Scalable Construction and Querying of Massive Knowledge Bases

Part II: Schema-agnostic Knowledge Base Querying

Yu Su
Department of Computer Science
University of California, Santa Barbara
Growing Gap between Human and Data

What disease does the patient have?
- (EMR) Similar patients?
- (Literature) New findings?
- (Gene sequence) Suspicious mutations?
- ...

Ad-hoc information needs for on-demand decision making

Massive, heterogeneous data

86.9% adoption (NEHRS 2015)
27M+ papers, >1M new/year (PubMed)
$1000 gene sequencing
24x7 monitoring
How can AI Bridge the Gap?

Bottleneck #1: Knowledge

Bottleneck #2: Access

Bottleneck #3: Reasoning

Insights
Discoveries
Solutions
## Structured Query: RDF + SPARQL

### Triples in an RDF Graph

<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barack_Obama</td>
<td>parentOf</td>
<td>Malia_Obama</td>
</tr>
<tr>
<td>Barack_Obama</td>
<td>parentOf</td>
<td>Natasha_Obama</td>
</tr>
<tr>
<td>Barack_Obama</td>
<td>spouse</td>
<td>Michelle_Obama</td>
</tr>
<tr>
<td>Barack_Obama_Sr.</td>
<td>parentOf</td>
<td>Barack_Obama</td>
</tr>
</tbody>
</table>

### RDF Graph

- Barack_Obama_Sr.
  - parentOf: Malia_Obama
  - parentOf: Natasha_Obama
- Barack_Obama
  - parentOf: Malia_Obama
  - spouse: Michelle_Obama
- Malia_Obama
- Natasha_Obama
- Michelle_Obama

### SPARQL Query

```
SELECT ?x WHERE {
  Barack_Obama_Sr. parentOf ?y.
  ?y parentOf ?x.
}
```

### Answer

```
<Malia_Obama>
<Natasha_Obama>
```
Why Structured Query Falls Short?

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th># Entities</th>
<th># Triples</th>
<th># Classes</th>
<th># Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freebase</td>
<td>45M</td>
<td>3B</td>
<td>53K</td>
<td>35K</td>
</tr>
<tr>
<td>DBpedia</td>
<td>6.6M</td>
<td>13B</td>
<td>760</td>
<td>2.8K</td>
</tr>
<tr>
<td>Google Knowledge Graph*</td>
<td>570M</td>
<td>18B</td>
<td>1.5K</td>
<td>35K</td>
</tr>
<tr>
<td>YAGO</td>
<td>10M</td>
<td>120M</td>
<td>350K</td>
<td>100</td>
</tr>
<tr>
<td>Knowledge Vault</td>
<td>45M</td>
<td>1.6B</td>
<td>1.1K</td>
<td>4.5K</td>
</tr>
</tbody>
</table>

* as of 2014

- It’s more than large: High heterogeneity of KBs
- If it’s hard to write SQL on simple relational tables, it’s only harder to write SPARQL on large knowledge bases
  - Even harder on automatically constructed KBs with a loosely-defined schema
“find all patients diagnosed with eye tumor”

WITH Traversed (cls,syn) AS (
  (SELECT R.cls, R.syn
   FROM XMLTABLE ('Document("Thesaurus.xml")
       /terminology/conceptDef/properties
       [property/name/text()='Synonym' and
        property/value/text()='Eye Tumor']
       /property[name/text()='Synonym']/value’
   COLUMNS
   cls CHAR(64) PATH ‘./*/parent::*/*/parent::*/
   /parent::*/*/name’,
   tgt CHAR(64) PATH’.’) AS R)

UNION ALL
  (SELECT CH.cls, CH.syn
   FROM Traversed PR,
      XMLTABLE ('Document("Thesaurus.xml")
       /terminology/conceptDef/definingConcepts/
       concept[./text()='$parent']/parent::*/*/parent::*/
       properties/property[name/text()='Synonym']/value’
   PASSING PR.cls AS "parent"
   COLUMNS
   cls CHAR(64) PATH ‘./*/parent::*/
   parent::*/*/parent::*/*/name’,
   syn CHAR(64) PATH’.’) AS CH))

SELECT DISTINCT V.*
FROM Visit V
WHERE V.diagnosis IN
  (SELECT DISTINCT syn FROM Traversed)

“Semantic queries by example”,
Lim et al., EDBT (2014)
In Pursue of Efficiency

**find all patients diagnosed with eye tumor**
In Pursue of Efficiency

find all patients diagnosed with eye tumor

Schema-agnostic Querying

WITH Transposed (clsרפ) AS (  
SELECT r FROM r)  
FROM Transposed  
WHERE r.parent="Eye Tumor"  
SELECT DISTINCT r.parent AS CHARS

WHERE r.parent IN  
(SELECT DISTINCT r FROM Transposed)
Outline

- Schema-agnostic Graph Query
- Natural Language Interface (a.k.a., Semantic Parsing)
  - A little history
  - Cold-start with crowdsourcing
  - Cold-start with neural transfer learning
Schemaless and Structureless Graph Querying

Shengqi Yang, Yinghui Wu, Huan Sun and Xifeng Yan
UC Santa Barbara

VLDB’14
“Find a professor, ~70 yrs., who works in Toronto and joined Google recently.”

Search intent

Graph query

A match (result)
## Query-KB Mismatch

<table>
<thead>
<tr>
<th>Knowledge Base</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>“University of Washington”</td>
<td>“UW”</td>
</tr>
<tr>
<td>“neoplasm”</td>
<td>“tumor”</td>
</tr>
<tr>
<td>“Doctor”</td>
<td>“Dr.”</td>
</tr>
<tr>
<td>“Barack Obama”</td>
<td>“Obama”</td>
</tr>
<tr>
<td>“Jeffrey Jacob Abrams”</td>
<td>“J. J. Abrams”</td>
</tr>
<tr>
<td>“teacher”</td>
<td>“educator”</td>
</tr>
<tr>
<td>“1980”</td>
<td>“~30”</td>
</tr>
<tr>
<td>“3 mi”</td>
<td>“4.8 km”</td>
</tr>
<tr>
<td>“Hinton” - “DNNresearch” - “Google”</td>
<td>“Hinton” - “Google”</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Schemaless Graph Querying (SLQ)

Query
Prof., 70 yrs.

A Match
Geoffrey Hinton (1947-)

✓ Acronym transformation: ‘UT’ → ‘University of Toronto’
✓ Abbreviation transformation: ‘Prof.’ → ‘Professor’
✓ Numeric transformation: ‘~70’ → ‘1947’
✓ Structural transformation: an edge → a path
## Transformations for KB-Query Mismatch

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Category</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>First/Last token</td>
<td>String</td>
<td>“Barack Obama” &gt; “Obama”</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>String</td>
<td>“Jeffrey Jacob Abrams” &gt; “J. J. Abrams”</td>
</tr>
<tr>
<td>Prefix</td>
<td>String</td>
<td>“Doctor” &gt; “Dr”</td>
</tr>
<tr>
<td>Acronym</td>
<td>String</td>
<td>&quot;International Business Machines&quot; &gt; &quot;IBM&quot;</td>
</tr>
<tr>
<td>Synonym</td>
<td>Semantic</td>
<td>“tumor&quot; &gt; “neoplasm&quot;</td>
</tr>
<tr>
<td>Ontology</td>
<td>Semantic</td>
<td>&quot;teacher&quot; &gt; &quot;educator&quot;</td>
</tr>
<tr>
<td>Range</td>
<td>Numeric</td>
<td>“~30” &gt; “1980”</td>
</tr>
<tr>
<td>Unit Conversion</td>
<td>Numeric</td>
<td>&quot;3 mi&quot; &gt; &quot;4.8 km&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Candidate Match Ranking

Query: $Q$

Candidate Match: $\varphi(Q)$

Features

- Node matching features: $F_V(v, \varphi(v)) = \sum_i \alpha_i f_i(v, \varphi(v))$
- Edge matching features: $F_E(e, \varphi(e)) = \sum_j \beta_j g_j(e, \varphi(e))$

Overall Matching Score

$$P(\varphi(Q) | Q) \propto \exp(\sum_{v \in V_Q} F_V(v, \varphi(v)) + \sum_{e \in E_Q} F_E(e, \varphi(e)))$$
Exploiting Relevance Feedback in Knowledge Graph Search

Yu Su, Shengqi Yang, Huan Sun, Mudhakar Srivatsa, Sue Kase, Michelle Vanni, and Xifeng Yan
UC Santa Barbara, IBM Research, Army Research Lab

KDD’15
Query-specific Ranking via Relevance Feedback

- Generic ranking: sub-optimal for specific queries
  - By “Washington”, user A means Washington D.C., while user B might mean University of Washington

- Query-specific ranking: tailored for each query
  - But need additional query-specific information for further disambiguation

Relevance Feedback:
1. Given user query, generate initial ranking results
2.1. Explicit feedback: Users indicate the (ir)relevance of a handful of answers
2.2. Pseudo feedback: Blindly assume top-10 initial results are correct
3. Improve ranking with feedback information
Problem Definition

\( Q \) : A graph query
\( G \): A knowledge graph
\( \phi(Q) \): A candidate match to \( Q \)
\( F(\phi(Q) \mid Q, \theta) \): A generic ranking function
\( \mathcal{M}^+ \): A set of positive/relevant matches of \( Q \)
\( \mathcal{M}^- \): A set of negative/non-relevant matches of \( Q \)

**Graph Relevance Feedback (GRF):**
Generate a query-specific ranking function \( \tilde{F} \) for \( Q \) based on \( \mathcal{M}^+ \) and \( \mathcal{M}^- \)
Query-specific Tuning

- $\theta$ represents (query-independent) feature weights. However, each query carries its own view of feature importance.
- Find query-specific $\theta^*$ that better aligned with the query using user feedback.

$$g(\theta^*) = (1 - \lambda) \left( \frac{\sum_{\phi(Q) \in M^+} F(\phi(Q) | Q, \theta^*)}{|M^+|} - \frac{\sum_{\phi(Q) \in M^-} F(\phi(Q) | Q, \theta^*)}{|M^-|} \right) + \lambda R(\theta, \theta^*)$$

User Feedback

Regularization
Type Inference

- Infer the implicit type of each query node
- The types of the positive entities constitute a composite type for each query node

Query

Positive Feedback

Candidate Nodes

- Babak Parviz (Professor)
- Brian Otis (Professor)
- Barack Obama (Politician)
- Steven Seitz (Professor)
- ProFlowers (Corporation)
- University of Washington (University, Organization)
- Google (Corporation)
Context Inference

- **Entity context**: 1-hop neighborhood of the entity
- The contexts of the positive entities constitute a composite context for each query node
Experimental Setup

- Knowledge base: DBpedia (4.6M nodes, 100M edges)
- Graph query sets: WIKI and YAGO

**YAGO Class**

Naval Battles of World War II Involving the United States

**Graph Query**

- **Structured Information need**
- **Links between YAGO and DBpedia**

**Answer**

- Battle of Midway
- Battle of the Caribbean

... ...
Evaluation with Explicit Feedback

- Explicit feedback: User gives relevance feedback on top-10 results
- GRF boosts SLQ by over 100%
- Three GRF components complement each other

Metric: mean average precision (MAP)@K

(a) WIKI
(b) YAGO
Evaluation with Pseudo Feedback

- Pseudo feedback: Blindly assume top-10 results from initial run are correct
- Erroneous feedback information but zero user effort

<table>
<thead>
<tr>
<th>MAP@K</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLQ_WIKI</td>
<td>0.23</td>
<td>0.21</td>
<td>0.24</td>
<td>0.25</td>
<td>0.27</td>
<td>0.28</td>
</tr>
<tr>
<td>GRF_WIKI</td>
<td>0.73</td>
<td>0.58</td>
<td>0.52</td>
<td>0.50</td>
<td>0.49</td>
<td>0.49</td>
</tr>
<tr>
<td>SLQ_YAGO</td>
<td>0.40</td>
<td>0.35</td>
<td>0.33</td>
<td>0.32</td>
<td>0.36</td>
<td>0.39</td>
</tr>
<tr>
<td>GRF_YAGO</td>
<td>0.82</td>
<td>0.66</td>
<td>0.60</td>
<td>0.57</td>
<td>0.58</td>
<td>0.61</td>
</tr>
</tbody>
</table>
Outline

- Schema-agnostic Graph Query
- Natural Language Interface (a.k.a., Semantic Parsing)
  - A little history
  - Cold-start with crowdsourcing
  - Cold-start with neural transfer learning
Natural Language Interface $\approx$ Model-Theoretic Semantics

Language Variations

Utterance

Symbol Grounding

Executable logical form (SQL, $\lambda$-calculus, ...)

World (knowledge base, database, ...)

Denotation

find the first kid of Queen Elizabeth II

semantic parsing

execution

Charles, Prince of Wales
Rule-based Natural Language Interface

editor> add verb
what is your verb? exceed
what is its third sing. pres? exceeds
what is its past form? exceeded
what is its perfect form? exceeded
what is its participle form? exceeding
to what set does the subject belong? numeric
is there a direct object? yes
to what set does it belong? numeric
is there an indirect object? no
is it linked to a complement? no
what is its predicate? greater_than
do you really wish to add this verb? y
<table>
<thead>
<tr>
<th>Period</th>
<th>Rule-based</th>
<th>Statistical</th>
<th>Neural</th>
</tr>
</thead>
<tbody>
<tr>
<td>1960s-1990s</td>
<td>• Manually designed rules</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Deterministic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1990s-2010s</td>
<td></td>
<td>• Low</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Few</td>
<td></td>
</tr>
<tr>
<td>2015-present</td>
<td></td>
<td></td>
<td>• Mostly applied on relational databases</td>
</tr>
</tbody>
</table>
Statistical Natural Language Interface

[Berant et al., 2013]

[Berant and Liang, 2015]
<table>
<thead>
<tr>
<th>Semantic Mapping</th>
<th>Rule-based</th>
<th>Statistical</th>
<th>Neural</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naturalness</td>
<td>• Manually designed rules</td>
<td>• Manually designed rules/features</td>
<td>• More</td>
</tr>
<tr>
<td></td>
<td>• Deterministic</td>
<td>• Learn weights from data</td>
<td>• Better</td>
</tr>
<tr>
<td>Training Data</td>
<td>• Low</td>
<td>• Few</td>
<td>• Better</td>
</tr>
<tr>
<td>Portability</td>
<td>• Few</td>
<td>• More</td>
<td>• Relational databases, knowledge bases</td>
</tr>
<tr>
<td></td>
<td>• Low</td>
<td>• Low</td>
<td>• Better</td>
</tr>
</tbody>
</table>

Mostly applied on relational databases.
Deep Learning

Accurate, Generic, Simple

Object recognition: Krizhevsky, Sutskever, Hinton 2012

Speech recognition: Graves, Mohamed, Hinton 2013

“Hey Siri, play some jazz music”


He loved to eat

Er liebte zu essen
Neural Natural Language Interface

Encoder

Decoder

\[ p(y_1 = \text{"argmin"} \mid v) \]
\[ y_2 = \text{"("} \mid v, y_1 \]
\[ y_3 = \text{"child"} \mid v, y_{1:2} \]
\[ \cdots \]

[first kid of Elizabeth II]
[eldest kid of Elizabeth II]
[Her Majesty’s first child]
[Queen Elizabeth’s firstborn]

[Dong & Lapata 2016], [Jia & Liang 2016], [Mei et al., 2016]
<table>
<thead>
<tr>
<th>1960s-1990s</th>
<th>1990s-2010s</th>
<th>2015-present</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rule-based</strong></td>
<td><strong>Statistical</strong></td>
<td><strong>Neural</strong></td>
</tr>
<tr>
<td>Semantic mapping</td>
<td>• Manually designed rules</td>
<td>• Manually designed Rules/features</td>
</tr>
<tr>
<td>Naturalness</td>
<td>• Deterministic</td>
<td>• Learn weights from data</td>
</tr>
<tr>
<td>Training Data</td>
<td>• Low</td>
<td>• Better</td>
</tr>
<tr>
<td>Portability</td>
<td>• Few</td>
<td>• More</td>
</tr>
<tr>
<td></td>
<td>• Low</td>
<td>• Best</td>
</tr>
<tr>
<td></td>
<td>• Mostly applied on relational databases</td>
<td>• Better</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Relational databases, knowledge bases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Both features and weights learned from data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Best</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• A LOT more</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Relational databases, knowledge bases, web tables, APIs, ...</td>
</tr>
</tbody>
</table>
Outline

- Schema-agnostic Graph Query
- Natural Language Interface (a.k.a., Semantic Parsing)
  - A little history
  - Cold-start with crowdsourcing
  - Cold-start with neural transfer learning
Portability: the Cold Start Problem

“I want to build an NLI for my domain, but I don’t have any user and training data’’
How to Build NLI for New Domain

- 1950s-1990s: Rule engineering (for rule-based NLI)
- 1990s-2010s: Feature engineering (for statistical NLI)
- 2015-present: Data engineering (for neural NLI)

- Crowdsourcing
- Neural transfer learning

[Auxerre and Inder, 1986]
Deep Learning with Weak Supervision

**Strong Supervision**
- In-domain, on-task

**Weak Supervision**
- In-domain, off-task
- Out-of-domain, on-task
- Out-of-domain, off-task

Text → Knowledge
How to Collect NLI Training Data?

- Training data: \{(\text{utterance}, \text{logical form})\}

- (“Who did nine-eleven?”, $\lambda x.\text{involved}\_\text{in}\_\text{attach}(x, \text{September}\_11\_\text{attacks})$)

- (“How many children of Eddard Stark were born in Winterfell?”,
  \[\text{count}(\lambda x.\text{children}(\text{Eddard}\_\text{Stark}, x) \land \text{place}\_\text{of}\_\text{birth}(x, \text{Winterfell}))\]

...
How to Collect NLI Training Data?

- If we already have utterances (questions/commands/queries/...) from users...

“How many children of Eddard Stark were born in Winterfell?”

\[
\text{count}(\lambda x. \text{children}(\text{Eddard}\_\text{Stark}, x) \land \text{place}\_\text{of}\_\text{birth}(x, \text{Winterfell}))
\]
How to Collect NLI Training Data?

- But for most domains we are interested in, there is yet any user, nor any utterance
- Ask domain experts to do everything?

“How many children of Eddard Stark were born in Winterfell?”

\[
\text{count}(\lambda x. \text{children}(\text{Eddard_Stark}, x) \land \text{place_of_birth}(x, \text{Winterfell}))
\]

- Do not scale
- Not representative
How to Collect NLI Training Data?

- Can we only use crowd workers?
- Crowd workers do not understand formal languages!

\[
\text{count}(\lambda x. \text{children}(\text{Eddard_Stark}, x) \land \text{place_of_birth}(x, \text{Winterfell}))
\]
How to Collect NLI Training Data?

- Can we only use crowd workers?
- Crowd workers do not understand formal languages!

1: Logical form generation

\[ \text{count}(\lambda x. \text{children}(\text{Eddard_Stark}, x) \land \text{place_of_birth}(x, \text{Winterfell})) \]
A General Framework for Crowdsourcing NLI Data

1: Logical form generation

\[
\text{count}(\lambda x.\text{children}(\text{Eddard Stark}, x) \land \text{place_of_birth}(x, \text{Winterfell}))
\]

2: Canonical utterance generation

“How many children of Eddard Stark were born in Winterfell?”

3: Paraphrasing via crowdsourcing

“What is the number of person who is born in Winterfell, and who is child of Eddard Stark?”

[Building a Semantic Parser Overnight, Wang et al. 2015]
Advantages

- Scalable
  - Low-cost annotation, applicable to many domains

- Configurable
  - Full control on what to annotate and how many to get

- Complete coverage
  - Fully exercise the formal language and data

- Representative (partially)
  - Natural wording
  - Do not capture distribution of user interests

\[
\text{count}(\lambda x. \text{children}(\text{Eddard Stark}, x) \land \text{place_of_birth}(x, \text{Winterfell}))
\]

"What is the number of person who is born in Winterfell, and who is child of Eddard Stark?"

"How many children of Eddard Stark were born in Winterfell?"
Challenges

- Logical form generation
  - How to automate and configure?
  - What logical forms are “relevant”?
  - How many to generate (huge candidate space)

- Canonical utterance generation
  - How to minimize the expertise requirement and workload for grammar design

- Paraphrasing via crowdsourcing
  - How to optimize the crowdsourcing process, i.e., select the right logical forms to annotate
  - How to control and improve result quality
  - How to encourage diversity

\[
\text{count(} \lambda x. \text{children(Eddard_Stark, x)} \ \land \ \text{place_of_birth}(x, \text{Winterfell})\text{)}
\]

“How many children of Eddard Stark were born in Winterfell?”

“What is the number of person who is born in Winterfell, and who is child of Eddard Stark?”

“What is the number of person who is born in Winterfell, and who is child of Eddard Stark?”
On Generating Characteristic-rich Question Sets for QA Evaluation

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gur, Zenghui Yan, Xifeng Yan
UCSB, OSU, Army Research Lab, IBM Research

EMNLP’16
Motivation

- Existing datasets for knowledge based question answering (KBQA) mainly contain *simple questions*
  - WebQuestions, SimpleQuestions, etc.

  “Where was Obama born?”

  “What party did Clay establish?”

  “What kind of money to take to bahamas?”

  … …
Multi-dimensional Benchmarking

☐ Structural complexity
  ■ “People who are on a gluten-free diet can’t eat what cereal grain that is used to make challah?”

☐ Quantitative analysis (function)
  ■ “In which month does the average rainfall of New York City exceed 86 mm?”

☐ Commonness
  ■ “Where was Obama born?” vs.
  ■ “What is the tilt of axis of Polestar?”

☐ Paraphrase
  ■ “What is the nutritional composition of coca-cola?”
  ■ “What is the supplement information for coca-cola?”
  ■ “What kind of nutrient does coke have?”

☐ ...

☐ ...
Configurable Benchmark Construction

Freebase
53K classes, 35K relations, 45M entities, 3B facts

Natural Language Paraphrases

- “Find people who died from lung cancer, same as their parent.”
- “From those lung cancer deaths, list the ones whose parent has the same cause of death”

Logical Form

V1: Graduate students
V2: Crowdsourcing (multi-stage quality control), 10x scale
# Functions

<table>
<thead>
<tr>
<th>Category</th>
<th>Counting</th>
<th>Superlative</th>
<th>Comparative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions</td>
<td>count</td>
<td>max and min</td>
<td>argmax and argmin</td>
</tr>
<tr>
<td>Domain</td>
<td>Question node</td>
<td>Question node of numeric class</td>
<td>Template/grounded node of numeric class</td>
</tr>
<tr>
<td>Example</td>
<td>count</td>
<td>min</td>
<td>Concert Venue</td>
</tr>
<tr>
<td></td>
<td>spaceports</td>
<td>internal storage</td>
<td>capacity</td>
</tr>
<tr>
<td></td>
<td>NASA</td>
<td>Ipad</td>
<td>Integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distilled Spirit</td>
</tr>
<tr>
<td></td>
<td>Rocket Launch Site</td>
<td>Float</td>
<td>alcoholByVolume</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>40.0</td>
</tr>
<tr>
<td>Question</td>
<td>How many launch sites does nasa have?</td>
<td>What’s the smallest internal storage of ipad?</td>
<td>Find the largest concert venue.</td>
</tr>
</tbody>
</table>
Too Many Graph Queries

- Freebase: 24K classes, 65K relations, 41M entities, 596M facts
- Easily generate millions of graph queries
- Which ones correspond to *relevant* questions?

1: Logical form generation
Commonness Checking

ClueWeb+FACC1:
1B documents, 10B entity mentions

<table>
<thead>
<tr>
<th>Entity Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
</tr>
<tr>
<td>James_Southam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation Probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location.contains</td>
</tr>
<tr>
<td>Chromosome.identifier</td>
</tr>
</tbody>
</table>
### GraphQuestions

- 5166 questions, 148 domains, 506 classes, 596 relations

| Question                                                                 | Domain       | Answer     | # of edges | Function | $\log_{10}(p(q))$ | $|A|$ |
|--------------------------------------------------------------------------|--------------|------------|------------|----------|-------------------|------|
| Find terrorist organizations involved in **September 11 attacks**.       | Terrorism    | alQaeda    | 1          | none     | -16.67            | 1    |
| The **September 11 attacks** were carried out with the involvement of what terrorist organizations? |             |            |            |          |                   |      |
| Who did **nine eleven**?                                                 |              |            |            |          |                   |      |
| How many children of **Eddard Stark** were born in **Winterfell**?       | Fictional Universe | 3          | 2          | count    | -23.34            | 1    |
| **Winterfell** is the home of how many of **Eddard Stark**’s children?   |              |            |            |          |                   |      |
| What’s the number of **Ned Stark**’s children whose birthplace is **Winterfell**? |              |            |            |          |                   |      |
| In which month does the average rainfall of **New York City** exceed 86 mm? | Travel       | March, August... | 3          | comp.    | -37.84            | 7    |
| Rainfall averages more than 86 mm in **New York City** during which months? |              |            |            |          |                   |      |
| List the calendar months when **NYC** averages in excess of 86 millimeters of rain? |              |            |            |          |                   |      |
Testbest for Research Progress


<table>
<thead>
<tr>
<th>F1-score</th>
<th>Berant and Liang EMNLP’13</th>
<th>Yao and Van ACM’14</th>
<th>Durme ACL’14</th>
<th>Berant and Liang ACL’14</th>
<th>Peng et al. ACL’17</th>
<th>Reddy et al. EMNLP’17</th>
<th>Li et al. EMNLP’17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.8</td>
<td>5.1</td>
<td>12.8</td>
<td>17.02</td>
<td>17.7</td>
<td>20.4</td>
<td></td>
</tr>
</tbody>
</table>
“What is the nutritional composition of coca-cola?”
“What is the supplement information for coca-cola?”
“What kind of nutrient does coke have?”

Benchmark Results on Paraphrasing

“Learning to Paraphrase for Question Answering”
Dong et al., *EMNLP* (2017)
(Su et al., 2016)
The Quest of Compositionality

\[
\text{[people who are on a gluten-free diet]}_{\text{rel1}} \quad \text{[can’t eat]}_{\text{rel2}} \quad \text{[what cereal grain that is used to make challah]}_{\text{rel3}}
\]

Benchmark Results on Structural Complexity

Further study on compositionality in CIKM’17 and SIGIR’18 (under review) (Su et al., 2016)
GraphQuestions V2 (coming soon)

- 10 to 20 times larger in scale
- Support more benchmarking scenarios
  - Cross-domain transfer learning, few- or zero-shot learning, compositionality, etc.
Which logical forms are of a high value for training NLI?

- \( \text{GET-Messages\{COUNT()\}} \)
  - “How many emails do I have?”

- \( \text{GET-Messages\{FILTER(isRead=False)\}} \)
  - “unread emails”

- \( \text{GET-Messages\{COUNT(), FILTER(isRead=False)\}} \)
  - “How many unread emails do I have”

Utterances follow the composition structure of API calls

Predict the language model of an API call without annotating it!

Crowdsourcing optimization
Outline

- Schema-agnostic Graph Query
- Natural Language Interface (a.k.a., Semantic Parsing)
  - A little history
  - Cold-start with crowdsourcing
  - Cold-start with neural transfer learning
How to Build NLI for New Domain

- 1950s-1990s: Rule engineering (for rule-based NLI)
- 1990s-2010s: Feature engineering (for statistical NLI)
- 2015-present: Data engineering (for neural NLI)
  - Crowdsourcing
  - Neural transfer learning

Out-of-domain, on-task supervision!
What is **Transferrable** in NLI across Domains?

**Source Domain: Basketball**

*In which season did Kobe Bryant play for the Lakers?*

\[ p(\text{rel(team)} | "play for") \]

**Target Domain: Social**

*When did Alice start working for Mckinsey?*

\[ p(\text{employee} | "work for") \]

\[ R[\text{season}]. (\text{player.KobeBryant } \sqcap \text{team.Lakers}) \]

\[ R[\text{start}]. (\text{employee.Alice } \sqcap \text{employer.Mckinsey}) \]

[EMNLP’17]
Cross-domain NLI via Paraphrasing

In which season did Kobe Bryant play for the Lakers?

\[ p(\text{"whose team is"}|\text{"play for"}) \]

\[ \text{play} \approx \text{work}, \ \text{team} \approx \text{employer} \]

\[ p(\text{"whose employer is"}|\text{"work for"}) \]

When did Alice start working for Mckinsey?

\[ p(\text{"play for"}|\text{"work for"}) \]

\[ \text{When did Alice start working for Mckinsey?} \]

Start date of employee Alice whose employer is Mckinsey

\[ \text{Start date of employee Alice whose employer is Mckinsey} \]

\[ \text{Season of Player Kobe Bryant whose team is Lakers} \]

\[ \text{R[season]. (player.KobeBryant \ \land \ \text{team.Lakers})} \]

\[ \text{automatic} \]

\[ \text{R[start]. (employee.Alice \ \land \ \text{employer.Mckinsey})} \]

\[ \text{When did Alice start working for Mckinsey?} \]

[EMNLP’17], inspired by [Berant & Liang, 2014], [Wang et al., 2015]
Seq2Seq Model for Paraphrasing

- Seq2Seq + Bi-directional encoder + Attentive decoder
- Learn to predict whether input utterance paraphrases canonical utterance
- Deterministic mapping between canonical utterance and logical form
Word Embedding

- Word $\triangleq$ Dense vector (typically 50-1000 dimensional)
- Word similarity $\triangleq$ Vector similarity
- Pre-trained on huge external text corpora

Fine-grained Similarity

```
“play” = [0.2, 0.4, 0.3]  
“work” = [0.1, 0.6, 0.2]
```

Linguistic Regularity

Out-of-domain, off-task supervision!
Pre-trained Word Embedding Alleviates Vocabulary Shifting

Vocabulary shifting: Only 45%~70% target domain vocabulary are covered by source domains\[1\]

Pre-trained word embedding can alleviate the vocabulary shifting problem

- **Word2vec**: 300-d vectors pre-trained on the 100B-token Google News Corpus; vocabulary size = 3M

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Calendar</th>
<th>Housing</th>
<th>Restaurants</th>
<th>Social</th>
<th>Publications</th>
<th>Recipes</th>
<th>Basketball</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>71.1</td>
<td>60.7</td>
<td>55.8</td>
<td>46.0</td>
<td>65.6</td>
<td>71.9</td>
<td>45.6</td>
<td>61.7</td>
</tr>
<tr>
<td>+word2vec</td>
<td>93.9</td>
<td>90.9</td>
<td>90.4</td>
<td>89.3</td>
<td>95.6</td>
<td>97.3</td>
<td>89.4</td>
<td>93.8</td>
</tr>
</tbody>
</table>

Overnight Dataset: 8 KBs

\[1\] Wang et al. Building a Semantic Parser Overnight. 2015

(Su et al., 2017)
Neural Transfer Learning for NLI

Source Domain

Output

Target Domain

ENCODER

STATE

DECODER

θ

ENCODER

STATE

DECODER

Word Embedding $\phi$

- Input utterance $x = (x_1, ..., x_m)$, canonical utterance $y = (y_1, ..., y_n)$
- Embedding: $\phi(x) = (\phi(x_1), ..., \phi(x_m))$, $\phi(y) = (\phi(y_1), ..., \phi(y_n))$
- Learning on source domain: $p(\phi(y)|\phi(x), \theta)$
- Warm start on target domain: $p(\phi(y)|\phi(x), \theta)$
- Fine-tuning on target domain: $p(\phi(y)|\phi(x), \theta^*)$

[EMNLP’17]
Experimental Setup

- **Dataset:** Overnight [Wang et al., 2015]
  - 8 domains (Social, Basketball, Restaurant, etc.)
- **Metric:** average accuracy
- **Transfer learning setup**
  - For each target domain, use the other 7 domains as source
- **Word embedding initialization**
  - **Random:** Randomly draw from uniform distribution with unit variance $U(-\sqrt{3}, \sqrt{3})$
  - **Word2vec:** 300-dimensional word2vec (skip-gram) embedding pre-trained on 100B-word News corpus
Direct Use of Word2vec Fails Dramatically...

- Transfer learning works (new state of the art)
- Word2vec brings 6.2% absolute decrease in accuracy

<table>
<thead>
<tr>
<th></th>
<th>In-domain</th>
<th>Cross-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wang et al. (2015)</td>
<td>58.8</td>
<td></td>
</tr>
<tr>
<td>Xiao et al. (2016)</td>
<td>72.7</td>
<td></td>
</tr>
<tr>
<td>Jia and Liang (2016)</td>
<td>75.8</td>
<td></td>
</tr>
<tr>
<td>Ours + Random</td>
<td>75.7</td>
<td>76.9</td>
</tr>
<tr>
<td>Ours + Word2vec</td>
<td>74.9</td>
<td></td>
</tr>
</tbody>
</table>

[EMNLP’17]
Problems of Pre-trained Word Embedding

- **Small micro variance**: hurt optimization
  - Activation variances $\approx$ input variances [Glorot & Bengio, 2010]
  - Small input variance implies poor exploration in parameter space

- **Large macro variance**: hurt generalization
  - Distribution discrepancy between training and testing

**Micro Variance**
Variance of the values comprising a vector

**Macro Variance**
Variance among different vectors

$$n = |V|$$

<table>
<thead>
<tr>
<th>Initialization</th>
<th>L2 norm</th>
<th>Variance</th>
<th>Cosine Sim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>$17.3 \pm 0.45$</td>
<td>$1.00 \pm 0.05$</td>
<td>$0.00 \pm 0.06$</td>
</tr>
<tr>
<td>WORD2VEC</td>
<td>$2.04 \pm 1.08$</td>
<td>$0.02 \pm 0.02$</td>
<td>$0.13 \pm 0.11$</td>
</tr>
</tbody>
</table>
Proposed Solution: Standardization

- Standardize each word vector to unit variance
- But it was unclear before why standardization should be applied on pre-trained word embedding
  - Obvious downside: make loss function of word embedding sub-optimal

<table>
<thead>
<tr>
<th>Initialization</th>
<th>L2 norm</th>
<th>Variance</th>
<th>Cosine Sim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>17.3 ± 0.45</td>
<td>1.00 ± 0.05</td>
<td>0.00 ± 0.06</td>
</tr>
<tr>
<td>Word2vec</td>
<td>2.04 ± 1.08</td>
<td>0.02 ± 0.02</td>
<td>0.13 ± 0.11</td>
</tr>
<tr>
<td>Word2vec + ES</td>
<td>17.3 ± 0.05</td>
<td>1.00 ± 0.00</td>
<td>0.13 ± 0.11</td>
</tr>
</tbody>
</table>

Random: randomly draw from uniform distribution with unit variance
Word2vec: pre-trained word2vec embedding
ES: per-example standardization (per column)
Standardization Fixes the Variance Problems

- Standardization brings 8.7% absolute increase
- Transfer learning brings another 2.4% increase

[Wang et al. (2015)] [Xiao et al. (2016)] [Jia and Liang (2016)] [Ours + Random] [Ours + Word2vec] [Ours + Word2vec+ES]

[EMNLP’17]
Recap

“I want to build an NLI for my domain, but I don’t have any training data”

Can I collect training data via crowdsourcing?
- Yes, and it’s not so expansive
- Cost can be further reduced by crowdsourcing optimization

Can I leverage existing training data from other domains?
- Yes, if you turn it into a paraphrasing problem
- Pre-trained word embedding can greatly help neural transfer learning, but only when properly standardized
FUTURE RESEARCH
How can AI Bridge the Gap?

Bottleneck #1: Knowledge

Bottleneck #2: Access

Bottleneck #3: Reasoning

Insights
Discoveries
Solutions
#3: Knowledge-based Machine Reasoning

similar molecular structure

target same gene

treat

treat

similar root cause
Methodological Exploration

- Inherent structure of the NLI problem space
  - Strong prior for learning
  - Key: compositionality of natural & formal languages

- Integration of neural and symbolic computation
  - Neural network modularized over symbolic structures
  - (Cognitive science) neural encoding of symbolic structures

- Goal-oriented human-computer conversation
  - Accommodate dynamic hypothesis generation and verification in a natural conversation
End-to-end General-purpose Knowledge Engine

“Which cement stocks go up the most when a Category 3 hurricane hits Florida?”
Thanks &