

QAnswer

how it works?

ESWC 2020 - tutorial
Online
by

Dennis Diefenbach



03.06.2020

Towards a Question Answering System over the Semantic Web*

Converting natural language questions to queries

*Dennis Diefenbach, Andreas Both, Kamal Singh, Pierre Maret, *Towards a Question Answering system over the Semantic Web* Semantic Web Journal
(under review)



Main idea: We do not perform natural language processing (NLP) in the traditional sense, but we rely on the graph structure of the KB!

Running example: Give me actors born in Strasbourg.

Step 1: Expansion

- **actors** <http://dbpedia.org/ontology/starring> dbpedia actor 0 PR
- **actors** <http://dbpedia.org/property/actor> dbpedia actor 0 PR
- **actors** <http://dbpedia.org/ontology/Actor> dbpedia actors 0 CL
- **actors** <http://dbpedia.org/property/actors> dbpedia actors 0 PR
- ...
- **actors** [http://dbpedia.org/resource/Actor_\(2016_film\)](http://dbpedia.org/resource/Actor_(2016_film)) dbpedia actor 65 IN
- **actors** [http://dbpedia.org/resource/Actor_\(UML\)](http://dbpedia.org/resource/Actor_(UML)) dbpedia actor 21 IN
- **actors** [http://dbpedia.org/resource/Actor_\(album\)](http://dbpedia.org/resource/Actor_(album)) dbpedia actor 111 IN
- ...
- **born** <http://dbpedia.org/ontology/hometown> dbpedia born 0 PR
- **born** <http://dbpedia.org/property/birthPlace> dbpedia born 0 PR
-
- **born** http://dbpedia.org/resource/Lucien-Hubert_Borne dbpedia borne 62 IN
- **born** <http://dbpedia.org/resource/Milestone> dbpedia borne 69 IN
- **born** http://dbpedia.org/resource/Max_Born dbpedia born 252 IN
-
- **strasbourg** <http://dbpedia.org/resource/Category:Strasbourg> dbpedia strasbourg 33 CL
- **strasbourg** <http://dbpedia.org/resource/Strasbourg> dbpedia strasbourg 1480 IN

127 Possible Meanings

Step 2: Query Generation

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
```

- SELECT DISTINCT ?s0 where {
 VALUES ?s0 { dbr:Strasbourg }
} limit 1000
- SELECT DISTINCT ?s1 where {
 ?s1 dbo:birthPlace dbr:Strasbourg .
 ?s1 ?p1 dbo:Actor .
} limit 1000
- SELECT DISTINCT ?o2 where {
 dbr:Actor_(1993_film) dbo:starring ?o1 .
 ?o1 dbp:birthPlace ?o2 .
} limit 1000
- SELECT DISTINCT ?s1 where {
 ?s1 dbo:hometown dbr:Strasbourg .
} limit 1000

500 Possible Interpretations

Step 3: Query Ranking

PREFIX dbr: <<http://dbpedia.org/resource/>>

PREFIX dbo: <<http://dbpedia.org/ontology/>>

PREFIX dbp: <<http://dbpedia.org/property/>>

1. SELECT DISTINCT ?s1 where {
 ?s1 dbo:birthPlace dbr:Strasbourg .
 ?s1 ?p1 dbr:Actor .
} limit 1000
2. SELECT DISTINCT ?o2 where {
 dbr:Actor_(1993_film) dbo:starring ?o1 .
 ?o1 dbp:birthPlace ?o2 .
} limit 1000
3. SELECT DISTINCT ?s1 where {
 ?s1 dbo:hometown dbr:Strasbourg .
} limit 1000
4. SELECT DISTINCT ?s0 where {
 VALUES ?s0 { dbr:Strasbourg }
} limit 1000

Step 4: Answer Decision

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
SELECT DISTINCT ?s1 where {
    ?s1 dbo:birthPlace dbr:Strasbourg .
    ?s1 ?p1 dbr:Actor .
} limit 1000
```

Is this query matching the intended meaning of the user?

Some technical details

Step 1: Expansion

Example: Give me actors born in Strasbourg.

- Index all labels in the dataset
- Analyze every n-gram in the question and try to find a corresponding URI
- We rely on a Lucene index: fast and low memory footprint
- Stemming is important! (adapted to the language)

Step 2: Query Generation

From step 1 we have a list of resources r_1, \dots, r_n (e.g. Max_Born, Actor)

Idea:

- Traverses the RDF graph starting from the identified resources in a breadth-search manner
- Find the distance between the resources
- Use the distances to build triple patterns

Step 3: Query Ranking

It's a learning to rank problem

Idea:

- Construct some features: the number of words covered in the question, the edit distance to the label, relevance score
- Construct a training dataset
- Learn

We use linear models, Coordinate Ascent

Step 4: Answer Decision

It's a binary classification problem. Fire or don't fire.

Idea:

- Reuse the same features as in the previous step
- Construct a training dataset
- Learn

We use logistic regression to have a number between 0 and 1, like a confidence.

Advantages

R1 Multilinguality



QAnswer

bars in borgomasino (/qa/full?query=bars%20in%20borgomasino&tags=[&lang=en&kb=wikidata,openstreetmap&user=open,open])

What is the birth date of George Washington? (/qa/full?query=What%20is%20the%20birth%20date%20of%20George%20Washington?&tags=[&lang=en&kb=wikidata,openstreetmap&user=open,open])

Who wrote Dubliners? (/qa/full?query=Who%20wrote%20Dubliners?&tags=[&lang=en&kb=wikidata,openstreetmap&user=open,open])

R2 Scalability

Hardware

We can run it on this Laptop!
16 Gb of RAM, 2,8 GHz Intel Core i7, 500 Gb



QAnswer

Who is the director of Big Fish? (/qa/full?query=Who%20is%20the%20director%20of%20Big%20Fish?&tags=%E2%80%9D&lang=en&kb=wikidata,openstreetmap&user=open,open)

What is the surface of Liechtenstein? (/qa/full?query=What%20is%20the%20surface%20of%20Liechtenstein?&tags=%E2%80%9D&lang=en&kb=wikidata,openstreetmap&user=open,open)

Who is the king of Jordan? (/qa/full?query=Who%20is%20the%20king%20of%20Jordan?&tags=%E2%80%9D&lang=en&kb=wikidata,openstreetmap&user=open,open)

R3 Portability



QAnswer

Who is Bach? (/qa/full?query=Who%20is%20Bach?&tags=[]&lang=en&kb=wikidata,openstreetmap&user=open,open)

Who are the members of Green Day? (/qa/full?query=Who%20are%20the%20members%20of%20Green%20Day?&tags=[]&lang=en&kb=wikidata,openstreetmap&user=open,open)

Who is the king of Jordan? (/qa/full?query=Who%20is%20the%20king%20of%20Jordan?%20&tags=[]&lang=en&kb=wikidata,openstreetmap&user=open,open)

paintinas bv monet (/aa/full?auerv=paintinas%20bv%20monet&taas=



R4 Robustness



QAnswer

communes in the province of biella (/qa/full?query=communes%20in%20the%20province%20of%20biella&tags= []&lang=en&kb=wikidata,openstreetmap&user=open,open)

Who is the king of Jordan? (/qa/full?query=Who%20is%20the%20king%20of%20Jordan?&tags= []&lang=en&kb=wikidata,openstreetmap&user=open,open)

How many inhabitants has Southampton? (/qa/full?query=How%20many%20inhabitants%20has%20Southampton?&tags= []&lang=en&kb=wikidata,openstreetmap&user=open,open)

R5 Multiple Knowledge Bases

QAnswer

Who produced the Life is Beautiful? (/qa/full?query=Who%20produced%20the%20Life%20is%20Beautiful?&tags=%E2%80%9C&lang=en&kb=wikidata,openstreetmap&user=open,open)

How many inhabitants has Southampton? (/qa/full?query=How%20many%20inhabitants%20has%20Southampton?&tags=%E2%80%9C&lang=en&kb=wikidata,openstreetmap&user=open,open)

Who wrote Dubliners? (/qa/full?query=Who%20wrote%20Dubliners?&tags=%E2%80%9C&lang=en&kb=wikidata,openstreetmap&user=open,open)

Evaluation

QALD

QA system	Lang	Type	P	R	F	Time
QALD-3						
WDAqua-core1	en	full	0.58	0.46	0.51	1.08s
WDAqua-core1	en	key	0.65	0.44	0.52	0.81s
WDAqua-core1	de	full	0.88	0.29	0.44	0.22s
WDAqua-core1	de	key	0.92	0.29	0.44	0.18s
WDAqua-core1	fr	full	0.86	0.28	0.43	0.31s
WDAqua-core1	fr	key	0.92	0.29	0.44	0.20s
WDAqua-core1	it	full	0.89	0.29	0.44	0.20s
WDAqua-core1	it	key	0.92	0.29	0.44	0.16s
WDAqua-core1	es	full	0.90	0.29	0.44	0.18s
WDAqua-core1	es	key	0.91	0.29	0.44	0.17s
gAnswer [28]*	en	full	0.40	0.40	0.40	$\approx 1\text{ s}$
RTV [14]	en	full	0.32	0.34	0.33	-
Intui2 [11]	en	full	0.32	0.32	0.32	-
SINA [21]*	en	full	0.32	0.32	0.32	$\approx 10\text{-}20\text{s}$
DEANNA [25]*	en	full	0.21	0.21	0.21	$\approx 1\text{-}50\text{s}$
SWIP [19]	en	full	0.16	0.17	0.17	-
Zhu et al. [27]*	en	full	0.38	0.42	0.38	-

QA system	Lang	Type	P	R	F	Time
QALD-4						
Xser [24]	en	full	0.72	0.71	0.72	-
WDAqua-core1	en	key	0.76	0.40	0.52	0.32s
WDAqua-core1	en	full	0.56	0.30	0.39	0.46s
gAnswer [28]	en	full	0.37	0.37	0.37	0.973 s
CASIA [16]	en	full	0.32	0.40	0.36	-
WDAqua-core1	de	key	0.92	0.20	0.33	0.04s
WDAqua-core1	fr	key	0.92	0.20	0.33	0.06s
WDAqua-core1	it	key	0.92	0.20	0.33	0.04s
WDAqua-core1	es	key	0.92	0.20	0.33	0.05s
WDAqua-core1	de	full	0.90	0.20	0.32	0.06s
WDAqua-core1	it	full	0.92	0.20	0.32	0.16s
WDAqua-core1	es	full	0.90	0.20	0.32	0.06s
WDAqua-core1	fr	full	0.86	0.18	0.29	0.09s
Intui3 [12]	en	full	0.23	0.25	0.24	-
ISOFT [18]	en	full	0.21	0.26	0.23	-
Hakimov [15]*	en	full	0.52	0.13	0.21	-

QA system	Lang	Type	P	R	F	Time
QALD-5						
Xser [24]	en	full	0.74	0.72	0.73	-
WDAqua-core1	en	full	0.56	0.41	0.47	0.62s
WDAqua-core1	en	key	0.60	0.27	0.37	0.50s
AskNow[13]	en	full	0.32	0.34	0.33	-
QAnswer[20]	en	full	0.34	0.26	0.29	-
WDAqua-core1	de	full	0.92	0.16	0.28	0.20s
WDAqua-core1	de	key	0.90	0.16	0.28	0.19s
WDAqua-core1	fr	full	0.90	0.16	0.28	0.19s
WDAqua-core1	fr	key	0.90	0.16	0.28	0.18s
WDAqua-core1	it	full	0.88	0.18	0.30	0.20s
WDAqua-core1	it	key	0.90	0.16	0.28	0.18s
WDAqua-core1	es	full	0.88	0.14	0.25	0.20s
WDAqua-core1	es	key	0.90	0.14	0.25	0.20s
SemGraphQA[2]	en	full	0.19	0.20	0.20	-
YodaQA[1]	en	full	0.18	0.17	0.18	-
QuerioDali[17]	en	full		0.48	?	?

QA system	Lang	Type	P	R	F	Time
QALD-6						
UTQA [23]	en	full	0.82	0.69	0.75	-
UTQA [23]	es	full	0.76	0.62	0.68	-
UTQA [23]	fs	full	0.70	0.61	0.65	-
WDAqua-core1	en	full	0.62	0.40	0.49	0.93s
WDAqua-core1	en	key	0.52	0.32	0.40	0.68s
SemGraphQA [2]	en	full	0.70	0.25	0.37	-
WDAqua-core1	de	full	0.95	0.17	0.29	0.12s
WDAqua-core1	de	key	0.96	0.17	0.29	0.08s
WDAqua-core1	fr	full	0.91	0.16	0.27	0.37s
WDAqua-core1	fr	key	0.96	0.17	0.29	0.12s
WDAqua-core1	it	full	0.96	0.17	0.29	0.16s
WDAqua-core1	it	key	0.96	0.17	0.29	0.07s
WDAqua-core1	es	full	0.96	0.17	0.29	0.19s
WDAqua-core1	es	key	0.96	0.17	0.29	0.19s

QA System	Lang	Type	Total	P	R	F	Runtime	Ref
QALD-7 task 4, training dataset								
WDAqua-core1	en	full	100	0.37	0.39	0.37	1.68s	-
WDAqua-core1	en	key	100	0.35	0.38	0.35	0.80s	-
WDAqua-core1	es	key	100	0.31	0.32	0.31	0.45s	-
Sorokin et al. [39]	en	full	100	-	-	0.29	-	[39]
WDAqua-core1	de	key	100	0.27	0.28	0.27	1.13s	-
WDAqua-core1	fr	key	100	0.27	0.30	0.27	1.14s	-
WDAqua-core1	fr	full	100	0.27	0.31	0.27	1.05s	-
WDAqua-core1	es	full	100	0.24	0.26	0.24	0.65s	-
WDAqua-core1	de	full	100	0.18	0.20	0.18	0.82s	-
WDAqua-core1	it	full	100	0.19	0.20	0.18	1.00s	-
WDAqua-core1	it	key	100	0.17	0.18	0.16	0.44s	-

Simple Question

QA System	Lang	Type	Total	Accuracy	Runtime	Ref
Lukovnikov et al.	en	full	21687	0.712	-	[29]
Golub and He	en	full	21687	0.709	-	[21]
Yin et al.	en	full	21687	0.683	-	[48]
Bordes et al.	en	full	21687	0.627	-	[4]
Dai et al.*	en	full	21687	0.626	-	[7]
WDAqua-core1*	en	full	21687	0.571	2.1 s	-

LC-QuAD & WDAqua-core0- Questions

Benchmark	Lang	Type	Total	P	R	F	Runtime
LC-QuAD	en	full	5000	0.59	0.38	0.46	1.5 s
WDAquaCore0Questions	mixed	mixed	689	0.79	0.46	0.59	1.3 s

Thank you for you attention!