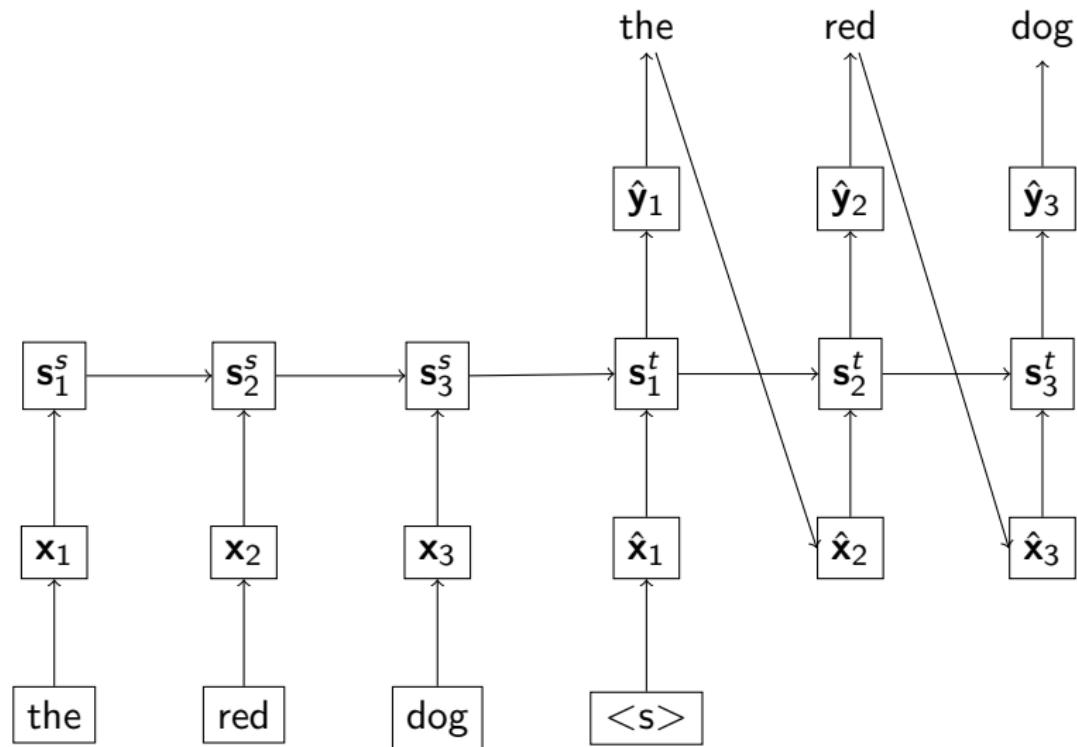


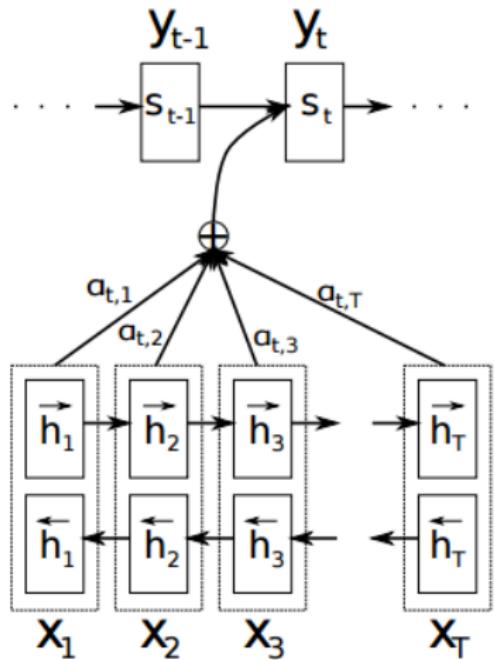
Question Answering/Semantic Parsing

CS 287

(Slides from Yoav Artzi, Cornell NY)

Sequence-to-Sequence





Question

It has become a popular strategy to handle sequential prediction problems using a seq2seq setup such as attention-based translation. How might you handle the following problems:

1. Segmentation (HW4)
2. Rare Word Replacement (HW3)
3. Part-of-Speech Tagging (HW2-HW5)

Semantics

Branch of linguistics focused on meaning

- ▶ Lexical Semantics
 - ▶ Meaning of individual words
 - ▶ e.g. BoW models, word2vec
- ▶ Compositional Semantics
 - ▶ Meaning of utterances
 - ▶ Concerned with the relations of meaning
 - ▶ Often expressed with logical relations

Today's Lecture

Survey of Question Answering

- ▶ Semantic Parsing: Linguistic model of questions and answers
- ▶ Knowledge Bases and Datasets
- ▶ IR-style approaches: Watson and Jeopardy!

Contents

Lambda Calculus

Data and Problems

Semantic Parsing

Jeopardy!

Lambda Calculus

- Formal system to express computation
- Allows high-order functions

$$\lambda a. move(a) \wedge dir(a, LEFT) \wedge to(a, \iota y. chair(y)) \wedge \\ pass(a, \mathcal{A}y. sofa(y) \wedge intersect(\mathcal{A}z. intersection(z), y))$$

Lambda Calculus

Base Cases

- Logical constant
- Variable
- Literal
- Lambda term

Lambda Calculus

Logical Constants

- Represent objects in the world

NYC, CA, RAINIER, LEFT, ...
located_in, depart_date, ...

Lambda Calculus

Variables

- Abstract over objects in the world
- Exact value not pre-determined

x, y, z, \dots

Lambda Calculus

Literals

- Represent function application

city(AUSTIN)

located_in(AUSTIN, TEXAS)

Lambda Calculus

Literals

- Represent function application

city(AUSTIN)

located_in(AUSTIN, TEXAS)

Predicate

Arguments

Logical expression

List of logical expressions

Lambda Calculus

Lambda Terms

- Bind/scope a variable
- Repeat to bind multiple variables

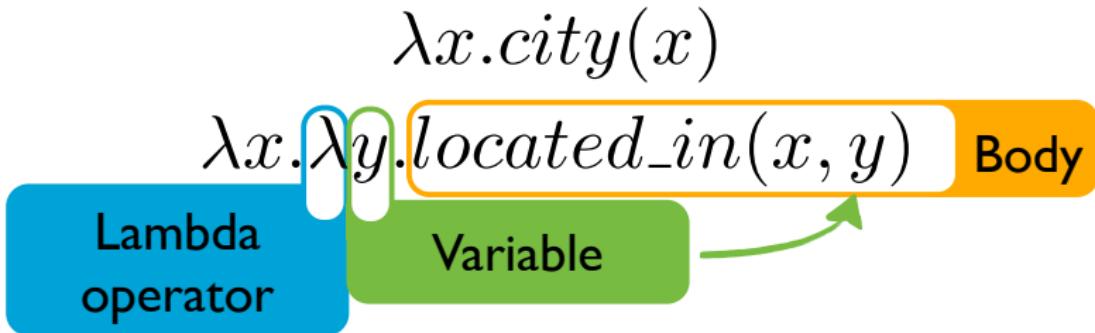
$$\lambda x.\text{city}(x)$$
$$\lambda x.\lambda y.\text{located_in}(x, y)$$

Lambda Calculus

Lambda Terms

- Bind/scope a variable
- Repeat to bind multiple variables

$$\lambda x. \lambda y. located_in(x, y)$$

The diagram illustrates a lambda term with three colored callouts. A blue callout labeled "Lambda operator" covers the first $\lambda x.$. A green callout labeled "Variable" covers the second $\lambda y.$ An orange callout labeled "Body" covers the expression $located_in(x, y)$. A green arrow points from the "Variable" callout to the y in the body expression, indicating it is a bound variable.

Lambda operator

Variable

Body

Lambda Calculus

Quantifiers?

- Higher order constants
- No need for any special mechanics
- Can represent all of first order logic

$$\forall(\lambda x.\text{big}(x) \wedge \text{apple}(x))$$
$$\neg(\exists(\lambda x.\text{lovely}(x)))$$
$$\iota(\lambda x.\text{beautiful}(x) \wedge \text{grammar}(x))$$

Lambda Calculus

Syntactic Sugar

$$\wedge(A, \wedge(B, C)) \Leftrightarrow A \wedge B \wedge C$$

$$\vee(A, \vee(B, C)) \Leftrightarrow A \vee B \vee C$$

$$\neg(A) \Leftrightarrow \neg A$$

$$Q(\lambda x. f(x)) \Leftrightarrow Qx. f(x)$$

$$\text{for } Q \in \{\iota, \mathcal{A}, \exists, \forall\}$$

$$\begin{aligned} & \lambda x. flight(x) \wedge to(x, move) \\ & \lambda x. flight(x) \wedge to(x, NYC) \\ & \lambda x. NYC(x) \wedge x(to, move) \end{aligned}$$

- ✗ $\lambda x.\text{flight}(x) \wedge \text{to}(x, \text{move})$
- ✓ $\lambda x.\text{flight}(x) \wedge \text{to}(x, \text{NYC})$
- ✗ $\lambda x.\text{NYC}(x) \wedge x(\text{to}, \text{move})$

Simply Typed Lambda Calculus

- Like lambda calculus
- But, typed

 $\lambda x. flight(x) \wedge to(x, move)$

 $\lambda x. flight(x) \wedge to(x, NYC)$

 $\lambda x. NYC(x) \wedge x(to, move)$

Lambda Calculus

Typing

- Simple types
- Complex types

t Truth-value
 e Entity

$\langle e, t \rangle$

$\langle\langle e, t \rangle, e \rangle$

Lambda Calculus

Typing

- Simple types
- Complex types

t Truth-value
 e Entity

Type constructor

$\langle e, t \rangle$

$\langle \langle e, t \rangle, e \rangle$

Domain

Range

Lambda Calculus

Typing

t
e
— tr
— loc
—

- Simple types
- Complex types

Type
constructor

$\langle e, t \rangle$

$\langle \langle e, t \rangle, e \rangle$

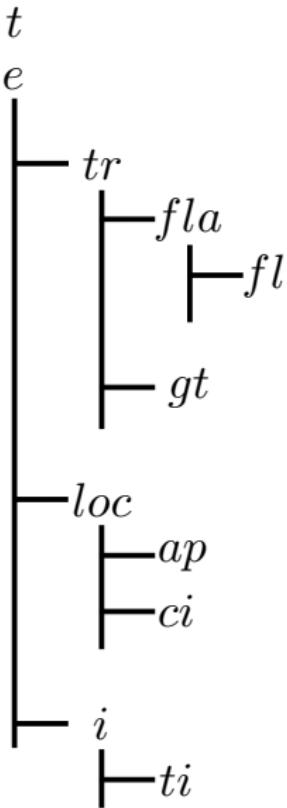
Domain

Range

- Hierarchical typing system

Lambda Calculus

Typing



- Simple types
- Complex types

Type
constructor

$\langle e, t \rangle$

$\langle \langle e, t \rangle, e \rangle$

Domain

Range

- Hierarchical typing system

Simply Typed Lambda Calculus

$$\lambda a. move(a) \wedge dir(a, LEFT) \wedge to(a, \iota y. chair(y)) \wedge \\ pass(a, \mathcal{A}y. sofa(y) \wedge intersect(\mathcal{A}z. intersection(z), y))$$

Type information usually omitted

Capturing Meaning with Lambda Calculus

State		
Abbr.	Capital	Pop.
AL	Montgomery	3.9
AK	Juneau	0.4
AZ	Phoenix	2.7

Border	
State1	State2
WA	OR
WA	ID
CA	OR
CA	NV
CA	AZ

Mountains	
Name	State
Bianca	CO
Antero	CO
Rainier	WA
Shasta	CA
Wrangell	AK
Snowy	CO
Rockies	CO



Show me mountains in states
bordering Texas

g show me flights from New
York to LA departing on
Thursday



Flights from **New York, NY** (all airports) to **Los Angeles, CA (LAX)**

Depart

Thu, Jan 30

Return

Mon, Feb 3

Nonstop only

- United from \$1,034
- Alaska from \$1,034
- American from \$1,034
- JetBlue from \$1,034
- Virgin America from \$1,034
- Delta from \$1,054

All flights Nonstop and connecting

- Delta from \$488
- AirTran from \$682
- Other airlines from \$803



Web



Images



News

MORE

Parsing as Structure Prediction

$$\begin{array}{ccccccc} \text{show} & \text{me} & \text{flights} & \text{to} & \text{Boston} \\ \hline S/N & & N & PP/NP & NP \\ \lambda f.f & & \lambda x.\text{flight}(x) & \lambda y.\lambda x.\text{to}(x,y) & BOSTON \\ & & & & \overrightarrow{PP} \\ & & & & \lambda x.\text{to}(x, BOSTON) \\ & & & & \overline{N \setminus N} \\ & & & & \lambda f.\lambda x.f(x) \wedge \text{to}(x, BOSTON) \\ & & & & \overleftarrow{N} \\ & & & & \lambda x.\text{flight}(x) \wedge \text{to}(x, BOSTON) \\ \hline & & & S & & & \\ & & & \lambda x.\text{flight}(x) \wedge \text{to}(x, BOSTON) & & & \end{array}$$

Contents

Lambda Calculus

Data and Problems

Semantic Parsing

Jeopardy!

Dataset: Geoquery

- ▶ Maps natural language questions to lambda calculus.
- ▶ Assume a database with numerical values as well
(`population(TEXAS)`)
- ▶ Queries contain rich nesting

```
size(stateid(X), S) :-  
area(stateid(X), S).  
size(cityid(X,St), S) :-  
population(cityid(X,St), S).  
size(riverid(X), S) :-  
len(riverid(X),S).  
size(placeid(X), S) :-  
elevation(placeid(X),S).  
size(X,X) :-  
number(X).
```

```
next_to(stateid(X),stateid(Y)) :-  
border(X,_,Ys),  
member(Y,Ys).
```

Quiz: Geoquery

- ▶ What states border Texas?
- ▶ What is the largest state?
- ▶ What states border the state that borders the most state?

(Syntactic Sugar: $\text{arg max}(f, g)$ returns max of element of g that satisfies predicate f .)

Geoquery (Zelle and Mooney, 1996)

a) What states border Texas

$$\lambda x. state(x) \wedge borders(x, \text{texas})$$

b) What is the largest state

$$\arg \max(\lambda x. state(x), \lambda x. size(x))$$

c) What states border the state that borders the most states

$$\begin{aligned} & \lambda x. state(x) \wedge borders(x, \arg \max(\lambda y. state(y), \\ & \quad \lambda y. count(\lambda z. state(z) \wedge borders(y, z)))) \end{aligned}$$

Figure 1: Examples of sentences with their logical forms.

WebQuestions (Berant, 2013)

Questions:

- ▶ what high school did president bill clinton attend?
- ▶ what form of government does russia have today?
- ▶ what movies does taylor lautner play in?

Answers:

- ▶ Hot Springs High School
http://www.freebase.com/view/en/bill_clinton
- ▶ Constitutional republic
<http://www.freebase.com/view/en/russia>
- ▶ Eclipse, Valentine's Day, The Twilight Saga: Breaking Dawn - Part 1, New Moon
http://www.freebase.com/view/en/taylor_lautner

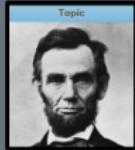
Here's an example:

We know that [William Shakespeare](#) wrote the play [Hamlet](#). We describe this in Freebase as such:

William Shakespeare		
is a	type	Author
has a	property	Works Written
with a	value	Hamlet

Important! Freebase is read-only and will be shut-down. [More...](#)

Created by book_bot on 5/7/2009

Topic [Abraham Lincoln](#) enmid: /m/0gzh notable type: /government/us_president notable for: /government/us_president on the web: [Wikipedia.org](#) ↗

Abraham Lincoln was the 16th president of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—its bloodiest war and its greatest moral, constitutional and political crisis. In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy. Reared in a poor family on the western frontier, Lincoln was a self-educated lawyer in Illinois, a Whig Party leader, state legislator during the 1830s. Lincoln was elected to Congress in 1846, where he promoted rapid modernization of the economy through banks, tariffs, and railroads. He had originally agreed not to run for a second term and his opposition to the Mexican–American War was unpopular among the voters. He returned to Springfield and concentrated on his successful law practice throughout central Illinois. He returned to politics in 1854, and was a leader in building up the new Republican Party, which was had a statewide majority. After a series of highly publicized debates in 1858, during which Lincoln spoke out against the expansion of slavery, he lost the U.S. [−]

Properties

I18n

Keys

Links



View and edit specific domains, types, or properties...

Filter options: Show all domains and properties

Common /common

Freebase Commons

Topic /common/topic

▼ Types:

Common

Topic

Film

Film story contributor

Film subject

Government

US President

Politician

Political Appointee

U.S. Congressperson

Medicine

Public figure with medical condition

Military

Military Person

Military Commander

Books

Literature Subject

Also known as /common/topic/alias

Also known as

Honest Abe

Abe Lincoln

The Buffoon

Caesar

Father Abraham

The Flatboat Man

The Grand Wrestler

The Great Emancipator

The Illinois Baboon

The Jester

58 values total »

Description /common/topic/description

Abraham Lincoln was the 16th president of the United States, serving from March 1861 until his assassination in April 1865. Lincoln led the United States through its Civil War—the bloodiest war and its greatest moral, constitutional and political crisis. In doing so, he preserved the Union, abolished slavery, strengthened the federal government, and modernized the economy. Reared in a poor family on the western frontier, Lincoln was a self-educated lawyer in Illinois, a Whig Party leader, state legislator during the 1830s. Lincoln was elected to Congress in 1846, where he promoted rapid modernization of the economy through banks, tariffs, and railroads. He



Item Discussion

Read View history

Search



Abraham Lincoln (Q91)

16th President of the United States

Abe Lincoln | Lincoln | Honest Abe

▼ In more languages Configure

Language	Label	Description	Also known as
English	Abraham Lincoln	16th President of the United States	Abe Lincoln Lincoln Honest Abe
German	Abraham Lincoln	US-amerikanischer Präsident	
Spanish	Abraham Lincoln	decimosexto presidente de los Estados Unidos	
Traditional Chinese	亞伯拉罕·林肯	第16任美國總統	林肯

[More languages](#)

- [Main page](#)
 - [Community portal](#)
 - [Project chat](#)
 - [Create a new item](#)
 - [Item by title](#)
 - [Recent changes](#)
 - [Random item](#)
 - [Query Service](#)
 - [Nearby](#)
 - [Help](#)
 - [Donate](#)
-
- [Print/export](#)
 - [Create a book](#)
 - [Download as PDF](#)
 - [Printable version](#)

Tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)
- [Concept URI](#)
- [Cite this page](#)

Statements

family name	Lincoln	<small>0 references</small>
-----------------------------	-------------------------	-----------------------------

given name	Abraham	<small>0 references</small>
----------------------------	-------------------------	-----------------------------

manner of death	homicide	<small>0 references</small>
---------------------------------	--------------------------	-----------------------------

WebQuestions

To collect this dataset, we used the Google Suggest API to obtain questions that begin with a whword and contain exactly one entity. We started with the question Where was Barack Obama born? and performed a breadth-first search over questions (nodes), using the Google Suggest API supplying the edges of the graph. Specifically, we queried the question excluding the entity, the phrase before the entity, or the phrase after it; each query generates 5 candidate questions, which are added to the queue. We iterated until 1M questions were visited; a random 100K were submitted to Amazon Mechanical Turk.

WebQuestions

The AMT task requested that workers answer the question using only the Freebase page of the questions entity, or otherwise mark it as unanswerable by Freebase. The answer was restricted to be one of the possible entities, values, or list of entities on the page. As this list was long, we allowed the user to filter the list by typing. We paid the workers \$0.03 per question. Out of 100K questions, 6,642 were annotated identically by at least two AMT workers.

Dataset	# examples	# word types
GeoQuery	880	279
ATIS	5,418	936
FREE917	917	2,036
WEBQUESTIONS	5,810	4,525

Table 3: Statistics on various semantic parsing datasets. Our new dataset, WEBQUESTIONS, is much larger than FREE917 and much more lexically diverse than ATIS.

Contents

Lambda Calculus

Data and Problems

Semantic Parsing

Jeopardy!

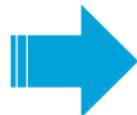
Language to Meaning



Example Task

Database Query

What states
border Texas?



Oklahoma
New Mexico
Arkansas
Louisiana

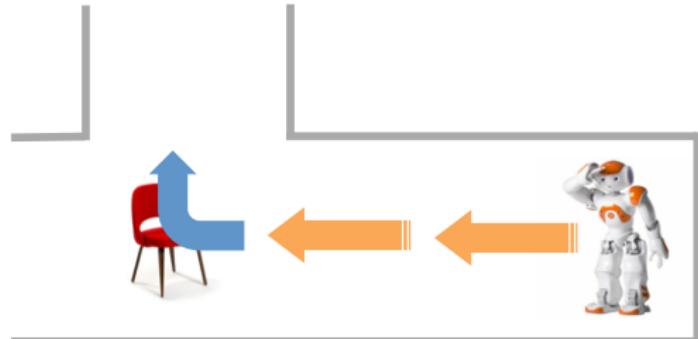
Language to Meaning



Example Task

Instructing a Robot

at the chair,
turn right



Language to Meaning



at the chair, move forward three steps past the sofa

$$\lambda a. \text{pre}(a, \iota x. \text{chair}(x)) \wedge \text{move}(a) \wedge \text{len}(a, 3) \wedge \\ \text{dir}(a, \text{forward}) \wedge \text{past}(a, \iota y. \text{sofa}(y))$$

Language to Meaning



at the chair, move forward three steps past the sofa

$$\lambda a. \text{pre}(a, \iota x. \text{chair}(x)) \wedge \text{move}(a) \wedge \text{len}(a, 3) \wedge \\ \text{dir}(a, \text{forward}) \wedge \text{past}(a, \iota y. \text{sofa}(y))$$

Language to Meaning

at the chair, move forward three steps past the sofa

$$\lambda a. \text{pre}(a, \iota x. \text{chair}(x)) \wedge \text{move}(a) \wedge \text{len}(a, 3) \wedge \\ \text{dir}(a, \text{forward}) \wedge \text{past}(a, \iota y. \text{sofa}(y))$$


Learn

$f : \text{sentence} \rightarrow \text{logical form}$

Language to Meaning

at the chair, move forward three steps past the sofa



$f : \text{sentence} \rightarrow \text{logical form}$

Supervised Data

$$\frac{\begin{array}{c} \text{show} \quad \text{me} \\ \hline S/N \\ \lambda f.f \end{array} \quad \frac{\begin{array}{c} \text{flights} \\ \hline N \\ \lambda x.\text{flight}(x) \end{array} \quad \frac{\begin{array}{c} \text{to} \\ \hline PP/NP \\ \lambda y.\lambda x.\text{to}(x,y) \end{array} \quad \frac{\begin{array}{c} \text{Boston} \\ \hline NP \\ BOSTON \end{array}}{\overrightarrow{PP} \\ \lambda x.\text{to}(x,BOSTON)} \\ \hline N \setminus N \\ \lambda f.\lambda x.f(x) \wedge \text{to}(x,BOSTON) \end{array} < \\ \hline \frac{\begin{array}{c} N \\ \lambda x.\text{flight}(x) \wedge \text{to}(x,BOSTON) \end{array} \longrightarrow \\ \hline S \\ \lambda x.\text{flight}(x) \wedge \text{to}(x,BOSTON) \end{array}$$

Supervised Data

show	me	flights	to	Boston
S/N		N	PP/NP	NP
$\lambda f.f$		$\lambda x.flight(x)$	$\lambda y.\lambda x.to(x, y)$	$BOSTON$
				\longrightarrow
			$\lambda x.to(x, BOSTON)$	
			$\lambda f.\lambda x.f(x) \wedge to(x, BOSTON)$	$N \setminus N$
			$\lambda x.flight(x) \wedge to(x, BOSTON)$	\longleftarrow
				N
				\longrightarrow
			$\lambda x.flight(x) \wedge to(x, BOSTON)$	S

Latent

Supervised Data

Supervised learning is done from pairs
of sentences and logical forms

Show me flights to Boston

$$\lambda x. flight(x) \wedge to(x, BOSTON)$$

I need a flight from baltimore to seattle

$$\lambda x. flight(x) \wedge from(x, BALTIMORE) \wedge to(x, SEATTLE)$$

what ground transportation is available in san francisco

$$\lambda x. ground_transport(x) \wedge to_city(x, SF)$$

Weak Supervision

- Logical form is latent
- “Labeling” requires less expertise
- Labels don’t uniquely determine correct logical forms
- Learning requires executing logical forms within a system and evaluating the result

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

New Mexico

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

New Mexico

$\text{argmax}(\lambda x. \text{state}(x)$
 $\wedge \text{border}(x, TX), \lambda y. \text{size}(y))$

$\text{argmax}(\lambda x. \text{river}(x)$
 $\wedge \text{in}(x, TX), \lambda y. \text{size}(y))$

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

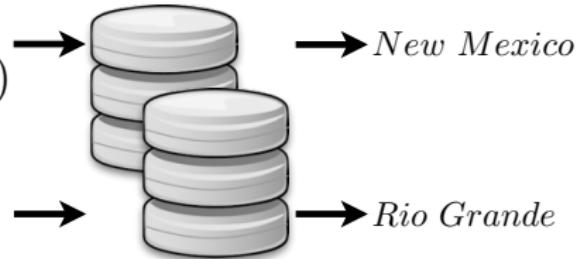
New Mexico

$\text{argmax}(\lambda x. \text{state}(x)$

$\wedge \text{border}(x, TX), \lambda y. \text{size}(y))$

$\text{argmax}(\lambda x. \text{river}(x)$

$\wedge \text{in}(x, TX), \lambda y. \text{size}(y))$



Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

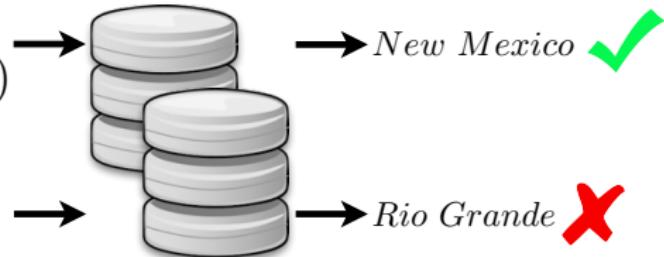
New Mexico

$\text{argmax}(\lambda x. \text{state}(x)$

$\wedge \text{border}(x, TX), \lambda y. \text{size}(y))$

$\text{argmax}(\lambda x. \text{river}(x)$

$\wedge \text{in}(x, TX), \lambda y. \text{size}(y))$



Weak Supervision

- Logical form is latent
- “Labeling” requires less expertise
- Labels don’t uniquely determine correct logical forms
- Learning requires executing logical forms within a system and evaluating the result

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

New Mexico

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

New Mexico

$\text{argmax}(\lambda x. \text{state}(x)$
 $\wedge \text{border}(x, TX), \lambda y. \text{size}(y))$

$\text{argmax}(\lambda x. \text{river}(x)$
 $\wedge \text{in}(x, TX), \lambda y. \text{size}(y))$

Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

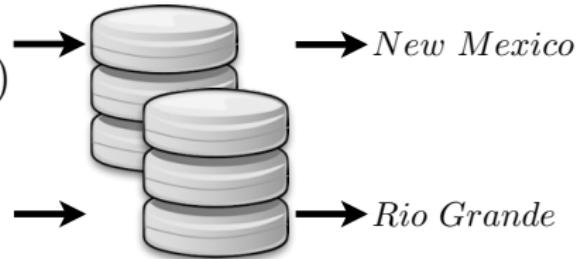
New Mexico

$\text{argmax}(\lambda x. \text{state}(x)$

$\wedge \text{border}(x, TX), \lambda y. \text{size}(y))$

$\text{argmax}(\lambda x. \text{river}(x)$

$\wedge \text{in}(x, TX), \lambda y. \text{size}(y))$



Weak Supervision

Learning from Query Answers

What is the largest state that borders Texas?

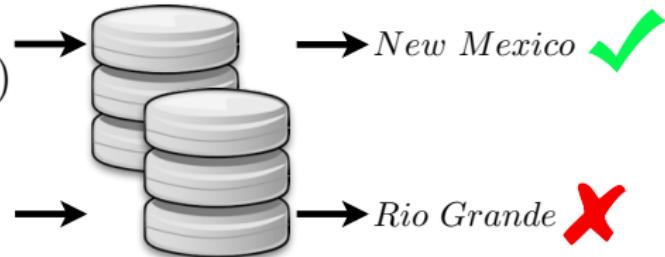
New Mexico

$\text{argmax}(\lambda x. \text{state}(x)$

$\wedge \text{border}(x, TX), \lambda y. \text{size}(y))$

$\text{argmax}(\lambda x. \text{river}(x)$

$\wedge \text{in}(x, TX), \lambda y. \text{size}(y))$



Weak Supervision

Learning from Demonstrations

at the chair, move forward three steps past the sofa



Weak Supervision

Learning from Demonstrations

at the chair, move forward three steps past the sofa



Some examples from other domains:

- Sentences and labeled game states [Goldwasser and Roth 2011]
- Sentences and sets of physical objects [Matuszek et al. 2012]

Weak Supervision

Learning from Conversation Logs

SYSTEM how can I help you ? (OPEN_TASK)

USER i ' d like to fly to new york

SYSTEM flying to new york . (CONFIRM: *from(f1, ATL)*) leaving what city ?
(ASK: $\lambda x.\text{from}(f1, x)$)

USER from boston on june seven with american airlines

SYSTEM flying to new york . (CONFIRM: *to(f1, NYC)*) what date would you
like to depart boston ? (ASK: $\lambda x.\text{date}(f1, x) \wedge \text{to}(f1, BOS)$)

USER june seventh

[CONVERSATION CONTINUES]

Parsing

Learning

Modeling

- Structured perceptron
- A unified learning algorithm
- Supervised learning
- Weak supervision

Structured Perceptron

- Simple additive updates
 - Only requires efficient decoding (argmax)
 - Closely related to MaxEnt and other feature rich models
 - Provably finds linear separator in finite updates, if one exists
- Challenge: learning with hidden variables

Structured Perceptron

Data: $\{(x_i, y_i) : i = 1 \dots n\}$

For $t = 1 \dots T$:

[iterate epochs]

For $i = 1 \dots n$:

[iterate examples]

$$y^* \leftarrow \arg \max_y \langle \theta, \Phi(x_i, y) \rangle$$

[predict]

$$\text{If } y^* \neq y_i:$$

[check]

$$\theta \leftarrow \theta + \Phi(x_i, y_i) - \Phi(x_i, y^*)$$

[update]

One Derivation of the Perceptron

Log-linear model: $p(y|x) = \frac{e^{w \cdot f(x,y)}}{\sum_{y'} e^{w \cdot f(x,y')}}$

Step 1: Differentiate, to maximize data log-likelihood

$$update = \sum_i f(x_i, y_i) - E_{p(y|x_i)} f(x_i, y)$$

Step 2: Use online, stochastic gradient updates, for example i :

$$update_i = f(x_i, y_i) - E_{p(y|x_i)} f(x_i, y)$$

Step 3: Replace expectations with maxes (Viterbi approx.)

$$update_i = f(x_i, y_i) - f(x_i, y^*) \text{ where } y^* = \arg \max_y w \cdot f(x_i, y)$$

The Perceptron with Hidden Variables

Log-linear

model: $p(y|x) = \sum_h p(y, h|x)$ $p(y, h|x) = \frac{e^{w \cdot f(x, h, y)}}{\sum_{y', h'} e^{w \cdot f(x, h', y')}}$

Step 1: Differentiate marginal, to maximize data log-likelihood

$$update = \sum_i E_{p(h|y_i, x_i)}[f(x_i, h, y_i)] - E_{p(y, h|x_i)}[f(x_i, h, y)]$$

Step 2: Use online, stochastic gradient updates, for example i :

$$update_i = E_{p(y_i, h|x_i)}[f(x_i, h, y_i)] - E_{p(y, h|x_i)}[f(x_i, h, y)]$$

Step 3: Replace expectations with maxes (Viterbi approx.)

$$update_i = f(x_i, h', y_i) - f(x_i, h^*, y^*) \text{ where}$$

$$y^*, h^* = \arg \max_{y, h} w \cdot f(x_i, h, y) \quad \text{and} \quad h' = \arg \max_h w \cdot f(x_i, h, y_i)$$

Hidden Variable Perceptron

Data: $\{(x_i, y_i) : i = 1 \dots n\}$

For $t = 1 \dots T$: [iterate epochs]

For $i = 1 \dots n$: [iterate examples]

$$y^*, h^* \leftarrow \arg \max_{y, h} \langle \theta, \Phi(x_i, h, y) \rangle \quad \text{[predict]}$$

If $y^* \neq y_i$: [check]

$$h' \leftarrow \arg \max_h \langle \theta, \Phi(x_i, h, y_i) \rangle \quad \text{[predict hidden]}$$

$$\theta \leftarrow \theta + \Phi(x_i, h', y_i) - \Phi(x_i, h^*, y^*) \quad \text{[update]}$$

Contents

Lambda Calculus

Data and Problems

Semantic Parsing

Jeopardy!



DeepQA

- ▶ (not actually deep learning)

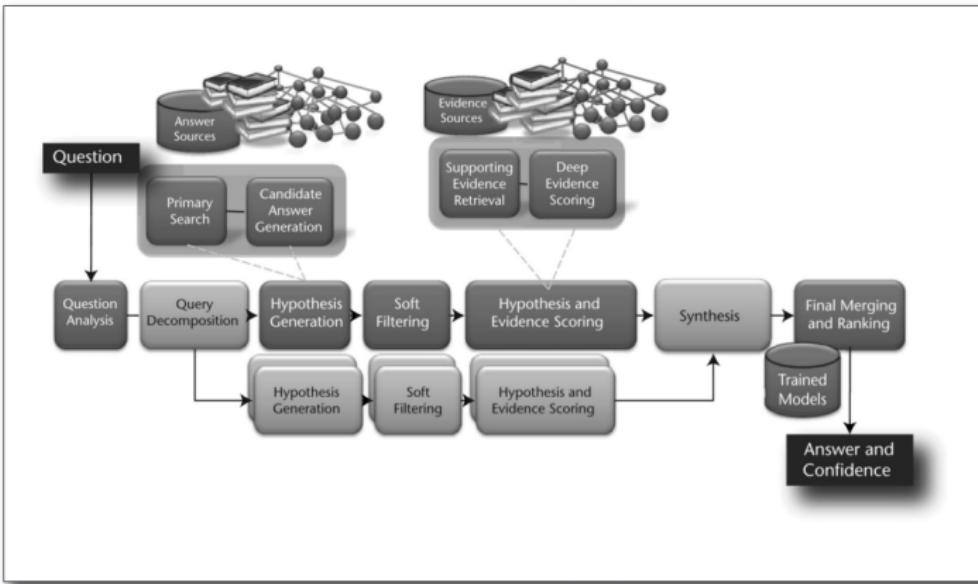


Figure 6. DeepQA High-Level Architecture.

Candidate Generation

This motivates an approach that merges answer scores before ranking and confidence estimation. Using an ensemble of matching, normalization, and coreference resolution algorithms, Watson identifies equivalent and related hypotheses (for example, Abraham Lincoln and Honest Abe) and then enables custom merging per feature to combine scores.

Given the kinds of questions and broad domain of the Jeopardy Challenge, the sources for Watson include a wide range of encyclopedias, dictionaries, thesauri, newswire articles, literary works, and so on.

A variety of search techniques are used, including the use of multiple text search engines with different underlying approaches (for example, Indri and Lucene), document search as well as passage search, knowledge base search using SPARQL on triple stores, the generation of multiple search queries for a single question, and backfilling hit lists to satisfy key constraints identified in the question.

Hypothesis Ranking

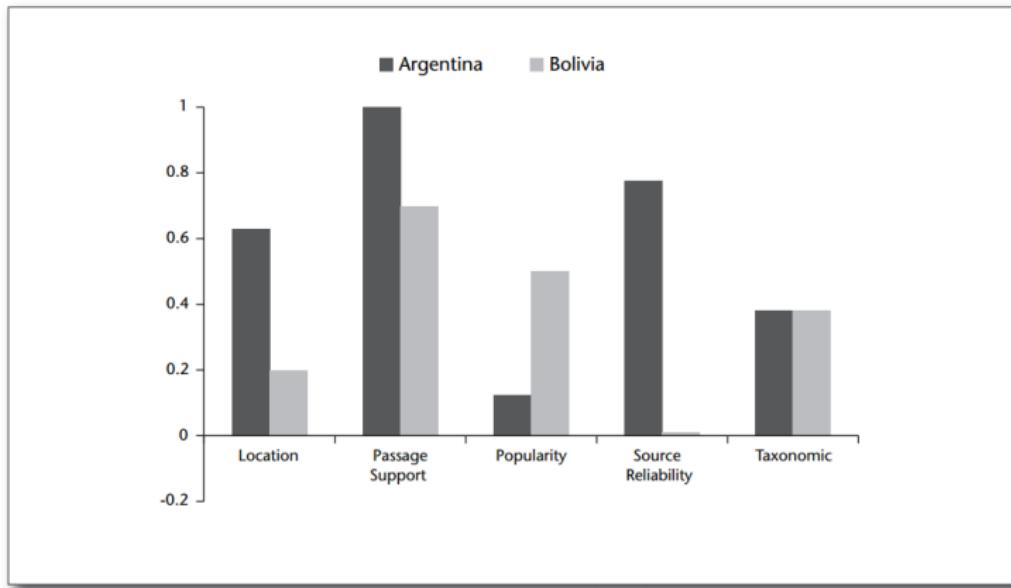


Figure 8. Evidence Profiles for Two Candidate Answers.
Dimensions are on the x-axis and relative strength is on the y-axis.

Other Aspects

- ▶ Parsing algorithm
- ▶ Inducing rules and parameters
- ▶ Relationship between syntax and semantics

CCG Categories

$$ADJ : \lambda x. fun(x)$$

- Basic building block
- Capture syntactic and semantic information jointly

CCG Categories

Syntax

ADJ

$\lambda x.fun(x)$

Semantics

- Basic building block
- Capture syntactic and semantic information jointly

CCG Categories

Syntax

$ADJ : \lambda x. fun(x)$

$(S \setminus NP) / ADJ : \lambda f. \lambda x. f(x)$

$NP : CCG$

- Primitive symbols: N, S, NP, ADJ and PP
- Syntactic combination operator ($/$, \backslash)
- Slashes specify argument order and direction

CCG Categories

$ADJ : \lambda x. fun(x)$ Semantics

$(S \setminus NP) / ADJ : \lambda f. \lambda x. f(x)$

$NP : CCG$

- λ -calculus expression
- Syntactic type maps to semantic type

CCG Lexical Entries

$\text{fun} \vdash ADJ : \lambda x. fun(x)$

- Pair words and phrases with meaning
- Meaning captured by a CCG category

CCG Lexical Entries

fun

Natural
Language

←

$ADJ : \lambda x. fun(x)$

CCG Category

- Pair words and phrases with meaning
- Meaning captured by a CCG category