Outline

- Learning
  - Overview
  - Details
Outline

• Learning
  – Overview
  – Details
Supervision in syntactic parsing

Input:

```
S
NP
NP
ESSLLI 2016
NP
the known summer school
VP
V
is
VP
V
located
PP
in Bolzano
```

Output:

```
They play football
S
NP
They
VP
V
play
NP
football
```
### Supervision in semantic parsing

**Input:**

<table>
<thead>
<tr>
<th><strong>Heavy supervision</strong></th>
<th><strong>Light supervision</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><em>How tall is Lebron James?</em></td>
<td><em>How tall is Lebron James?</em></td>
</tr>
<tr>
<td><em>HeightOf.LebronJames</em></td>
<td><em>203cm</em></td>
</tr>
<tr>
<td><em>What is Steph Curry’s daughter called?</em></td>
<td><em>What is Steph Curry’s daughter called?</em></td>
</tr>
<tr>
<td><em>ChildrenOf.StephCurry □ Gender.Female</em></td>
<td><em>Riley Curry</em></td>
</tr>
<tr>
<td><em>Youngest player of the Cavaliers</em></td>
<td><em>Youngest player of the Cavaliers</em></td>
</tr>
<tr>
<td><em>arg min(PLAYEROF.Cavaliers, BIRTHDATEOF)</em></td>
<td><em>Kyrie Irving</em></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

[Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005; Clarke et al. 2010; Liang et al., 2011]
Supervision in semantic parsing

Input:

**Heavy supervision**

How tall is Lebron James?
HeightOf.LebronJames

What is Steph Curry’s daughter called?
ChildrenOf.StephCurry \( \cap \) Gender.Female

Youngest player of the Cavaliers
\( \text{arg min}(\text{PlayerOf.Cavaliers}, \text{BirthDateOf}) \)

... 

**Light supervision**

How tall is Lebron James?
203cm

What is Steph Curry’s daughter called?
Riley Curry

Youngest player of the Cavaliers
Kyrie Irving

... 

Output:

Clay Thompson’s weight

ClayThompson’s weight → ClayThompson’s Weight → Weight.ClayThompson → 205 lbs
Learning in a nutshell

*utterance*

0. Define model for derivations
Learning in a nutshell

0. Define model for derivations
1. Generate candidate derivations (later)
0. Define model for derivations
1. Generate candidate derivations (later)
2. Label as correct and incorrect
0. Define model for derivations
1. Generate candidate derivations (later)
2. Label as correct and incorrect
3. Update model to favor correct trees
Training intuition

Where did Mozart tupress?

Vienna
Training intuition

*Where did Mozart tupress?*

PlaceOfBirth.WolfgangMozart
PlaceOfDeath.WolfgangMozart
PlaceOfMarriage.WolfgangMozart

**Vienna**
Training intuition

Where did Mozart tupress?

PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna
Training intuition

*Where did Mozart tupress?*

- PlaceOfBirth.WolfgangMozart ⇒ Salzburg
- PlaceOfDeath.WolfgangMozart ⇒ Vienna
- PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna
Training intuition

Where did Mozart tupress?
PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna

Where did Hogarth tupress?
**Training intuition**

*Where did Mozart tupress?*

- **PlaceOfBirth.WolfgangMozart** ⇒ Salzburg
- **PlaceOfDeath.WolfgangMozart** ⇒ Vienna
- **PlaceOfMarriage.WolfgangMozart** ⇒ Vienna

**Vienna**

*Where did Hogarth tupress?*

- **PlaceOfBirth.WilliamHogarth**
- **PlaceOfDeath.WilliamHogarth**
- **PlaceOfMarriage.WilliamHogarth** ⇒ London

**London**
Training intuition

Where did Mozart tupress?

PlaceOfBirth.WolfgangMozart $\Rightarrow$ Salzburg

PlaceOfDeath.WolfgangMozart $\Rightarrow$ Vienna

PlaceOfMarriage.WolfgangMozart $\Rightarrow$ Vienna

Vienna

Where did Hogarth tupress?

PlaceOfBirth.WilliamHogarth $\Rightarrow$ London

PlaceOfDeath.WilliamHogarth $\Rightarrow$ London

PlaceOfMarriage.WilliamHogarth $\Rightarrow$ Paddington

London
Training intuition

Where did Mozart tuples?
PlaceOfBirth.WolfgangMozart ⇒ Salzburg
PlaceOfDeath.WolfgangMozart ⇒ Vienna
PlaceOfMarriage.WolfgangMozart ⇒ Vienna

Vienna

Where did Hogarth tuples?
PlaceOfBirth.WilliamHogarth ⇒ London
PlaceOfDeath.WilliamHogarth ⇒ London
PlaceOfMarriage.WilliamHogarth ⇒ Paddington

London
Training intuition

Where did Mozart t\textsuperscript{up}ress?

\text{PlaceOfBirth}.\text{WolfgangMozart} \Rightarrow \text{Salzburg}

\text{PlaceOfDeath}.\text{WolfgangMozart} \Rightarrow \text{Vienna}

\text{PlaceOfMarriage}.\text{WolfgangMozart} \Rightarrow \text{Vienna}

\textbf{Vienna}

Where did Hogarth t\textsuperscript{up}ress?

\text{PlaceOfBirth}.\text{WilliamHogarth} \Rightarrow \text{London}

\text{PlaceOfDeath}.\text{WilliamHogarth} \Rightarrow \text{London}

\text{PlaceOfMarriage}.\text{WilliamHogarth} \Rightarrow \text{Paddington}

\textbf{London}
Outline

• Learning
  – Overview
  – Details
Constructing derivations
Many possible derivations!

\[ x = \text{people who have lived in Chicago} \]

set of candidate derivations \( \mathcal{D}(x) \)
Feature vector and parameters in $\mathbb{R}^F$:

\[ \phi(x, d) \quad \theta \quad \leftarrow \text{learned} \]

- apply join: \(1\) \(1.2\)
- apply intersect: \(1\) \(0.6\)
- apply lexicon: \(3\) \(2.1\)
- \textit{lived} maps to \textit{PlacesLived}: \(1\) \(3.1\)
- \textit{lived} maps to \textit{PlaceOfBirth}: \(0\) \(-0.4\)
- \textit{born} maps to \textit{PlaceOfBirth}: \(0\) \(2.7\)
- ... ... ...
\( x \): utterance
\( d \): derivation

Feature vector and parameters in \( \mathbb{R}^F \):

\[
\phi(x, d) \quad \theta \quad \Leftarrow \text{learned}
\]

- apply join 1 1.2
- apply intersect 1 0.6
- apply lexicon 3 2.1
- \textit{lived} maps to \textit{PlacesLived} 1 3.1
- \textit{lived} maps to \textit{PlaceOfBirth} 0 -0.4
- \textit{born} maps to \textit{PlaceOfBirth} 0 2.7
- ...

\[
\text{Score}_\theta(x, d) = \phi(x, d)^\top \theta = 1.2 \cdot 1 + 0.6 \cdot 1 + 2.1 \cdot 3 + 3.1 \cdot 1 + -0.4 \cdot 0 + 2.7 \cdot 0 + \ldots
\]
The feature vector $\phi(x, d)$ is constructed by hand.
Deep learning alert!

The feature vector $\phi(x, d)$ is constructed by hand

Constructing good features is hard
Deep learning alert!

The feature vector $\phi(x, d)$ is constructed by hand

Constructing good features is hard

Algorithms are likely to do it better
Deep learning alert!

The feature vector $\phi(x, d)$ is constructed by hand

Constructing good features is hard

Algorithms are likely to do it better

Perhaps we can train $\phi(x, d)$

$$\phi(x, d) = F_\psi(x, d), \text{ where } \psi \text{ are the parameters}$$
Log-linear model

Candidate derivations: $D(x)$

Model: distribution over derivations $d$ given utterance $x$

$$p_\theta(d \mid x) = \frac{\exp(\text{Score}_\theta(x,d))}{\sum_{d' \in D(x)} \exp(\text{Score}_\theta(x,d'))}$$
Log-linear model

Candidate derivations: $\mathcal{D}(x)$

Model: distribution over derivations $d$ given utterance $x$

$$p_\theta(d \mid x) = \frac{\exp(\text{Score}_\theta(x,d))}{\sum_{d' \in \mathcal{D}(x)} \exp(\text{Score}_\theta(x,d'))}$$

score$_\theta(x,d)$

$$[1, 2, 3, 4]$$

$$p_\theta(d \mid x) \left[ \frac{e}{e + e^2 + e^3 + e^4}, \frac{e^2}{e + e^2 + e^3 + e^4}, \frac{e^3}{e + e^2 + e^3 + e^4}, \frac{e^4}{e + e^2 + e^3 + e^4} \right]$$
Log-linear model

Candidate derivations: $\mathcal{D}(x)$

Model: distribution over derivations $d$ given utterance $x$

$$p_\theta(d \mid x) = \frac{\exp(\text{Score}_\theta(x,d))}{\sum_{d' \in \mathcal{D}(x)} \exp(\text{Score}_\theta(x,d'))}$$

$\text{score}_\theta(x, d)$

$$[1, 2, 3, 4]$$

$$p_\theta(d \mid x) \quad \begin{bmatrix} \frac{e}{e+e^2+e^3+e^4}, & \frac{e^2}{e+e^2+e^3+e^4}, & \frac{e^3}{e+e^2+e^3+e^4}, & \frac{e^4}{e+e^2+e^3+e^4} \end{bmatrix}$$

Parsing: find the top-$K$ derivation trees $\mathcal{D}_\theta(x)$
Features

Dense features:

- intersection=0.67
- ent-popularity:HIGH
- denoation-size:1
Features

Dense features:

- intersection=0.67
- ent-popularity:HIGH
- denoation-size:1

Sparse features:

- bridge-binary:STUDY
- born:PlaceOfBirth
- city:Type.Location
Features

Dense features:

• intersection=0.67
• ent-popularity:HIGH
• denotation-size:1

Sparse features:

• bridge-binary:STUDY
• born:PlaceOfBirth
• city:Type.Location

Syntactic features:

• ent-pos:NNP NNP
• join-pos:V NN
• skip-pos:IN
Features

Dense features:

- intersection=0.67
- ent-popularity:HIGH
- denoation-size:1

Sparse features:

- bridge-binary:STUDY
- born:PlaceOfBirth
- city:Type.Location

Syntactic features:

- ent-pos:NNP NNP
- join-pos:V NN
- skip-pos:IN

Grammar features:

- Binary->Verb
Learning $\theta$: marginal maximum-likelihood

Training data:

<table>
<thead>
<tr>
<th>What’s Bulgaria’s capital?</th>
<th>Sofia</th>
</tr>
</thead>
<tbody>
<tr>
<td>What movies has Tom Cruise been in?</td>
<td>TopGun, VanillaSky,…</td>
</tr>
<tr>
<td>What’s Bulgaria’s capital?</td>
<td>CapitalOf.Bulgaria</td>
</tr>
<tr>
<td>What movies has Tom Cruise been in?</td>
<td>Type.Movie $\sqcap$ HasPlayed.TomCruise</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Learning $\theta$: marginal maximum-likelihood

Training data:

<table>
<thead>
<tr>
<th>What's Bulgaria's capital?</th>
<th>Sofia</th>
</tr>
</thead>
<tbody>
<tr>
<td>What movies has Tom Cruise been in?</td>
<td>TopGun, VanillaSky,...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$$\arg\max_\theta \sum_{i=1}^n \log p_\theta(y^{(i)} \mid x^{(i)}) =$$

$$\arg\max_\theta \sum_{i=1}^n \log \sum_{d^{(i)}} p_\theta(d^{(i)} \mid x^{(i)}) R(d^{(i)})$$
Learning $\theta$: marginal maximum-likelihood

Training data:

- What's Bulgaria's capital?
  - Sofia
- What movies has Tom Cruise been in?
  - TopGun, VanillaSky, ...

$$\arg\max_{\theta} \sum_{i=1}^{n} \log p_\theta(y^{(i)} \mid x^{(i)}) =$$

$$\arg\max_{\theta} \sum_{i=1}^{n} \log \sum_{d^{(i)}} p_\theta(d^{(i)} \mid x^{(i)}) R(d^{(i)})$$

$$R(d) = \begin{cases} 1 & d.z = z^{(i)} \\ 0 & o/w \end{cases}$$

$$R(d) = \begin{cases} 1 & [d.z]_\kappa = y^{(i)} \\ 0 & o/w \end{cases}$$

$$R(d) = F_1([d.z]_\kappa, y^{(i)})$$
Optimization: stochastic gradient descent

For every example:

\[ O(\theta) = \log \sum_d p_\theta(d \mid x) R(d) \]
\[ \nabla O(\theta) = E_{q_\theta(d \mid x)}[\phi(x, d)] - E_{p_\theta(d \mid x)}[\phi(x, d)] \]
\[ p_\theta(d \mid x) \propto \exp(\phi(x, d)^\top \theta) \]
\[ q_\theta(d \mid x) \propto \exp(\phi(x, d)^\top \theta) \cdot R(d) \]
Optimization: stochastic gradient descent

For every example:

\[ O(\theta) = \log \sum_d p_\theta(d \mid x) R(d) \]
\[ \nabla O(\theta) = E_{q_\theta(d \mid x)}[\phi(x, d)] - E_{p_\theta(d \mid x)}[\phi(x, d)] \]
\[ p_\theta(d \mid x) \propto \exp(\phi(x, d)^\top \theta) \]
\[ q_\theta(d \mid x) \propto \exp(\phi(x, d)^\top \theta) \cdot R(d) \]

\[ p_\theta(D(x)) = [0.2, 0.1, 0.1, 0.6] \]
\[ R(D(x)) = [1, 0, 0, 1] \]
Optimization: stochastic gradient descent

For every example:

\[
O(\theta) = \log \sum_d p_\theta(d \mid x) R(d)
\]
\[
\nabla O(\theta) = E_{q_\theta(d|x)}[\phi(x, d)] - E_{p_\theta(d|x)}[\phi(x, d)]
\]
\[
p_\theta(d \mid x) \propto \exp(\phi(x, d) \top \theta)
\]
\[
q_\theta(d \mid x) \propto \exp(\phi(x, d) \top \theta) \cdot R(d)
\]

\[
p_\theta(\mathcal{D}(x)) = [0.2, 0.1, 0.1, 0.6]
\]
\[
R(\mathcal{D}(x)) = [1, 0, 0, 1]
\]
\[
q_\theta(\mathcal{D}(x)) = [0.25, 0, 0, 0.75]
\]
\[
q_\theta = \frac{p_\theta}{p_\theta \cdot R}
\]
Optimization: stochastic gradient descent

For every example:

\[
O(\theta) = \log \sum_d p_\theta(d | x) R(d)
\]
\[
\nabla O(\theta) = E_{q_\theta(d|x)}[\phi(x, d)] - E_{p_\theta(d|x)}[\phi(x, d)]
\]

\[
p_\theta(d | x) \propto \exp(\phi(x, d)^\top \theta)
\]
\[
q_\theta(d | x) \propto \exp(\phi(x, d)^\top \theta) \cdot R(d)
\]

\[
p_\theta(\mathcal{D}(x)) = [0.2, 0.1, 0.1, 0.6]
\]
\[
R(\mathcal{D}(x)) = [1, 0, 0, 1]
\]
\[
q_\theta(\mathcal{D}(x)) = [0.25, 0, 0, 0.75]
\]
\[
q_\theta = \frac{p_\theta}{p_\theta R}
\]

Gradient:

\[
0.05 \cdot \phi(x, d_1) - 0.1 \cdot \phi(x, d_2) - 0.1 \cdot \phi(x, d_3) + 0.15 \cdot \phi(x, d_4)
\]
Training

Input: $\{x_i, y_i\}_{i=1}^{n}$

Output: $\theta$
Training

Input: \( \{x_i, y_i\}_{i=1}^{n} \)

Output: \( \theta \)

\( \theta \leftarrow 0 \)
Training

Input: \( \{x_i, y_i\}^n_{i=1} \)

Output: \( \theta \)

\( \theta \leftarrow 0 \)

for iteration \( \tau \) and example \( i \)

\( \mathcal{D}(x_i) \leftarrow \arg\max^K (p_\theta(d \mid x_i)) \)
Training

Input: \( \{ x_i, y_i \}_{i=1}^{n} \)

Output: \( \theta \)

\( \theta \leftarrow 0 \)

for iteration \( \tau \) and example \( i \)

\[
D(x_i) \leftarrow \arg \max^K (p_\theta(d \mid x_i))
\]

\[
\theta \leftarrow \theta + \eta_{\tau,i} (E_{q_\theta(d \mid x_i)}[\phi(x_i, d)] - E_{p_\theta(d \mid x_i)}[\phi(x_i, d)])
\]
Training

Input: \( \{x_i, y_i\}_{i=1}^n \)

Output: \( \theta \)

\[
\theta \leftarrow 0
\]

for iteration \( \tau \) and example \( i \)

\[
D(x_i) \leftarrow \text{arg max}_K (p_\theta (d \mid x_i))
\]

\[
\theta \leftarrow \theta + \eta_{\tau,i} (E_{q_\theta (d \mid x_i)} [\phi(x_i, d)] − E_{p_\theta (d \mid x_i)} [\phi(x_i, d)])
\]

\( \eta_{\tau,i} \): learning rate

Regularization often added (L2, L1, ...)

Training (structured perceptron)

Input: \{x_i, y_i\}_{i=1}^n

Output: \theta
Training (structured perceptron)

Input: \( \{ x_i, y_i \}_{i=1}^{n} \)

Output: \( \theta \)

\( \theta \leftarrow 0 \)
Training (structured perceptron)

Input: \( \{x_i, y_i\}_{i=1}^n \)

Output: \( \theta \)

\( \theta \leftarrow 0 \)

for iteration \( \tau \) and example \( i \)

\( \hat{d} \leftarrow \text{arg max}(p_\theta(d \mid x_i)) \)

\( d^* \leftarrow \text{arg max}(q_\theta(d \mid x_i)) \)
Training (structured perceptron)

Input: \( \{ x_i, y_i \}_{i=1}^n \)

Output: \( \theta \)

\( \theta \leftarrow 0 \)

for iteration \( \tau \) and example \( i \)

\( \hat{d} \leftarrow \arg \max (p_\theta(d \mid x_i)) \)

\( d^* \leftarrow \arg \max (q_\theta(d \mid x_i)) \)

if \( [d^*]_K \neq [\hat{d}]_K \)

\( \theta \leftarrow \theta + \phi(x_i, d^*) - \phi(x_i, \hat{d}) \)
Training (structured perceptron)

Input: \( \{x_i, y_i\}_{i=1}^n \)

Output: \( \theta \)

\( \theta \leftarrow 0 \)

for iteration \( \tau \) and example \( i \)

\( \hat{d} \leftarrow \arg \max(p_\theta(d \mid x_i)) \)

\( d^* \leftarrow \arg \max(q_\theta(d \mid x_i)) \)

if \( [d^*]_K \neq [\hat{d}]_K \)

\( \theta \leftarrow \theta + \phi(x_i, d^*) - \phi(x_i, \hat{d}) \)

Regularization often added with weight averaging
Training

Other simple variants exist:

- E.g., cost-sensitive max-margin training

- That is, find pairs of good and bad derivations that look different but have similar scores and update on those