Qsearch: Answering Quantity Queries from Text

Vinh Thinh Ho, Yusra Ibrahim, Koninika Pal, Klaus Berberich and Gerhard Weikum

Max Planck Institute for Informatics, Germany
Motivation

• Cars with price less than $100k

Results from a single page
Under some special structure (e.g., list)
Motivation

- Cars with price less than $99k in 2017 in Germany more...
Problem Statement

- **Input:**
  - Quantity query:
    - Cars with price less than 100k Euro in Germany
  - Text corpus:

- **Output: List of entity results**
  - Chevrolet Corvette Z06
    - $79,495
  - Chevrolet Camaro ZL1
    - $61,500
  - Ford Shelby GT350R
    - $67,135
  - etc.

from multiple pages
Related Work: QALD (Question Answering over Linked Data)

Typical QALD query:

fuel consumption of Toyota Prado

```
SELECT ?e ?x WHERE {
  ?e type SportUtilityVehicle .
  ?e hasFuelConsumption ?x
}
```

Can return any literal for ?x

Limited amount of quantities in KBs

Quantity query:

SUVs with fuel consumption below 20 MPG in city

```
SELECT ?e ?x WHERE {
  ?e type SportUtilityVehicle .
  ?e hasFuelConsumption ?x .
  ?x < 20 MPG
}
```

Needs to understand the quantity (measure, value, unit)
Related Work: QALD (Question Answering over Linked Data)

**Typical QALD query:**

```
fuel consumption of Toyota Prado
SELECT ?e ?x WHERE {
  ?e type SportUtilityVehicle .
  ?e hasFuelConsumption ?x .
} .
```

*Can return any literal for ?x*

**Quantity query:**

```
SUVs with fuel consumption below 20 MPG
SELECT ?e ?x WHERE {
  ?e type SportUtilityVehicle .
  ?e hasFuelConsumption ?x .
  ?x < 20 MPG .
} .
```

*Needs to understand the quantity (measure, value, unit)*

**Objectives:**

- High recall of quantities
- Understanding quantities
- Complex quantity conditions

➤ Tap directly into the text source

Limited amount of quantities in KBs
Outline

• Motivation

➤ Approach

• System Overview

• Evaluation
Computational Framework

“Cars with price under 100k Euro in Germany”

**Quantity-query (Qquery):**
- Semantic type ($t^*$)
- Quantity constraint ($q^*$): {value ($v$), unit ($u$), comparison operator ($o$)}
  - (equal, approximate, less/more than, interval)
- Context constraint ($X^*$): a bag of **important words** expressing the relation between $t^*$ and $q^*$

**Example:**
- Qquery: $t^*$: Car; $q^*$: (< 100.000 €); $X^*$: { price, Germany }
Computational Framework

**Quantity-fact (Qfact):**

- Entity \( (e) \)
- Quantity \( (q) \): {value \( (v) \), unit \( (v) \), value resolution \( (r) \)}
  
  (equal, approximate, less/more than, interval)

- Context \( (X) \): a bag of important words describing the connection between \( e \) and \( q \)

**Example:**

“BMW i8 has price about €138k in Germany and range from 50 to 60km battery only”

- Qfact 1: \( e: \) BMW i8; \( q: (= 138.000 \text{ €}) \); \( X: \) { price, Germany }
- Qfact 2: \( e: \) BMW i8; \( q: ([50-60] \text{ km}) \); \( X: \) { range, battery }
Outline

• Motivation

➢ Approach

➢ Qfact Extraction from Text

• Query Matching

• System Overview

• Evaluation
Qfact Extraction from Text

- Two steps: **Pre-processing** and **Extracting**

- **Step 1: Pre-processing**: Detecting entities (AIDA [Hoffart et al., 2011]) and quantities (Illinois Quantifier [Roy et al. 2015])

**Example:**

“According to BBC, BMW i8 has price of €138k in Germany and range from 50 to 60 km battery only”

- \(<\text{KB:BCC}\rangle\) (138.000 €)
- \(<\text{KB:BMW_i8}\rangle\) ([50-60] km)
- \(<\text{KB:Germany}\rangle\)
Qfact Extraction from Text

• Step 2: Extracting:
  • For each quantity in the pre-processed sentence, detect its linked entity and context

Q₁ in the input:
“According to <KB:BBC>, <KB:BMW_i8> has price of Q₁ in <KB:Germany> and range Q₂ battery only”

Q₂ in the input:
“According to <KB:BBC>, <KB:BMW_i8> has price of Q₁ in <KB:Germany> and range Q₂ battery only”
Qfact Extraction from Text

• **Step 2: Extracting:**
  • For each quantity in the pre-processed sentence, detect its linked entity and context
  • Tag the sentence with 3 tags: E, X, O

**Q₁ in the input:**
“According to <KB:BBC>, <KB:BMW_i8> has price of Q₁ in <KB:Germany> and range Q₂ battery only”

**Q₂ in the input:**
“According to <KB:BBC>, <KB:BMW_i8> has price of Q₁ in <KB:Germany> and range Q₂ battery only”
LSTM Tagging Model

- Process one quantity at a time

[He et al, ACL 2017]
Training Data?

• Match OpenIE outputs with the extracted entities and quantities.

“<KB:BMW_i8> has price of \(Q_1\) in <KB:Germany> and range \(Q_2\) battery only”

• Keep tuples with one quantity \(Q\).
• The entity \(e\) in the subject is chosen to link with \(Q\).
• Context tokens \(X\) are chosen based on POS tags.
Motivation

Approach
- Qfact Extraction from Text
  - Query Answering
- System Overview

Evaluation
Query Answering

“Cars with price under $100k”

Qquery \( (t^*; q^*; X^*) \)

Qfact \( (e; q; X) \)

- Match \( (t^* \text{ with } e) \) \&\& \( (q^* \text{ with } q) \) \&\& \( (X^* \text{ with } X) \)

- **Context matching:**
  - Probabilistic approach (KL-divergence)
  - **Embedding-based approach** (See paper for detail)
Outline

• Motivation

• Approach

➤ System Overview

• Evaluation
Qsearch Overview

Extract

1. Text
2. Extract Facts
   - Detect Entities
   - Detect Quantities
   - LSTM
3. Quantity Facts
   - \( L_{e_1} : (q_{11}, X_{11}); (q_{12}, X_{12}); \ldots \)
   - \( L_{e_2} : (q_{21}, X_{21}); (q_{22}, X_{22}); \ldots \)
   - \( L_{e_3} : (q_{31}, X_{31}); (q_{32}, X_{32}); \ldots \)
4. Qfact Storage

Answer

5. Quantity Query
   \( (t^*, q^*, X^*) \)
6. Candidate Entities
   - \( L_{c_1} : (q_{11}, X_{11}); \ldots \)
   - \( L_{c_2} : (q_{21}, X_{21}); \ldots \)
   - \( L_{c_3} : (q_{31}, X_{31}); \ldots \)
7. Match & Score
   - Quantity Matching
   - Context Matching
   - Aggregating
8. Final Result
# Result | Source
--- | ---
1 | Cadillac XLR
2 | Mercedes-Benz E-Class
3 | Cadillac SRX
4 | Nissan Leaf
5 | Nissan Murano

<table>
<thead>
<tr>
<th>#</th>
<th>Result</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cadillac XLR</td>
<td>The XLR <strong>base price</strong> was <strong>$ 98,000</strong> and by 2008 it reached $ 101,300 making it one of General Motors’ most expensive vehicles of all time.</td>
</tr>
<tr>
<td>2</td>
<td>Mercedes-Benz E-Class</td>
<td>I also drove an <strong>E500</strong>, which is powered by a 5 - liter 302 - horsepower V - 8 and has a <strong>base price</strong> of <strong>$ 55,515</strong>.</td>
</tr>
<tr>
<td>3</td>
<td>Cadillac SRX</td>
<td>Cadillac also currently sells a mid-size SUV, the <strong>SRX</strong>, which is <strong>priced from just over</strong> <strong>$ 38,000</strong>.</td>
</tr>
<tr>
<td>4</td>
<td>Nissan Leaf</td>
<td>For a more affordable option, we suggest the <strong>Nissan Leaf</strong>, at a <strong>base price</strong> of <strong>$ 30,000</strong>.</td>
</tr>
<tr>
<td>5</td>
<td>Nissan Murano</td>
<td>While <strong>Murano</strong> pricing starts <strong>below $ 29,000</strong>, upgrading to the higher SL trim level, all - wheel drive and options like a navigation system, antiskid control and leather seating will quickly run up the tab.</td>
</tr>
</tbody>
</table>
Outline

- Motivation
- Approach
- System Overview

▶ Evaluation
Experimental Setup

• **Datasets:** news articles - 7.6M documents
  - STICS [Hoffart et al, SIGIR 2014]
  - New York Times

• **Query domains:**
  - Finance
    - e.g., Companies whose stock value increased by at least 30 % in a year
  - Transport
    - e.g., Sport utility vehicles with engine power above 150 horsepower
  - Sports
    - e.g., Sprinters who ran 100 meter in less than 10 seconds
  - Technology
    - e.g., Power companies which generate more than 1000 MW electricity

• **Metrics:** Precision@K, Hit@K, MRR
Results

- **Two baselines:** Google & Elasticsearch

- See the paper for full evaluation results
Summary

- End-to-end QA system for Answering Quantity Queries from Text.
  - *Query* and *Fact* models
  - Extracting *Facts* & Matching *Facts* against *Query*

- *Qsearch* outperforms baselines for answering quantity queries.

- Working system in the following link:
  
  https://qsearch.mpi-inf.mpg.de/  
  (Ho et al, WSDM 2020)
Appendix
### Amount of Quantities in Wikidata & DBpedia

**Table 1.** Statistics on exemplary quantitative properties from Wikidata and DBpedia. 
- **#E**: number of entities; **#P**: with property present; **#Q**: with explicit data type for the property.

<table>
<thead>
<tr>
<th>Entity type/property</th>
<th>Wikidata</th>
<th></th>
<th></th>
<th>DBpedia</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#E</td>
<td>#P</td>
<td>#Q</td>
<td>#E</td>
<td>#P</td>
<td>#Q</td>
</tr>
<tr>
<td>Car model/range</td>
<td>3195</td>
<td>4</td>
<td>4</td>
<td>6705</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Car model/engine power</td>
<td>3195</td>
<td>0</td>
<td>0</td>
<td>6705</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mobile phone/display size</td>
<td>291</td>
<td>0</td>
<td>0</td>
<td>1358</td>
<td>1309</td>
<td>0</td>
</tr>
<tr>
<td>Marathon runner/best time</td>
<td>1629</td>
<td>18</td>
<td>18</td>
<td>3426</td>
<td>1346</td>
<td>601</td>
</tr>
</tbody>
</table>
## Evaluation – Qfact Extraction Model

Table 1: Evaluation of quantity fact extraction model of Qsearch.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Length</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Money</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Percentage</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>General</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>0.860</td>
<td>0.860</td>
<td>0.860</td>
<td>0.850</td>
<td>0.850</td>
<td>0.850</td>
<td>0.794</td>
<td>0.770</td>
<td>0.782</td>
<td>0.882</td>
<td>0.820</td>
<td>0.850</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>0.650</td>
<td>0.849</td>
<td>0.736</td>
<td>0.717</td>
<td>0.844</td>
<td>0.776</td>
<td>0.659</td>
<td>0.827</td>
<td>0.734</td>
<td>0.728</td>
<td>0.713</td>
<td>0.721</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>0.958</td>
<td>0.886</td>
<td>0.920</td>
<td>0.942</td>
<td>0.886</td>
<td>0.913</td>
<td>0.947</td>
<td>0.888</td>
<td>0.917</td>
<td>0.895</td>
<td>0.906</td>
<td>0.900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macro-avg.</td>
<td>0.823</td>
<td>0.865</td>
<td>0.839</td>
<td>0.836</td>
<td>0.860</td>
<td>0.846</td>
<td>0.800</td>
<td>0.828</td>
<td>0.811</td>
<td>0.835</td>
<td>0.813</td>
<td>0.824</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Qsearch with different scoring models

- **Datasets**: news articles (STICS + NYT) – 7.6M documents
- **Metrics**: Precision@K, Hit@K, MRR

<table>
<thead>
<tr>
<th>Metric</th>
<th>Finance</th>
<th>Transport</th>
<th>Sports</th>
<th>Technology</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr.@1</td>
<td>0.720</td>
<td>0.800</td>
<td>0.480</td>
<td>0.600</td>
<td>0.560</td>
</tr>
<tr>
<td>Pr.@3</td>
<td>0.667</td>
<td>0.747</td>
<td>0.480</td>
<td>0.480</td>
<td>0.507</td>
</tr>
<tr>
<td>Pr.@5</td>
<td>0.632</td>
<td>0.672</td>
<td>0.412</td>
<td>0.412</td>
<td>0.480</td>
</tr>
<tr>
<td>Pr.@10</td>
<td>0.604</td>
<td>0.608</td>
<td>0.333</td>
<td>0.379</td>
<td>0.412</td>
</tr>
<tr>
<td>Hit@3</td>
<td>0.880</td>
<td>0.920</td>
<td>0.760</td>
<td>0.760</td>
<td>0.760</td>
</tr>
<tr>
<td>Hit@5</td>
<td>0.880</td>
<td>0.960</td>
<td>0.760</td>
<td>0.760</td>
<td>0.920</td>
</tr>
<tr>
<td>MRR</td>
<td>0.792</td>
<td>0.870</td>
<td>0.621</td>
<td>0.678</td>
<td>0.685</td>
</tr>
</tbody>
</table>
Embedding-based Context Matching

- Match each term in query with its closest term in fact:

\[
\sum_{u \in X^*} W(u) \min_{v \in X} (dist(u, v)) + 1
\]

**Query context** \( X^* = \{ \text{price} \} \)

**Fact context** \( X = \{ \text{cost, Germany} \} \)

*directed embedding distance*: \( \text{ded}(X^* \rightarrow X) \)
Embedding-based Context Matching

• Penalize additional context terms of the facts

\[
ced(X^*, X) = ded(X^* \rightarrow X) \times ded(X \rightarrow X^*)^\alpha \\
= \left( \frac{\sum_{u \in X^*} W(u) \min_{v \in X} (dist(u, v))}{\sum_{u \in X^*} W(u)} + 1 \right) \times \left( \frac{\sum_{u \in X} W(u) \min_{v \in X^*} (dist(u, v))}{\sum_{u \in X} W(u)} + 1 \right)^\alpha
\]

*context embedding distance*: \( ced(X^*, X) \)

How well \( X^* \) matches \( X \)

How much additional terms of \( X \) shift its meaning
Related Work

• QA over Knowledge Graphs
  • [Abujabal et al, WWW 2018], [Zheng et al, PVLDB 2018]
  • [Xu et al, ACL 2016], [Bast et al, CIKM 2015]

• QA on Text
  • [Yang et al, EMNLP 2018], [Clark et al, ACL 2018], [Chen et al, ACL 2017]

• Recognizing numeric expressions from text
  • [Alonso et al, SIGIR 2018], [Saha et al, ACL 2017]
  • [Madaan et al, AAAI 2016], [Roy et al, TACL 2015]

• Neural-based Sequence Tagging
  • [He et al, ACL 2017], [FitzGerald et al, EMNLP 2015], [Zhou et al, ACL 2015]