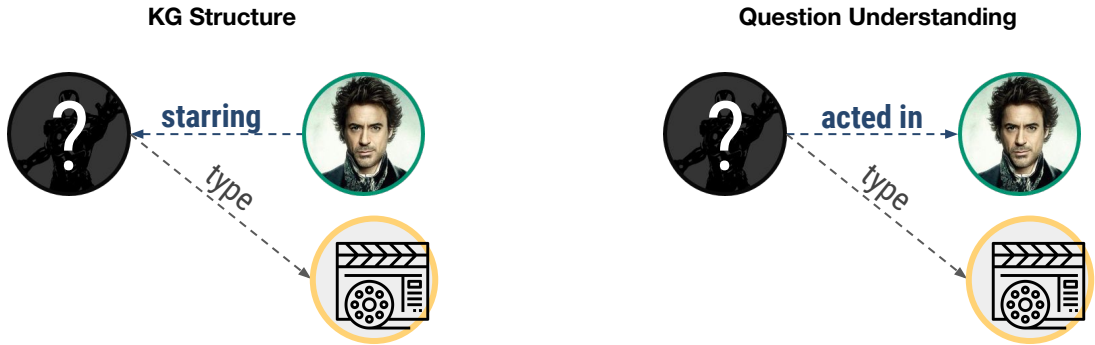
Find me all the films **casting** **Rober Downey Jr** ?

List all the movies **starring** **Robert Downey Junior**?

RDJ **has acted** in which movies?

KG Structure Mismatch

Name all the movies in which **Robert Downey Jr** **Acted**?



Auxiliary Constraints

Name everything where Robert Downey Jr Acted?

```
SELECT ?uri WHERE {
    ?uri dbo:starring dbr:Robert_Downey_Jr .
}
```

Auxiliary Constraints

Name everything where Robert Downey Jr Acted?

Questions might ask for:

```
SELECT ?uri WHERE {
    ?uri dbo:starring dbr:Robert_Downey_Jr .
}
```

Auxiliary Constraints

Name all the **movies** in which **Robert Downey Jr** **Acted**?

Questions might ask for:

Specific **type** of answers

```
SELECT ?uri WHERE {
    ?uri dbo:starring dbr:Robert_Downey_Jr .
    ?uri rdf:type dbo:Film
}
```

Auxiliary Constraints

How many movies has RDJ **acted** in?

Questions might ask for:

Count the number of results.

```
SELECT count(?uri) WHERE {
    ?uri dbo:starring dbr:Robert_Downey_Jr .
}
```

Auxiliary Constraints

Name the movies RDJ acted in after 2005?

Questions might ask for:

Filter the results based on some arbitrary metric.

```
SELECT ?uri WHERE {
    ?uri dbo:starring dbr:Robert_Downey_Jr .
    ?uri dbo:releaseDate ?date .
    FILTER (?date >= xsd:date("2005-01-01"))
}
```

Challenges

Entity Linking

Predicate Linking

KG structure mis-match

Auxiliary Constraints

Solution

Recap

QA can be seen as a task of converting natural language question to formal queries.

Two major approaches

creating formal queries from question

selecting a formal query based on question

Image created by Dashu83 - Freepik.com

1. **Creating** queries

Closely understand the question structure.

Find KG resources referenced in it.

Generate a parse composed of KG resources.

Semantic Parsing. (Grounded)

1. **Creating** queries

Closely understand the question structure.

Find KG resources referenced in it.

Generate a parse composed of KG resources.

Semantic Parsing. (Grounded)

1. Creating queries

Closely understand the question structure.

Find KG resources referenced in it.

Generate a parse composed of KG resources.

Semantic Parsing. (Grounded)

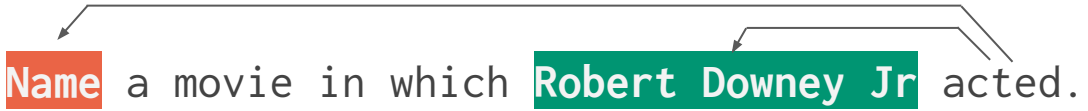
Semantic Parsing

```
Name a movie in which Robert Downey Jr acted.
```

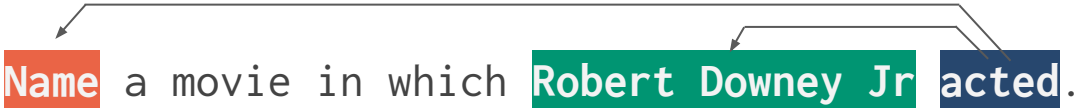
Semantic Parsing

Semantic Parsing

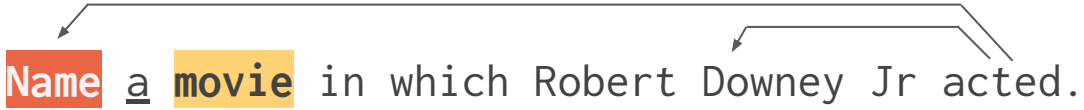
Semantic Parsing

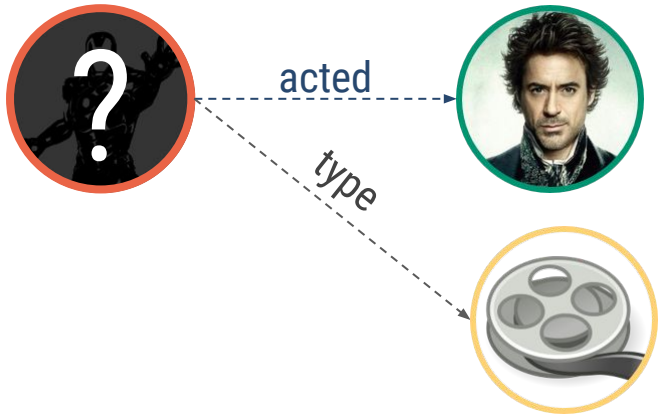


Semantic Parsing



Semantic Parsing

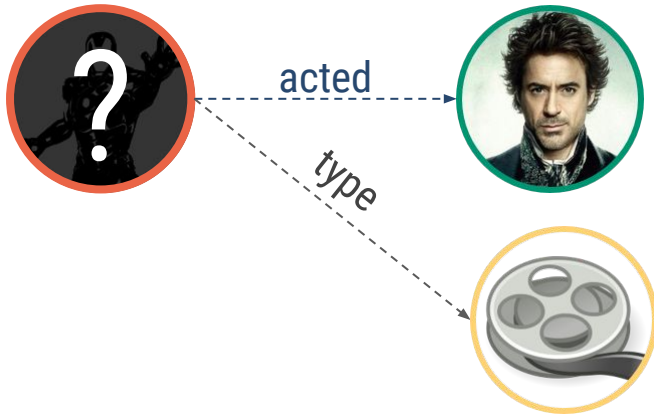




Semantic Parsing

Name a movie in which Robert Downey Jr acted.

```
SELECT ?uri WHERE {
  ?uri dbo:starring dbr:Robert_Downey_Jr .
  ?uri rdf:type dbo:Film .
}
```



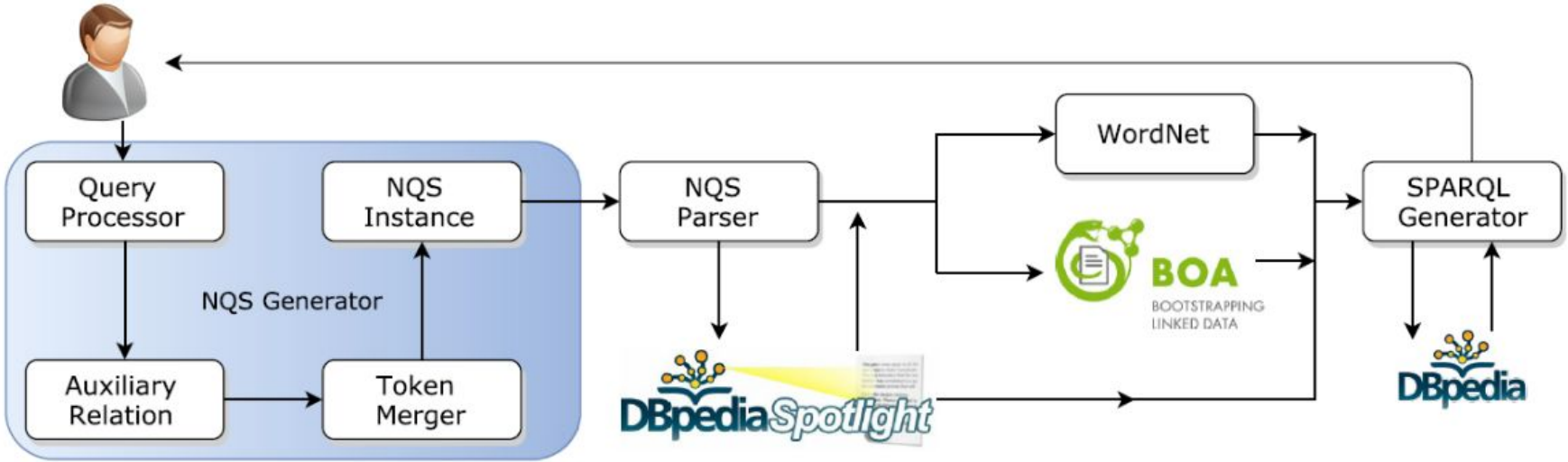
Approaches

Template Matching

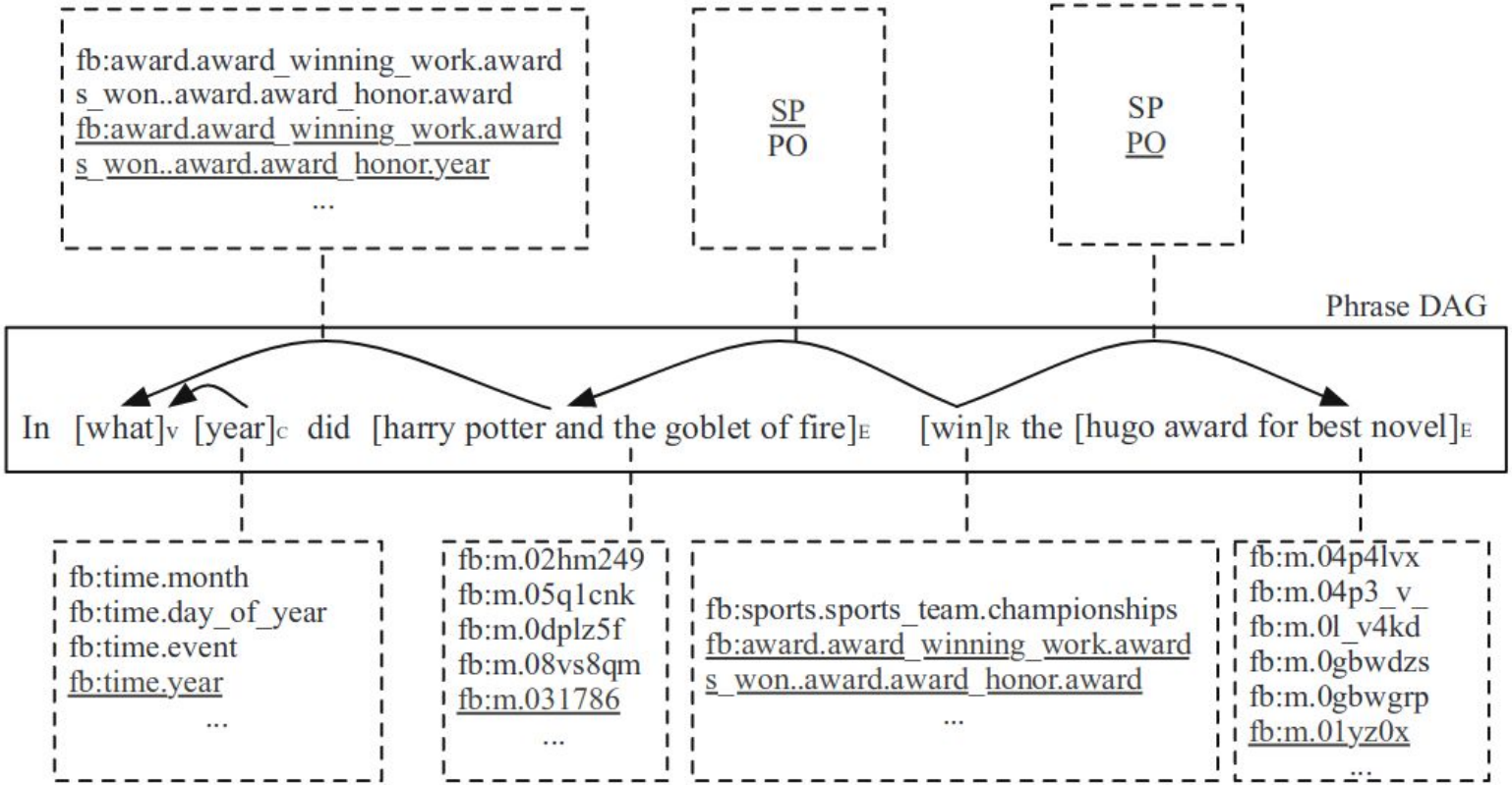
Template Learning

Grounding Open Semantic Parsers

Neural Generative Approaches



Dubey, Mohnish, et al. "Asknow: A framework for natural language query formalization in sparql." *European Semantic Web Conference*. Springer, Cham, 2016.



Xu, Kun, Yansong Feng, and Dongyan Zhao. "Xser@ qald-4: Answering natural language questions via phrasal semantic parsing." Working Notes for CLEF 2014 Conference. 2014.

2. Selecting a query

Closely follow the KG structure.

Make the query and question comparable.

Discriminative Model (sp. ranking)

2. Selecting a query

Specifically

- Creates a set of formal query candidates
- Rank them with respect to Question

Problems

But there could be

- **Millions** of queries
- Query syntaxes are difficult to **represent**.

Millions of SPARQL

Only choose queries which have **all the entities** mentioned in the question a.k.a **topic entities**.

Achieved by entity linking system

Restrict the number of predicates in queries to just few.

Millions of SPARQL

```
Give me a list of everything where Robert Downey Jr Acted?
```

Millions of SPARQL

Give me a list of everything where **Robert Downey Jr** Acted?

Topic Entity

Millions of SPARQL

Give me a list of everything where **Robert Downey Jr** Acted?

✅ `SELECT ?uri WHERE {dbr:Robert_Downey_Jr dbp:lives ?uri.}`

✅ `SELECT ?uri WHERE {dbr:Robert_Downey_Jr dbp:resides ?uri.}`

✅
```
SELECT ?uri WHERE {?uri dbp:studio ?x.
                   ?uri dbp:starring  dbr:Robert_Downey_Jr.}
```

❌ `SELECT ?uri WHERE {dbr:Iron_Man dbp:studio ?uri.}`

❌ `SELECT ?uri WHERE {?uri dbp:studio dbr:Warner_Bros.}`

SPARQL is difficult to represent

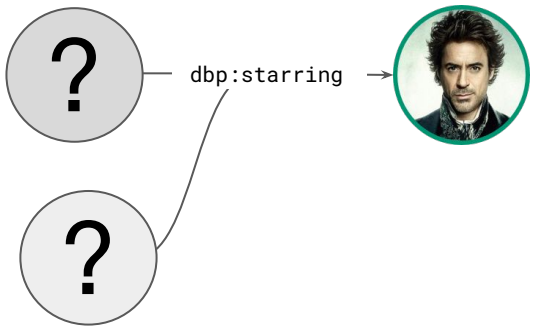
```
SELECT ?uri WHERE {
    ?uri dbp:starring dbr:Robert_Downey_Jr
}
```

SPARQL is difficult to represent

```
?uri dbp:starring dbr:Robert_Downey_Jr
```

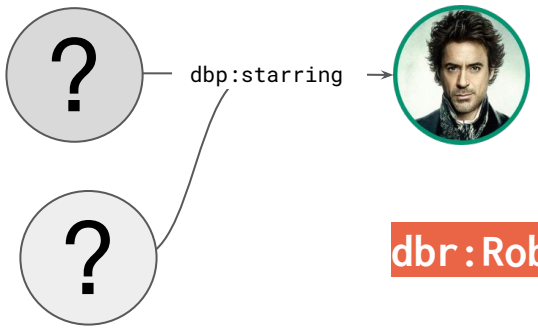
SPARQL is difficult to represent

```
?uri  dbp:starring  dbr:Robert_Downey_Jr
```



SPARQL is difficult to represent

```
?uri  dbp:starring  dbr:Robert_Downey_Jr
```



`dbr:Robert_Downey_Jr` - `dbp:starring`

SPARQL is difficult to represent

```
dbr:Robert_Downey_Jr - dbp:starring
dbr:Robert_Downey_Jr + dbp:spouse
dbr:Robert_Downey_Jr + dbp:spouse + dbo:birthplace
dbr:Robert_Downey_Jr - dbo:child - dbo:director
```

SPARQL is difficult to represent

```
- dbp:starring
+ dbp:spouse
+ dbp:spouse + dbo:birthplace
- dbo:child - dbo:director
```

Problems

- **Millions** of SPARQL

- SPARQL are difficult to **represent**.

Solutions

- Millions of SPARQL - Use **topic entity** and **limited** number of predicate
- SPARQL is difficult to represent - linearize it as **core chain**

Approach

Overview

Give me a list of everything where Robert Downey Jr Acted?

Overview

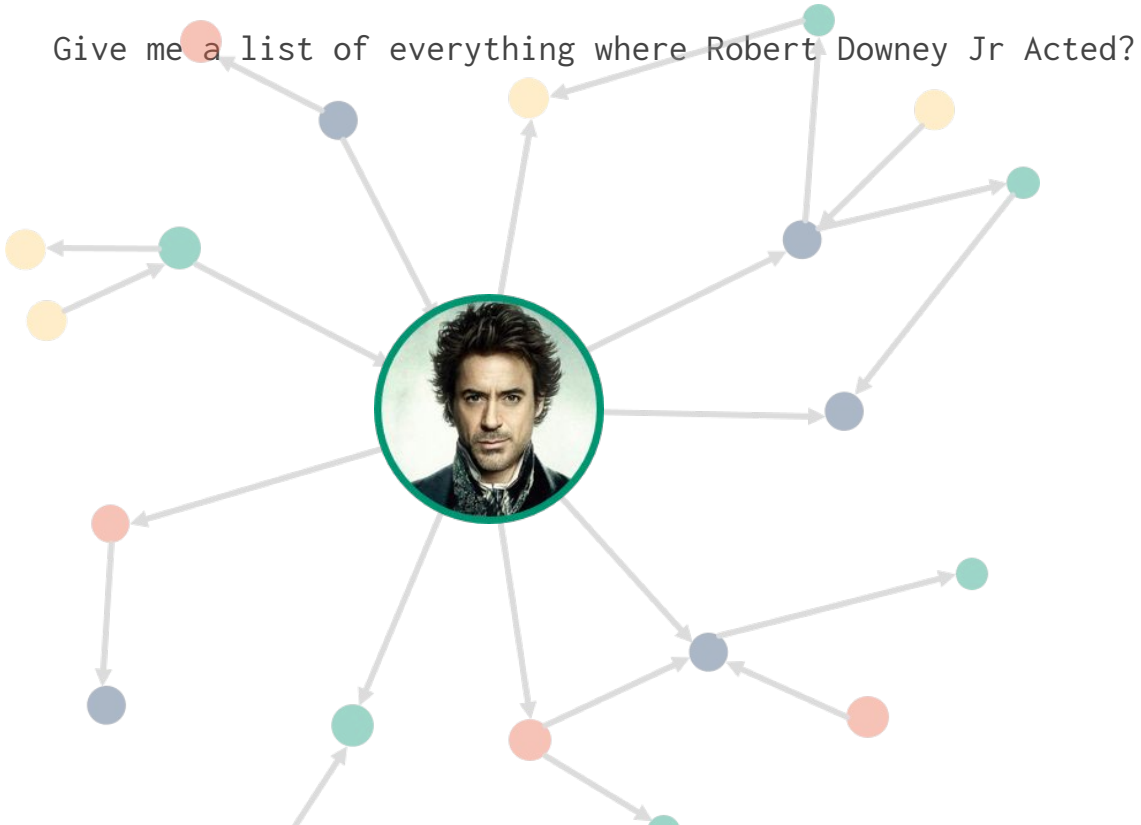
Given: Question, Topic Entity.

`Give me a list of everything where` **`Robert Downey Jr`** `Acted?`

Overview

Given: Question, Topic Entity.

Collect 2-hop subgraph around it.



Overview

Given: Question, Topic Entity.

Collect 2-hop subgraph around it.

Generate core-chain candidates

```
Give me a list of everything where Robert Downey Jr Acted?
```

```
+ dbp:birthplace
+ dbp:parent
+ dbp:spouse - dbp:foundedBy
- dbp:starring
- dbp:starring + dbp:director
...
```

Overview

Given: Question, Topic Entity.

Collect 2-hop subgraph around it.

Generate core-chain candidates

Rank Candidates based on similarity with questions

```
Give me a list of everything where Robert Downey Jr Acted?

0.10 + dbp:birthplace
0.23 + dbp:parent
0.04 + dbp:spouse - dbp:foundedBy
0.73 - dbp:starring
0.41 - dbp:starring + dbp:director
     ...
```

Ranking Framework

Encode core chain and question to a **vector space** such that the correct core chain and the question are **aligned** with one another.

Ranking Framework

Encode core chain and question to a **vector space** such that the correct core chain and the question are **aligned** with one another.

$$\text{ans} = \underset{c}{\text{argmax}}\Big(\text{compare}\big(\text{enc}_q(q), \text{enc}_c(c)\big)\Big)$$

Setup

References.

Images of **manhattan and marvel cinematic universe** have been taken from wikipedia
https://en.wikipedia.org/wiki/Manhattan#/media/File:New_York_City_location_Manhattan.svg
https://en.wikipedia.org/wiki/Marvel_Cinematic_Universe#/media/File:Marvel_Cinematic_Universe_logo.png

Robert Downey Jr. image from slide 40 ownwards has been taken from
https://commons.wikimedia.org/wiki/File:Robert_Downey,_Jr._2012.jpg

References (Icons in KG)

Sherlock holmes by Matthew Davis from the Noun Project

Empire State Building by Jake Dunham from the Noun Project

Golden gate bridge by icon 54 from the Noun Project

Statue of Liberty by Berkah Icon from the Noun Project

MARVEl is from flaticon

Sherlock Holmes is from FlatIcon

Iron Man by Tatyana Kyul from the Noun Project

Ferguson by priyanka from the Noun Project