Improved Simple Question Answering over Wikidata

Bachelor's thesis presentation

David Otte

University of Freiburg

June 28, 2023

Introduction to Wikidata

- Simplified subset of Wikidata in RDF format:

Subject	Predicate	Object
"Eiffel Tower"	"named after"	"Gustave Eiffel"
"Eiffel Tower"	"visitors per year"	6,207,303
"Gustave Eiffel"	"place of birth"	"Dijon"

Introduction to Wikidata

- Simplified subset of Wikidata in RDF format:

Subject	Predicate	Object
"Eiffel Tower"	"named after"	"Gustave Eiffel"
"Eiffel Tower"	"visitors per year"	6,207,303
"Gustave Eiffel"	"place of birth"	"Dijon"

- Simplified example query:

```
SELECT ?o WHERE {
  "Eiffel Tower" "named after" ?o .
}
```

Introduction to Wikidata

- Simplified subset of Wikidata in RDF format:

Subject	Predicate	Object
"Eiffel Tower"	"named after"	"Gustave Eiffel"
"Eiffel Tower"	"visitors per year"	6,207,303
"Gustave Eiffel"	"place of birth"	"Dijon"

- Simplified example query:

```
SELECT ?o WHERE {
  "Eiffel Tower" "named after" ?o .
}
```

- Results:

?o
"Gustave Eiffel"

Introduction to Wikidata

- Subset of Wikidata in RDF format (Prefixes omitted):

Subject	Predicate	Object
Q243	P138	Q20882
Q243	P1174	6,207,303
Q20882	P19	Q7003

- Example query:

```
SELECT ?o WHERE {
  wd:Q243 wdt:P138 ?o .
}
```

- Results:

?o
Q20882

Problem: Motivation

- Question: What is the height of Mount Everest?

Problem: Motivation

- Question: What is the height of Mount Everest?
- *What are the required Wikidata IDs?*
  How is the required data organized in Wikidata?
  How to fomulate the correct query?

Problem: Motivation

- Question: What is the height of Mount Everest?
- *What are the required Wikidata IDs?*
  How is the required data organized in Wikidata?
  How to fomulate the correct query?
- Query that answers question:

```
SELECT ?o WHERE {
  wd:Q513 wdt:P2044 ?o .
}
```

Problem: Definition

- Focus on Simple Questions

Problem: Definition

- Focus on Simple Questions
- Given: Natural language question q
- Goal: Find query that answers q using one of the following two patterns:

Target: Object

```
SELECT ?o WHERE {
  <entity> <relation> ?o .
}
```

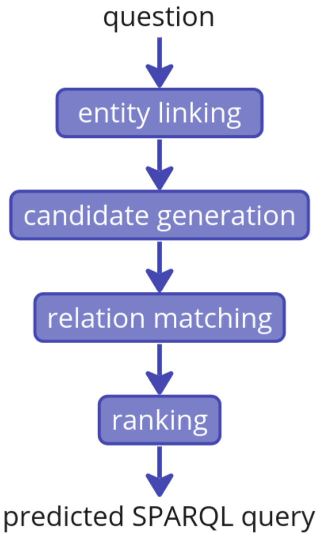
Target: Subject

```
SELECT ?s WHERE {
  ?s <relation> <entity> .
}
```

Problem

Questions?

Approach: Pipeline



Approach: Entity linking

- Question: In which city was Leonhard Euler born?
- Identified entities for each subsequence:

s	E_s
"Leonhard Euler"	$\{\underline{\texttt{Q7604}}, \texttt{Q58118685}, ...\}$
"city"	$\{\texttt{Q515}, ...\}$
...	...

Approach: Entity linking

- Question: In which city was Leonhard Euler born?
- Identified entities for each subsequence:

s	E_s
"Leonhard Euler"	$\{\underline{\texttt{Q7604}}, \texttt{Q58118685}, \ldots\}$
"city"	$\{\texttt{Q515}, \ldots\}$
...	...

- Get final set E' by combining all E_s and by dropping less promising entities
- $E' = \{\underline{\texttt{Q7604}}, \texttt{Q58118685}, \texttt{Q515}, \ldots\}$

Approach: Candidate generation

- For each entity in E', we generate all possible query candidates:

Entity	Relations Target: Object	Relations Target: Subject
Q7604	$\{\underline{\texttt{P19}}, \texttt{P937}, \ldots\}$	$\{\texttt{P138}, \ldots\}$
Q515	$\{\texttt{P135}, \ldots\}$	$\{\texttt{P31}, \texttt{P1813}, \ldots\}$
...	...	...

Approach: Candidate generation

- For each entity in E', we generate all possible query candidates:

Entity	Relations Target: Object	Relations Target: Subject
Q7604	$\{$P19, P937, ...$\}$	$\{$P138, ...$\}$
Q515	$\{$P135, ...$\}$	$\{$P31, P1813, ...$\}$
...	...	...

- In this case 930 query candidates are generated, including the correct query:

```
SELECT ?o WHERE {
  wd:Q7604 wdt:P19 ?o .
}
```

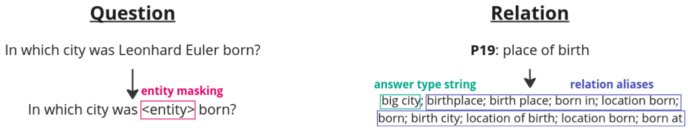
Approach: Relation Matching

- Illustration of relation scorer for the correct candidate:

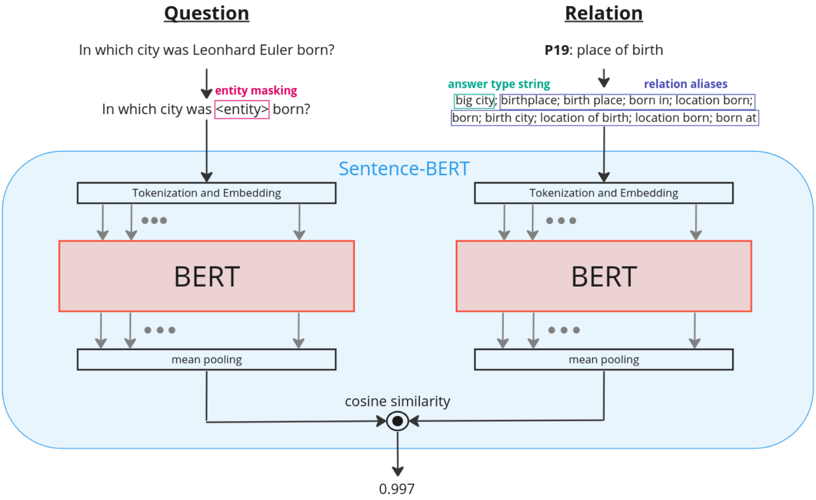
Question	Relation
In which city was Leonhard Euler born?	**P19**: place of birth

Approach: Relation Matching

- Illustration of relation scorer for the correct candidate:

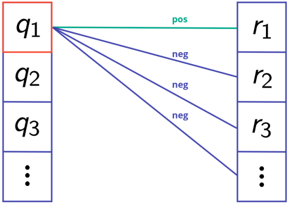


- Illustration of relation scorer for the correct candidate:



Approach: Relation Matching

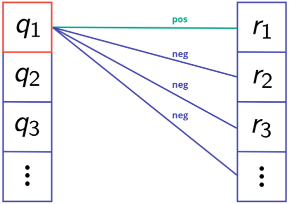
- Fine-tune relation scorer with the Multiple Negatives Ranking (MNR) loss function:
  - Create batches without duplicates, $q_1, \ldots, q_b$ question representations, $r_1, \ldots, r_b$ relation representations



  - Use cross entropy loss

Approach: Relation Matching

- Fine-tune relation scorer with the Multiple Negatives Ranking (MNR) loss function:
  - Create batches without duplicates, $q_1, ..., q_b$ question representations, $r_1, ..., r_b$ relation representations



  - Use cross entropy loss
- Alternative if few relations: contrastive loss function

Approach: Ranking

- Create feature vector for each candidate. Vector of correct candidate: $[1, 2, 174, 1, 2, 1, 4, 0, 0.997, 0.57, 3288499]$

Approach: Ranking

- Create feature vector for each candidate. Vector of correct candidate: $[1, 2, 174, 1, 2, 1, 4, 0, 0.997, 0.57, 3288499]$
- Use random forest model for binary classification to infer a pairwise ranking

Approach: Ranking

- Create feature vector for each candidate. Vector of correct candidate: $[1, 2, 174, 1, 2, 1, 4, 0, 0.997, 0.57, 3288499]$
- Use random forest model for binary classification to infer a pairwise ranking
- Compare each pair of candidates and sort candidates by number of "won" comparisons

Approach

Questions?

Evaluation: Datasets

- Three different benchmarks, all provide simple questions together with the corresponding gold query

Evaluation: Datasets

- Three different benchmarks, all provide simple questions together with the corresponding gold query
- SimpleQuestions-Wikidata: Translated from SimpleQuestions dataset, low variety in questions

Evaluation: Datasets

- Three different benchmarks, all provide simple questions together with the corresponding gold query
- SimpleQuestions-Wikidata: Translated from SimpleQuestions dataset, low variety in questions
- LC-QuAD 2.0 SQ: Simple questions of LC-QuAD 2.0 dataset

Evaluation: Datasets

- Three different benchmarks, all provide simple questions together with the corresponding gold query
- SimpleQuestions-Wikidata: Translated from SimpleQuestions dataset, low variety in questions
- LC-QuAD 2.0 SQ: Simple questions of LC-QuAD 2.0 dataset
- Own questions: 50 own questions, high variety

Evaluation: Results

- Accuracy: Fraction of questions, for which the answers of the predicted query are the same as the answers of the gold query

Evaluation: Results

- Accuracy: Fraction of questions, for which the answers of the predicted query are the same as the answers of the gold query
- Main results on the three benchmarks (AD is the average duration per question):

Dataset	Accuracy	AD
SimpleQuestions-Wikidata	0.816	0.49
LC-QuAD 2.0 SQ	0.825	0.57
Own questions	0.820	0.46

Evaluation: Results

- Accuracy on SimpleQuestions-Wikidata compared to the accuracies of other QA systems:

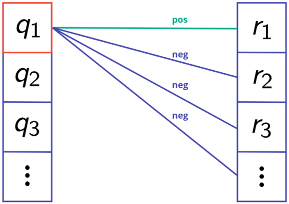
QA System	SimpleQuestions (FB2M)	SimpleQuestions-Wikidata
Yu et al. (2017)	0.787	-
Petrochuk et al. (2018)	0.781	-
Oliya et al. (2021)	-	0.682
Goette (2021)	-	0.586
Aqqu Wikidata (2023)	-	0.816

Questions?

Appendix: All features

ID	Name
1	Exact entity match
2	Exact entity token matches
3	Entity popularity score
4	Exact relation match
5	Literal score
6	Content literal score
7	Exact token matches
8	Similarity score
9	Relation score
10	Proportion matched/total tokens
11	Occurrences relation KG

Appendix: MNR loss



$$L_{MNR}(\mathbf{q}_i, \mathbf{r}_1, ..., \mathbf{r}_b) = -\log\left(\frac{\exp(s \cdot sim(\mathbf{q}_i, \mathbf{r}_i))}{\sum_{j=1}^{b} \exp(s \cdot sim(\mathbf{q}_i, \mathbf{r}_j))}\right),$$

$$\text{with } sim(\mathbf{q}, \mathbf{r}) = \frac{\mathbf{q} \cdot \mathbf{r}}{\|\mathbf{q}\| \|\mathbf{r}\|}$$

Appendix: Contrastive loss

Loss for single question-relation pair (embeddings $\mathbf{q}_i$, $\mathbf{r}_i$) and label y_i can be computed with

$$L_{CL}(\mathbf{q}_i, \mathbf{r}_i, y_i) = y_i \frac{1}{2} \|\mathbf{q}_i - \mathbf{r}_i\|_2 + (1 - y_i) \frac{1}{2} max(0, m - \|\mathbf{q}_i - \mathbf{r}_i\|_2)^2.$$

with m being a parameter that controls the influence of negative pairs.

Appendix: Results for different loss functions

	SimpleQuestions-Wikidata	LC-QuAD 2.0 SQ
MNR loss fine-tuning	0.799	0.825
contrastive loss fine-tuning	0.816	0.807

Appendix: Detailed evaluation

	SimpleQuestions-Wikidata	LC-QuAD 2.0 SQ	Own questions	AD
Full Pipeline	0.816	0.825	0.820	0.50
w/o rel score	0.673	0.808	0.760	0.44
w/o rel occs, w/o sim score	0.811	0.823	0.760	0.40
only rel and popularity score	0.792	0.785	0.740	0.38
entity sentence: marking	0.795	0.826	0.820	0.59
fine-tuning WikiQuestions	0.813	0.823	0.820	0.52
entity pruning: 200/500	0.818	0.819	0.820	1.76
no candidate pruning	0.816	0.825	0.820	2.01

Appendix: Results including top-k scores

Dataset	Accuracy	Top-2	Top-3	Top-5	Top-10	AD
SimpleQuestions-Wikidata	0.816	0.863	0.879	0.889	0.895	0.49
LC-QuAD 2.0 SQ	0.825	0.860	0.865	0.873	0.877	0.57
Own questions	0.820	0.880	0.920	0.960	0.960	0.46