Answering Questions with Complex Semantic Constraints on Open Knowledge Bases

Pengcheng Yin\# Nan Duan$ Ben Kao\# Junwei Bao$ Ming Zhou$

\# The University of Hong Kong $ Microsoft Research Asia

Speaker: Pengcheng Yin

pcyin@cs.hku.hk

The University of Hong Kong
Agenda

• **Background & Motivation**
  • Knowledge-based Question Answering (KB-QA) systems
  • How to answer *complex* questions?
  • Problems with the state-of-the-arts
  • Our contribution

• Approach

• Results

• Summary
Knowledge-based Question Answering (KB-QA)

• Task: Answer factoid questions in natural languages

Who wrote Harry Potter?

Transformation

Select ?author.
Where{
  Harry_Potter written_by ?x,
  ?x name ?author
}

Sparql query

Execution

Knowledge Bases (KBs)

1. Transform questions into structured queries
2. Execute the query against the KB to retrieve answers

J. K. Rowling
How to answer complex questions?

• Many existing KB-QA systems focus on answering questions with simple semantic constraints:

\[ Q_1: \text{What } [\text{is the currency of}]_{rel} \text{ Spain?} \]

\text{Answers: } \{ \text{Euro, Peseta} \}

• How to answer questions with complex semantic constraints?

\[ Q_2: \text{What } [\text{was the currency of}]_{rel} \text{ Spain } \text{before 2002?} \]

Expressed via prepositional/adverbial modifiers

Ex: Euro, Peseta
Two Families of KBs

curated KBs

open KBs
Curated KBs

- Curated KBs
  - Manually created, structured KBs based on predefined schema
  - Accurate and precise
- Problem: non-trivial to transform complex questions to structured queries!

*what was James K. Polk before he was president?*

```sql
SELECT ?job_title.
FROM Freebase
WHERE{
  James K. Polk government_position ?job.

  ?job to ?to_date.
  FILTER(?to_date < (  
    SELECT ?start_date.  
    WHERE{  
      ?job1 title President.  
      ?job1 from ?start_date.  
    })  
  ))
}
```
Open KBs

- open KBs [Galárraga et al., 2014; Fader et al., 2014]: Open domain KBs automatically extracted using open Information Extraction (IE) techniques.

(massive) Web Docs → Open Information Extraction → Open Knowledge Base

*Peseta was replaced by Euro as official tender of Spain in 2002.*

Open Information Extraction

**n-tuple assertion**

<table>
<thead>
<tr>
<th>Subject</th>
<th>Relation Phrase</th>
<th>Arguments</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peseta</td>
<td>was replaced</td>
<td>by Euro; as official tender of Spain; in 2002</td>
<td>0.89</td>
</tr>
</tbody>
</table>

- Each assertion has a subject field, a relation phrase field, and multiple argument fields.

*Different arguments are separated by semicolons.*
Open KBs

Open KB is a large collection of n-tuple assertions

- Knowledge is modeled as \( n \)-tuple \((n \geq 3)\) assertions
- Open KBs are unnormalized

<table>
<thead>
<tr>
<th>Subject</th>
<th>Relation Phrase</th>
<th>Arguments</th>
<th>Freq</th>
<th>Conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>James K. Polk</td>
<td>was</td>
<td>a governor; before he was president</td>
<td>2</td>
<td>0.87</td>
</tr>
<tr>
<td>the currency of Spain</td>
<td>was</td>
<td>the Peseta; before 2002</td>
<td>3</td>
<td>0.95</td>
</tr>
<tr>
<td>Peseta</td>
<td>was replaced</td>
<td>by Euro; as official tender of Spain; in 2002</td>
<td>3</td>
<td>0.81</td>
</tr>
<tr>
<td>Barack Obama</td>
<td>graduated</td>
<td>from Harvard Law School; in 1979 and 1991</td>
<td>4</td>
<td>0.77</td>
</tr>
<tr>
<td>Obama</td>
<td>graduated</td>
<td>magna cum laude; from Harvard Law School; in 1991</td>
<td>5</td>
<td>0.93</td>
</tr>
<tr>
<td>Barack Obama</td>
<td>attended</td>
<td>Harvard Law School</td>
<td>3</td>
<td>0.90</td>
</tr>
</tbody>
</table>

[Galárraga et al., 2014; Fader et al., 2014]
Open KB-QA and Open KBs

- $n$-tuple assertions contain rich semantic information
- Naturally handles complex questions

What was James K. Polk before he was president?

$n$-tuple query: 

\[ \langle \text{James K. Polk; was; } ?x, \text{ before he was president} \rangle \]

Open KB assertions

<table>
<thead>
<tr>
<th>Subject</th>
<th>Relation Phrase</th>
<th>Arguments</th>
<th>Freq.</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>James K. Polk</td>
<td>was</td>
<td>a governor; before he was a president</td>
<td>2</td>
<td>0.93</td>
</tr>
<tr>
<td>Peseta</td>
<td>was replaced</td>
<td>by Euro; as official tender of Spain; in 2002</td>
<td>3</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Governor
Existing Open KB-QA systems

- Existing open KB-QA systems are designed to work on open KBs of *triplet* assertions (with *single* argument)
  - Triplet assertions are NOT semantically rich!

<table>
<thead>
<tr>
<th>Subject</th>
<th>Relation Phrase</th>
<th>Argument</th>
<th>Freq.</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>James K. Polk</td>
<td>was</td>
<td>a governor</td>
<td>2</td>
<td>0.93</td>
</tr>
<tr>
<td>Peseta</td>
<td>was replaced</td>
<td>by Euro</td>
<td>3</td>
<td>0.89</td>
</tr>
</tbody>
</table>

- Cannot handle complex questions
  - *What was James K. Polk before he was president?*
  - *What was the currency of Spain before 2012?*
- It’s non-trivial to directly extend existing systems to work on *n*-tuple open KBs
Our Contribution

nOKB: \( n \)-tuple open KB

lavage rich semantic information in \( n \)-tuple assertions to answer complex questions

TAQA: \( n \)-Tuple Assertion-based Question Answering
Agenda

• Background & Motivation

• **Approach**
  • **TAQA**: n-**Tuple** Assertion-based **Question Answering**
  • Question Paraphrasing
  • Question Parsing
  • Open KB Querying
  • Answering Ranking

• Results

• Summary
TAQA’s workflow – An example

Q: Where did Barack Obama go to college in 1991?

Paraphrased Questions
Which university did Barack Obama attend in 1991?
F=[PMI=0.6, go to college ↔ attend=1.0, …]

Question Parsing
(Tarack Obama; attend; ?x) ∧ (?x; is-a; university)
F=[PMI=0.6, …, is_relaxed_query=1.0, …]

Open KB Querying
Candidate Answers
Harvard Law School F=[align_score=0.6, …]
Columbia University F=[align_score=0.6, …]
Punahou School F=[align_score=0.6, …]

Answer Ranking
Harvard Law School
Columbia University

Paraphrased Questions
Where did Barack Obama graduate from in 1991
F=[PMI=0.5, go to college ↔ graduate from=1.0, …]

Tuple Query
(Barack Obama; graduate; from ?x, in 1991)
F=[PMI=0.5, …, not_relaxed_query=1.0, …]
Question Paraphrasing

• paraphrase (rewrite) input question into multiple semantically similar questions

  • Motivation: bridge the lexical/syntactical gap between input questions and relevant KB assertions

**Question** Where did [Barack Obama] go to college in 1991?

**Noun Phrase**

**Prepositional Phrase**

**Relevant assertion in open KB**

<table>
<thead>
<tr>
<th>Subject</th>
<th>Rel. Phrase</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack Obama</td>
<td>graduated</td>
<td>from Harvard Law School; in 1979 and 1991</td>
</tr>
</tbody>
</table>

5 million Paraphrasing Templates

- Where did [None Phrase] go to college
- Where did [None Phrase] graduate from

(Templates provided by Fader et al., 2014)

**Paraphrased Question** Where did Barack Obama graduate from in 1991?

\[1\text{Detected via phrase chunking}\]
 Question Parsing

• Parse natural language questions into **tuple queries**
• Use *dependency parsing* to parse questions into **tuple queries**
  • Dep. parsing: powerful tool to analyze semantic relations between constituents
• Parse a question into tuple query by traveling through its *dependency tree*
• Generate fields in the tuple query from the descendants of the root node

*Where did Barack Obama graduate from in 1991?*

---

**Dependency Parse Tree**

**tuple query**

<table>
<thead>
<tr>
<th>Subject</th>
<th>Rel. Phrase</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack Obama</td>
<td>graduate; from ?x, in 1991</td>
<td></td>
</tr>
</tbody>
</table>

nsbj: nominal subject
prep: preposition
pobj: prepositional object
advmod: adverbial modifiers
aux: auxiliary verb
Open KB Querying

• Given a tuple query, how to query the open KB to retrieve answers?

• Key challenge: both the query and assertions in the open KB may have arbitrary number/order of arguments

In which year did Germany invade Poland in World War 2?

**Query**: Germany invade Poland in World War 2 in ?x

**Assertion**: Nazi Germany invaded Republic of Poland in 1939 in the beginning of World War 2

**Solution**: answer extraction as a matching problem on weighted bipartite graph
Open KB Querying

In which year did Germany invade Poland in World War 2?

- Fields in the query and assertion form two sets of nodes in the bipartite graph
- Define pairwise similarity between fields in query and assertion (1-to-1 matching)
- Get optimal matching solutions by maximizing global matching score

Answer is extracted from the field aligned with \(?x\) in optimal solutions

\(?x = 1939\) matching score = 2.35
Answer Ranking

- **Answer consolidation:**
  - The same answers can be derived from paraphrasing-parsing-querying pipeline
  - Merge the feature vectors of the answers with the same surface text form

- **Log-linear ranking model**
  - Use over 20K features to measure the probability that a candidate answer $a$ is an answer of $Q$

$$p(a|Q) = \frac{\exp\left\{ \sum_{k=1}^{M} \lambda_k \cdot f_k(a) \right\}}{\sum_{a' \in A} \exp\left\{ \sum_{k=1}^{M} \lambda_k \cdot f_k(a') \right\}}$$

  - Training the model by maximizing the log-likelihood on a set of question-answer pairs

$$\mathcal{L}(\mathcal{D}; \lambda) = \sum_{t=1}^{N} \log p(a_t|Q_t; \lambda)$$

  - Stochastic Gradient Descent learning (AdaGrad)
Agenda

• Background & Motivation
• Approach
• Results
  • \( n \)-tuple Open KB
  • Questions Sets
  • Experiment results
• Summary
All the existing open KBs are in triplet form (assertions contain only ONE argument)

We build nOKB+: ensemble of all extracted $n$-tuple assertions and existing triplet open KBs (Probase, NELL, etc)

Size of nOKB+: 0.8 billion assertions
Evaluation Question Sets

• **WebQuestions** [Berant et al., 2013]:
  • 5810 questions, mostly (95%+) with simple constraints
  • Manually answered on Freebase by crowdsourcing workers

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who played Jacob Black in twilight?</td>
<td>Taylor Lautner</td>
</tr>
<tr>
<td>Who did Roger Federer married?</td>
<td>Dominica</td>
</tr>
</tbody>
</table>

• **ComplexQuestions** [our work]: 300 questions with complex semantic constraints
  • 80 from WebQuestions, 220 newly collected
  • Manually labeled with gold-standard answers

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>What team did Shaq play for before the Lakers?</td>
<td>Orlando Magic</td>
</tr>
<tr>
<td>What country gained its independence from Britain in 1960?</td>
<td>Cyprus</td>
</tr>
<tr>
<td>What did France lose to the British in the treaty of Paris in 1763?</td>
<td>Dominica</td>
</tr>
</tbody>
</table>
End-to-End Evaluation

Accuracies \((\text{acc})\) on WebQuestions

Accuracies \((\text{acc})\) on ComplexQuestions

\[
\text{acc} = \frac{\text{number of correctly answered questions}}{\text{total number of questions}}
\]

- Baseline systems:
  - ParaSempre [Berant et al., 2014]: SOTA curated KB-QA system
  - OQA [Fader et al., 2014]: SOTA open KB-QA system
- TAQA performs well on both simple and complex questions
How can we further improve?

- Oracle Accuracy approximately 1.5-2.0x higher than (actual) accuracy
- Ideally TAQA is capable of answering at most 60% questions in two question sets

<table>
<thead>
<tr>
<th>Rank:</th>
<th>1</th>
<th>2-5</th>
<th>6-10</th>
<th>11-20</th>
<th>&gt;20</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComplexQuestions</td>
<td>59.9%</td>
<td>19.8%</td>
<td>8.1%</td>
<td>4.1%</td>
<td>8.1%</td>
</tr>
<tr>
<td>WebQuestions</td>
<td>62.7%</td>
<td>15.8%</td>
<td>6.4%</td>
<td>4.9%</td>
<td>10.2%</td>
</tr>
</tbody>
</table>

distribution of rank positions for oracle answers

- **Insight:** we can effectively improve TAQA’s accuracy by $\frac{1}{3}$ if we can improve the answer ranker to rank correct answers within top-5 up to top-1
**Summary**

**nOKB**: \( n \)-tuple open KB

**TAQA**: \( n \)-Tuple Assertion-based Question Answering

<table>
<thead>
<tr>
<th>Subject</th>
<th>Relation Phrase</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>James K. Polk</td>
<td>was</td>
<td>a governor; before he was president</td>
</tr>
<tr>
<td>the currency of Spain</td>
<td>was</td>
<td>the Peseta; before 2002</td>
</tr>
<tr>
<td>Peseta</td>
<td>was replaced</td>
<td>by Euro; as official tender of Spain; in 2002</td>
</tr>
<tr>
<td>Barack Obama</td>
<td>graduated</td>
<td>from Harvard Law School; in 1979 and 1991</td>
</tr>
<tr>
<td>Obama</td>
<td>graduated magna cum laude</td>
<td>from Harvard Law School; in 1991</td>
</tr>
<tr>
<td>Barack Obama</td>
<td>attended</td>
<td>Harvard Law School</td>
</tr>
</tbody>
</table>

**Question Paraphrasing**

Q: Where did Barack Obama go to college in 1991?

Paraphrased Questions:

- Which university did Barack Obama attend in 1991?
  \( F = [\text{PMI}=0.6, \text{go to college} \leftrightarrow \text{attend}=1.0, \ldots] \)

**Question Parsing**

Barack Obama; attend; ?x; a; university

\( F = [\text{PMI}=0.6, \ldots, \text{is_released}\_query=1.0, \ldots] \)

**Open KB Querying**

Candidate Answers:

- Harvard Law School
- Columbia University
- Peking University

Answer Ranking:

- Harvard Law School
- Columbia University
Thanks!

**TAQA**: n-Tuple Assertion-based Question Answering

Demo, Open KB and question set available at: [http://taqa.pcyin.me](http://taqa.pcyin.me)