Question Answering/Semantic Parsing

CS 287

(Slides from Yoav Artzi, Cornell NY)
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 linguistic 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!
Lambda Calculus

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

\[ \lambda a.\text{move}(a) \land \text{dir}(a, LEFT) \land \text{to}(a, \forall y.\text{chair}(y)) \land \text{pass}(a, \forall y.\text{sofa}(y) \land \text{intersect}(\forall z.\text{intersection}(z), y)) \]

[Church 1932]
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

\( x, y, z, \ldots \)
Lambda Calculus

Literals

- Represent function application

\[
\text{city}(AUSTIN)
\]

\[
\text{located_in}(AUSTIN, TEXAS)
\]
Lambda Calculus

Literals

- Represent function application

\[
\text{city}(\text{AUSTIN})
\]

\[
\text{located_in}(\text{AUSTIN, TEXAS})
\]

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

\[ \lambda x.\text{city}(x) \]

\[ \lambda x.\lambda y.\text{located\_in}(x, y) \]
Higher order constants
No need for any special mechanics
Can represent all of first order logic

\( \forall (\lambda x. \text{big}(x) \land \text{apple}(x)) \)
\( \neg (\exists (\lambda x. \text{lovely}(x))) \)
\( \impliedby (\lambda x. \text{beautiful}(x) \land \text{grammar}(x)) \)
\[ \land (A, \land (B, C)) \iff A \land B \land C \]
\[ \lor (A, \lor (B, C)) \iff A \lor B \lor C \]
\[ \lnot (A) \iff \lnot A \]
\[ Q(\lambda x. f(x)) \iff Qx. f(x) \]
for \( Q \in \{ \forall, \exists \} \)
\[ \lambda x. \text{flight}(x) \land \text{to}(x, \text{move}) \]
\[ \lambda x. \text{flight}(x) \land \text{to}(x, \text{NYC}) \]
\[ \lambda x. \text{NYC}(x) \land x(\text{to}, \text{move}) \]
\[ \lambda x. \text{flight}(x) \land \text{to}(x, \text{move}) \]

\[ \lambda x. \text{flight}(x) \land \text{to}(x, \text{NYC}) \]

\[ \lambda x. \text{NYC}(x) \land x(\text{to, move}) \]
Simply Typed Lambda Calculus

- Like lambda calculus
- But, typed

\[ \lambda x. \text{flight}(x) \land \text{to}(x, \text{move}) \]

\[ \lambda x. \text{flight}(x) \land \text{to}(x, \text{NYC}) \]

\[ \lambda x. \text{NYC}(x) \land x(\text{to, move}) \]
Lambda Calculus

Typing

• Simple types

• Complex types

\(< e, t >\)

\(<<< e, t >, e >>\)
Lambda Calculus

Typing

- Simple types
- Complex types

\(<e, t>\)

Type constructor

\(<<e, t>, e>\)

Domain

Range
Lambda Calculus

Typing

- Simple types
- Complex types

Hierarchical typing system

Type constructor

\[ \langle e, t \rangle \]

\[ \langle \langle e, t \rangle, e \rangle \]

- Hierarchical typing system
Lambda Calculus

Typing

- Simple types
- Complex types
- Hierarchical typing system

Type constructor

<e, t>

[Green] Domain

<< e, t >, e >

[Orange] Range
Simply Typed Lambda Calculus

$$\lambda a.\text{move}(a) \land \text{dir}(a, \text{LEFT}) \land \text{to}(a, \nu y.\text{chair}(y)) \land \text{pass}(a, \forall y.\text{sofa}(y) \land \text{intersect}(\forall z.\text{intersection}(z), y))$$

Type information usually omitted
Capturing Meaning with Lambda Calculus

**State**

<table>
<thead>
<tr>
<th>Abbr.</th>
<th>Capital</th>
<th>Pop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Montgomery</td>
<td>3.9</td>
</tr>
<tr>
<td>AK</td>
<td>Juneau</td>
<td>0.4</td>
</tr>
<tr>
<td>AZ</td>
<td>Phoenix</td>
<td>2.7</td>
</tr>
</tbody>
</table>

**Border**

<table>
<thead>
<tr>
<th>State1</th>
<th>State2</th>
</tr>
</thead>
<tbody>
<tr>
<td>WA</td>
<td>OR</td>
</tr>
<tr>
<td>WA</td>
<td>ID</td>
</tr>
<tr>
<td>CA</td>
<td>OR</td>
</tr>
<tr>
<td>CA</td>
<td>NV</td>
</tr>
</tbody>
</table>

**Mountains**

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bianca</td>
<td>CO</td>
</tr>
<tr>
<td>Antero</td>
<td>CO</td>
</tr>
<tr>
<td>Rainier</td>
<td>WA</td>
</tr>
<tr>
<td>Shasta</td>
<td>CA</td>
</tr>
<tr>
<td>Wrangel</td>
<td>AK</td>
</tr>
<tr>
<td>Sill</td>
<td>CA</td>
</tr>
<tr>
<td>Bona</td>
<td>AK</td>
</tr>
<tr>
<td>Elbert</td>
<td>CO</td>
</tr>
</tbody>
</table>

Show me mountains in states bordering Texas

[Zettlemoyer and Collins 2005]
show me flights from New York to LA departing on Thursday

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

<table>
<thead>
<tr>
<th>Depart</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thu, Jan 30</td>
<td>Mon, Feb 3</td>
</tr>
</tbody>
</table>

**Nonstop only**

<table>
<thead>
<tr>
<th>Airline</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>United</td>
<td>from $1,034</td>
</tr>
<tr>
<td>Alaska</td>
<td>from $1,034</td>
</tr>
<tr>
<td>American</td>
<td>from $1,034</td>
</tr>
<tr>
<td>JetBlue</td>
<td>from $1,034</td>
</tr>
<tr>
<td>Virgin America</td>
<td>from $1,034</td>
</tr>
<tr>
<td>Delta</td>
<td>from $1,054</td>
</tr>
</tbody>
</table>

**All flights** Nonstop and connecting

<table>
<thead>
<tr>
<th>Airline</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delta</td>
<td>from $488</td>
</tr>
<tr>
<td>AirTran</td>
<td>from $682</td>
</tr>
<tr>
<td>Other airlines</td>
<td>from $803</td>
</tr>
</tbody>
</table>
Parsing as Structure Prediction

\[
\begin{align*}
\text{show me flights to Boston} & \\
S/N & N & PP/NP & NP \\
\lambda f.f & \lambda x.\text{flight}(x) & \lambda y.\lambda x.\text{to}(x, y) & BOSTON \\
\text{S/N N P P/NP NP} & \text{S/N N P P/NP NP} & \text{S/N N P P/NP NP} & \text{S/N N P P/NP NP} \\
\end{align*}
\]
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
\[ \lambda x. state(x) \land borders(x, texas) \]

b) What is the largest state
\[ \text{arg max}(\lambda x. state(x), \lambda x. size(x)) \]

c) What states border the state that borders the most states
\[ \lambda x. state(x) \land borders(x, \text{arg max}(\lambda y. state(y), \lambda y. count(\lambda z. state(z) \land borders(y, z)))) \]

Figure 1: Examples of sentences with their logical forms.
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
▶ Constitutional republic
▶ Eclipse, Valentine’s Day, The Twilight Saga: Breaking Dawn - Part 1, New Moon
Here's an example:

We know that William Shakespeare wrote the play Hamlet. We describe this in Freebase as such:

<table>
<thead>
<tr>
<th>William Shakespeare</th>
<th>is a type Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>has a property Works Written</td>
<td></td>
</tr>
<tr>
<td>with a value Hamlet</td>
<td></td>
</tr>
</tbody>
</table>
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. Raised 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 most of Illinois. He returned to politics in 1854, and was a leader in building up the new Republican Party, which was held 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.
## Abraham Lincoln (Q91)

16th President of the United States
Abe Lincoln | Lincoln | Honest Abe

### In more languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Label</th>
<th>Description</th>
<th>Also known as</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Abraham Lincoln</td>
<td>16th President of the United States</td>
<td>Abe Lincoln, Lincoln, Honest Abe</td>
</tr>
<tr>
<td>German</td>
<td>Abraham Lincoln</td>
<td>US-amerikanischer Präsident</td>
<td></td>
</tr>
<tr>
<td>Spanish</td>
<td>Abraham Lincoln</td>
<td>decimosexto presidente de los Estados Unidos</td>
<td></td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>亚伯拉罕·林肯</td>
<td>第16任美国总统</td>
<td>林肯</td>
</tr>
</tbody>
</table>

### More languages

### Statements

<table>
<thead>
<tr>
<th>property</th>
<th>value</th>
<th>references</th>
</tr>
</thead>
<tbody>
<tr>
<td>family name</td>
<td>Lincoln</td>
<td>0</td>
</tr>
<tr>
<td>given name</td>
<td>Abraham</td>
<td>0</td>
</tr>
<tr>
<td>manner of death</td>
<td>homicide</td>
<td>0</td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th>Dataset</th>
<th># examples</th>
<th># word types</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoQuery</td>
<td>880</td>
<td>279</td>
</tr>
<tr>
<td>ATIS</td>
<td>5,418</td>
<td>936</td>
</tr>
<tr>
<td>FREE917</td>
<td>917</td>
<td>2,036</td>
</tr>
<tr>
<td>WebQUESTIONS</td>
<td>5,810</td>
<td>4,525</td>
</tr>
</tbody>
</table>

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

More informative

Semantic Parsing

Recover complete meaning representation

Example Task

Database Query

What states border Texas?

Oklahoma!

New Mexico!

Arkansas!

Louisiana
Language to Meaning

More informative
Semantic Parsing
Recover complete meaning representation
More informative

Example Task
Instructing a Robot
at the chair, turn right
at the chair, move forward three steps past the sofa

\[ \lambda a. \text{pre}(a, \forall x. \text{chair}(x)) \land \text{move}(a) \land \text{len}(a, 3) \land \text{dir}(a, \text{forward}) \land \text{past}(a, \forall y. \text{sofa}(y)) \]
Language to Meaning

Semantic Parsing

Recover complete meaning representation

More informative

at the chair, move forward three steps past the sofa

\( \lambda a. pre(a, \forall x. chair(x)) \land move(a) \land len(a, 3) \land dir(a, forward) \land past(a, \forall y. sofa(y)) \)
Language to Meaning

at the chair, move forward three steps past the sofa

\[
\lambda a. \text{pre}(a, \forall x. \text{chair}(x)) \land \text{move}(a) \land \text{len}(a, 3) \land \text{dir}(a, \text{forward}) \land \text{past}(a, \forall y. \text{sofa}(y))
\]

Learn

\[f : \text{sentence} \rightarrow \text{logical form}\]
Language to Meaning

at the chair, move forward three steps past the sofa

Learn

\[ f : \text{sentence} \rightarrow \text{logical form} \]
Supervised Data

\[
\begin{array}{cccc}
\text{show} & \text{me} & \text{flights} & \text{to} \\
S/N & \lambda f.f & \lambda x.\text{flight}(x) & PP/NP \\
\lambda y.\lambda x.\text{to}(x,y) & NP & BOSTON \\
\end{array}
\]

\[
\begin{align*}
&\lambda x.\text{to}(x, BOSTON) \\
&\lambda f.\lambda x.f(x) \wedge \text{to}(x, BOSTON) \\
&\lambda x.\text{flight}(x) \wedge \text{to}(x, BOSTON) \\
\end{align*}
\]

\[
\begin{align*}
&P P \\
&N \backslash N \\
&N \\
&S \\
\end{align*}
\]
### Supervised Data

<table>
<thead>
<tr>
<th>show</th>
<th>me</th>
<th>flights</th>
<th>to</th>
<th>Boston</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S/N$</td>
<td>$\lambda f. f$</td>
<td>$N$</td>
<td>$\lambda x. flight(x)$</td>
<td>$\lambda y. \lambda x. to(x, y)$</td>
</tr>
<tr>
<td>$\lambda f. f$</td>
<td>$\lambda x. flight(x)$</td>
<td>$\lambda x. to(x, BOSTON)$</td>
<td>$\lambda x. flight(x) \land to(x, BOSTON)$</td>
<td>$\lambda x. flight(x) \land to(x, BOSTON)$</td>
</tr>
</tbody>
</table>

### Latent

$\lambda x. flight(x) \land to(x, BOSTON)$
Supervised Data

Supervised learning is done from pairs of sentences and logical forms

Show me flights to Boston
\[ \lambda x. \text{flight}(x) \land \text{to}(x, BOSTON) \]

I need a flight from baltimore to seattle
\[ \lambda x. \text{flight}(x) \land \text{from}(x, BALTIMORE) \land \text{to}(x, SEATTLE) \]

what ground transportation is available in san francisco
\[ \lambda x. \text{ground}_\text{transport}(x) \land \text{to}_\text{city}(x, SF) \]

[Zettlemoyer and Collins 2005; 2007]
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*

[Clarke et al. 2010; Liang et al. 2011]
Weak Supervision
Learning from Query Answers

What is the largest state that borders Texas?

New Mexico

\[
\text{argmax}(\lambda x. \text{state}(x)) \\
\land \text{border}(x, TX), \lambda y. \text{size}(y))
\]

\[
\text{argmax}(\lambda x. \text{river}(x)) \\
\land \text{in}(x, TX), \lambda y. \text{size}(y))
\]

[Clarke et al. 2010; Liang et al. 2011]
Weak Supervision
Learning from Query Answers

What is the largest state that borders Texas?

*New Mexico*

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

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

[Clarke et al. 2010; Liang et al. 2011]
Weak Supervision
Learning from Query Answers

What is the largest state that borders Texas?

New Mexico

argmax(λx.state(x)
∧ border(x, TX), λy.size(y))

argmax(λx.river(x)
∧ in(x, TX), λy.size(y))

New Mexico

Rio Grande

[Clarke et al. 2010; Liang et al. 2011]
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

[Clarke et al. 2010; Liang et al. 2011]
Weak Supervision
Learning from Query Answers

What is the largest state that borders Texas?

*New Mexico*

\[
\text{argmax} (\lambda x. \text{state}(x) \land \text{border}(x, TX), \lambda y. \text{size}(y))
\]

\[
\text{argmax} (\lambda x. \text{river}(x) \land \text{in}(x, TX), \lambda y. \text{size}(y))
\]

[Clarke et al. 2010; Liang et al. 2011]
What is the largest state that borders Texas?

New Mexico

\[
\text{argmax}(\lambda x. \text{state}(x) \land \text{border}(x, TX), \lambda y. \text{size}(y))
\]

\[
\text{argmax}(\lambda x. \text{river}(x) \land \text{in}(x, TX), \lambda y. \text{size}(y))
\]

[Clarke et al. 2010; Liang et al. 2011]
Weak Supervision
Learning from Query Answers

What is the largest state that borders Texas?

*New Mexico*

\[
\text{argmax}(\lambda x. \text{state}(x) \\
\land \text{border}(x, TX), \lambda y. \text{size}(y))
\]

\[
\text{argmax}(\lambda x. \text{river}(x) \\
\land \text{in}(x, TX), \lambda y. \text{size}(y))
\]

[Clarke et al. 2010; Liang et al. 2011]
Weak Supervision
Learning from Demonstrations

at the chair, move forward three steps past the sofa

[Chen and Mooney 2011; Kim and Mooney 2012; Artzi and Zettlemoyer 2013b]
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(fl, ATL)) leaving what city? *(ASK: \(\lambda x. \text{from}(fl,x)\))

**User**  from boston on june seven with american airlines

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

**User**  june seventh

[CONVERSATION CONTINUES]

[Artzi and Zettlemoyer 2011]
• 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: \( \{(x_i, y_i) : i = 1 \ldots n\} \)

For \( t = 1 \ldots T \):

For \( i = 1 \ldots n \):

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

If \( y^* \neq y_i \):

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

[iterate epochs]
[iterate examples]
[predict]
[check]
[update]

[Collins 2002]
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

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

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

\[
\text{update}_i = f(x_i, y_i) - E_{p(y|x_i)} f(x_i, y) 
\]

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

\[
\text{update}_i = f(x_i, y_i) - f(x_i, y^*) \text{ where } y^* = \arg \max_y w \cdot f(x_i, y) 
\]
The Perceptron with Hidden Variables

Log-linear model:

\[ p(y|x) = \sum_h p(y, h|x) \quad p(y, h|x) = \frac{e^{w \cdot f(x, h, y)}}{\sum_{y', h'} e^{w \cdot f(x, h', y')}} \]

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

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

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

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

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

update \( i \) = \[ f(x_i, h^*, y^*) \]

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: \( \{(x_i, y_i) : i = 1 \ldots n\} \)

For \( t = 1 \ldots T \):

For \( i = 1 \ldots n \):

\[
\begin{align*}
  y^*, h^* & \leftarrow \arg\max_{y,h} \langle \theta, \Phi(x_i, h, y) \rangle \\
  \text{If } y^* & \neq y_i:
    h' & \leftarrow \arg\max_h \langle \theta, \Phi(x_i, h, y_i) \rangle \\
    \theta & \leftarrow \theta + \Phi(x_i, h', y_i) - \Phi(x_i, h^*, y^*)
\end{align*}
\]

[iterate epochs]

[iterate examples]

[predict]

[check]

[predict hidden]

[update]

[Liang et al. 2006; Zettlemoyer and Collins 2007]
Contents

Lambda Calculus

Data and Problems

Semantic Parsing

Jeopardy!
DeepQA

- Information Retrieval based factoid QA
- Combines search, rule-based extraction, and ranking
- Kitchen-sink type of approach
- (Nothing to do with deep learning)
DeepQA (Ferrucci, 2010)

- Massive parallelism: Exploit massive parallelism in the consideration of multiple interpretations and hypotheses.
- Many experts: Facilitate the integration, application, and contextual evaluation of a wide range of loosely coupled probabilistic question and content analytics.
- Pervasive confidence estimation: No component commits to an answer; all components produce features and associated confidences, scoring different question and content interpretations. An underlying confidence-processing substrate learns how to stack and combine the scores.
- Integrate shallow and deep knowledge: Balance the use of strict semantics and shallow semantics, leveraging many loosely formed ontologies.
Figure 6. DeepQA High-Level Architecture.
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: $ADJ : \lambda x. fun (x)$

• Basic building block
• Capture syntactic and semantic information jointly
CCG Categories

Syntax

\[
\text{ADJ} : \lambda x. \text{fun}(x)
\]

\[
(S\backslash NP)/\text{ADJ} : \lambda f. \lambda x. f(x)
\]

\[
\text{NP} : \text{CCG}
\]

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

\[ ADJ : \lambda x. fun(x) \]

\[ (S \backslash NP)/ADJ : \lambda f. \lambda x. f(x) \]

\[ NP : CCG \]

- \( \lambda \)-calculus expression
- Syntactic type maps to semantic type
CCG Lexical Entries

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

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

Meaning captured by a CCG category