Question answering: Structured vs Unstructured

AI project - Academic year 2020/2021

https://github.com/rpo19/AIQuestionAnswering

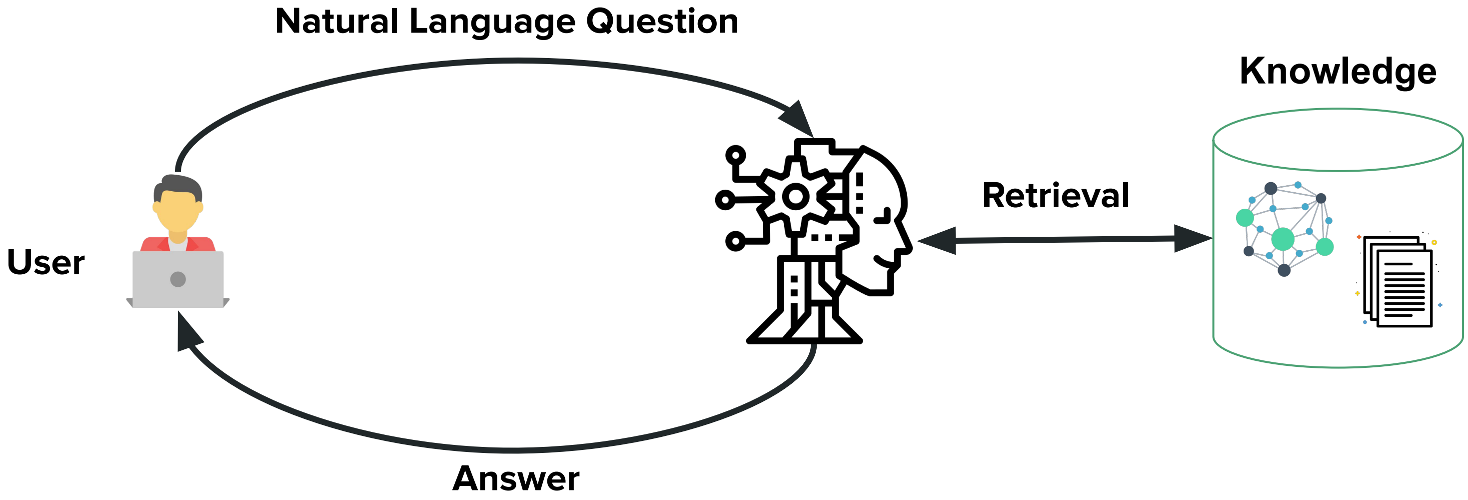
Christian Bernasconi 816423
Gabriele Ferrario 817518
Riccardo Pozzi 807857
Marco Ripamonti 806785

Objectives

- Explore Open Domain Question Answering approaches for both Knowledge Graph Question Answering (**KGQA**) and Free Text Question Answering (**FTQA**)
- Implementation of a KGQA approach (structured data)
- Implementation of a FTQA approach (unstructured data)
- Comparison between the two approaches

What is Question Answering?

Information Retrieval + Natural Language Processing



Domain, papers and references

Reference papers - KGQA

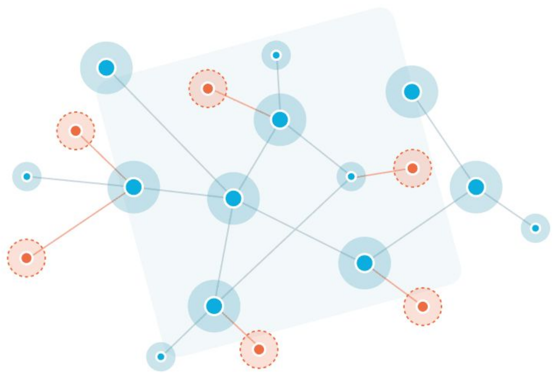
1. Bin Fu , Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, Jian Sun, "A Survey on Complex Question Answering over Knowledge Base: Recent Advances and Challenges" 2020 - https://arxiv.org/pdf/2007.13069.pdf
2. Nilesh Chakraborty, Denis Lukovnikov, Gaurav Maheshwari, Priyansh Trivedi, Jens Lehmann, Asja Fischer, "Introduction to Neural Network based Approaches for Question Answering over Knowledge Graphs" 2019 - https://arxiv.org/pdf/1907.09361.pdf
3. Weiguo Zheng, Mei Zhang, "Question Answering over Knowledge Graphs via Structural Query Patterns" 2019 - https://arxiv.org/pdf/1910.09760.pdf
4. Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, Tiejun Zhao "Constraint-Based Question Answering with Knowledge Graph" - https://www.aclweb.org/anthology/C16-1236.pdf
5. Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann, "LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs" - http://lc-quad.sda.tech/static/ISWC2017_paper_152.pdf

About Knowledge Graph QA

Information from a **Knowledge Graph**.

Three main categories:

- Traditional methods
- Information retrieval based
- Neural



KGQA approaches

- **Traditional methods**:
  - use of **rules** or **templates** (manually defined)
  - works well with **simple questions**
- **Information retrieval based**
  - NER + NEL ➔ subgraphs as candidate answers ➔ most relevant answer
  - **no manually defined templates + can't handles complex questions**
- **Neural**
  - **Classification based**: relation classification + entity prediction (NER + NEL)
    - works well with **simple questions**
  - **Ranking based**: most probable formal query wrt question, then ranked with a Neural Network.
    - can answer **complex queries + need to reduce possible queries' set cardinality**
  - **Translation based**: machine translation task (*e.g. seq2seq / Transformers*)
    - **simplified pipeline + need a large fully annotated dataset + not flexible to KB's changes**

Simple question:
Was Obama born in Hawaii?

[1] "A Survey on Complex Question Answering over Knowledge Base: Recent Advances and Challenges"

Benchmark datasets

- **WebQuestions**: questions fetched from the Google Suggest API. Only provides annotated answers without a formal query. Used as a benchmarking dataset for KGQA on **Freebase**

- **QALD-1to9**: from QALD-1 with easier questions to QALD-9 with more complex questions including comparative, superlative and inference constraints. Used as a benchmarking dataset for KGQA on **DBPedia**

- **LC-QuAD**: LargeScale Complex Question Answering Dataset is a benchmarking dataset for KGQA on **DBPedia**. It comprises 5000 pairs of question and its corresponding SPARQL query and 82% of its questions are complex questions.

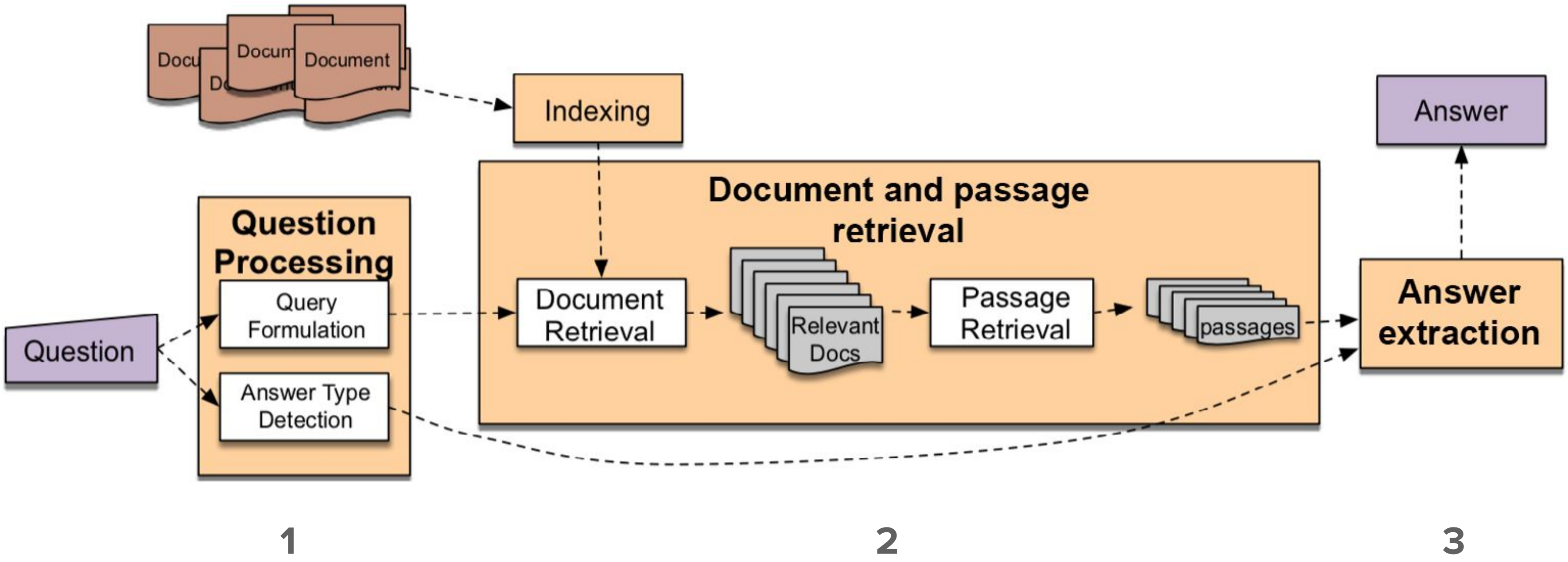

```
{
    "_id": "1501",
    "corrected_question": "How many movies did Stanley Kubrick direct?",
    "intermediary_question": "How many <movies> are there whose <director> is <Stanley Kubrick>?",
    "sparql_query": "SELECT DISTINCT COUNT(?uri) WHERE {?uri <http://dbpedia.org/ontology/director> <http://dbpedia.org/resource/Stanley_Kubrick>  . }",
    "sparql_template_id": 101
},
```


[1] “Introduction to Neural Network based Approaches for Question Answering over Knowledge Graphs”
[5] “LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs”

Reference papers - FTQA

6. Zahra Abbasiantaeb, Saeedeh Momtazi, “Text-based question answering from information retrieval and deep neural network perspectives: A survey” 2021 - https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1412
7. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding” 2018 - https://arxiv.org/abs/1810.04805
8. Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter” 2020 - https://arxiv.org/abs/1910.01108
9. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang “SQuAD: 100,000+ Questions for Machine Comprehension of Text” 2016 - https://arxiv.org/abs/1606.05250

About Free Text QA



[6] “Text-based question answering from information retrieval and deep neural network perspectives: A survey”

About Free Text QA

Answer extraction: compute the similarity of the question and the answer

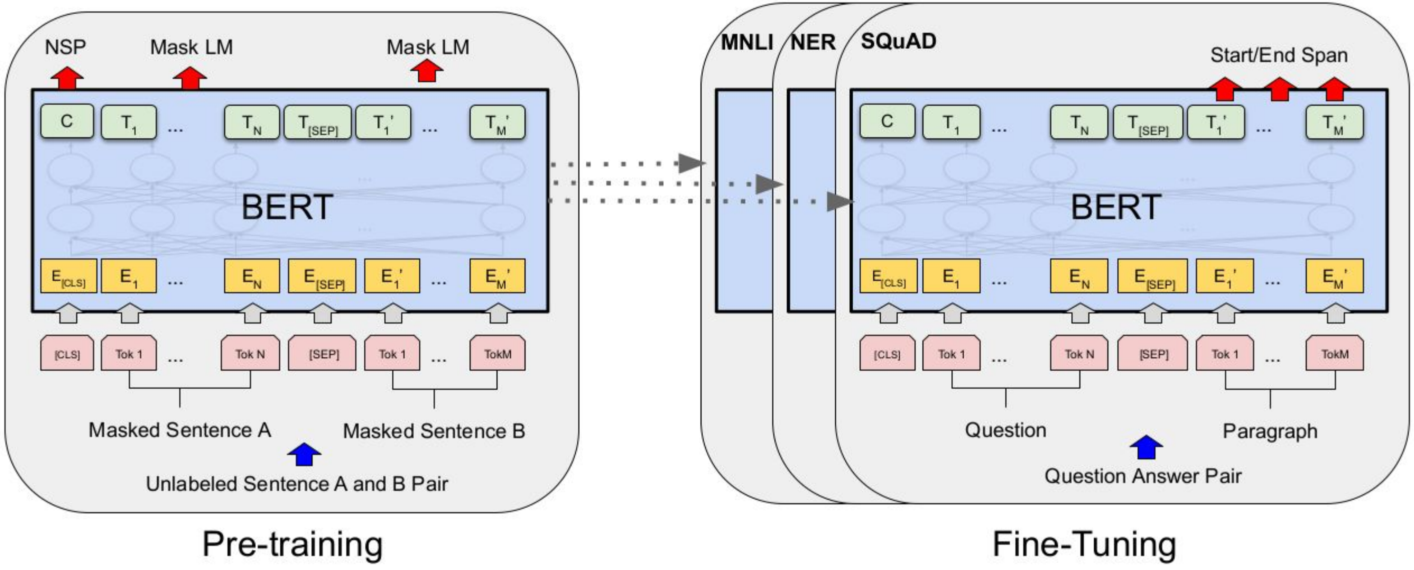
- **Information Retrieval:**
  - **Language Model** based: *P(Q|A)*

- **Deep Learning:**
  - **Representation** based: fixed-dimensional **vector representation** for both the question and the candidate answer sentences separately
  - **Interaction** based: compute the interaction between each term of the question and the candidate answer sentences
  - **Hybrid** (Representation + Interaction)

[6] "Text-based question answering from information retrieval and deep neural network perspectives: A survey"

About Free Text QA - BERT for Question Answering

"Bert Model with a span classification head on top" for Question Answering



[7] "Bert: Pre-training of deep bidirectional transformers for language understanding"

[10] https://huggingface.co/transformers/model_doc/bert.html?highlight=bertforquestionanswering#bertforquestionanswering

Benchmark datasets

- **WikiQA**: from **Bing** query logs
- **TREC-QA**: from the Text REtrieval Conference (TREC) 8–13 QA dataset
- **MovieQA**: from diverse data sources
- **InsuranceQA**: insurance domain (close domain)
- **Yahoo! Dataset:** collected from Yahoo! Answers QA system. Yahoo!
- **SQuAD**: questions posed by crowdworkers on a set of **Wikipedia** articles


```
{
    "title":"Super_Bowl_50",
    "context":"Super Bowl 50 was an American football game ...",
    "question":"Which NFL team represented the AFC at Super Bowl 50?",
    "answers":[
        "..."
    ]
}
```


[9] “SQuAD: 100,000+ Questions for Machine Comprehension of Text”

[10] “Text-based question answering from information retrieval and deep neural network perspectives: A survey”

Methodology - KGQA

Implementation of a KGQA approach

Question Answering over Knowledge Graphs via **Structural Query Patterns** is the implementation adopted for this project.

- Implementation was **feasible** with respect to other approaches (even though it was still hard to implement the whole pipeline)

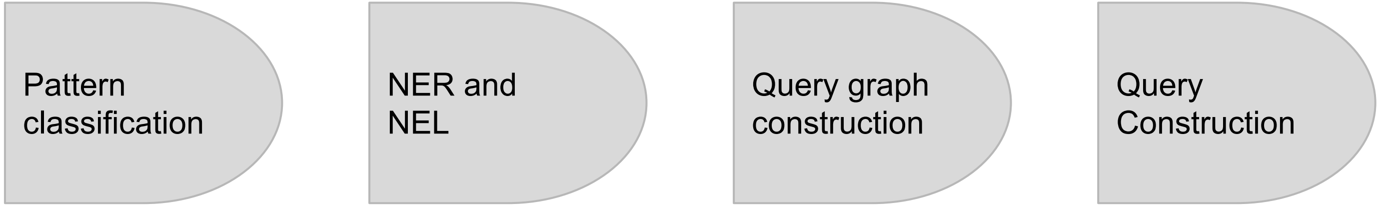
- **Performance** achieved by the approach are pretty good on **state of the art** benchmarking datasets

Method	LC-QuAD			QALD-8			QALD-9		
	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
Frankenstein	0.480	0.490	0.485	-	-	-	-	-	-
qaSearch	0.357	0.336	0.344	0.243	0.243	0.243	0.198	0.191	0.193
qaSQP	0.748	0.704	0.718	0.439	0.439	0.439	0.401	0.413	0.405
qaSQP-CE	0.835	0.813	0.827	0.558	0.663	0.620	0.522	0.625	0.568

[3] "Question Answering over Knowledge Graphs via Structural Query Patterns"

KGQA via Structural Query Patterns

The idea of the approach is to aid the construction of the query graph with the structural pattern of the given input question.



KGQA via Structural Query Patterns

Q: “Rashid Behbudov State Song Theatre and Baku Puppet Theatre can be found in which country?”

Pattern classification

NER and NEL

Query graph construction

Query Construction

[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA - Pattern classification

The first step of the process is the identification and **classification** of what is referenced as “**Structural Query Pattern**” of the question given as input.

Example:

Q: “What university campuses are situated in Indiana?”

SPARQL query:

```
SELECT DISTINCT ?uri WHERE {
   ?uri dbo:campus dbr:Indiana .
   ?uri rdf:type dbo:University .
}
```

[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA - Pattern classification

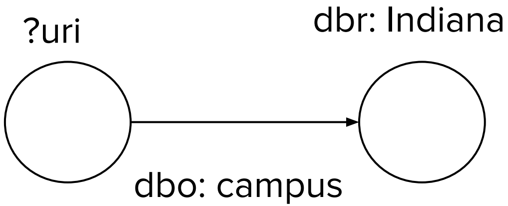
The first step of the process is the identification and **classification** of what is referenced as "**Structural Query Pattern**" of the question given as input.

Example:

Q: "What university campuses are situated in Indiana?"

SPARQL query:

```
SELECT DISTINCT ?uri WHERE {
    ?uri dbo:campus dbr:Indiana .
    ?uri rdf:type dbo:University .
}
```



[3] "Question Answering over Knowledge Graphs via Structural Query Patterns"

KGQA - Pattern classification

The first step of the process is the identification and **classification** of what is referenced as "**Structural Query Pattern**" of the question given as input.

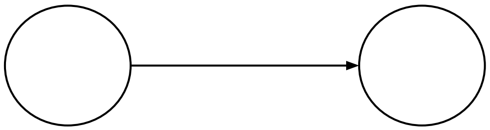
Example:

Q: "What university campuses are situated in Indiana?"

SPARQL query:

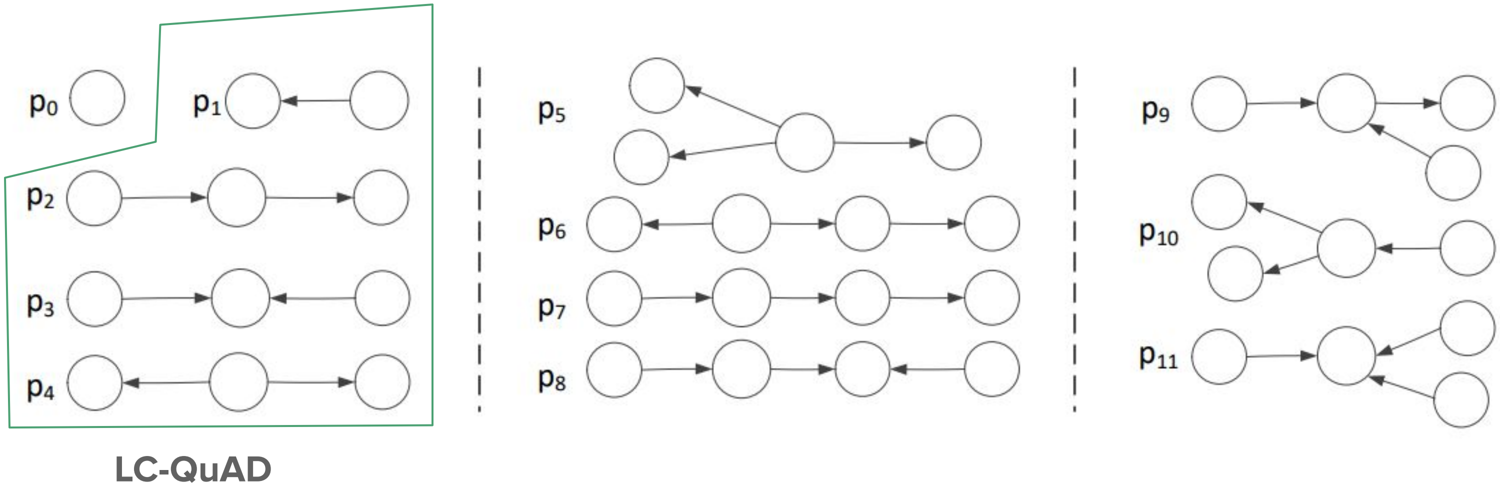
```
SELECT DISTINCT ?uri WHERE {
    ?uri dbo:campus dbr:Indiana .
    ?uri rdf:type dbo:University .
}
```

SQP: p1



[3] "Question Answering over Knowledge Graphs via Structural Query Patterns"

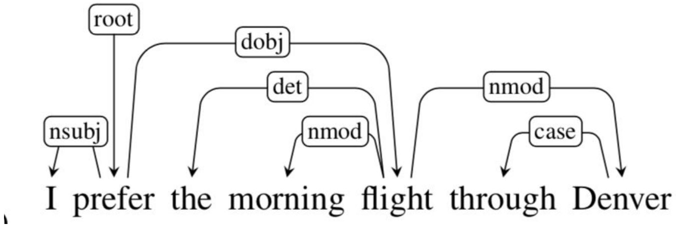
KGQA - Pattern classification



[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA - Pattern classification

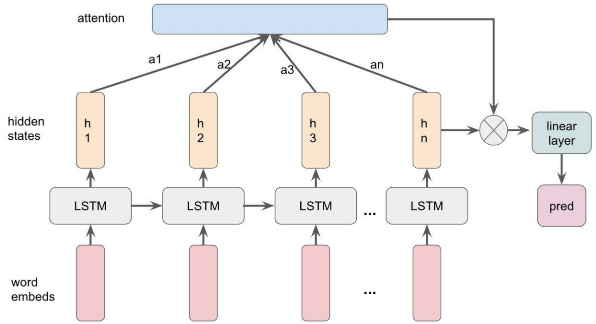
[3] Weiguo Zheng , Mei Zhang



Ensamble: LSTM with attention + CNN on text

Pattern

DistilBERT as a feature extractor



LSTM with attention

Pattern

[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”
[7] “Bert: Pre-training of deep bidirectional transformers for language understanding”

KGQA via Structural Query Patterns

Q: “Rashid Behbudov State Song Theatre and Baku Puppet Theatre can be found in which country?”

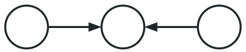
Pattern classification

NER and NEL

Query graph construction

Query Construction

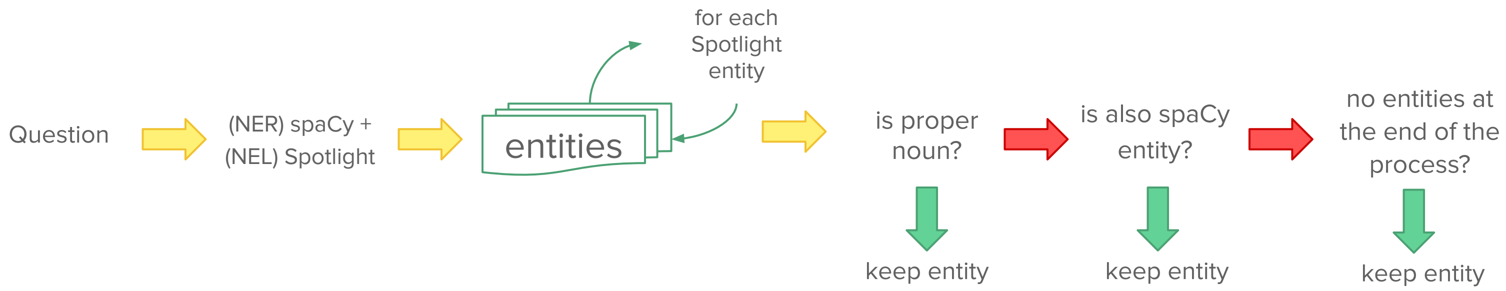
Classified Pattern: p3



[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA - NER and NEL

How to keep only useful entities for the query graph construction?



Example:

Q: “What is the lake of the city of Lecco?”

NER: Lecco

NEL: <http://dbpedia.org/resource/Lake>, <http://dbpedia.org/resource/City>, <http://dbpedia.org/resource/Lecco>

RESULT: <http://dbpedia.org/resource/Lecco>

[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA via Structural Query Patterns

Q: “Rashid Behbudov State Song Theatre and Baku Puppet Theatre can be found in which country?”

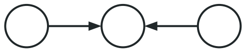
Pattern classification

NER and NEL

Query graph construction

Query Construction

Classified Pattern: p3

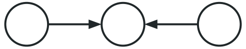


<http://dbpedia.org/resource/Rashid_Behbudov_State_Song_Theatre>
<http://dbpedia.org/resource/Baku_Puppet_Theatre>

[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

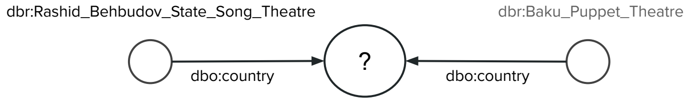
KGQA - Query Graph Construction

Input:

- empty query graph: 
- question: “Rashid Behbudov State Song Theatre and Baku Puppet Theatre can be found in which country?”
- entities:
  - dbr:Rashid_Behbudov_State_Song_Theatre
  - dbr:Baku_Puppet_Theatre

Output:

- labeled query graph:

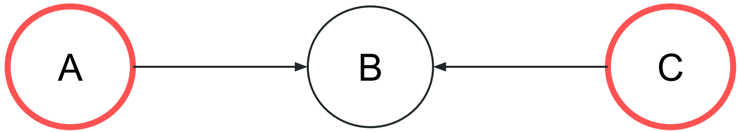


[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA - Query Graph Construction

Non-redundancy assumption: The entity e ∈ G identified for the question q is not an intermediate node in the structural query pattern p

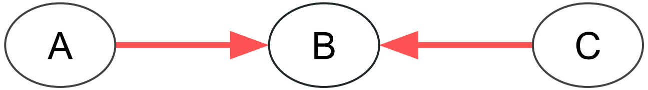
→ Select non-intermediate nodes *NS*



[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA - Query Graph Construction

- Identify type of relation of *NS*: outgoing



- Extract all possible relations according to a candidate entity ce and the relations direction. The query contains the entire pattern *p*.

```
SELECT DISTINCT ?pred WHERE {
    dbr:Baku_Puppet_Theatre ?pred ?obj.
    ?1 ?2 ?obj.
    FILTER( … )
}
```

http://dbpedia.org/ontology/abstract
http://dbpedia.org/ontology/address
http://dbpedia.org/ontology/alternativeName
http://dbpedia.org/ontology/seatingCapacity
http://dbpedia.org/ontology/architect
http://dbpedia.org/ontology/city
http://dbpedia.org/ontology/country
http://dbpedia.org/ontology/type
http://www.w3.org/ns/prov#wasDerivedFrom

[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA - Query Graph Construction

➜ Find the most similar relation *r* with respect to the question

Glove embeddings vectors

$$relevance(q, r) = \frac{\sum_{i=1}^{|q|} \sum_{j=1}^{|r|} \lambda \cdot cos(q_i, r_j) + (1 - \lambda) \cdot \frac{1}{lev(q_i, r_j)+1}}{len(relation\_tokens)}$$

birthDate ➜ [birth, date]
where ➜ place
when ➜ date

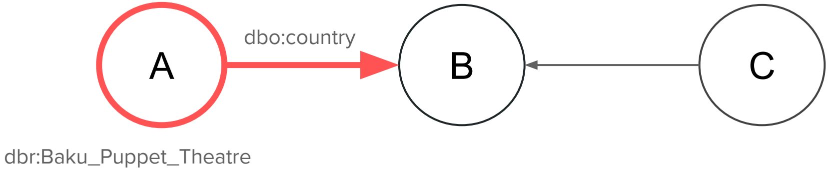
Entities are removed from text.

http://dbpedia.org/ontology/abstract
http://dbpedia.org/ontology/address
http://dbpedia.org/ontology/alternativeName
http://dbpedia.org/ontology/seatingCapacity
http://dbpedia.org/ontology/architect
http://dbpedia.org/ontology/city
http://dbpedia.org/ontology/country
http://dbpedia.org/ontology/type
http://www.w3.org/ns/prov#wasDerivedFrom

[3] "Question Answering over Knowledge Graphs via Structural Query Patterns"

KGQA - Query Graph Construction

- Assemble *ce* and *r* into the graph

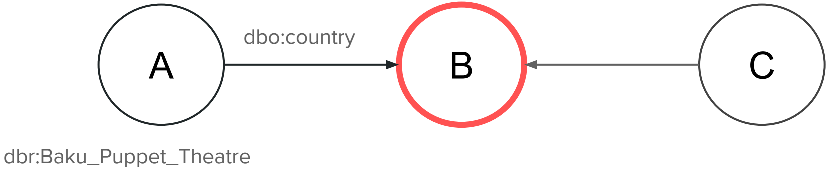


If an entity is present in the question it will be used to label the node. Otherwise it will be considered as a query variable.

[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

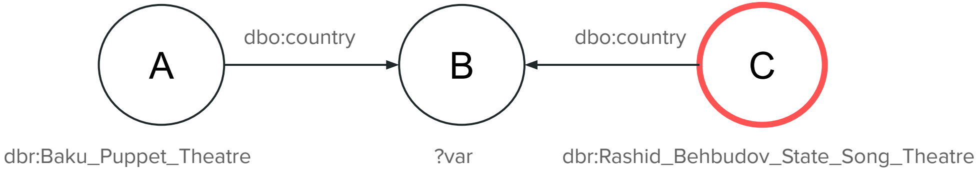
KGQA - Query Graph Construction

➜ If *p* has unlabeled edges start a **new iteration** considering *NS* as the adjacent node to the explored graph *B* and all entities found as object for the first relation



➜ *… repeat procedure …*

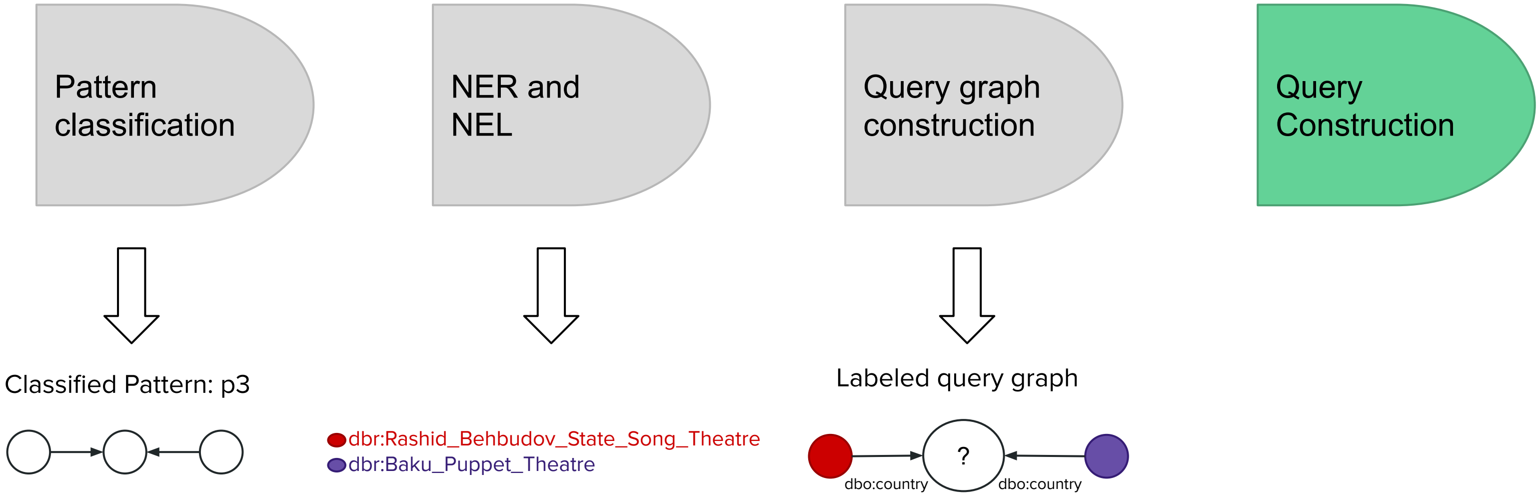
➜ Finally, when all edges are labeled, assemble last entities in the remaining node



[3] "Question Answering over Knowledge Graphs via Structural Query Patterns"

KGQA via Structural Query Patterns

Q: “Rashid Behbudov State Song Theatre and Baku Puppet Theatre can be found in which country?”

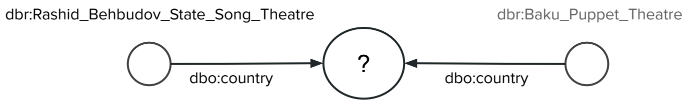


[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

KGQA - Query Construction

Input:

- labeled query graph:



- question: “Rashid Behbudov State Song Theatre and Baku Puppet Theatre can be found in which country?”

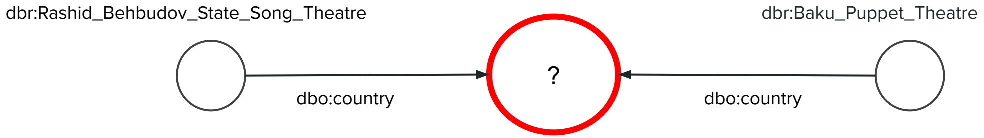
Output:

- Query:

```
SELECT DISTINCT ?target WHERE {
   dbr:Rashid_Behbudov_State_Song_Theatre  dbo:country ?target.
   dbr:Baku_Puppet_Theatre dbo:country ?target.
}
```

KGQA - Query Construction

- Target variable identification



- Body construction via generation of triple for each edge

```
{
    dbr:Rashid_Behbudov_State_Song_Theatre  dbo:country ?target.
    dbr:Baku_Puppet_Theatre dbo:country ?target.
}
```

KGQA - Query Construction

➔ Head construction

“Rashid Behbudov State Song Theatre and Baku Puppet Theatre can be found in which country?”

```
SELECT DISTINCT ?target WHERE
```

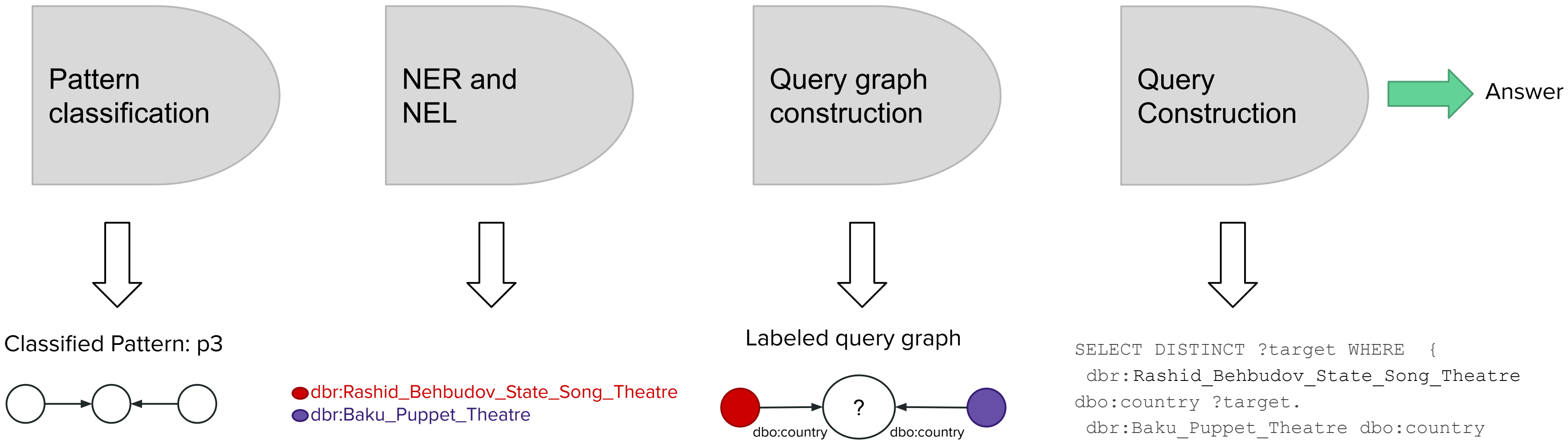
Other cases:

- “How many … ?” / “What is the number of … ?” / “Give me a count of … ?”
  ➔ `SELECT COUNT(DISTINCT ?target)`
- “Is Barack Obama … ?” ➔ [‘VBZ’, ‘NNP’, ‘NNP’, …] ➔ `ASK`

[4] “Constraint-Based Question Answering with Knowledge Graph”

KGQA via Structural Query Patterns

Q: “Rashid Behbudov State Song Theatre and Baku Puppet Theatre can be found in which country?”



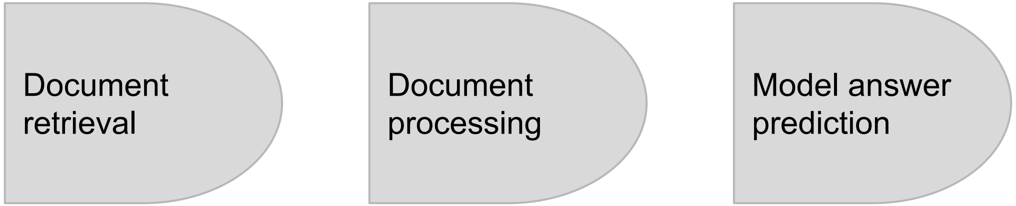

```
SELECT DISTINCT ?target WHERE  {
 dbr:Rashid_Behbudov_State_Song_Theatre
dbo:country ?target.
 dbr:Baku_Puppet_Theatre dbo:country
?target.
 }
```

[1] “Question Answering over Knowledge Graphs via Structural Query Patterns”

Methodology - FTQA

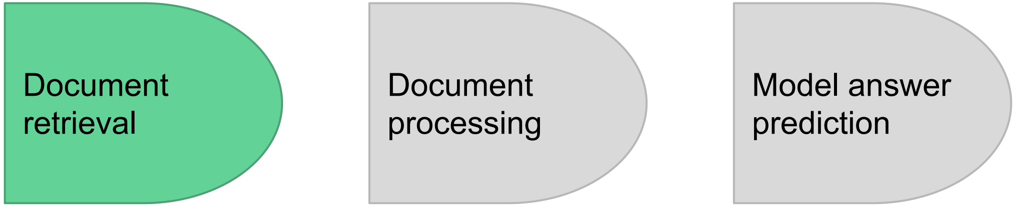
Implementation of a FTQA approach

The idea of the approach tries to answer questions using **Wikipedia** as retrieval base and an **extractive QA model**.



Implementation of a FTQA approach

Q: “How did Gandhi die?”



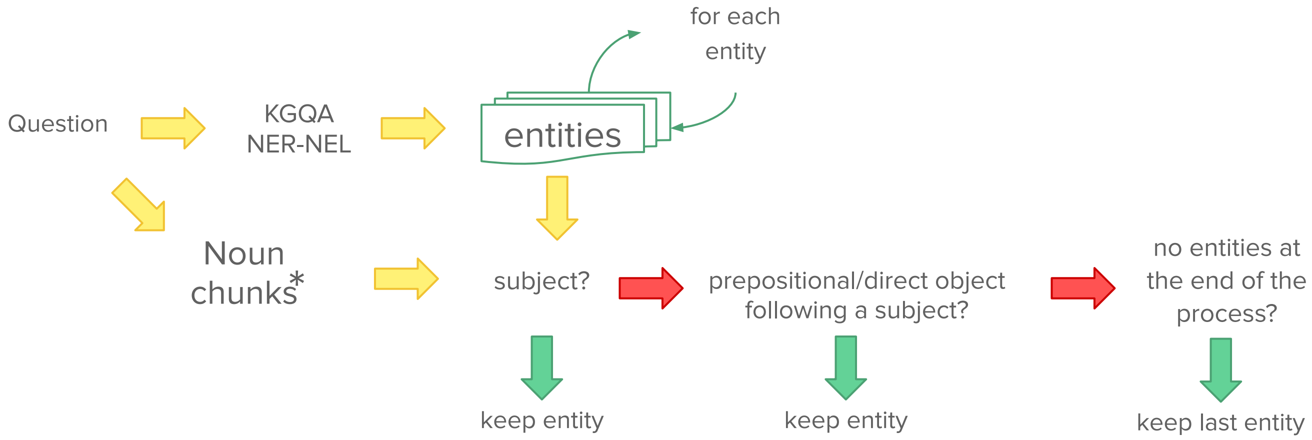
FTQA - Document retrieval

Modality 1: use Wikipedia search API

* https://spacy.io/usage/linguistic-features#noun-chunks

FTQA - Document retrieval

Modality 2: use NER and NEL with DBPedia Spotlight to locate the Wikipedia resource



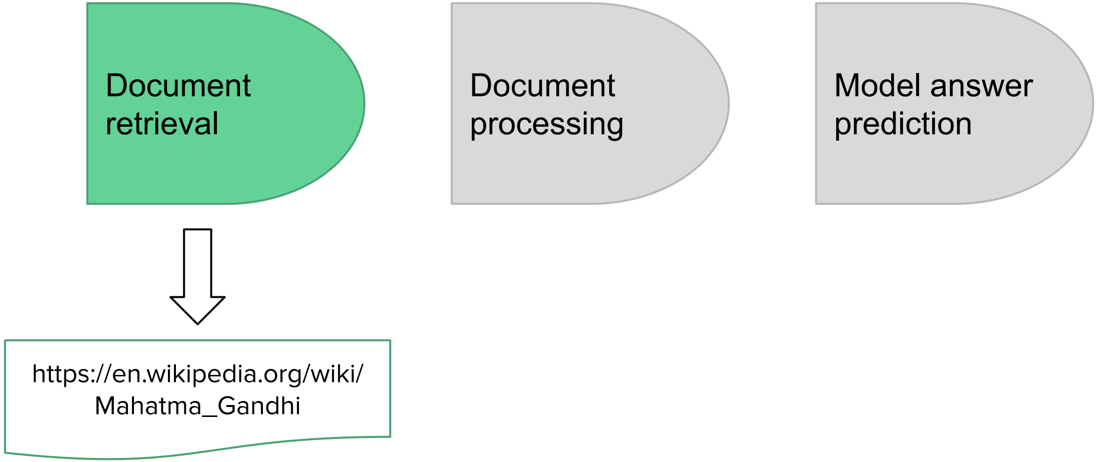
Example:

Q: "In which city of the Hawaii was Barack Obama born?"
Noun chunks: prepositional complement - prepositional object - subject
Main entity: http://dbpedia.org/resource/Barack_Obama ➔ https://en.wikipedia.org/wiki/Barack_Obama

* https://spacy.io/usage/linguistic-features#noun-chunks

Implementation of a FTQA approach

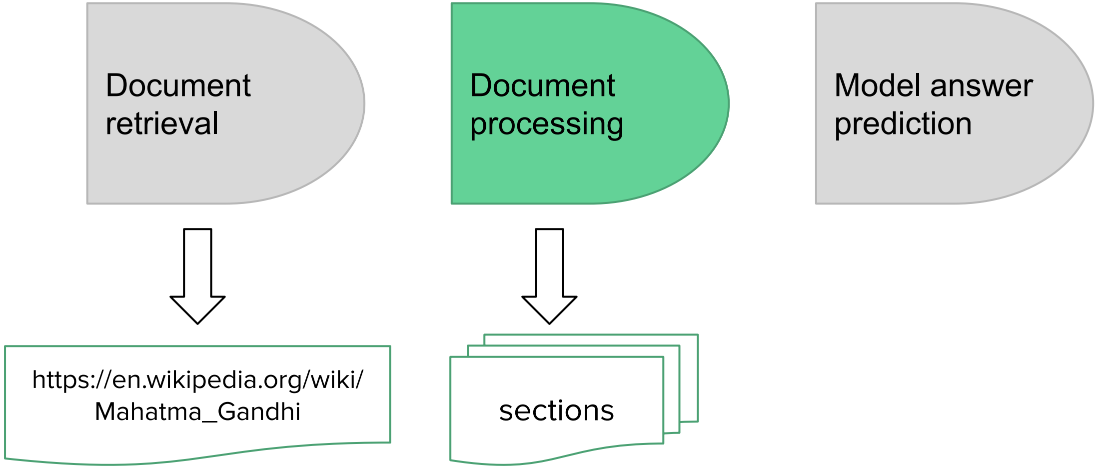
Q: “How did Gandhi die?”



[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

Implementation of a FTQA approach

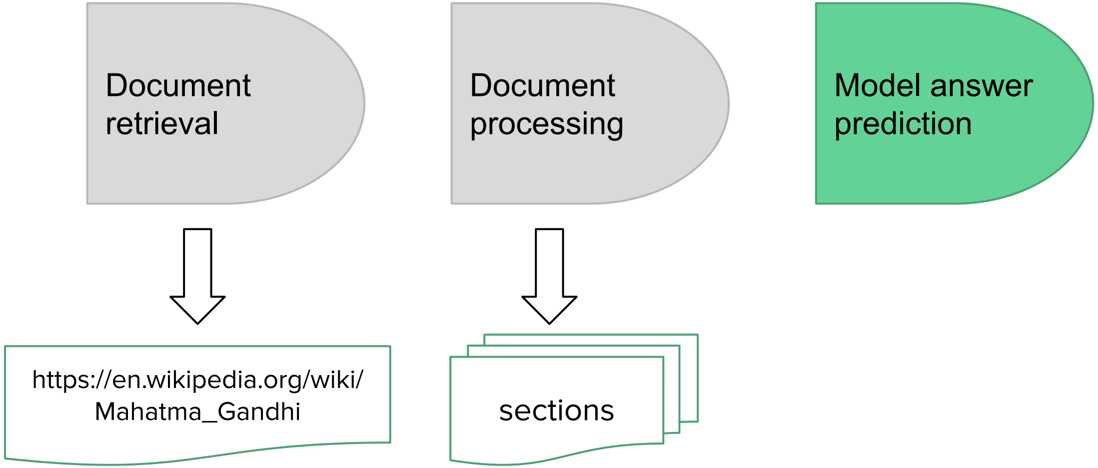
Q: “How did Gandhi die?”



[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

Implementation of a FTQA approach

Q: “How did Gandhi die?”



[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

FTQA - Model

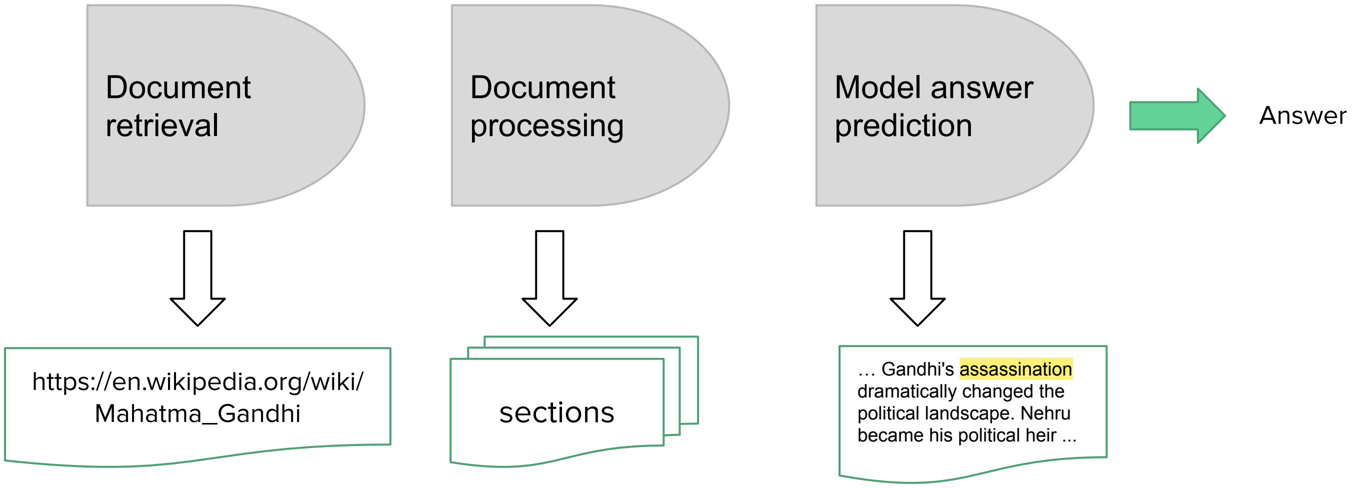
As a core component of the approach **DistilBert** has been chosen:

- a **distilled** version of **BERT** for faster inference and smaller model size
- with a **span classification head** for extractive question-answering
- fine tuned on **SQuAD**

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Feb 21, 2021	FPNet (ensemble) *Ant Service Intelligence Team*	**90.871**	**93.183**
2 May 16, 2021	IE-NetV2 (ensemble) *RICOH_SRCB_DML*	90.860	93.100
X	BERT		88.7
Y	Distilbert		87.1

Implementation of a FTQA approach

Q: “How did Gandhi die?”



[3] “Question Answering over Knowledge Graphs via Structural Query Patterns”

Evaluation and comparison

KGQA - Evaluation on LC-QuAD

LC-QuAD test	Precision	Recall	F1
Pattern classification	0.815	0.786	0.800
Question type classification	0.949	0.952	0.950
NEL (avg ± stddev) *	0.604 ± 0.461	0.582 ± 0.453	0.585 ± 0.448
Predicates choice (avg ± stddev) *	0.204 ± 0.376	0.185 ± 0.348	0.189 ± 0.348

Problem: *

dataset refers to *DBpedia 04-2016* ➡ differences in the KBs ➡ differences in entities/relations

KGQA - Evaluation on LC-QuAD

Problem: dataset refers to *DBpedia 04-2016* ➔ differences in the KB ➔ differences in answers

Proposed approach: try to partially avoid the problem by comparing labeled graphs

LC-QuAD test: predicted graph vs expected	**Accuracy**
Isomorphic graph	0.693
Permissive strict (different variable names accepted)	0.068
Permissive (variable instead of entity accepted)	0.092
Permissive (variable always accepted)	0.095
Permissive (variable always equal to variable)	0.068

FTQA - Evaluation on SQuAD

Problems:

- dataset refers to *Wikipedia 2016* ➜ differences in Wikipedia ➜ differences in answers

- the actual implementation requires too much time for a test on an entire dataset

Comparison - FTQA vs KGQA

To make a **fair comparison** an ad-hoc test set has been constructed including **44 questions** that both the approaches have the **potential** to answer correctly.

	KG top1	**FT span top1**	**FT wiki top1**	**FT wiki top3**	**FT nernel top1**	**FT nernel top3**
avg *	1.205	1.300	0.659	1.0	0.5	0.932
percentages *	36.4 6.8 56.8	34.1 2.3 63.6	63.6 6.8 29.5	40.9 18.2 40.9	75.0 0 25.0	43.2 20.5 36.4
avg time	1.382	0.572	15.453		24.358	

*
0 → **wrong**
1 → **acceptable**
2 → **correct**

KGQA - Pros & Cons

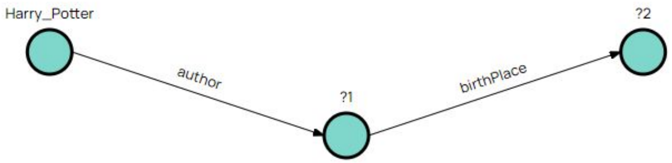
Pros:

- can answer to more complex questions

P2

Q:“What is the birthplace of the author of Harry Potter?”

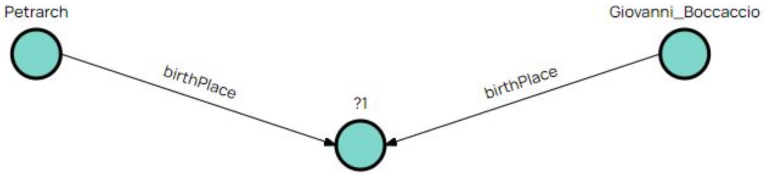
A: http://dbpedia.org/resource/Yate



P3

Q:“What is the birthplace of both Petrarch and Giovanni Boccaccio?”

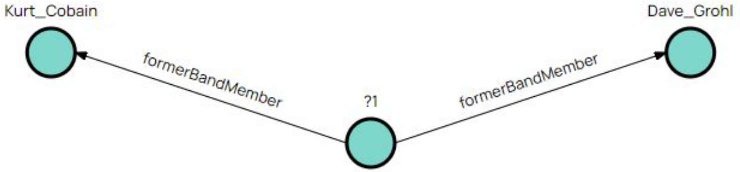
A: http://dbpedia.org/resource/Republic_of_Florence



P4

Q:“What is the band which has Dave Grohl and Kurt Cobain as former members?”

A: http://dbpedia.org/resource/Nirvana_(band)



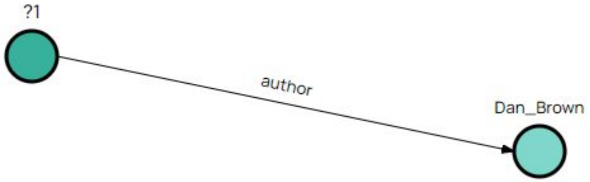
KGQA - Pros & Cons

Pros:

- can answer to more complex questions
- can answer to count-based and binary questions

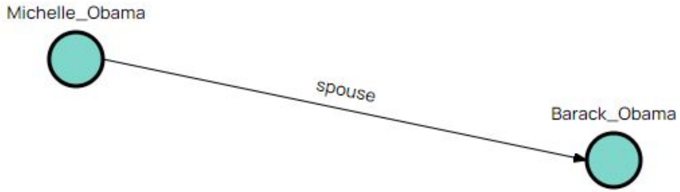
Q:“How many books has author Dan Brown written?”
A: 8

COUNT



Q:“Is Michelle Obama the spouse of Barack Obama?”
A: yes

ASK



KGQA - Pros & Cons

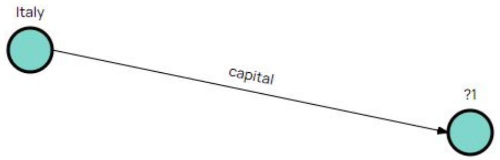
Pros:

- can answer to more complex questions
- can answer to count-based and binary questions

Cons:

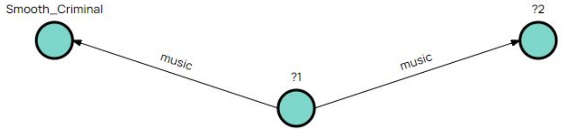
- patterns could be misclassified

Q:“What is the postal code of the capital of Italy?”



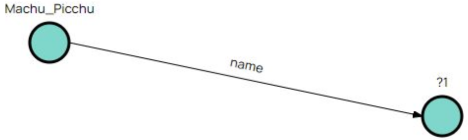
Correct pattern: P2
Predicted pattern: P1

Q:“Who is the producer of album which has the song Smooth Criminal?”



Correct pattern: P2
Predicted pattern: P4

Q:“What is the language spoken in the country of Machu Picchu?”



Correct pattern: P2
Predicted pattern: P1

KGQA - Pros & Cons

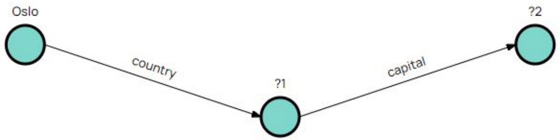
Pros:

- can answer to more complex questions
- can answer to count-based and binary questions

Cons:

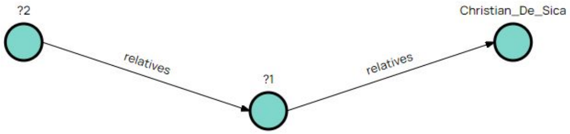
- patterns could be misclassified
- correct relations could be missed

Q:“How many universities are there whose country's capital is Oslo?”



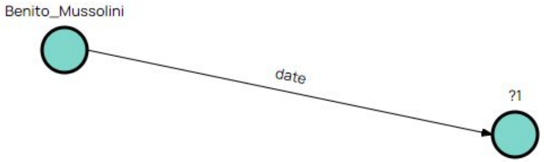
Problem: *Oslo*’s relation should be *capital*

Q:“List the places where the relatives of Christian De Sica died ?”



Problem: the second relation should be *deathPlace*

Q:“When did Mussolini die?”



Problem: *date* has a higher score than *deathDate*

KGQA - Pros & Cons

Pros:

- can answer to more complex questions
- can answer to count-based and binary questions

Cons:

- patterns could be misclassified
- correct relations could be missed
- multiple entities identified instead of one

Q:“Who is the producer of Sweeney Todd: The Demon Barber Of Fleet Street?”

Correct entity:
dbr:Sweeney_Todd:_The_Demon_Barber_of_Fleet_Street_(2007_film)

Linked entities:

- dbr:Sweeney_Todd
- dbr:Etrigan_the_Demon
- dbr:Fleet_Street

FTQA - Pros & Cons

Pros:

- can understand well the type of question

Q:“Who is Cleopatra's father?”
A: Ptolemy XII Auletes, Juba II, Caesarion, Ptolemy XII...

Q:“Where was Mussolini born?”
A: Italy, Milan, Czechoslovakia, Israel, Dovia di Predappio...

Q:“When was Bill Gates born?”
A: 1974, 1997, 1975, October 28 - 1955...

FTQA - Pros & Cons

Pros:

- can understand well the type of question
- text has more info for answering to more specific questions

Q:“What did gandhi study?”
A: law and jurisprudence

“At UCL, he studied law and jurisprudence and was invited to enroll at Inner Temple with the intention of becoming a barrister. His childhood shyness and self-withdrawal had continued through his teens. He retained these traits when he arrived in London, but joined a public speaking practice group and overcame his shyness sufficiently to practise law.”

Observation: the more specific relation about his studies which can be found in DBPedia is *almaMater*:

dbo:almaMater	▪ dbr:University_College_London ▪ dbr:Inner_Temple

FTQA - Pros & Cons

Pros:

- can understand well the type of question
- text has more info for answering to more specific questions

Cons:

- difficulties in answering questions which require lists

Q:“What companies did Elon Musk found?”
A: Neuralink and OpenAI

“Elon Reeve Musk FRS (/ˈiːlɒn/ EE-lon; born June 28, 1971) is an entrepreneur and business magnate. He is the founder, CEO, and chief engineer at SpaceX; early stage investor,[note 1] CEO, and product architect of Tesla, Inc.; founder of The Boring Company; and co-founder of Neuralink and OpenAI. A centibillionaire, Musk is one of the richest people in the world.”

Observation: answers should be contiguous in the text, but it isn’t a sufficient condition

FTQA - Pros & Cons

Pros:

- can understand well the type of question
- text has more info for answering to more specific questions

Cons:

- difficulties in answering questions which require lists
- can't answer to complex questions
- can't answer to binary questions
- can't answer to count-based questions

Conclusions & Future improvements

Conclusions

- implementing an open domain question answering system involves **many tasks and steps** (*i.e. NER/NEL, predictive models, SPARQL, …*)
- all the steps should be very **solid**
- relation **linking** is a crucial step of KGQA
- FTQA is less suitable to **complex questions** (*i.e. involving multiple entities)*
- KGQA has fastest **response time** than FTQA

Future improvements

FTQA:

- improve relevant documents retrieval
- filter the most relevant sections

KGQA:

- NER and NEL improvements
- improve choice of most relevant relation
- handle more Structured Query Patterns
- handle more question types (*e.g. “What is the highest mountain in Asia?”*)

Demo application

Thanks!