Towards Democratizing Data Science with AI-Powered Knowledge Engines

Yu Su
Microsoft Semantic Machines
The Ohio State University
Data-Driven Decision Making

What disease does the patient have?

$P(\text{Disease} \mid C_1, C_2, C_3)$

- **C1**: Inquiry
- **C2**: Examination
- **C3**: Literature
Growing Gap between Human and Data

*What disease does the patient have?*
- EMR => Similar patients?
- Literature => New discoveries?
- 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 to Democratize Data Science?
AI-Powered Knowledge Engine

Bottleneck #1: Knowledge

Bottleneck #2: Access

Bottleneck #3: Reasoning

Discoveries
Decisions
Actions
Knowledge Base

1970s-1990s

2000s-present

Non-expert user
Query
Advice

User Interface
Expert System
Inference Engine
Knowledge Base

Knowledge from an expert

Texts

bing
AOL
Yahoo!
Google

Freebase

Hi, I'm Cortana.
Ask me a question!

DBpedia
yago
select knowledge

Echo

Selecting knowledge
Knowledge Base

- Encyclopedic knowledge about concepts, entities and their relationships (facts)
  - Google Knowledge Graph: **570M entities** and **18B facts** (2014)
Methodology: 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
KNOWLEDGE HARVESTING FROM MASSIVE TEXT
Knowledge Base Construction from Text

- Entity recognition and linking
- Relation extraction: binary, n-ary (event)

High-throughput cell-based screening of 4910 known drugs and drug-like small molecules identifies **Disulfiram** as an inhibitor of prostate cancer cell growth

Relation: inhibit

<table>
<thead>
<tr>
<th>Subject</th>
<th>Object</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disulfiram</td>
<td>Prostate Cancer</td>
<td>0.85</td>
</tr>
</tbody>
</table>

"Alcohol-abuse drug disulfiram targets cancer via p97 segregase adaptor NPL4"
Scalable Relation Extraction with Distant Supervision

**place_of_birth**: (Michael Jackson, US)

*Training*

Michael Jackson was born in the US.

Born in the US, Michael Jackson was one of ...

I visited the birthplace of Michael Jackson in Gary, Indiana, United Stated.

*Learn & Generalize*

Barack Obama was born in the US.

... nearby Stratford, birthplace of Justin Bieber ...

The German-born American physicist Albert Einstein revolutionized ...

*Extraction*

(Barack Obama, US)

(Justin Bieber, Stratford)

(Albert Einstein, Germany)

E.g., [Mintz et al., 2009], [Riedel et al., 2010], [Zeng et al., 2015], [Lin et al., 2016], …
Global Statistics of Relations

- Number of co-occurrences of KB-textual relation pairs in the entire corpus.

<table>
<thead>
<tr>
<th>Relation</th>
<th>nsubjpass</th>
<th>born</th>
<th>nmod:in</th>
<th>nsubj</th>
<th>died</th>
<th>nmod:in</th>
</tr>
</thead>
<tbody>
<tr>
<td>place_of_birth</td>
<td></td>
<td>0.73</td>
<td></td>
<td></td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>nationality</td>
<td></td>
<td>0.15</td>
<td></td>
<td></td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>place_of_death</td>
<td></td>
<td>0.01</td>
<td></td>
<td></td>
<td>0.89</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Meaning of this textual relation:

Global

Local

Text Corpus

Knowledge Base

[NAACL’18]

Word embedding analogy: GloVe (global statistics) vs. Word2vec (local statistics)
Textual Relation Embedding with Global Statistics

ClueWeb: 500M web documents

Freebase: 45M entities, 3B facts
Evaluation on Newswire Corpus

- **Dataset:** New York Times corpus, 53 target relations
  - place_of_birth, place_of_death, founder_of, employee_of, etc.

- The learned textual relation embedding improves the STOA method by 5.9% (top 1,000 extracted facts)
Knowledge Base Construction: Food for Thought

- (Open-world) probabilistic KBs
  - Model uncertainties of the real world

- Multi-modal KBs
  - Images, audio, video, temporal-special info

- (Dynamic) distributed KBs
  - Personal KBs (at edge) + a public KB (in the cloud)
(Open-World) Probabilistic KBs

- KB: `place_of_birth(John, United States)`
- Query: “Does John speak English?”
- Closed-world assumption: “No.”
- Open-world assumption: “I don’t know.”
- Open-world probabilistic KB: “99% yes.”

- Challenges
  - Uncertainty modeling and probability calibration
  - Efficient querying
  - Combination of logic-based reasoning and machine learning based reasoning

Some examples: YAGO, NELL, Google Knowledge Vault
Multi-Model KBs

source: RoboBrain by Saxena et al.

source: visualgenome.org
(Dynamic) Distributed KBs
NATURAL LANGUAGE INTERFACE
Writing formal queries is a pain...

"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)
In Pursue of Efficiency

find all patients diagnosed with eye tumor
In Pursue of Efficiency

find all patients diagnosed with eye tumor

```
WITH Traversed (cls, spn) AS {
  @:'kr/kr' as spn
  FROM Terminology
  WHERE ('ancestor' = 'Entire/Anatomy') and ('descendant' = 'Eye')
  UNION ALL
  @:cls.CH as CH
  FROM Traversed CH
  UNION ALL
  @:cls.CH as CH
  FROM Traversed CH
  WHERE ('ancestor' = 'Entire/Anatomy') and ('descendant' = 'Eye')
  SELECT DISTINCT v
  FROM Terminology
  WHERE v.diagno = 'Eye Tumor'
}
```

Natural Language Interface
Natural Language Interface $\approx$ Model-Theoretic Semantics

Language Variations

Utterance

Symbolic Expression

Semantic Parsing

Execution

Denotation

Charles, Prince of Wales
COLD START
The Cold Start Problem

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

- 1950s-1990s: Rule engineering (rule-based systems)
- 1990s-2010s: Feature engineering (statistical ML)
- 2010s-present: Data engineering (neural models)

```
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
```

[Auxerre and Inder, 1986]
Crowdsourcing

User Interaction

Transfer Learning
Cross-domain Natural Language Interface

Source Domain

Natural Language Interface

Knowledge Transfer

Target Domain

Natural Language Interface

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?

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

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

Target Domain: Social

When did Alice start working for Mckinsey?

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

\[ p(\text{rel(employer)}|"\text{work for"}) \]
Cross-domain NLI via Paraphrasing

In which season did Kobe Bryant play for the Lakers?

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

\( \text{play} \approx \text{work}, \text{team} \approx \text{employer} \)

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

When did Alice start working for Mckinsey?

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

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

automatic

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

\[ \text{Start date of employee Alice whose employer is Mckinsey} \]
Pre-trained Word Embedding

- Word $\equiv$ Dense vector (typically 50-1000 dimensional)
- Word similarity $\equiv$ Vector similarity
- Pre-trained on large 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></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>Coverage</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>

Neural Transfer Learning for NLI

Source Domain

\[ \text{Encoder} \rightarrow \text{Decoder} \]

Target Domain

\[ \text{Encoder} \rightarrow \text{Decoder} \]

Word Embedding \( \phi \)

- Input utterance \( x = (x_1, \ldots, x_m) \), canonical utterance \( y = (y_1, \ldots, y_n) \)
- Embedding: \( \phi(x) = (\phi(x_1), \ldots, \phi(x_m)), \phi(y) = (\phi(y_1), \ldots, \phi(y_n)) \)
- 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: \( p(\phi(y)|\phi(x), \theta^*) \)
Direct Use of Word2vec Fails Dramatically...

- Cross-domain: for each target domain, use all others as source domain
- Word2vec brings 6.2% absolute decrease in accuracy
Pre-trained Word Embedding: What May Be Wrong?

- **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

$$n = |V|$$

$$\begin{bmatrix}
  \vdots \\
  \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_n
\end{bmatrix}$$

<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>
</tbody>
</table>

Micro Variance
Variance of the values comprising a vector

Macro Variance
Variance among different vectors
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

<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

<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>69.5</td>
<td>74.9</td>
</tr>
<tr>
<td>Ours + Word2vec+ES</td>
<td>80.6</td>
<td></td>
</tr>
</tbody>
</table>

In-domain vs. Cross-domain comparison.
Let machines understand human thinking
Don’t let humans think like machines

WHAT’S NEXT?
Bridging the Gap between Human and Data: AI-Powered Knowledge Engine

Discoveries
Decisions
Actions

Natural Language Interface

Knowledge-based Reasoning

Knowledge Harvesting
Natural Language Interface for Data Analytics

Study:
use nighttime luminosity observed by satellites as a proxy measure of development and welfare

Command (high-level):
calculate the average nighttime luminosity near roads in China in 1994

Command (implementation):

# Get China less gas flares polygon
arcpy.Select_analysis("countries_nogas", "china1.shp", "\"NAME\" = 'China'\")
# Average two satellites for 1994
outRaster = (Float("F101994") + Float("F121994")) / 2
outRaster.save("FXC1994")

# Use buffer tool and roads to make polygon of China close to roads, then clip chinal1 to this
arcpy.Buffer_analysis("a2010_final_proj", "roadbuff.shp", "0.5 DecimalDegrees", "FULL", "ROUND", "ALL", "")
arcpy.Clip_analysis("H:/Research/Data/Lights/china1.shp", "H:/Research/Data/Lights/roadbuff.shp", "china2.shp", "")

# Clip each lights raster to extent of china2
rasterList = arcpy.ListRasters("F*")
for raster in rasterList:
    arcpy.Clip_management(raster, "-179.9999 90.0 180.0 83.62741", "G"+str(raster[1:]), "H:/Research/Data/Lights/china2.shp", "", "ClippingGeometry")

# Create grid to extent of one of new light rasters
arcpy.CreateFishnet_management("ch_grid.shp", "73.55416 18.15416", "73.5541 28.15416", "0.1", "0.1", "0", "134.77916 53.5625", "NO_LABELS", "G101992", "POLYGON")
arcpy.RasterToPolygon_conversion("G101992", "G101992p.shp", "NO_SIMPLIFY", "Value")

# Process: Clip grid to perimeter of polygon
arcpy.Clip_analysis("H:/Research/Data/Lights/ch_grid.shp", "H:/Research/Data/Lights/G101992p.shp", "china_grid.shp", "")

# Zonal statistics on each year
rasterList = arcpy.ListRasters("G*")
for raster in rasterList:
arcpy.gp.ZonalStatisticsAsTable_sa("H:/Research/Data/Lights/china_grid.shp", "FID", raster, "1"+str(raster[5:])+".dbf", "DATA", "MEAN")
Natural Language Interface for Data Analytics

- Transduce natural language commands into programs

- Allow users to stay focused on high-level thinking and decision making, instead of overwhelmed by low-level implementation details

- Two steps
  - Simple commands → single function calls
    - [CIKM’17], [SIGIR’18]
  - Complex commands → programs of multiple function calls
Knowledge-based Machine Reasoning

- Similar molecular structure
- Target same gene

Similar root cause
Methodological Exploration

- Inherent structure of the NLI problem space
  - Strong prior for learning
  - Key: compositionality of natural & formal languages [CIKM’17]

- Integration of neural and symbolic computation
  - Neural network modularized over symbolic structures [SIGIR’18]
  - (Cognitive science) neural encoding of symbolic structures

- Goal-oriented human-computer conversation
  - Accommodate dynamic hypothesis generation and verification in a natural conversation
  - Challenge: open-ended, no fixed frames
AI-Powered Knowledge Engine: Applications

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