Outline

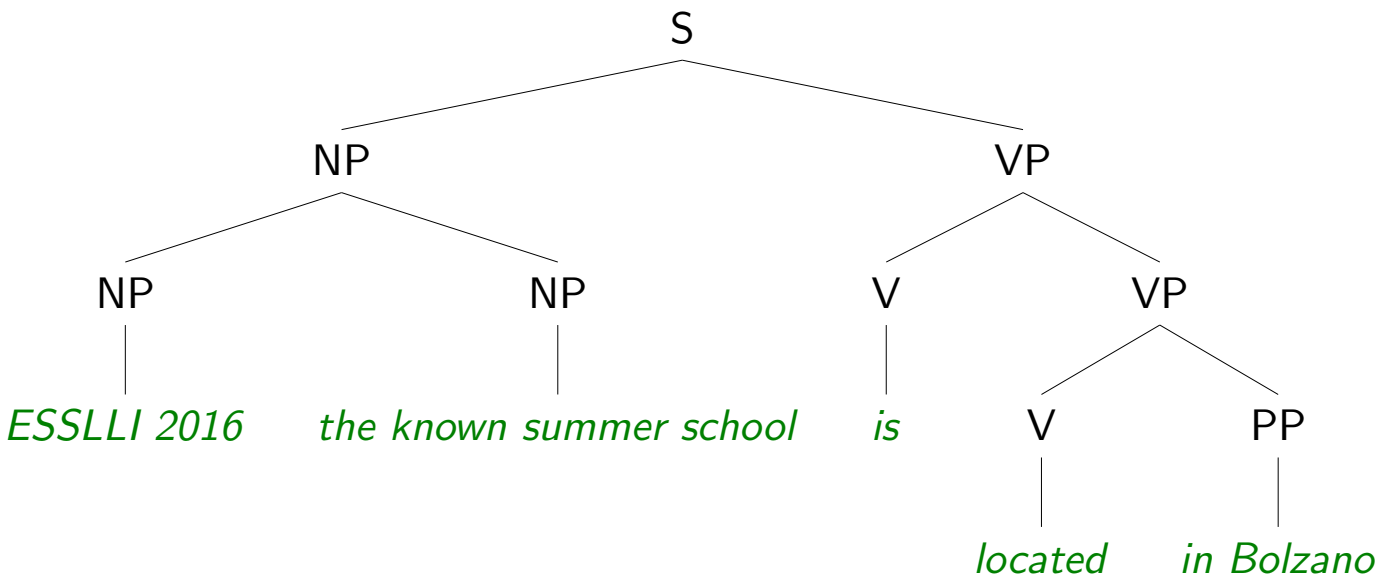
- **Learning**
  - Overview
  - Details
  - Example
  - Lexicon learning
  - Supervision signals

Outline

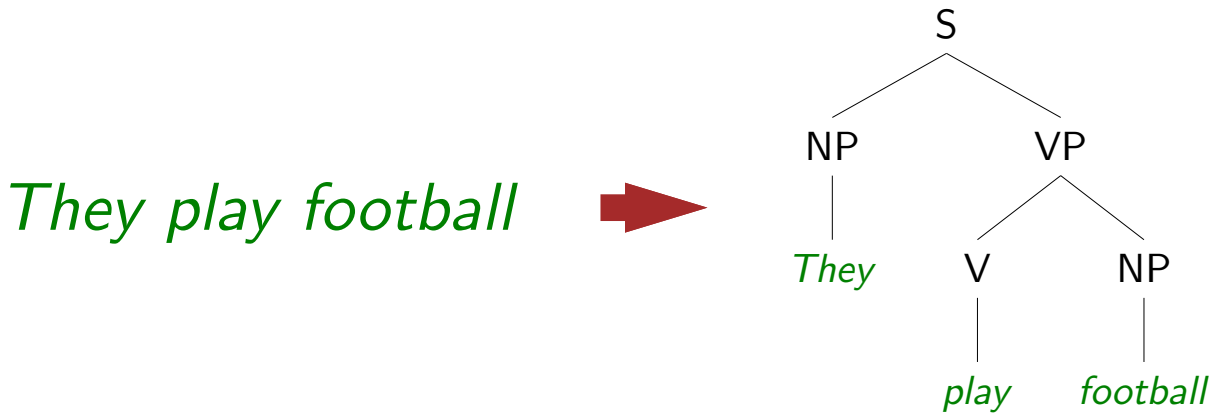
- Learning
  - **Overview**
  - Details
  - Example
  - Lexicon learning
  - Supervision signals

Supervision in syntactic parsing

Input:



Output:



[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 $\sqcap$ Gender.Female

Youngest player of the Cavaliers

$\arg\min(\text{PlayerOf.Cavaliers}, \text{BirthDateOf})$

...

Light supervision

How tall is Lebron James?

203cm

What is Steph Curry's daughter called?

Riley Curry

Youngest player of the Cavaliers

Kyrie Irving

...

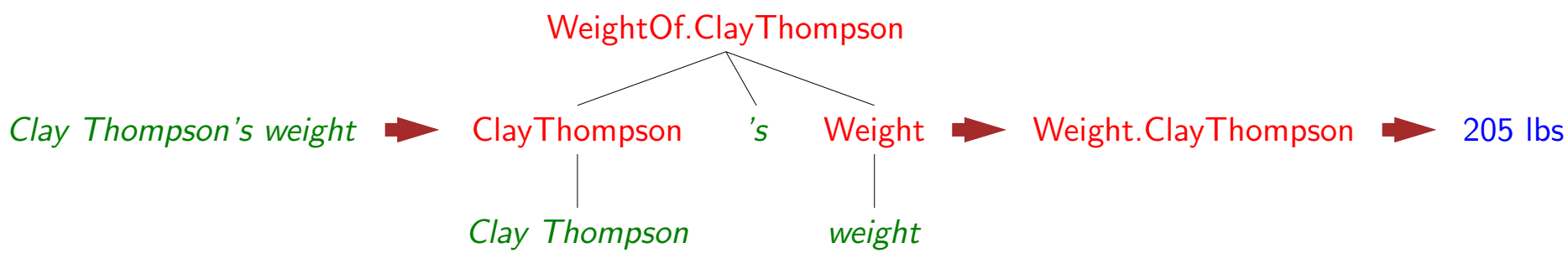
[Zelle & Mooney, 1996; Zettlemoyer & Collins, 2005; Clarke et al. 2010; Liang et al., 2011]

Supervision in semantic parsing

Input:

Heavy supervision	**Light supervision**
How tall is Lebron James?	*How tall is Lebron James?*
HeightOf.LebronJames	203cm
What is Steph Curry's daughter called?	*What is Steph Curry's daughter called?*
ChildrenOf.StephCurry $\sqcap$ Gender.Female	Riley Curry
Youngest player of the Cavaliers	*Youngest player of the Cavaliers*
$\arg\min$(PlayerOf.Cavaliers, BirthDateOf)	Kyrie Irving
...	...

Output:



Learning in a nutshell

utterance

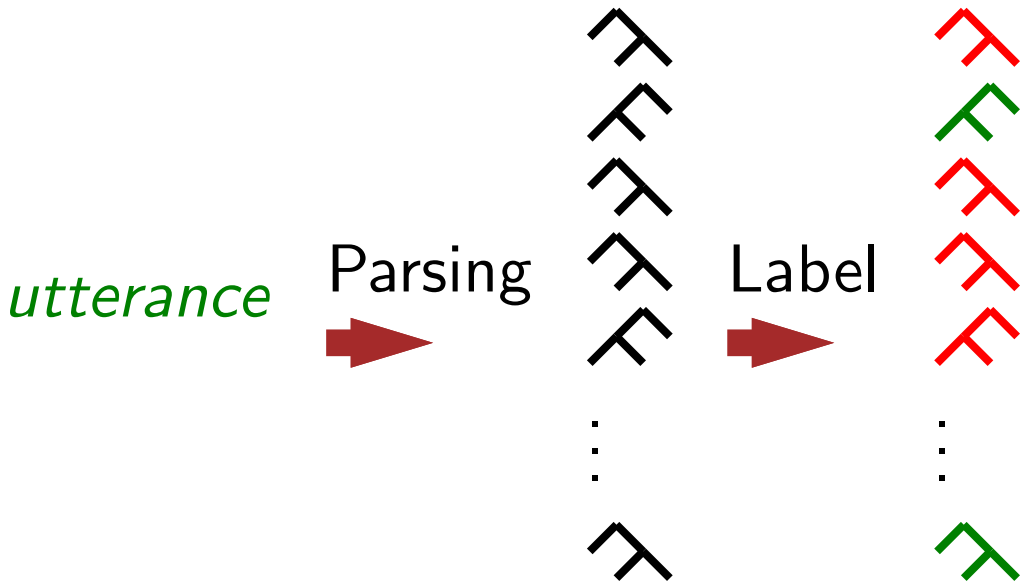
0. Define model for derivations

Learning in a nutshell

utterance Parsing

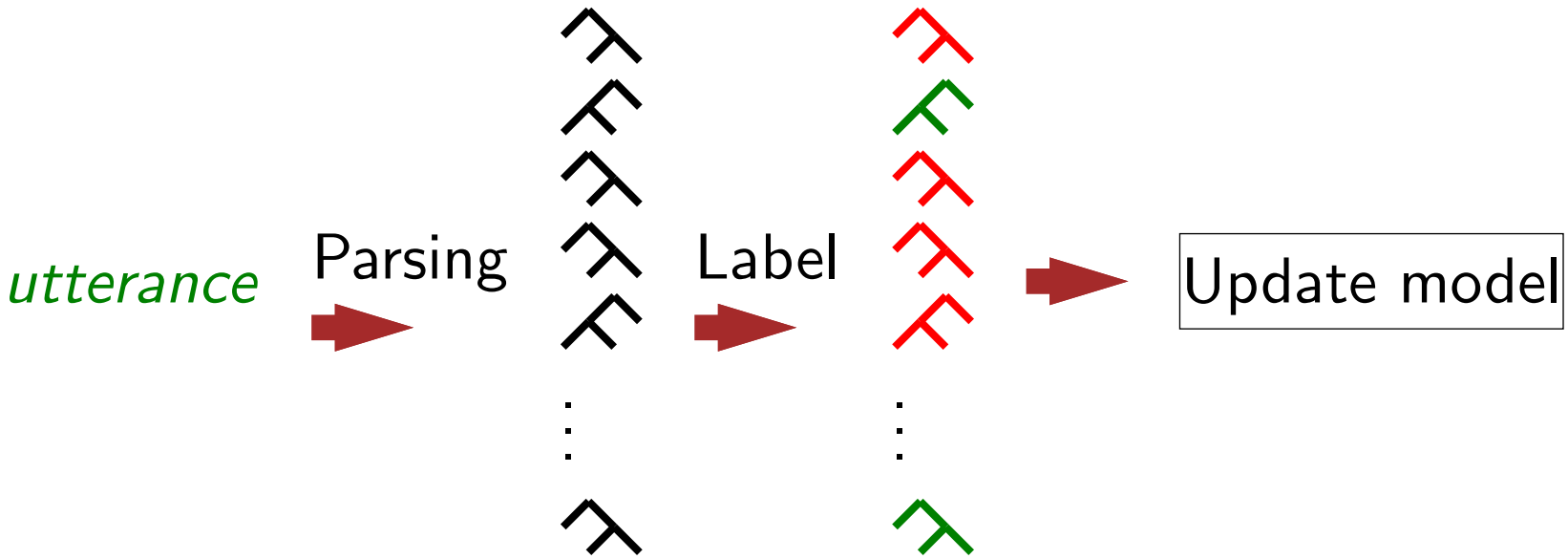
0. Define model for derivations
1. Generate candidate derivations (later)

Learning in a nutshell



0. Define model for derivations
1. Generate candidate derivations (later)
2. Label as correct and incorrect

Learning in a nutshell



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`	$\Rightarrow$ Salzburg
`PlaceOfDeath.WolfgangMozart`	$\Rightarrow$ Vienna
`PlaceOfMarriage.WolfgangMozart`	$\Rightarrow$ Vienna

Vienna

Training intuition

Where did Mozart tupress?

~~`PlaceOfBirth.WolfgangMozart` $\Rightarrow$ Salzburg~~

`PlaceOfDeath.WolfgangMozart` $\Rightarrow$ Vienna

`PlaceOfMarriage.WolfgangMozart` $\Rightarrow$ Vienna

Vienna

Training intuition

Where did Mozart tupress?

~~`PlaceOfBirth.WolfgangMozart` $\Rightarrow$ Salzburg~~

`PlaceOfDeath.WolfgangMozart` $\Rightarrow$ Vienna

`PlaceOfMarriage.WolfgangMozart` $\Rightarrow$ Vienna

Vienna

Where did Hogarth tupress?

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`

`PlaceOfDeath.WilliamHogarth`

`PlaceOfMarriage.WilliamHogarth`

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 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 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

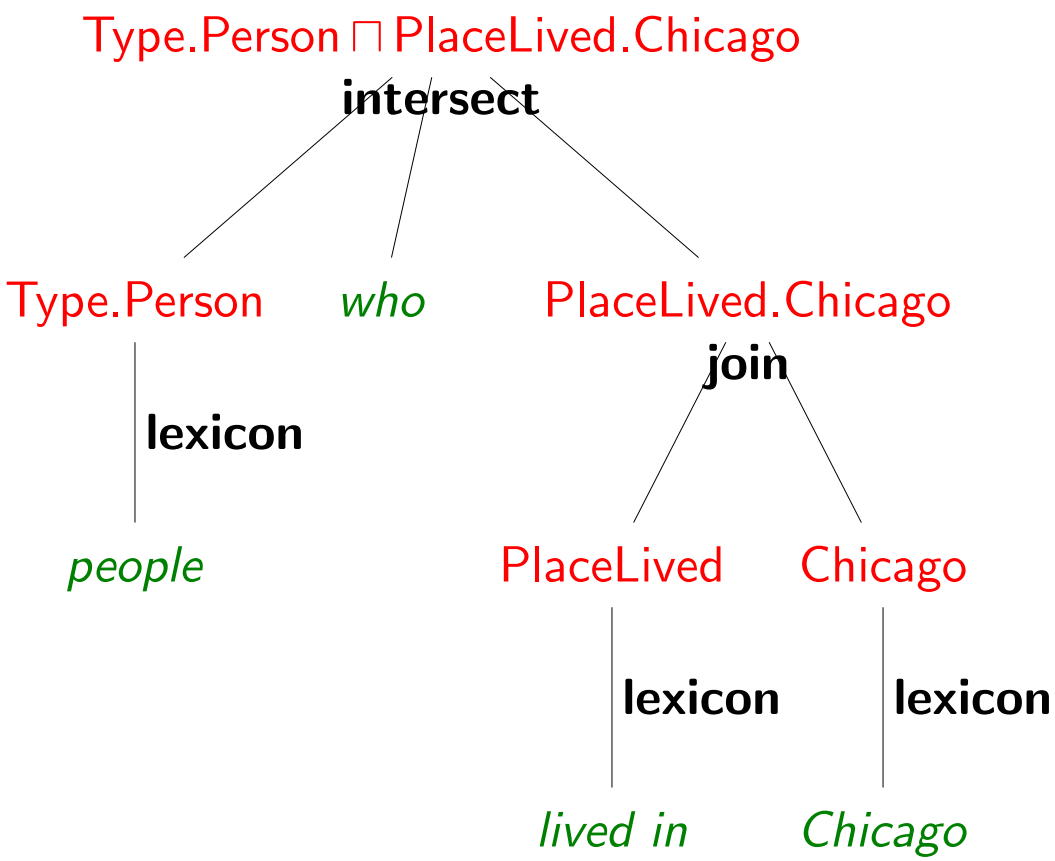
~~`PlaceOfMarriage.WilliamHogarth` $\Rightarrow$ Paddington~~

London

Outline

- Learning
  - Overview
  - **Details**
  - Example
  - Lexicon learning
  - Supervision signals

Constructing derivations

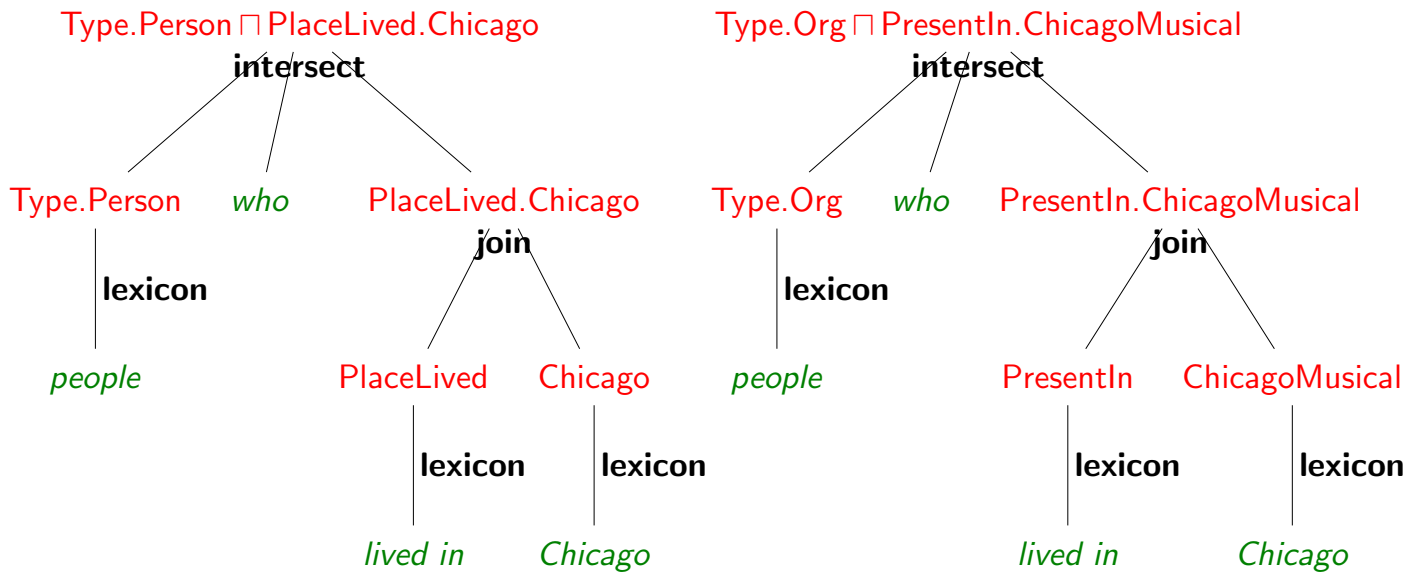


Many possible derivations!

$x =$ *people who have lived in Chicago*

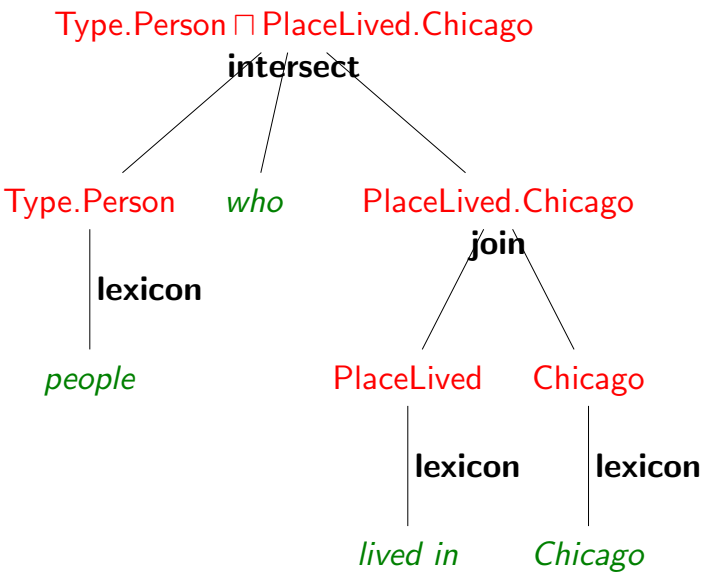
?

set of candidate derivations $\mathcal{D}(x)$



x: utterance

d: derivation

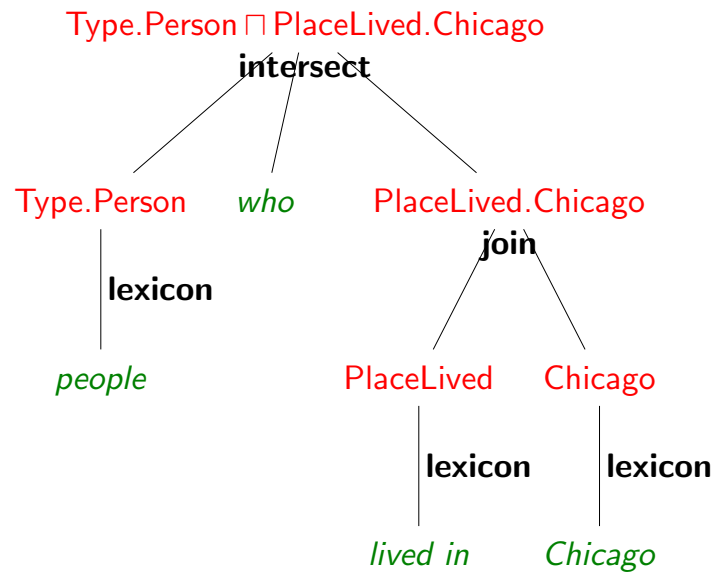


Feature vector and parameters in $\mathbb{R}^F$:

	$\phi(x,d)$	θ ⇐learned
apply join	1	1.2
apply intersect	1	0.6
apply lexicon	3	2.1
lived maps to PlacesLived	1	3.1
lived maps to PlaceOfBirth	0	-0.4
born maps to PlaceOfBirth	0	2.7
...	...	...

x: utterance

d: derivation



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

	$\phi(x,d)$	θ ⇐learned
apply join	1	1.2
apply intersect	1	0.6
apply lexicon	3	2.1
lived maps to PlacesLived	1	3.1
lived maps to PlaceOfBirth	0	-0.4
born maps to PlaceOfBirth	0	2.7
...	...	...

$$\text{Score}_\theta(x,d) = \phi(x,d)^\top \theta =$$

$$1.2 \cdot 1 + 0.6 \cdot 1 + 2.1 \cdot 3 + 3.1 \cdot 1 + -0.4 \cdot 0 + 2.7 \cdot 0 + \ldots$$

Deep learning alert!

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 $\phi(x, d)$

$\phi(x, d) = F_\psi(x, d)$, where ψ are the parameters

Log-linear model

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

Model: distribution over derivations d given utterance x

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

Log-linear model

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

Model: distribution over derivations d given utterance x

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

$\mathsf{score}_\theta(x, d)$ $\quad [1, 2, 3, 4]$

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

Log-linear model

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

Model: distribution over derivations d given utterance x

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

$\mathsf{score}_\theta(x, d)$ $\quad [1, 2, 3, 4]$

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

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`
- `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`

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 θ: maximum-likelihood

Training data:

What's Bulgaria's capital?	*What's Bulgaria's capital?*
Sofia	CapitalOf.Bulgaria
What movies has Tom Cruise been in?	*What movies has Tom Cruise been in?*
TopGun,VanillaSky,...	Type.Movie $\sqcap$ HasPlayed.TomCruise
...	...

Learning θ: maximum-likelihood

Training data:

What's Bulgaria's capital?	*What's Bulgaria's capital?*
Sofia	CapitalOf.Bulgaria
What movies has Tom Cruise been in?	*What movies has Tom Cruise been in?*
TopGun,VanillaSky,...	Type.Movie $\sqcap$ HasPlayed.TomCruise
...	...

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

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

Learning θ: maximum-likelihood

Training data:

What's Bulgaria's capital?	*What's Bulgaria's capital?*
Sofia	CapitalOf.Bulgaria
What movies has Tom Cruise been in?	*What movies has Tom Cruise been in?*
TopGun,VanillaSky,...	Type.Movie $\sqcap$ HasPlayed.TomCruise
...	...

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

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

$$R(d) = \begin{cases} 1 & d.z = z^{(i)} \\ 0 & \text{o/w} \end{cases} \quad R(d) = \begin{cases} 1 & [d.z]_{\mathcal{K}} = y^{(i)} \\ 0 & \text{o/w} \end{cases} \quad R(d) = F_1([d.z]_{\mathcal{K}}, y^{(i)})$$

Optimization: stochastic gradient descent

For every example:

$$O(\theta) = \log \sum_d p_\theta(d \mid x) R(d)$$

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

$$p_\theta(d \mid x) \propto \exp(\phi(x,d)^\top \theta)$$

$$q_\theta(d \mid x) \propto \exp(\phi(x,d)^\top \theta) \cdot R(d)$$

Optimization: stochastic gradient descent

For every example:

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

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

Optimization: stochastic gradient descent

For every example:

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

$p_\theta(\mathcal{D}(x)) = [0.2, 0.1, 0.1, 0.6]$

$R(\mathcal{D}(x)) = [1, 0, 0, 1]$

$q_\theta(\mathcal{D}(x)) = [0.25, 0, 0, 0.75]$

$q_\theta = \frac{p_\theta}{p_\theta^\top R}$

Optimization: stochastic gradient descent

For every example:

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

$p_\theta(\mathcal{D}(x)) = [0.2, 0.1, 0.1, 0.6]$

$R(\mathcal{D}(x)) = [1, 0, 0, 1]$

$q_\theta(\mathcal{D}(x)) = [0.25, 0, 0, 0.75]$

$q_\theta = \frac{p_\theta}{p_\theta^\top R}$

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: θ

Training

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

Output: θ

$\theta \leftarrow 0$

Training

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

Output: θ

$\theta \leftarrow 0$

for iteration τ 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: θ

$\theta \leftarrow 0$

for iteration τ and example i

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

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

Training

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

Output: θ

$\theta \leftarrow 0$

for iteration τ and example i

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

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

$\eta_{\tau,i}$: learning rate

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

Training (structured perceptron)

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

Output: θ

Training (structured perceptron)

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

Output: θ

$\theta \leftarrow 0$

Training (structured perceptron)

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

Output: θ

$\theta \leftarrow 0$

for iteration τ and example i

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

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

Training (structured perceptron)

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

Output: θ

$\theta \leftarrow 0$

for iteration τ 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 $[d^*]_\mathcal{K} \neq [\hat{d}]_\mathcal{K}$

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

Training (structured perceptron)

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

Output: θ

$\theta \leftarrow 0$

for iteration τ 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 $[d^*]_\mathcal{K} \neq [\hat{d}]_\mathcal{K}$

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

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

Outline

- Learning
  - Overview
  - Details
  - **Example**
  - Lexicon learning
  - Supervision signals

Example

```
./run @mode=simple-lambdadcs \

-Grammar.inPaths esslli_2016/class3_demo.grammar \

-SimpleLexicon.inPaths esslli_2016/class3_demo.lexicon
```

```
(loadgraph geo880/geo880.kg)
size of california
size capital california
size of capital of california
california size
```

Exercise

Find a pair of natural language utterances that cannot be distinguished using the current feature representation

- The utterances can be not fully grammatical in English
- You can ignore the denotation feature if that helps
- Verify this in sempre (ask me how to disable features)
- Design a feature that will solve this problem

Outline

- Learning
  - Overview
  - Details
  - Example
  - **Lexicon learning**
  - Supervision signals

The lexicon problem

How is the lexicon generated?

- Annotation
- Exhaustive search
- String matching
- Supervised alignment
- Unsupervised alignment
- **Learning**

Training

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

Output: θ

$\theta \leftarrow 0$

for iteration τ and example i

Add lexicon entries

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

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

$\eta_{\tau,i}$: learning rate

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

[Adapted from semantic parsing tutorial, Artzi et al.]

Adding lexicon entries

Input: training example (x_i, y_i), current lexicon Λ, model θ

$\Lambda_{\text{temp}} \leftarrow \Lambda \cup \text{GENLEX}(x_i, y_i)$ Create expanded temporaty lexicon

[Adapted from semantic parsing tutorial, Artzi et al.]

Adding lexicon entries

Input: training example (x_i, y_i), current lexicon Λ, model θ

$\Lambda_{\text{temp}} \leftarrow \Lambda \cup \text{GENLEX}(x_i, y_i)$	Create expanded temporaty lexicon
$\mathcal{D}(x_i) \leftarrow \arg\max^K (p_{\theta,\Lambda_{\text{temp}}}(d \mid x_i))$	Parse with temporary lexicon

[Adapted from semantic parsing tutorial, Artzi et al.]

Adding lexicon entries

Input: training example (x_i, y_i), current lexicon Λ, model θ

$\Lambda_{\text{temp}} \leftarrow \Lambda \cup \text{GENLEX}(x_i, y_i)$	Create expanded temporaty lexicon
$\mathcal{D}(x_i) \leftarrow \arg\max^K (p_{\theta,\Lambda_{\text{temp}}}(d \mid x_i))$	Parse with temporary lexicon
$\Lambda = \Lambda \cup \{l \mid l \in \hat{d}, \hat{d} \in \mathcal{D}(x_i), R(d) = 1\}$	Add entries from correct trees

[Adapted from semantic parsing tutorial, Artzi et al.]

Adding lexicon entries

Input: training example (x_i, y_i), current lexicon Λ, model θ

$\Lambda_{\text{temp}} \leftarrow \Lambda \cup \text{GENLEX}(x_i, y_i)$	Create expanded temporaty lexicon
$\mathcal{D}(x_i) \leftarrow \arg\max^K (p_{\theta,\Lambda_{\text{temp}}}(d \mid x_i))$	Parse with temporary lexicon
$\Lambda = \Lambda \cup \{l \mid l \in \hat{d}, \hat{d} \in \mathcal{D}(x_i), R(d) = 1\}$	Add entries from correct trees

Overgenerate lexical entries and add promising ones

[Zettlemoyer and Collins, 2005]

Lexicon generation

Logical form supervision:

Largest state bordering California

$\text{argmax}(\text{Type}(\text{State}) \sqcap \text{Border}(\text{California}), \text{Area})$

[Zettlemoyer and Collins, 2005]

Lexicon generation

Logical form supervision:

Largest state bordering California

$\text{argmax}(\text{Type}(\text{State}) \sqcap \text{Border}(\text{California}), \text{Area})$

Enumerate spans	Use rules to extract sub-forumulas
Largest	California
state	Border(California)
bordering	$\lambda f.f(\textbf{California})$
California	Area
Largest state	$\lambda x.\text{argmax}(x, \text{Area})$
...	...

[Zettlemoyer and Collins, 2005]

Lexicon generation

Logical form supervision:

Largest state bordering California

$\text{argmax}(\text{Type}(\text{State}) \sqcap \text{Border}(\text{California}), \text{Area})$

Enumerate spans	Use rules to extract sub-forumulas
Largest	California
state	Border(California)
bordering	$\lambda f.f(\textbf{California})$
California	Area
Largest state	$\lambda x.\text{argmax}(x, \text{Area})$
...	...

Add cross-product to lexicon

Lexicon generation

Denotation supervision:

Largest state bordering California

Arizona

Lexicon generation

Denotation supervision:

Largest state bordering California

Arizona

Enumerate spans	Generate sub-formulas from KB
Largest	California
state	Border(California)
bordering	Traverse
California	Type.Mountain
Largest state	$\lambda x.\text{argmax}(x, \text{Elevation})$
...	...

Restrict candidates with alignment, string matching, ...

Lexicon generation

Denotation supervision:

Largest state bordering California

Arizona

Enumerate spans	Generate sub-formulas from KB
Largest	California
state	Border(California)
bordering	Traverse
California	Type.Mountain
Largest state	$\lambda x.\text{argmax}(x, \text{Elevation})$
...	...

Restrict candidates with alignment, string matching, ...

Fancier methods exist (coarse-to-fine)

[Kwiatkowski et al, 2010]

Unification

Logical form supervision:

[Kwiatkowski et al, 2010]

Unification

Logical form supervision:

Initialize lexicon with (x_i, z_i):

States bordering California

$\text{Type(State)} \sqcap \text{Border(California)}$

Split lexical entry in all possible ways:

[Kwiatkowski et al, 2010]

Unification

Logical form supervision:

Initialize lexicon with (x_i, z_i):

States bordering California

$\text{Type(State)} \sqcap \text{Border(California)}$

Split lexical entry in all possible ways:

Enumerate spans	Generate sub-formulas from KB
(states, bordering california)	$(\text{Type(State)}, \lambda x.x \sqcap \text{Border(California)})$
(states bordering, california)	$(\lambda x.\text{Type(State)} \sqcap x, \text{Border(California)})$
	$(\lambda f.\text{Type(State)} \sqcap f(\text{California}), \text{California})$
	$\ldots$

[Kwiatkowski et al, 2010]

Unification

For example x_i, z_i:

[Kwiatkowski et al, 2010]

Unification

For example x_i, z_i:

Find highest scoring correct parse d^*

[Kwiatkowski et al, 2010]

Unification

For example x_i, z_i:

Find highest scoring correct parse d^*

Split all lexical entries in d^* in all possible ways

[Kwiatkowski et al, 2010]

Unification

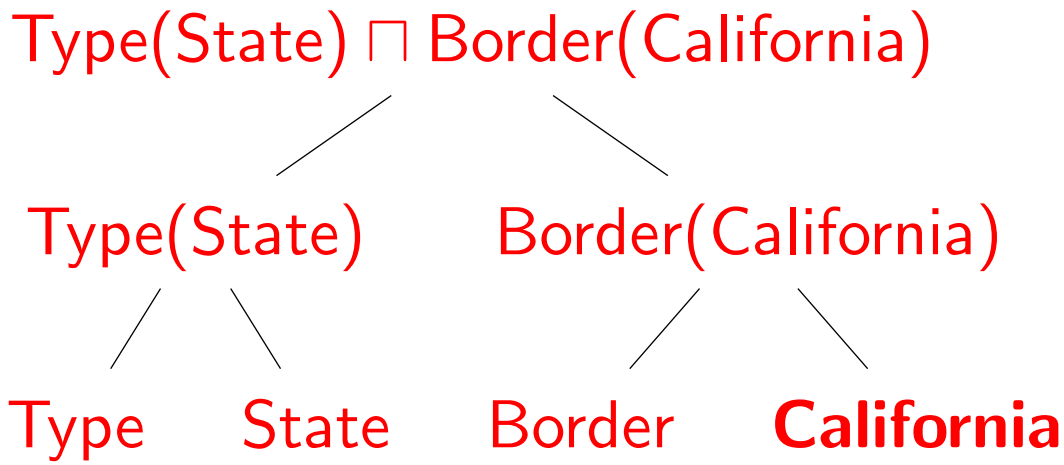
For example x_i, z_i:

Find highest scoring correct parse d^*

Split all lexical entries in d^* in all possible ways

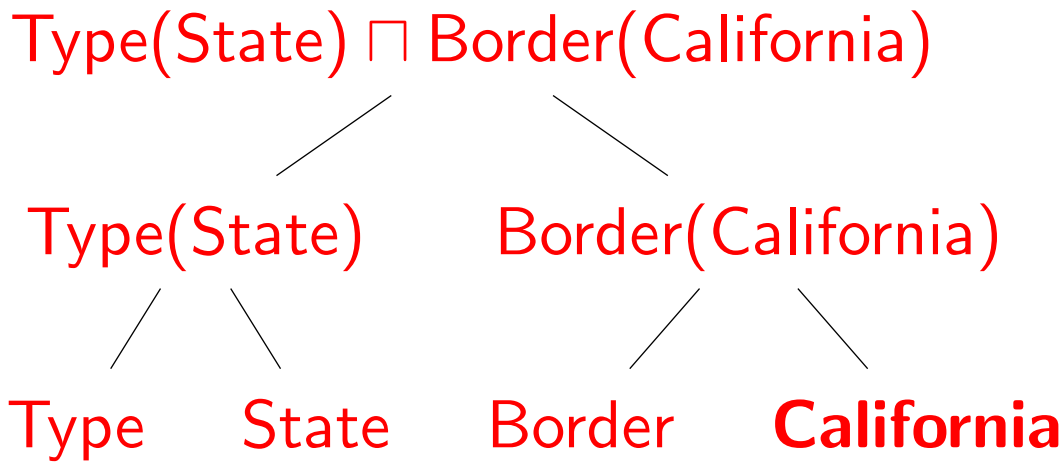
Add to lexicon lexical entry that improves parse score best

Do we need a lexicon?



California *neighbors*

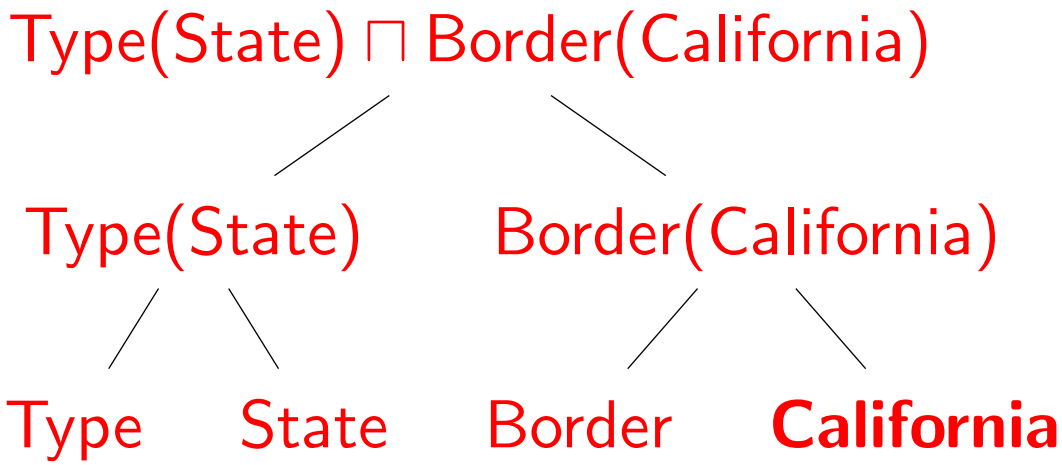
Do we need a lexicon?



California *neighbors*

Floating parse tree: generalization of bridging

Do we need a lexicon?



California *neighbors*

Floating parse tree: generalization of bridging

Perhaps with better learning and search not necessary?

Outline

- Learning
  - Overview
  - Details
  - Example
  - Lexicon learning
  - **Supervision signals**

Supervision signals

We discussed training from logical forms and denotations

Supervision signals

We discussed training from logical forms and denotations

Other forms of supervision have been proposed:

- Demonstrations
- Distant supervision
- Conversations
- Unsupervised
- Paraphrasing

[Artzi and Zettlemoyer, 2013]

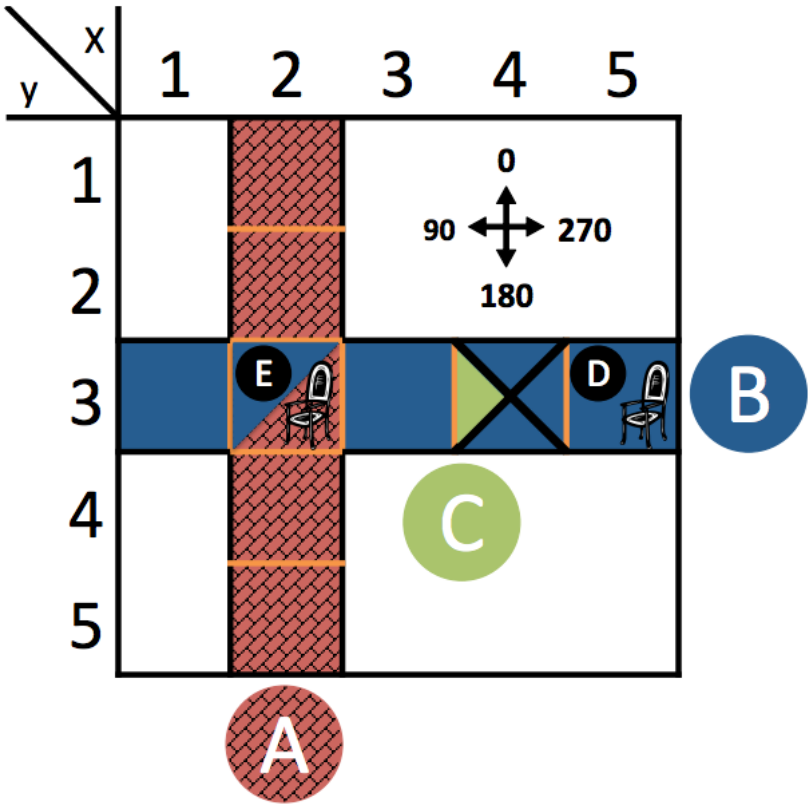
Training from demonstrations

Input: (x_i, s_i, t_i)

x_i: utterance

s_i: start state

t_i: end state



move forward until you reach the intersection

[Artzi and Zettlemoyer, 2013]

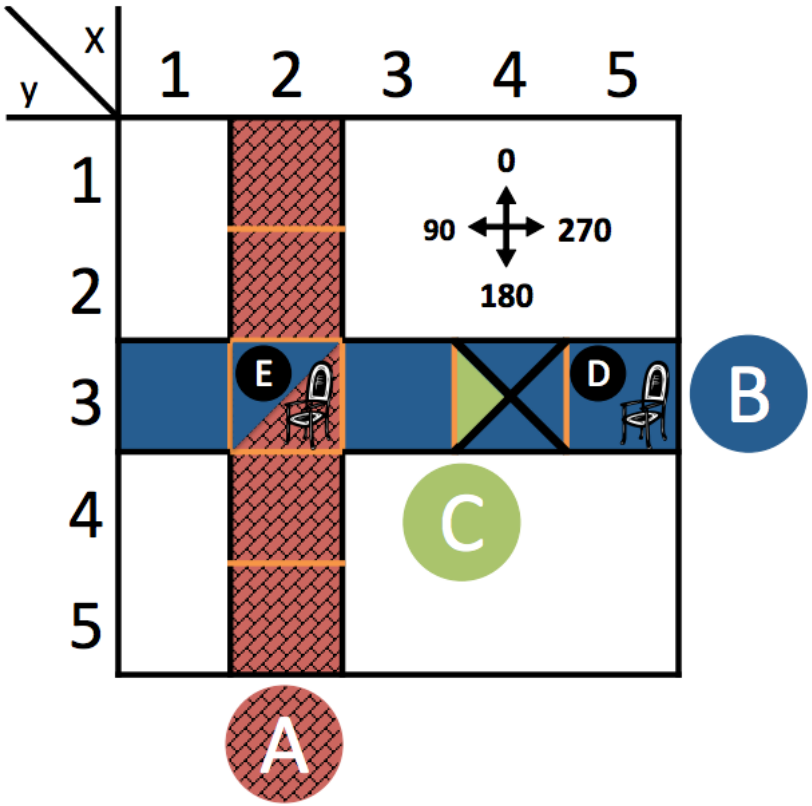
Training from demonstrations

Input: (x_i, s_i, t_i)

x_i: utterance

s_i: start state

t_i: end state



move forward until you reach the intersection

$\lambda a.\text{move}(a) \wedge \text{dir}(a.\text{forward}) \ldots$

[Artzi and Zettlemoyer, 2013]

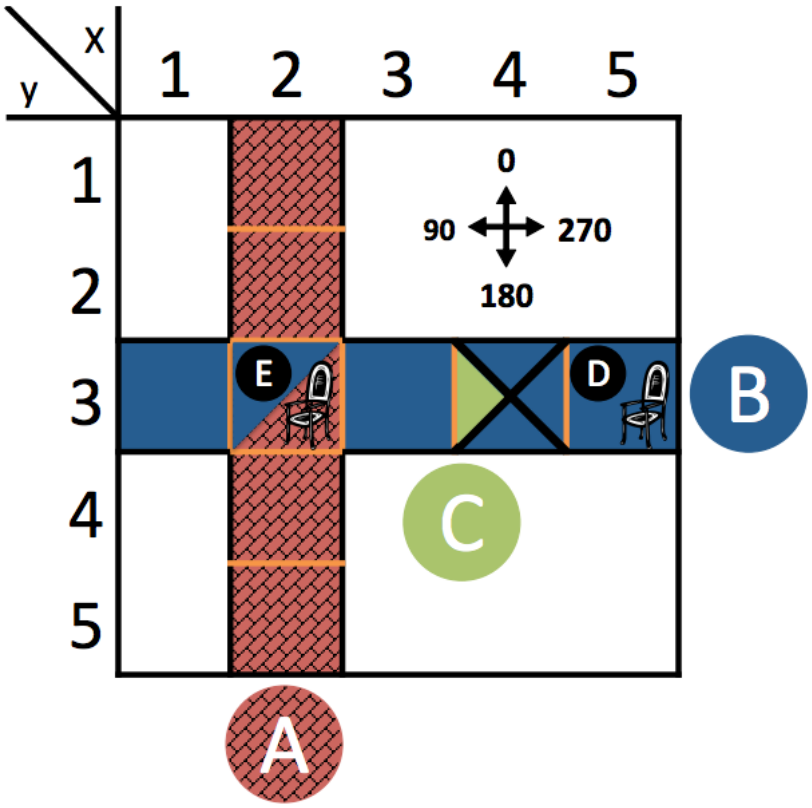
Training from demonstrations

Input: (x_i, s_i, t_i)

x_i: utterance

s_i: start state

t_i: end state



move forward until you reach the intersection

$\lambda a.\text{move}(a) \wedge \text{dir}(a.\text{forward}) \ldots$

An instance of learning from denotations

[Reddy et al., 2014]

Distant supervision

Data generation:

- Decompose declarative text to questions and answers

James Cameron is the director of Titanic

Q: X is the director of Titanic *A: James Cameron*

[Reddy et al., 2014]

Distant supervision

Data generation:

- Decompose declarative text to questions and answers

James Cameron is the director of Titanic

Q: X is the director of Titanic *A: James Cameron*

Declarative text is cheap!

[Reddy et al., 2014]

Distant supervision

Training:

- Use exising non-executable semantic parsers

X is the director of Titanic

[Reddy et al., 2014]

Distant supervision

Training:

- Use exising non-executable semantic parsers

X is the director of Titanic

$\lambda x.\text{director}(x) \wedge \text{director.of.arg1}(e, x) \wedge \text{director.of.arg2}(e, \text{Titanic})$

[Reddy et al., 2014]

Distant supervision

Training:

- Use exising non-executable semantic parsers

X is the director of Titanic

$\lambda x.\text{director}(x) \wedge \text{director.of.arg1}(e, x) \wedge \text{director.of.arg2}(e, \text{Titanic})$

$\lambda x.\text{Director}(x) \wedge \text{FilmDirectedBy}(e, x) \wedge \text{FileDirected}(e, \text{Titanic})$

$\lambda x.\text{Producer}(x) \wedge \text{FilmProducedBy}(e, x) \wedge \text{FileProduced}(e, \text{Titanic})$

[Reddy et al., 2014]

Distant supervision

Training:

- Use exising non-executable semantic parsers

X is the director of Titanic

$\lambda x.\text{director}(x) \wedge \text{director.of.arg1}(e, x) \wedge \text{director.of.arg2}(e, \text{Titanic})$

$\lambda x.\text{Director}(x) \wedge \text{FilmDirectedBy}(e, x) \wedge \text{FileDirected}(e, \text{Titanic})$

$\lambda x.\text{Producer}(x) \wedge \text{FilmProducedBy}(e, x) \wedge \text{FileProduced}(e, \text{Titanic})$

James Cameron	true
James Camerson, Jon Landau	false

Training from conversations

SYSTEM: how can i help you? (OPEN_TASK)

USER: i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning

SYSTEM: leaving what city? (ASK: $\lambda x.from(fl, x)$)

USER: atlanta georgia

SYSTEM: leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)

USER: london

[conversation continues]

Training from conversations

SYSTEM:	how can i help you? (OPEN_TASK)
USER:	i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning
SYSTEM:	leaving what city? (ASK: $\lambda x.from(fl, x)$)
USER:	atlanta georgia
SYSTEM:	leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)
USER:	london
[conversation continues]	

z_1 : From(Atlanta) $\sqcap$ To(London)

z_2 : From(Atlanta) $\sqcap$ From(London)

z_3 : To(Atlanta) $\sqcap$ To(London)

z_4 : To(Atlanta) $\sqcap$ From(London)

Training from conversations

SYSTEM:	how can i help you? (OPEN_TASK)
USER:	i would like to fly from atlanta georgia to london england on september twenty fourth in the early evening i would like to return on october first departing from london in the late morning
SYSTEM:	leaving what city? (ASK: $\lambda x.from(fl, x)$)
USER:	atlanta georgia
SYSTEM:	leaving atlanta. (CONFIRM: $from(fl, ATL)$) going to which city? (ASK: $\lambda x.to(fl, x)$)
USER:	london
[conversation continues]	

z_1 : From(Atlanta) $\sqcap$ To(London)

z_2 : From(Atlanta) $\sqcap$ From(London)

z_3 : To(Atlanta) $\sqcap$ To(London)

z_4 : To(Atlanta) $\sqcap$ From(London)

Define loss:

- Does z align with converation?
- Does z obey domain constraints?

[Goldwasser et al., 2011]

Unsupervised learning

Intuition: assume repeating patterns are correct

[Goldwasser et al., 2011]

Unsupervised learning

Intuition: assume repeating patterns are correct

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

Output: θ

[Goldwasser et al., 2011]

Unsupervised learning

Intuition: assume repeating patterns are correct

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

Output: θ

θ initialized manually, $S = \phi$

[Goldwasser et al., 2011]

Unsupervised learning

Intuition: assume repeating patterns are correct

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

Output: θ

θ initialized manually, $S = \phi$

Until stopping criterion met

for example x_i

$$S = S \cup (x_i, \arg\max p_\theta(d \mid x_i))$$

[Goldwasser et al., 2011]

Unsupervised learning

Intuition: assume repeating patterns are correct

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

Output: θ

θ initialized manually, $S = \phi$

Until stopping criterion met

for example x_i

$$S = S \cup (x_i, \arg\max p_\theta(d \mid x_i))$$

Compute statistics of S

[Goldwasser et al., 2011]

Unsupervised learning

Intuition: assume repeating patterns are correct

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

Output: θ

θ initialized manually, $S = \phi$

Until stopping criterion met

for example x_i

$$S = S \cup (x_i, \arg\max p_\theta(d \mid x_i))$$

Compute statistics of S

$S_{\text{conf}} \leftarrow$find confident subset

[Goldwasser et al., 2011]

Unsupervised learning

Intuition: assume repeating patterns are correct

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

Output: θ

θ initialized manually, $S = \phi$

Until stopping criterion met

for example x_i

$$S = S \cup (x_i, \arg\max p_\theta(d \mid x_i))$$

Compute statistics of S

$S_{\text{conf}} \leftarrow$find confident subset

Train on S_{conf}

[Goldwasser et al., 2011]

Unsupervised learning

Intuition: assume repeating patterns are correct

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

Output: θ

θ initialized manually, $S = \phi$

Until stopping criterion met

for example x_i

$$S = S \cup (x_i, \arg\max p_\theta(d \mid x_i))$$

Compute statistics of S

$S_{\text{conf}} \leftarrow$find confident subset

Train on S_{conf}

Substantially lower performance

Paraphrasing

What languages do people in Brazil use?

Paraphrasing

What languages do people in Brazil use?

Type.HumanLanguage $\sqcap$ LanguagesSpoken.Brazil ... CapitalOf.Brazil

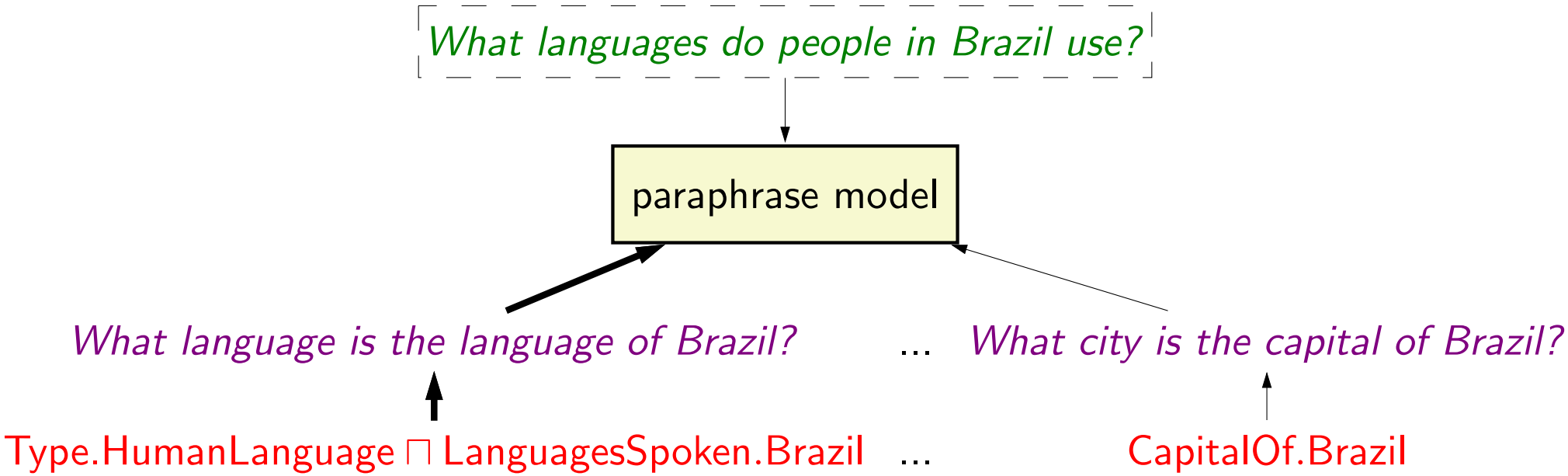
Paraphrasing

What languages do people in Brazil use?

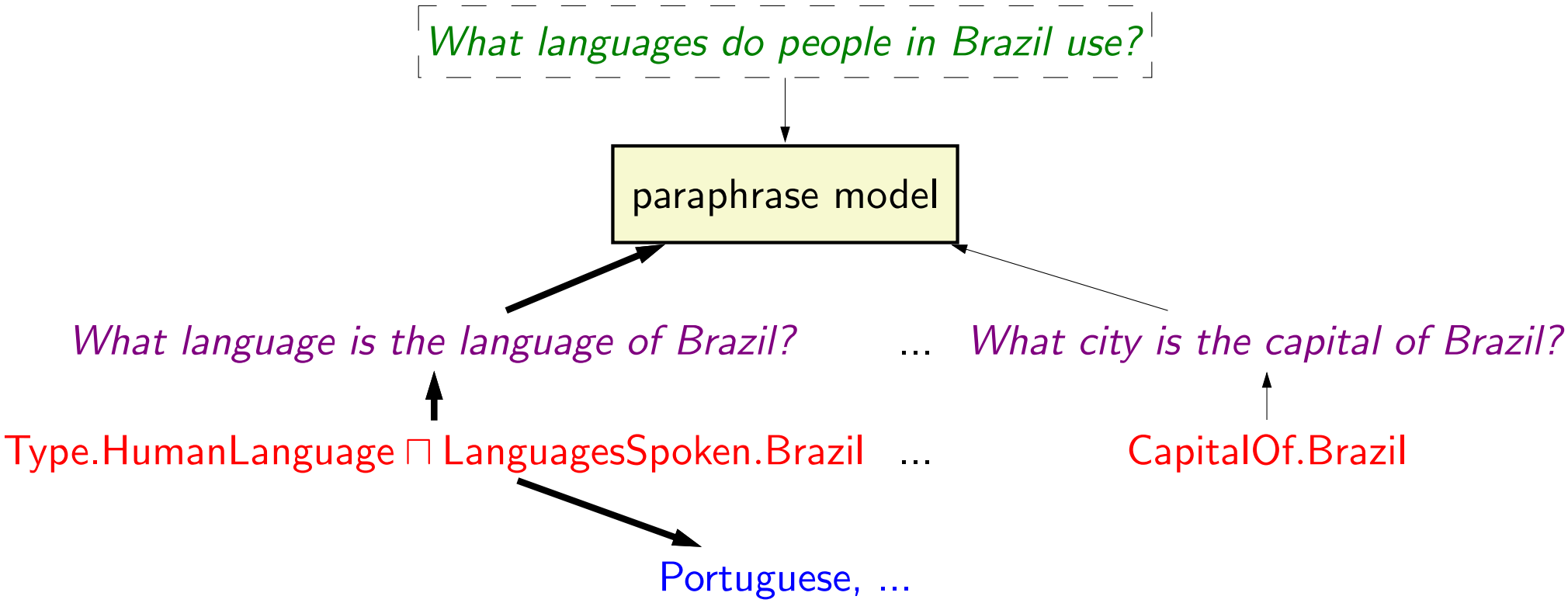
What language is the language of Brazil? ... *What city is the capital of Brazil?*

Type.HumanLanguage ⊓ LanguagesSpoken.Brazil ... CapitalOf.Brazil

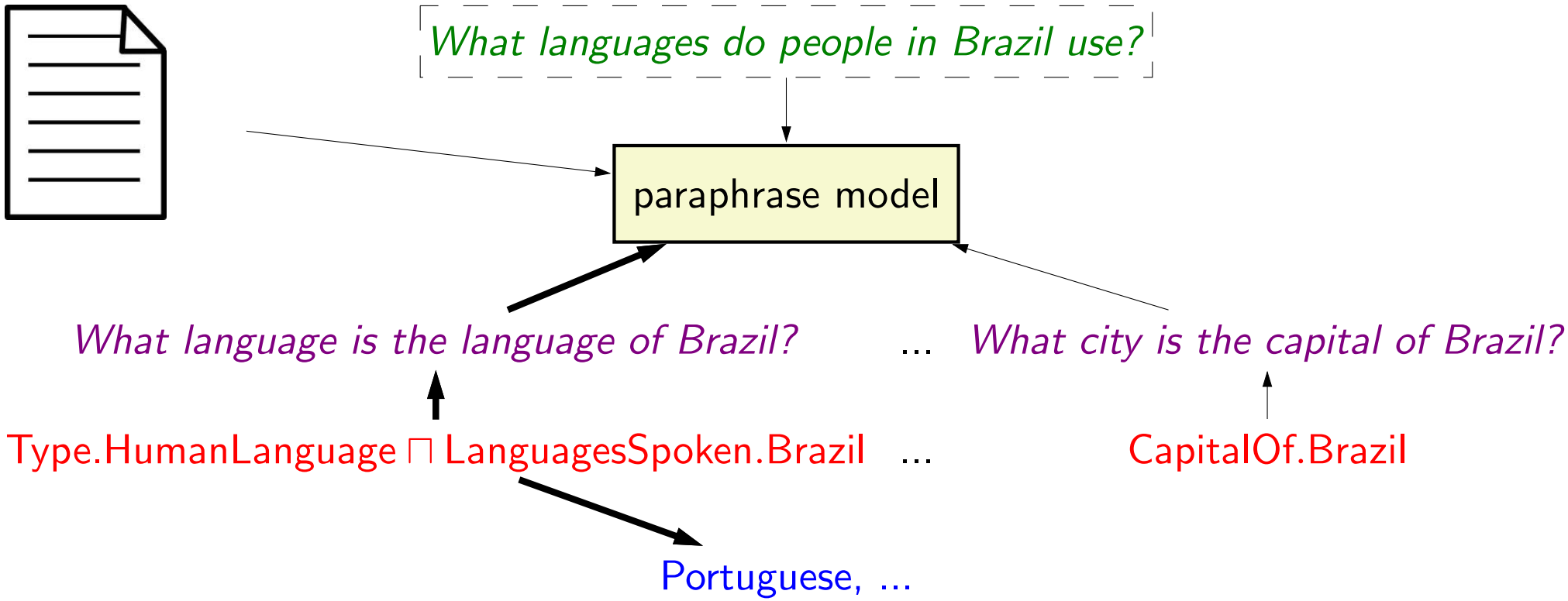
Paraphrasing



Paraphrasing



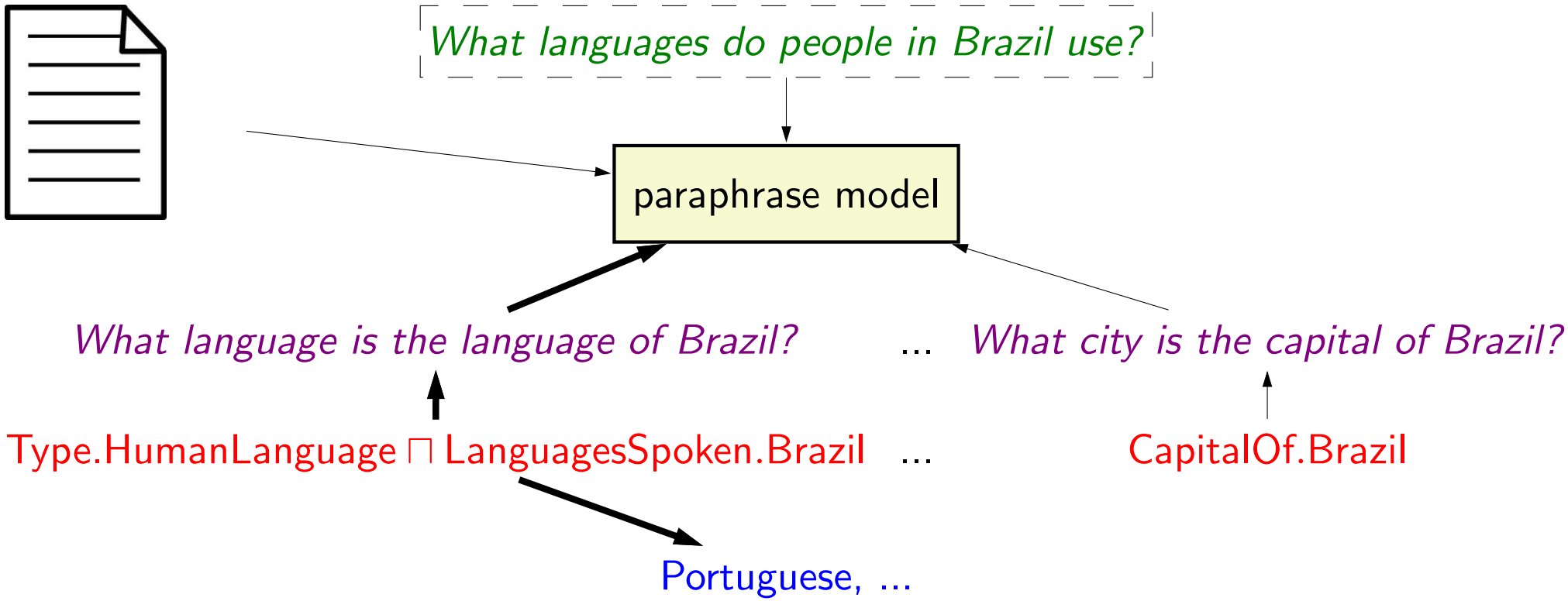
Paraphrasing



Model: $p_\theta(c, z \mid x) = p(z \mid x) \times p_\theta(c \mid x)$

Idea: train a large paraphrase model $p_\theta(c \mid x)$

Paraphrasing



Model: $p_\theta(c, z \mid x) = p(z \mid x) \times p_\theta(c \mid x)$

Idea: train a large paraphrase model $p_\theta(c \mid x)$

More later

Summary

We saw how to train from denotations and logical forms

We see methods for inducing lexicons during training

We reviewed work on how to use even weaker forms of supervision

Still an open problem