Carnegie Mellon University
School of Computer Science

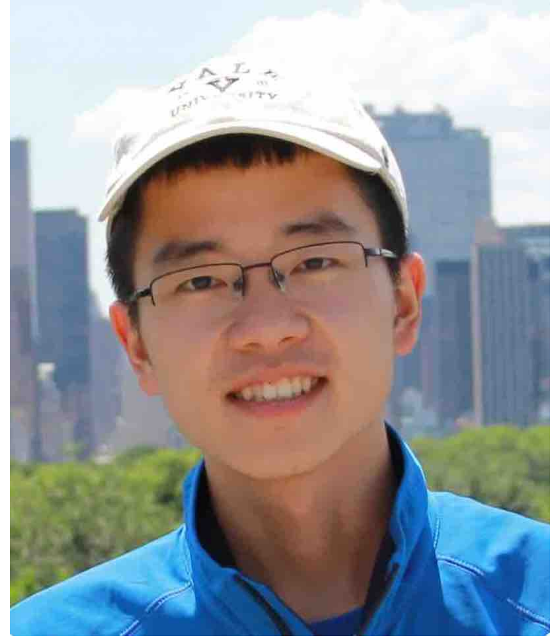
Language Technologies Institute

# Learning with Latent Linguistic Structure

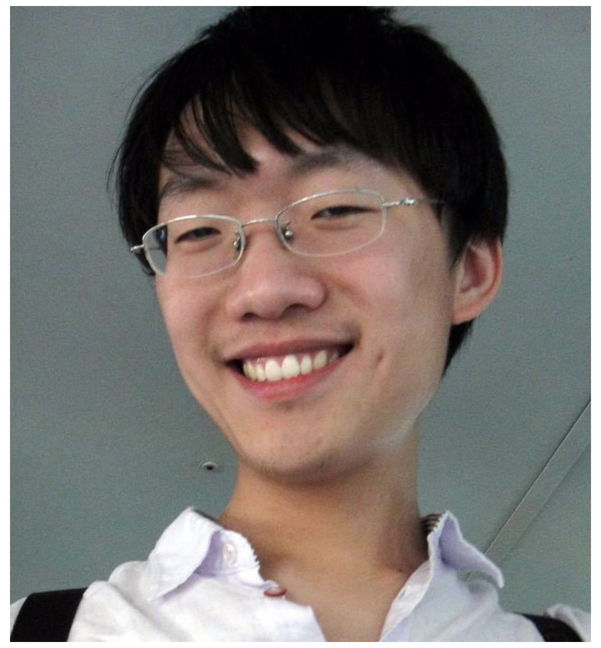
Graham Neubig

@ BlackBoxNLP 11/1/2018

with: Junxian He



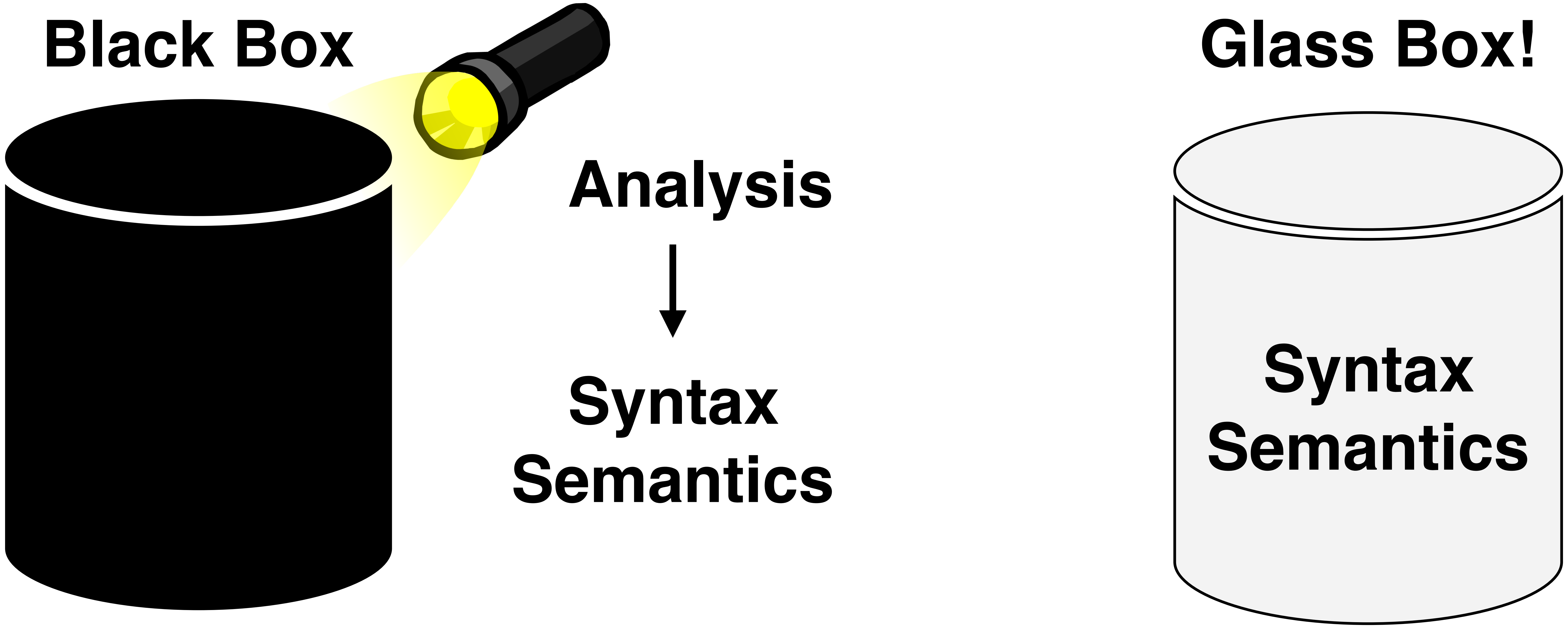
Pengcheng Yin



Chunting Zhou

Taylor Berg-Kirkpatrick

# How to Achieve Interpretability in Neural Nets?



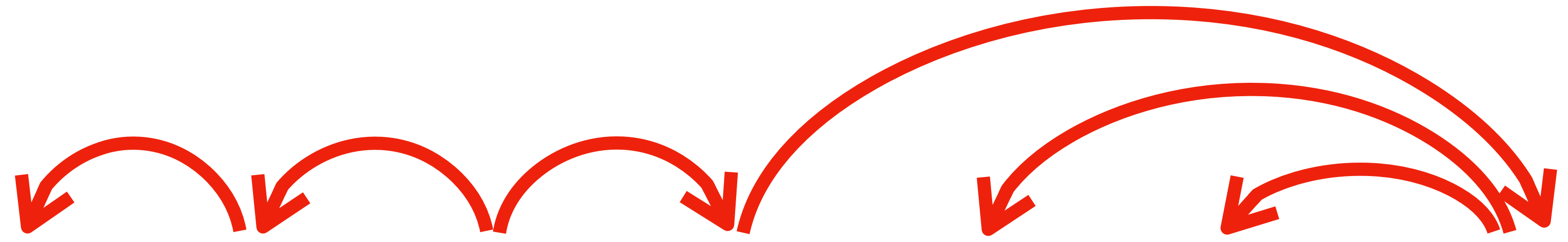
# Research Problems

- Fundamentally highly interpretable models (e.g. discrete HMMs) are not sufficiently powerful
- How can we harness the power of neural networks, with underlying interpretable representations?
- How can we learn them on unlabeled data?

Language Technologies Institute

# e.g. Syntactic Analysis

Dependency:

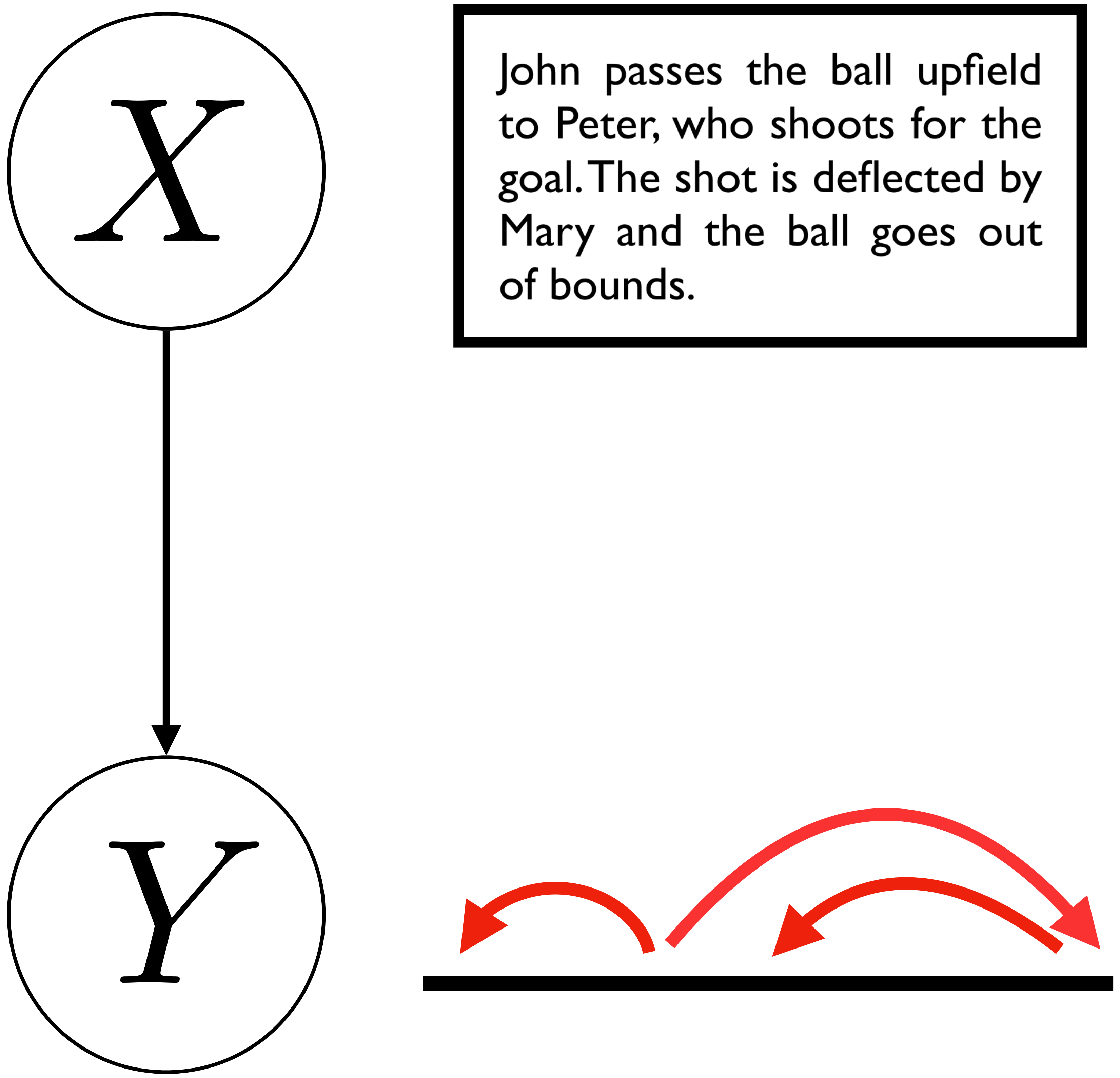


Parts-of-speech: DT NN VBD IN DT JJ NN

The cat sat on a green wall

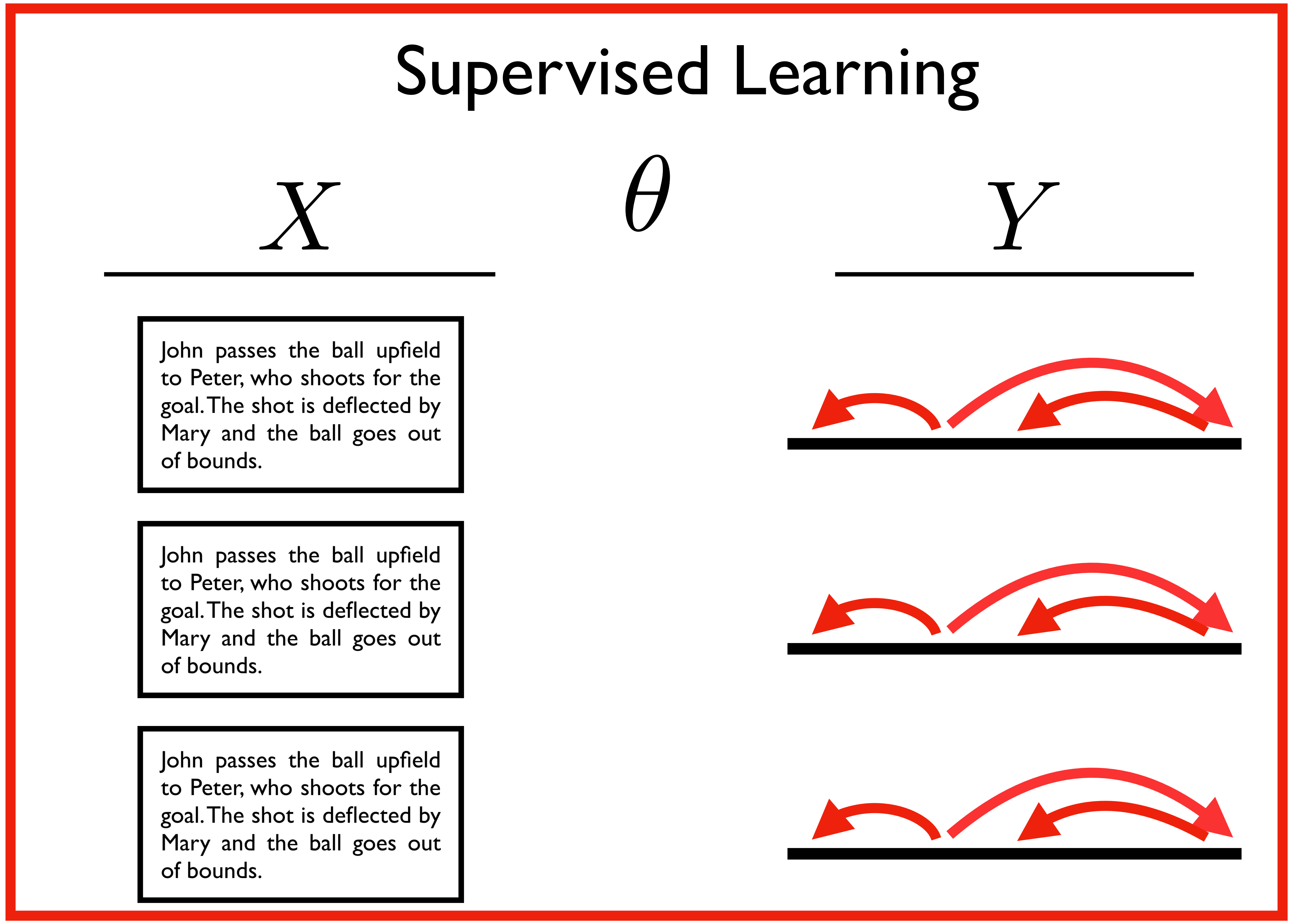
Language Technologies Institute

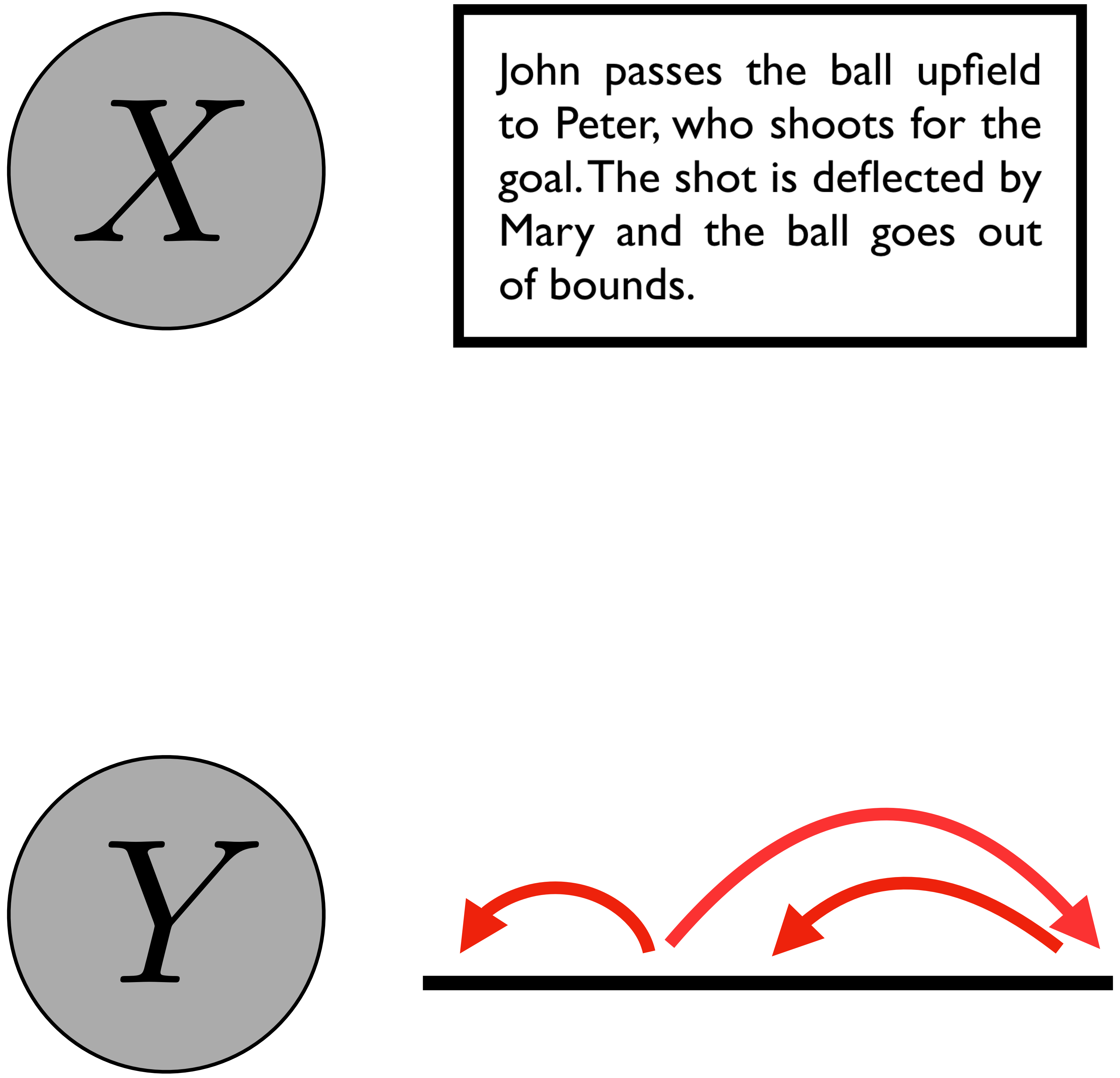
# Supervised Approach



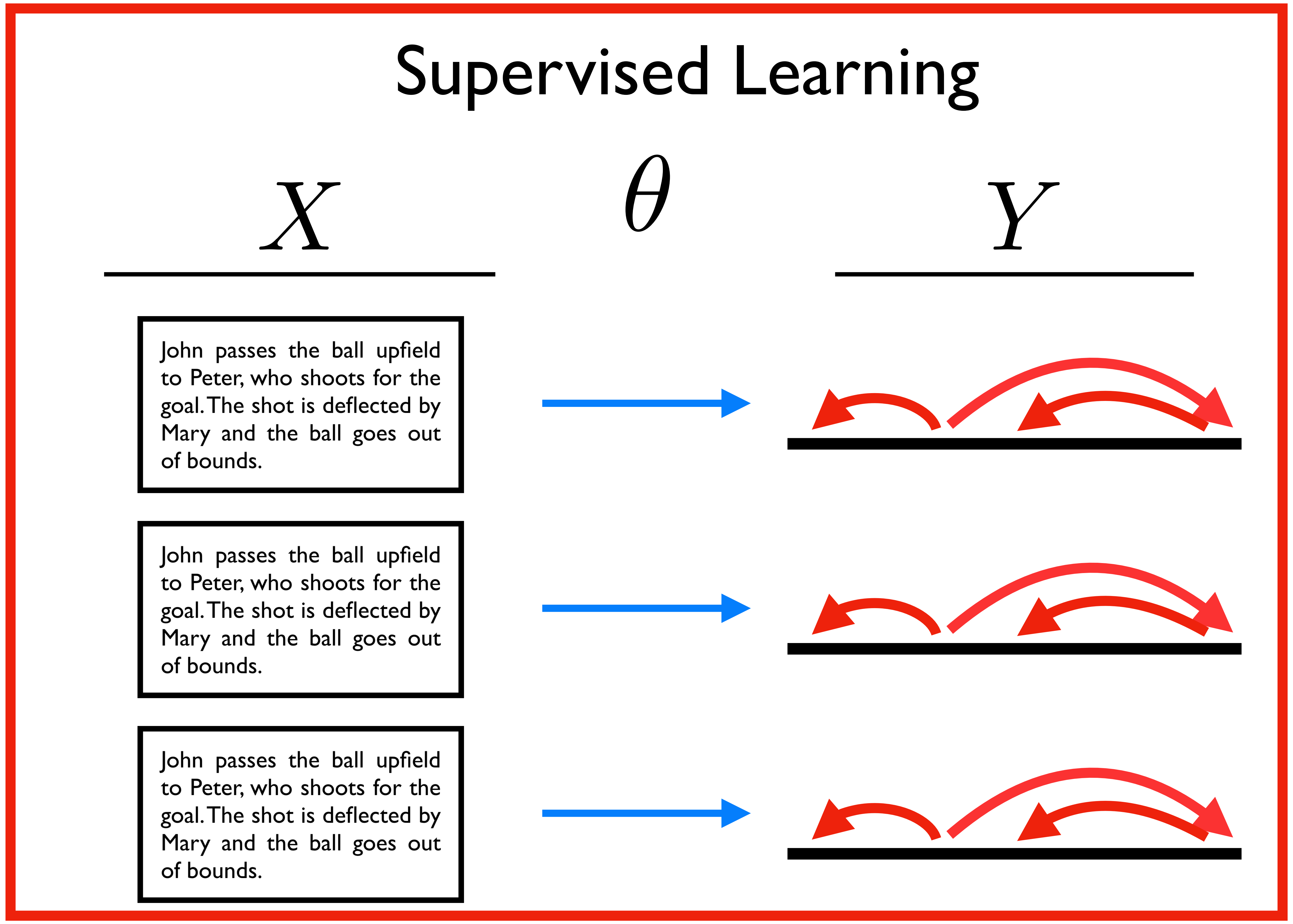
Language
Technologies
Institute

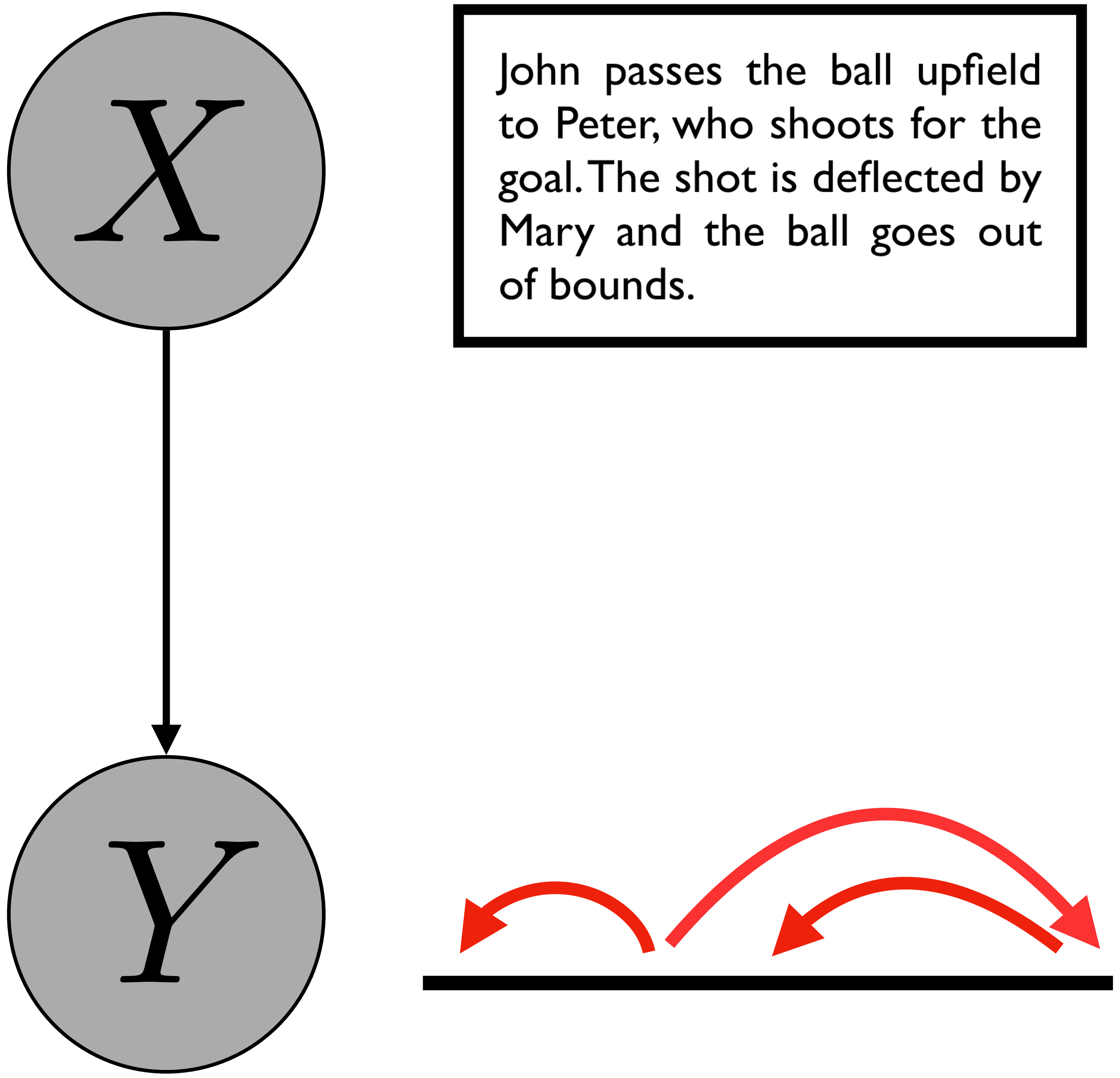
# Supervised Approach



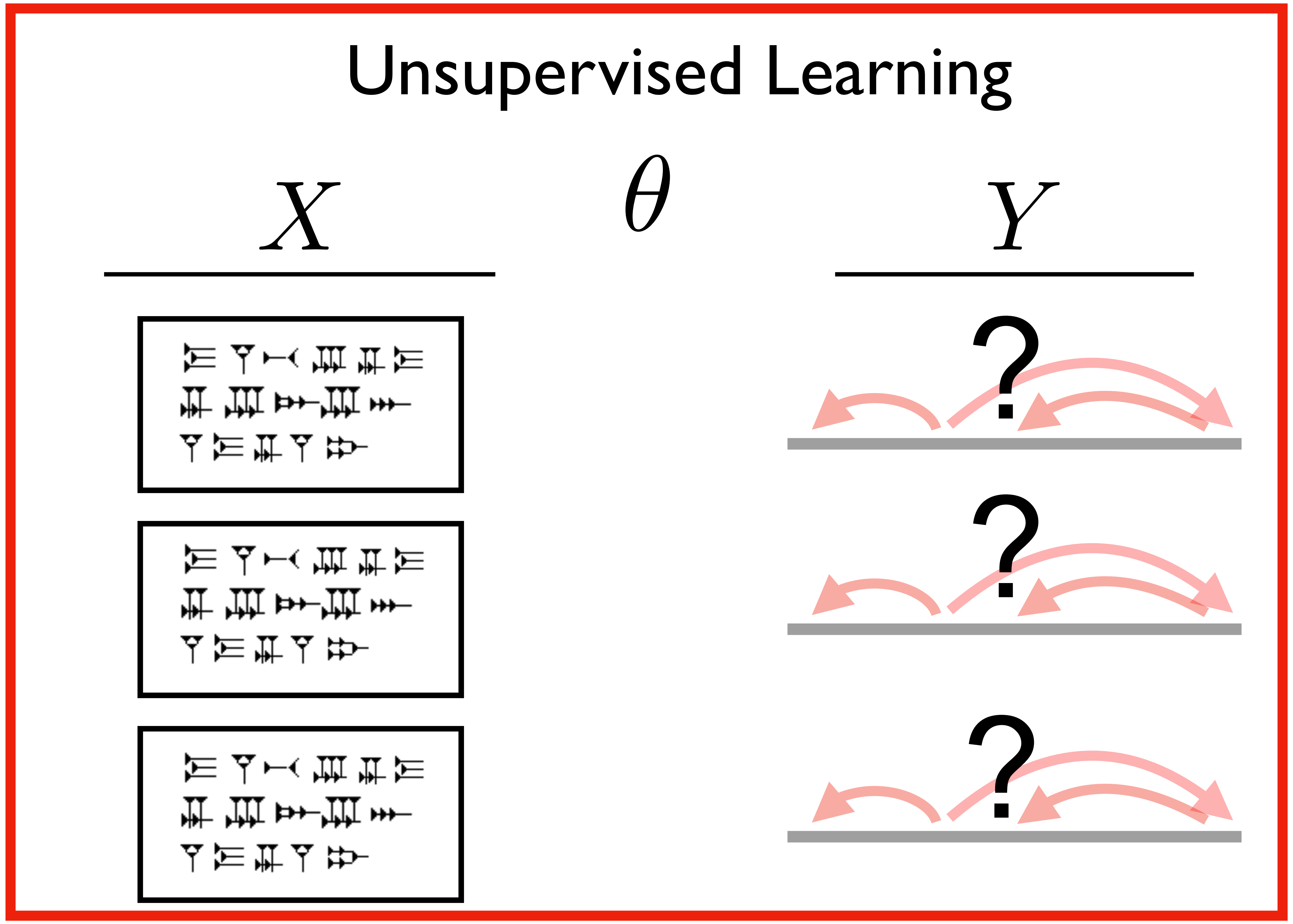
# Supervised Approach

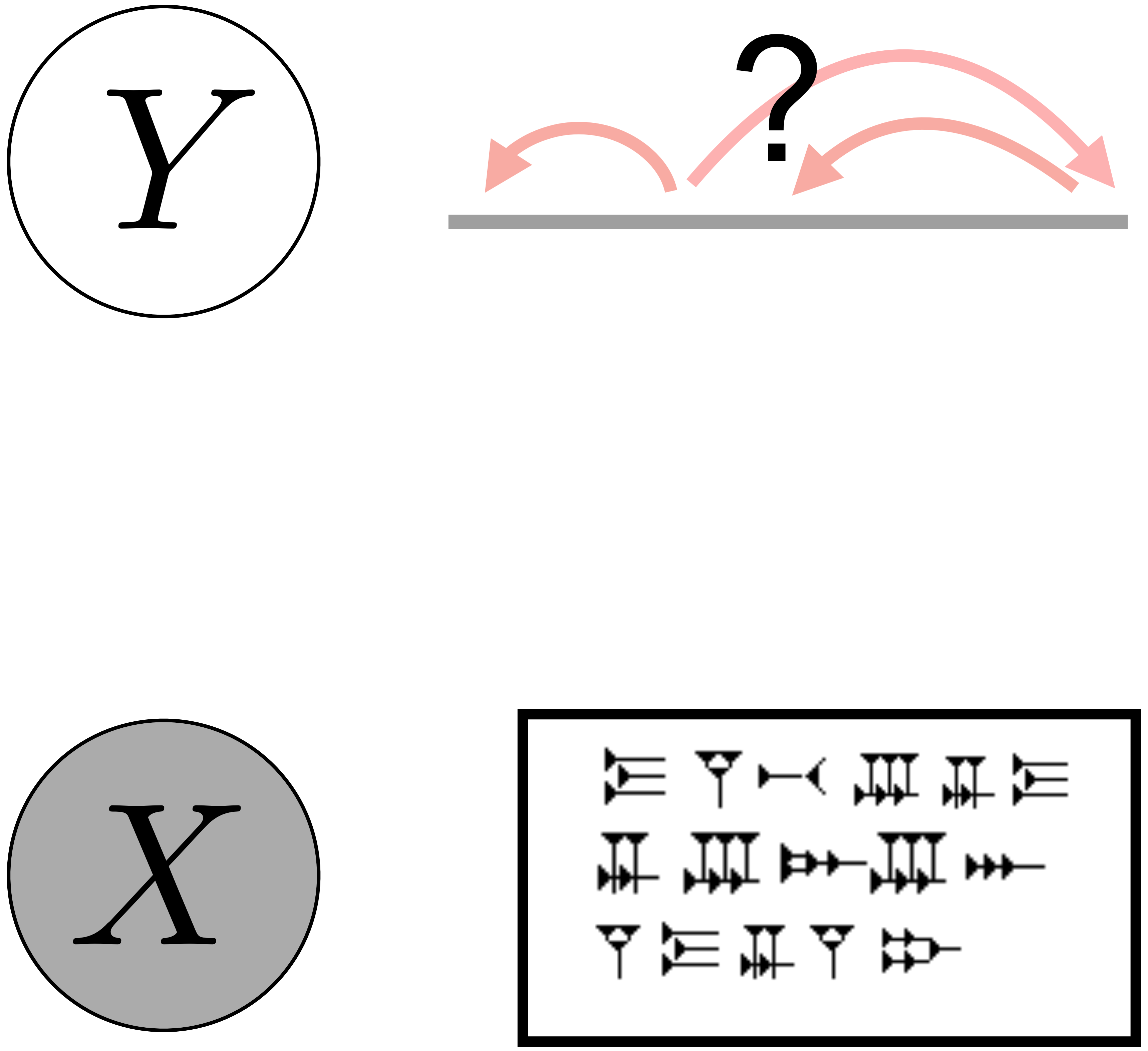


Language Technologies Institute

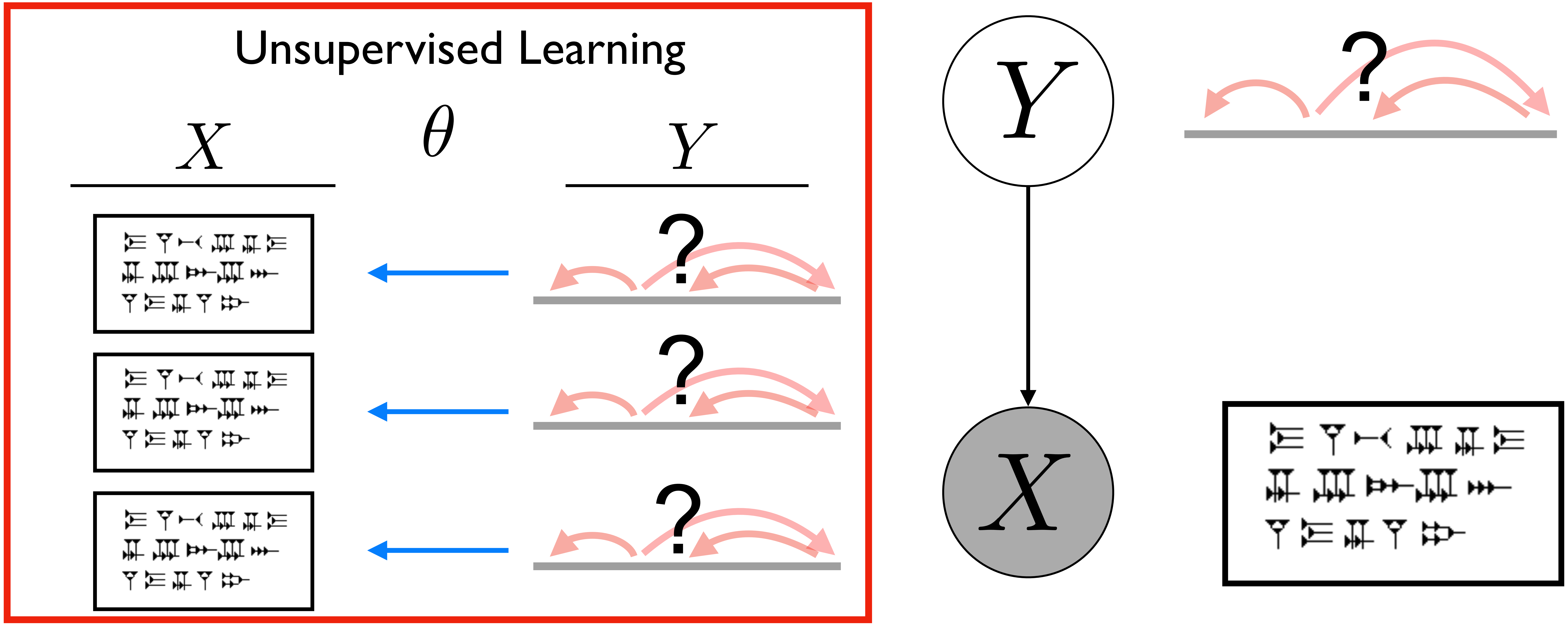
# Latent Variable Approach



Language Technologies Institute

# Latent Variable Approach



Carnegie Mellon University
School of Computer Science

Language
Technologies
Institute

# Multi-space Variational Encoder-Decoders

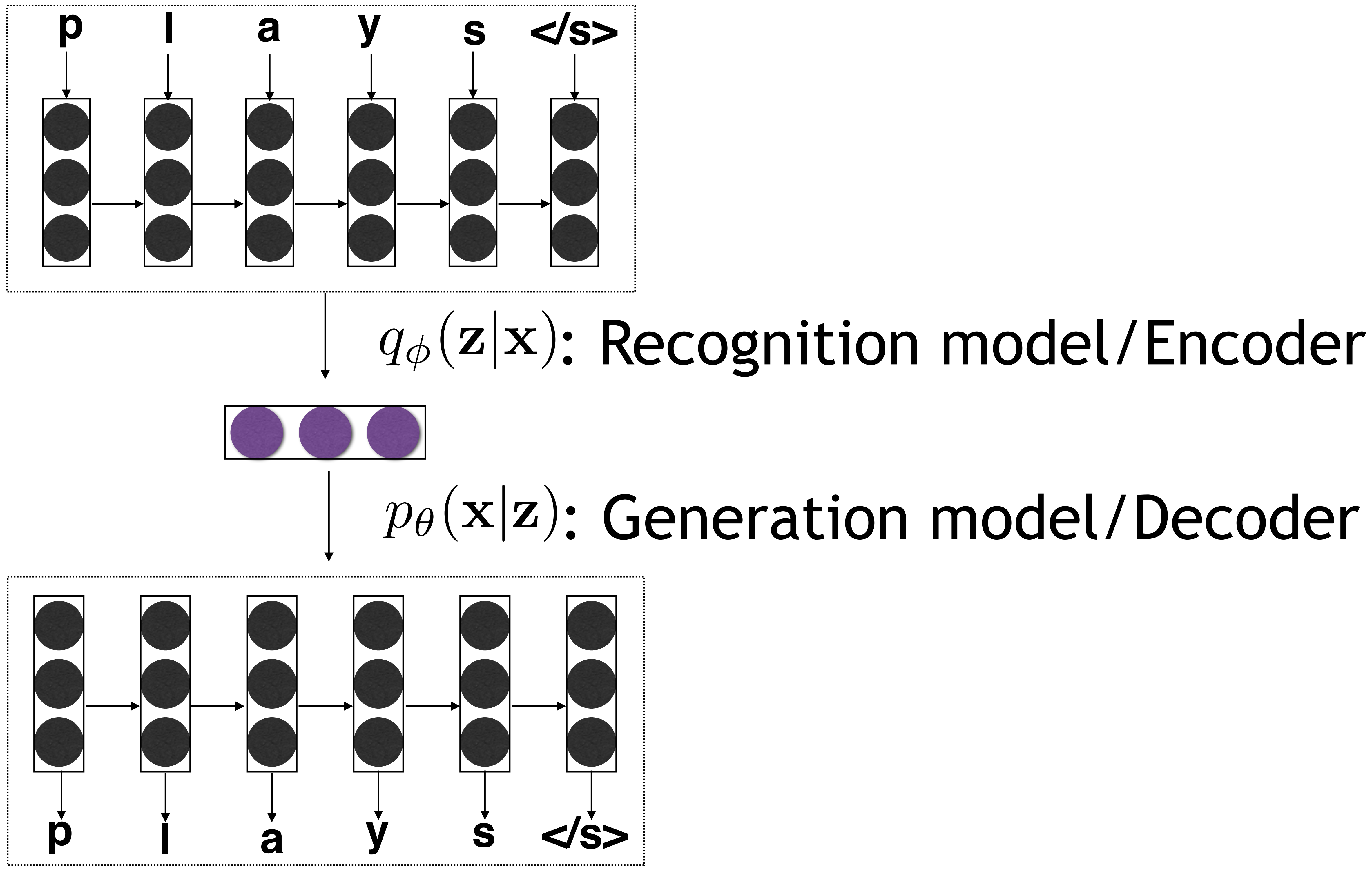
Chunting Zhou and Graham Neubig

(ACL 2017)

# Features of Words

- **Syntax:**
  - What syntactic features does the word have?
  - Closed-class, generally enumerable for a specific language.
- **Meaning/Symbol:**
  - What is the meaning of the word, how is it spelled/pronounced?
  - Open-class, complicated regularities and relationships.
- Can we create a model that elegantly models both?

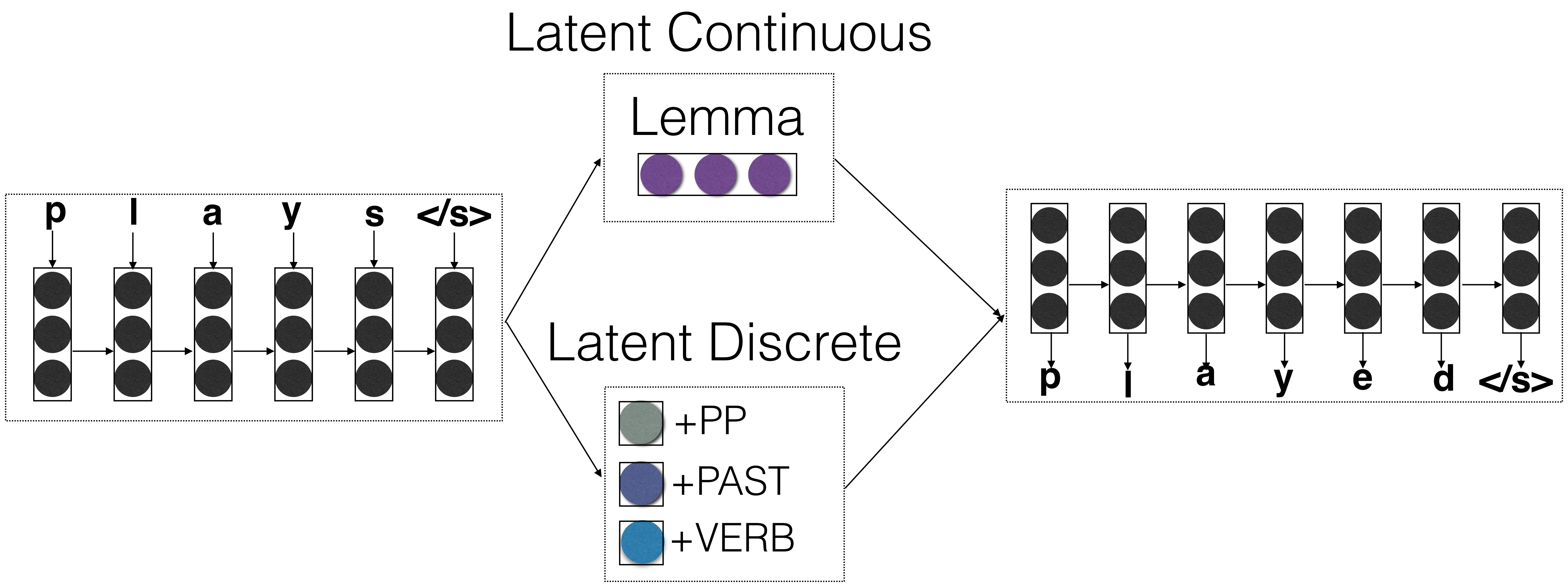
# Background: Variational Auto-encoder (Kingma et al., 2014, Bowman et al., 2016)




**Maximize the Variational lower bound:**

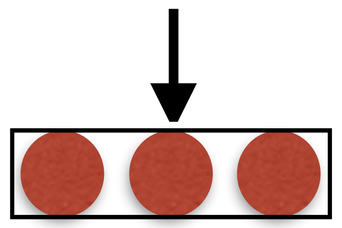
$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

# Proposed Model: Multi-space Variational Encoder-Decoders




- Modeling complicated higher-level structure (e.g. meaning or symbol of the word): incorporation of continuous latent variables
- Modeling closed-class and interpretable features (e.g. syntax): incorporation of discrete latent variables

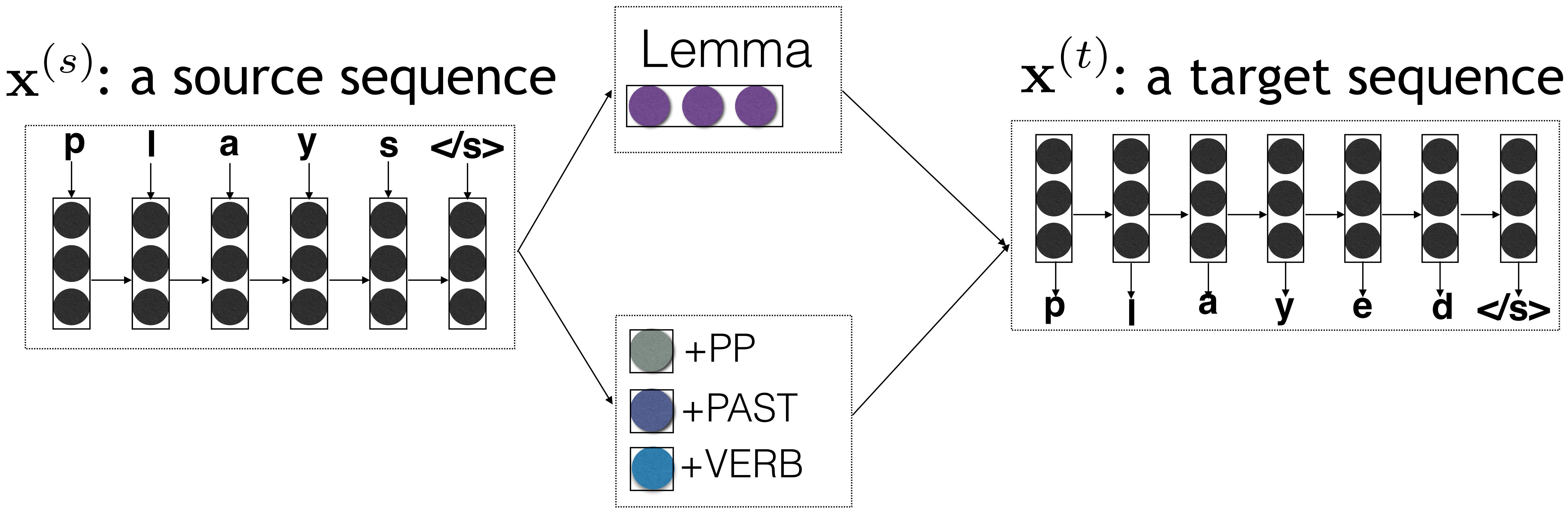
plays, played, playing



- How can we learn in a un- or semi-supervised way?

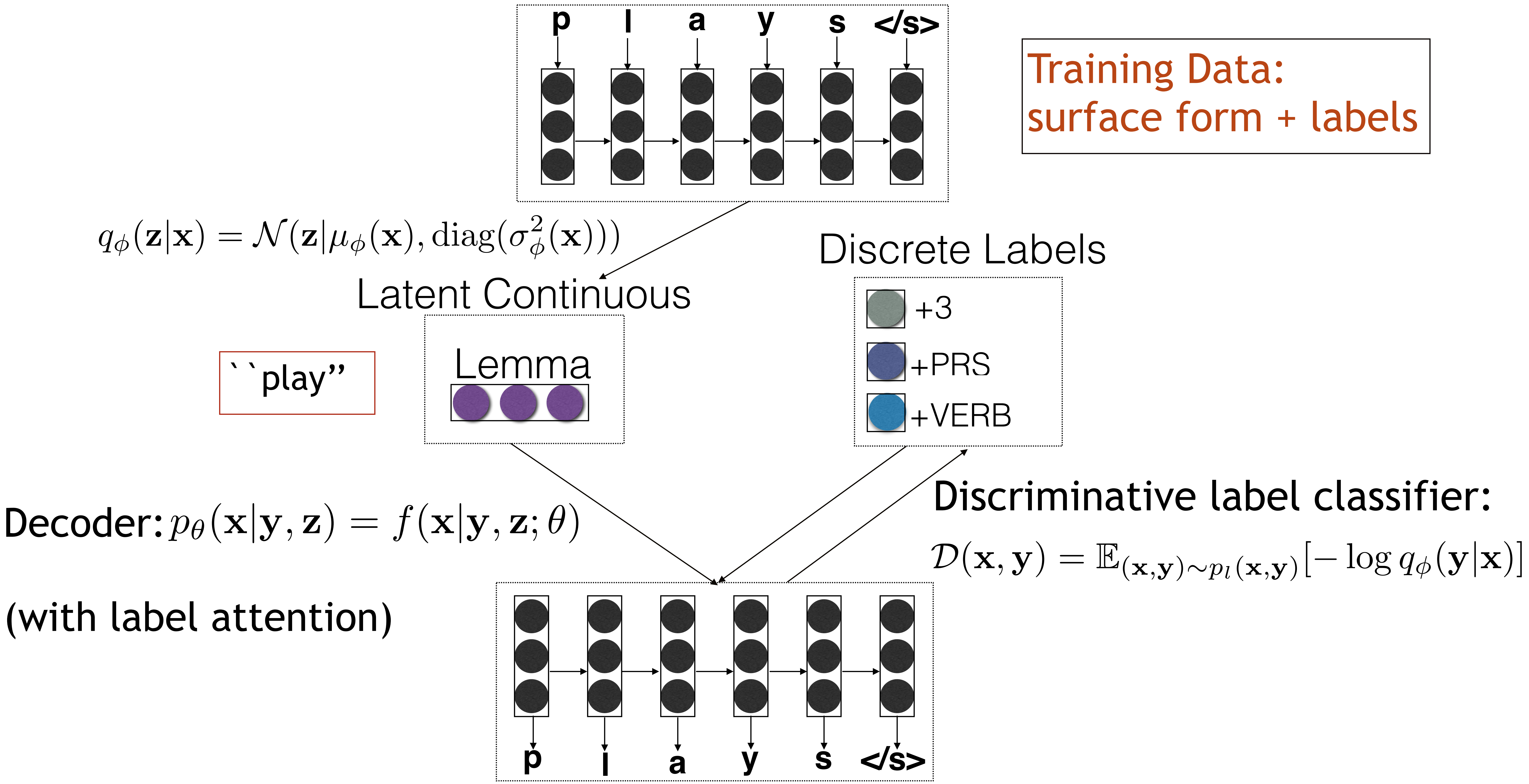
# Variable Definitions

$\mathbf{z}$ :continuous latent variable



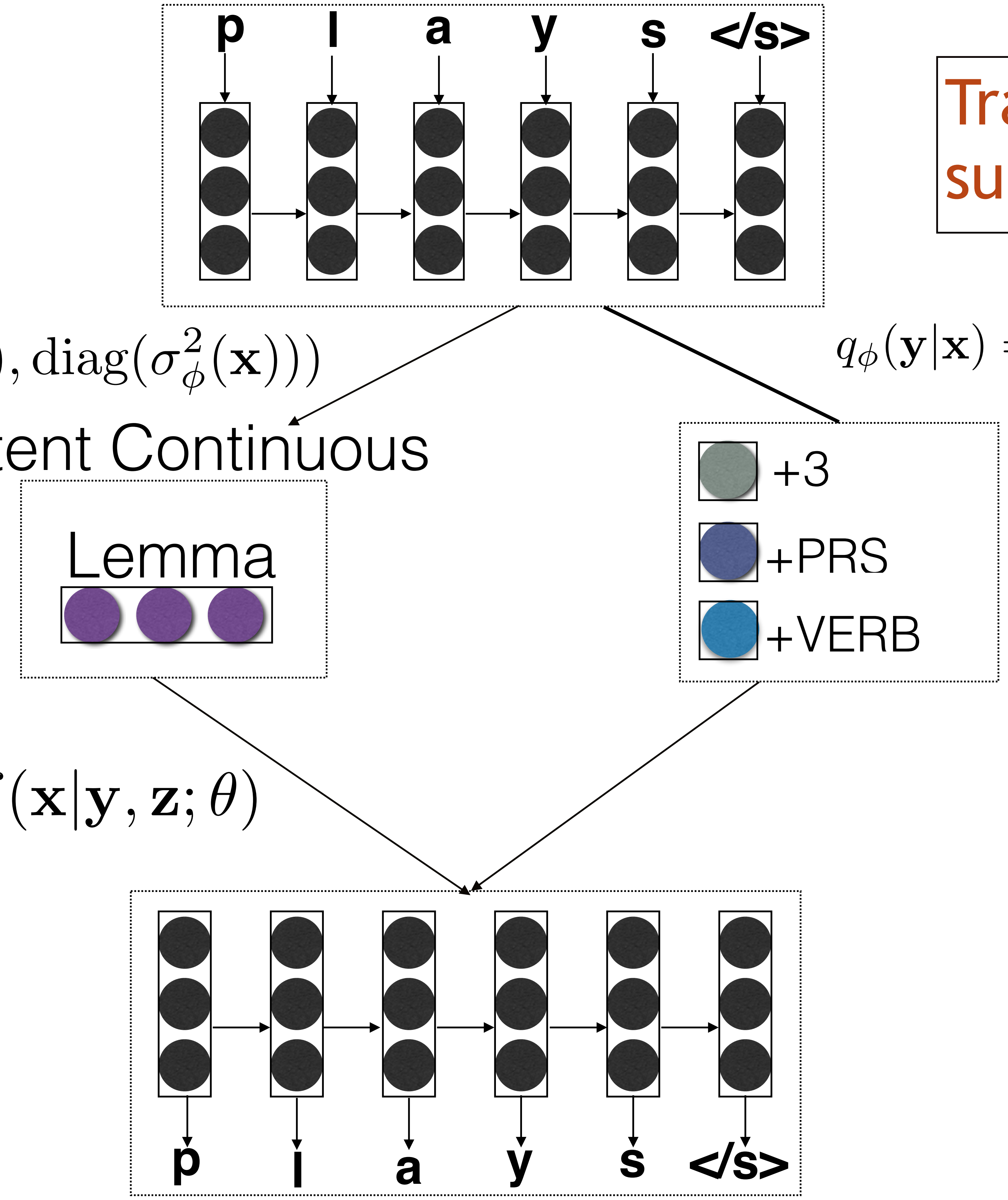

$\mathbf{y}^{(t)} = [y_1^{(t)}, y_2^{(t)}, \cdots y_K^{(t)}]$ :discrete labels for each target sequence

# Supervised Learning: Labeled Multi-space Variational Autoencoders




$$\text{Maximize}: \mathcal{U}(\mathbf{x}) = \text{Variational Lower Bound of } \log p(\mathbf{x}, \mathbf{y})$$

# Unsupervised Learning:
# Unlabeled Multi-space Variational Auto-encoders




Training Data:
surface form only

$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mu_\phi(\mathbf{x}), \mathrm{diag}(\sigma^2_\phi(\mathbf{x})))$

$q_\phi(\mathbf{y}|\mathbf{x}) = \prod_k q_\phi(y_k|\mathbf{x}) = \prod_k \mathrm{Cat}(y_k|\pi_\phi(\mathbf{x}))$

``play''

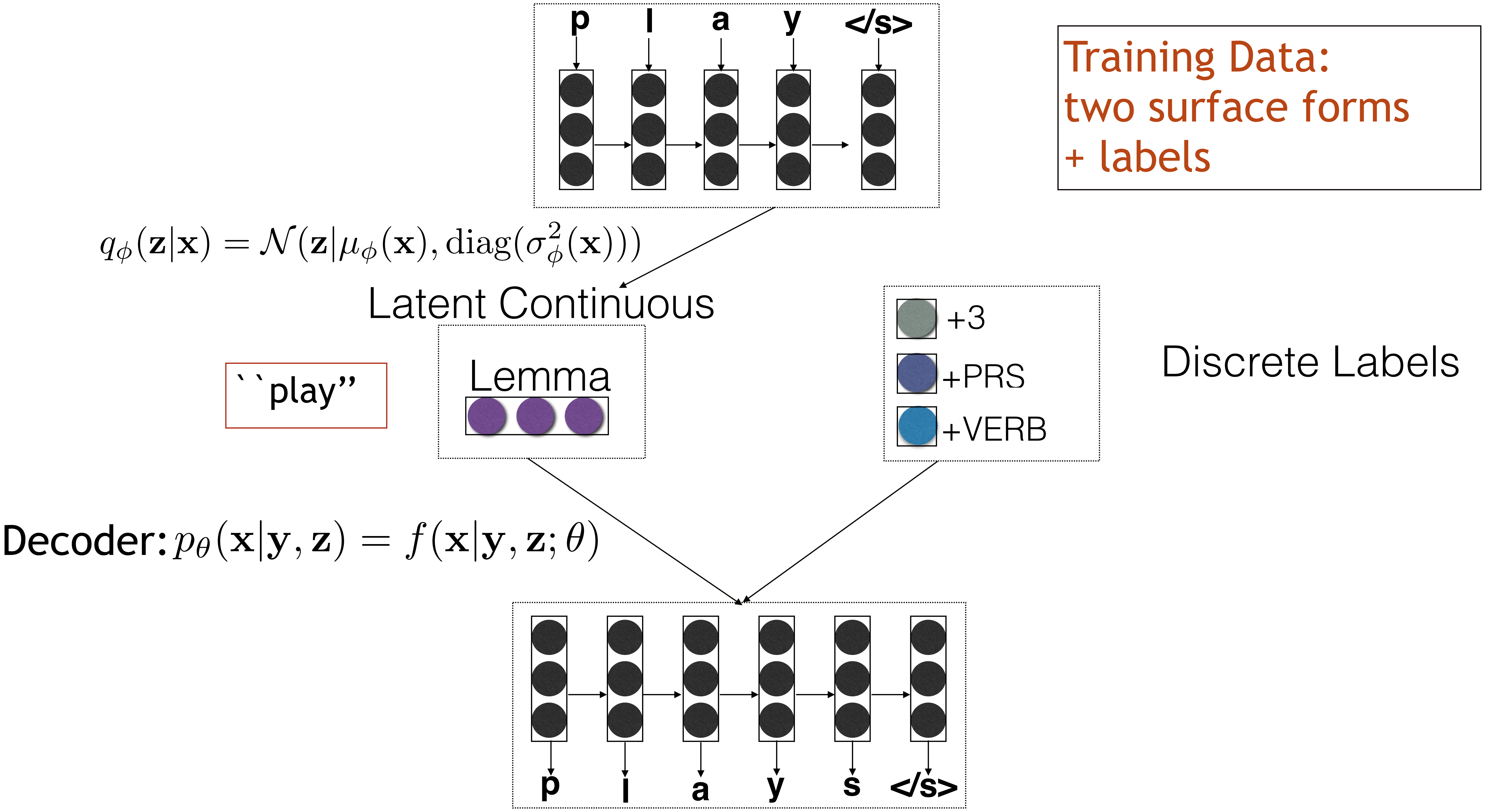
Latent Discrete

Decoder: $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) = f(\mathbf{x}|\mathbf{y}, \mathbf{z}; \theta)$

$\text{Maximize} : \mathcal{U}(\mathbf{x}) = \text{Variational Lower Bound of } \log p(\mathbf{x}, \mathbf{y})$

Language Technologies Institute

# Labeled Sequence-to-sequence Training: Multi-space Variational Encoder-Decoders




Training Data:
two surface forms
+ labels

$$\text{Maximize} : \mathcal{L}_l(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}|\mathbf{x}^{(s)}) = \text{Variational Lower Bound of } \log p(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}|\mathbf{x}^{(s)})$$

# Learning MSVED

- **Learning Continuous Latent Variables:**
  Reparameterization trick (Kingma et al., 2014):

$$\epsilon \sim \mathcal{N}(0,1), \quad \mathbf{z} = \mu_\phi(x) + \sigma_\phi(x) \circ \epsilon$$

- **Learning Discrete Latent Variables:**
  Gumbel-Softmax (Maddison et al., 2017)

$$\hat{y}_{ij} = \frac{\exp((\log(\pi_{ij}) + g_{ij})/\tau)}{\sum_{k=1}^{N_i} \exp((\log(\pi_{ik}) + g_{ik})/\tau}$$

- **Training tricks** (Bowman et al. 2016):
  - KL-divergence Annealing
  - Input dropout in the decoder

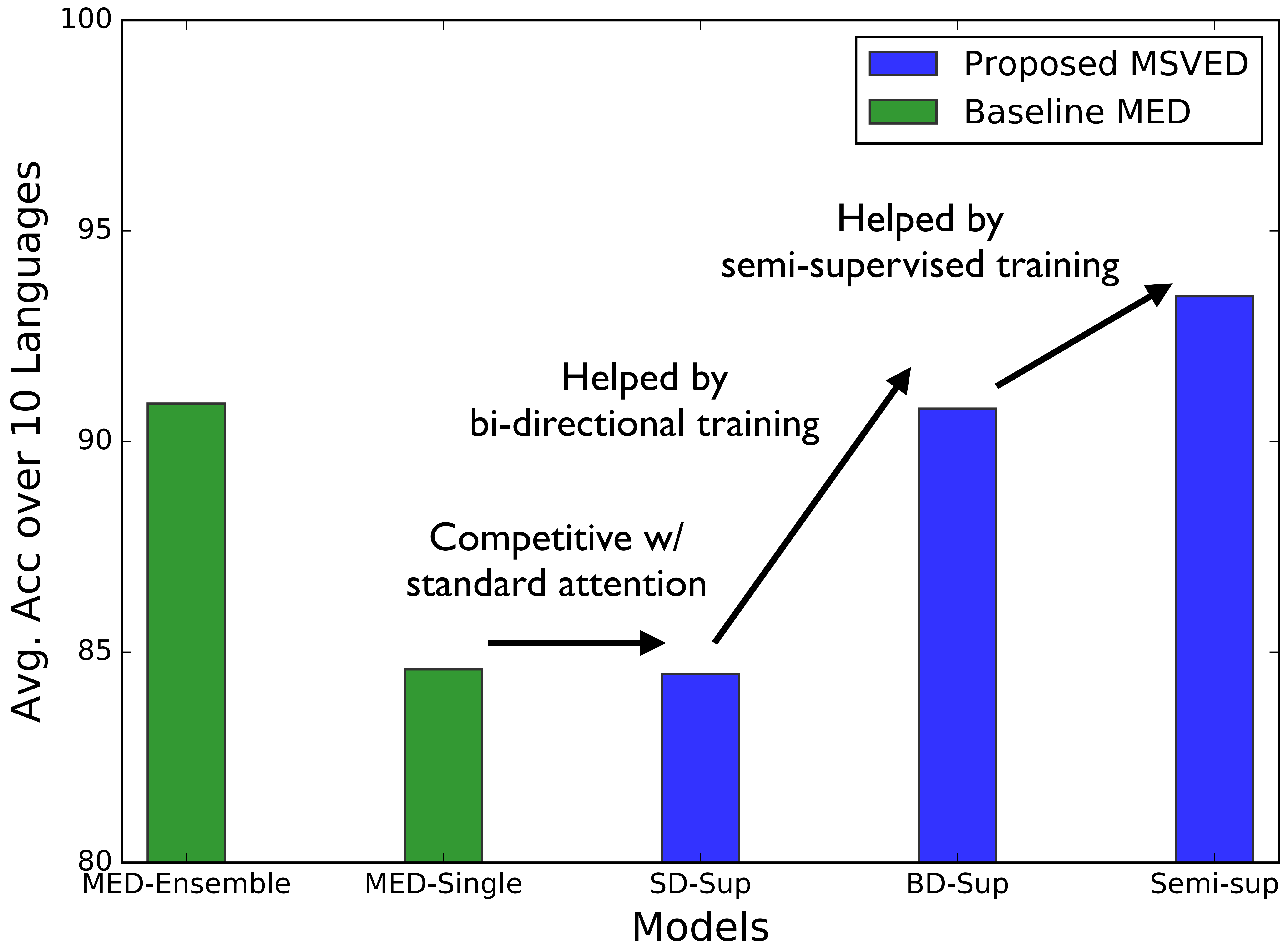
# Experimental Setup

Task: Morphology re-inflection

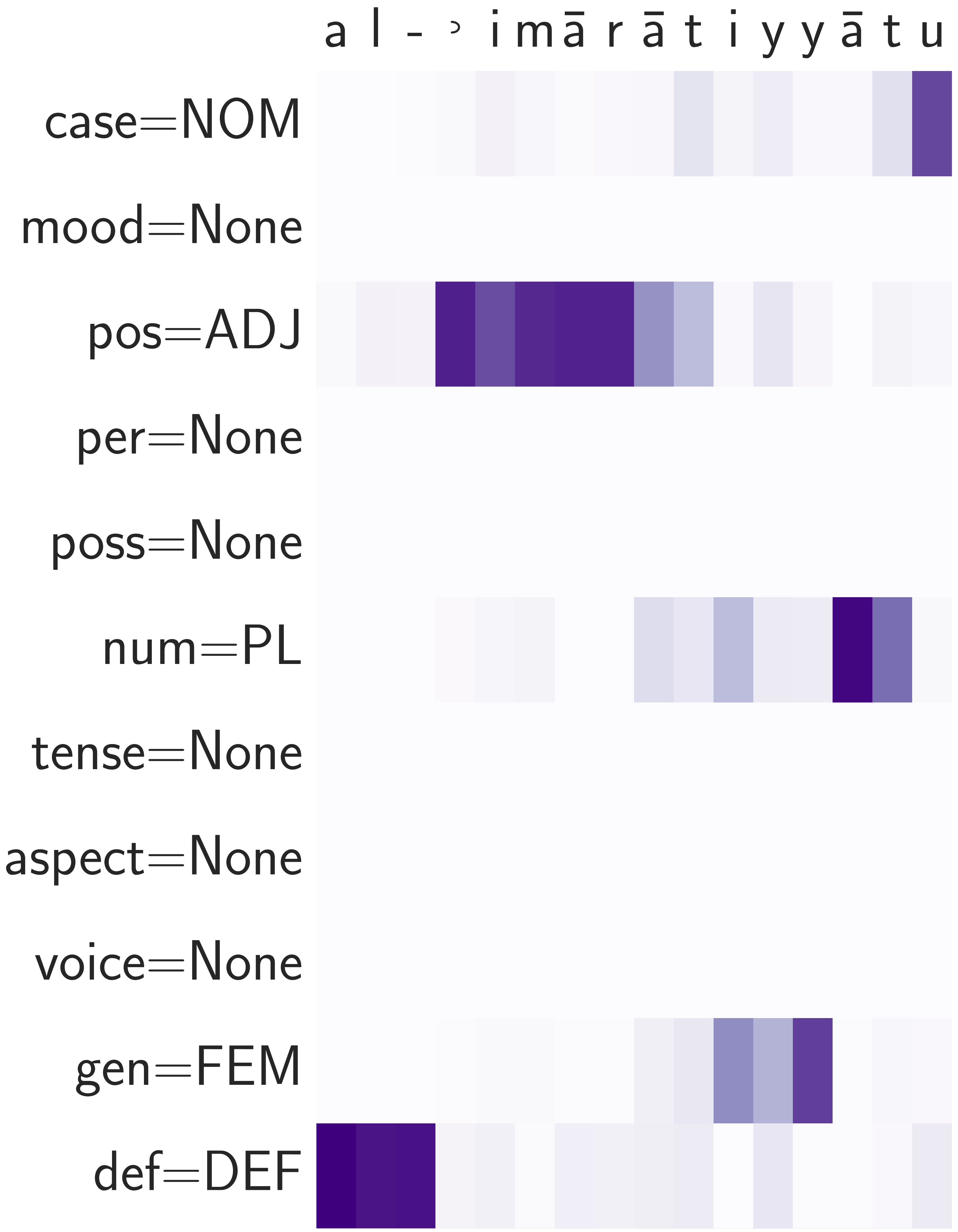
Dataset: SIGMORPHON 2016 task 3
source word: communicated
target word: communicates
target labels: V;3;SG;PRS

Language: Turkish, Arabic, Maltese, Finnish, Spanish, German, Hungarian, Navajo, Georgian, Russian

# Results and Analysis

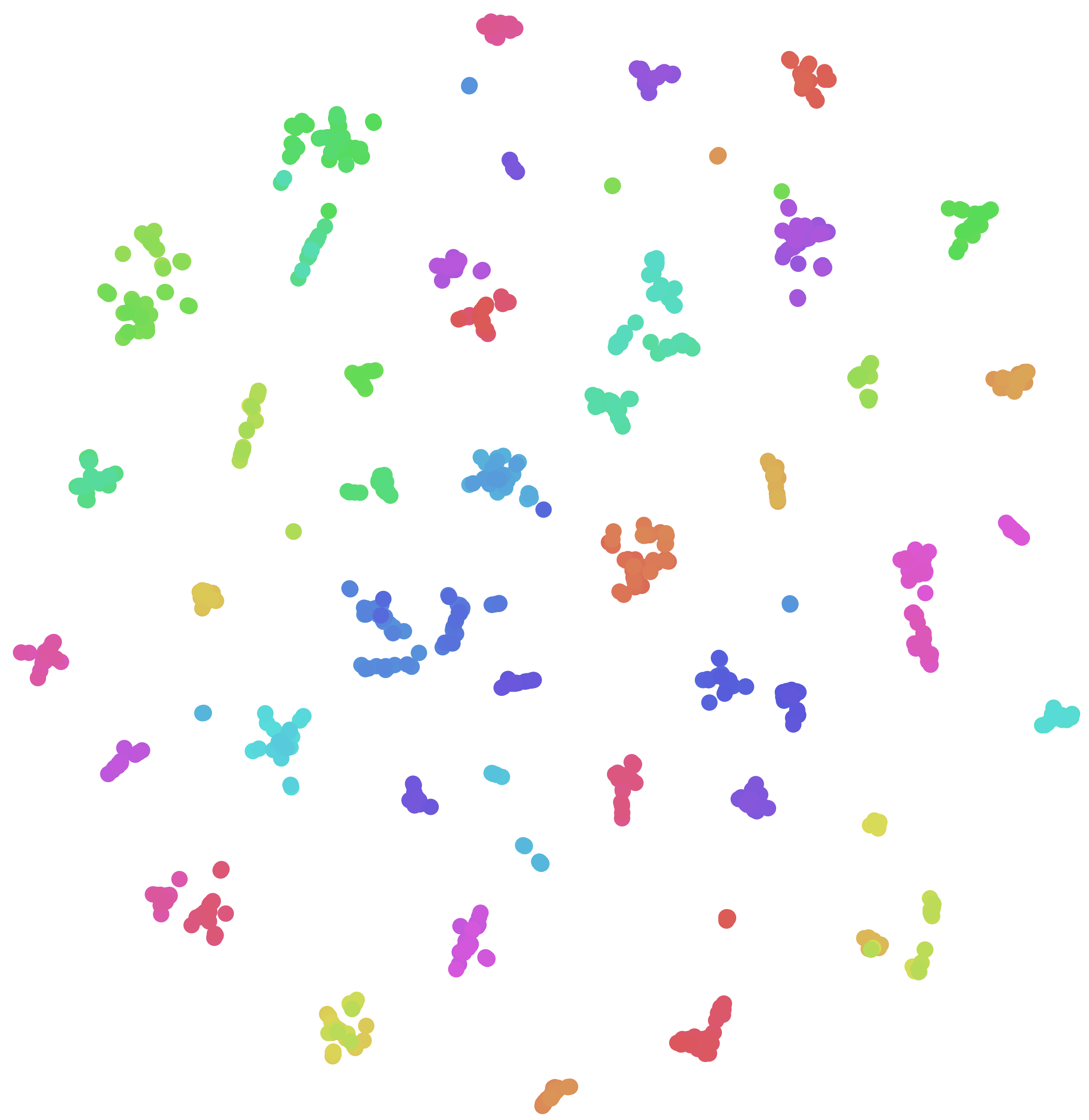


# Analysis on Tag Attention

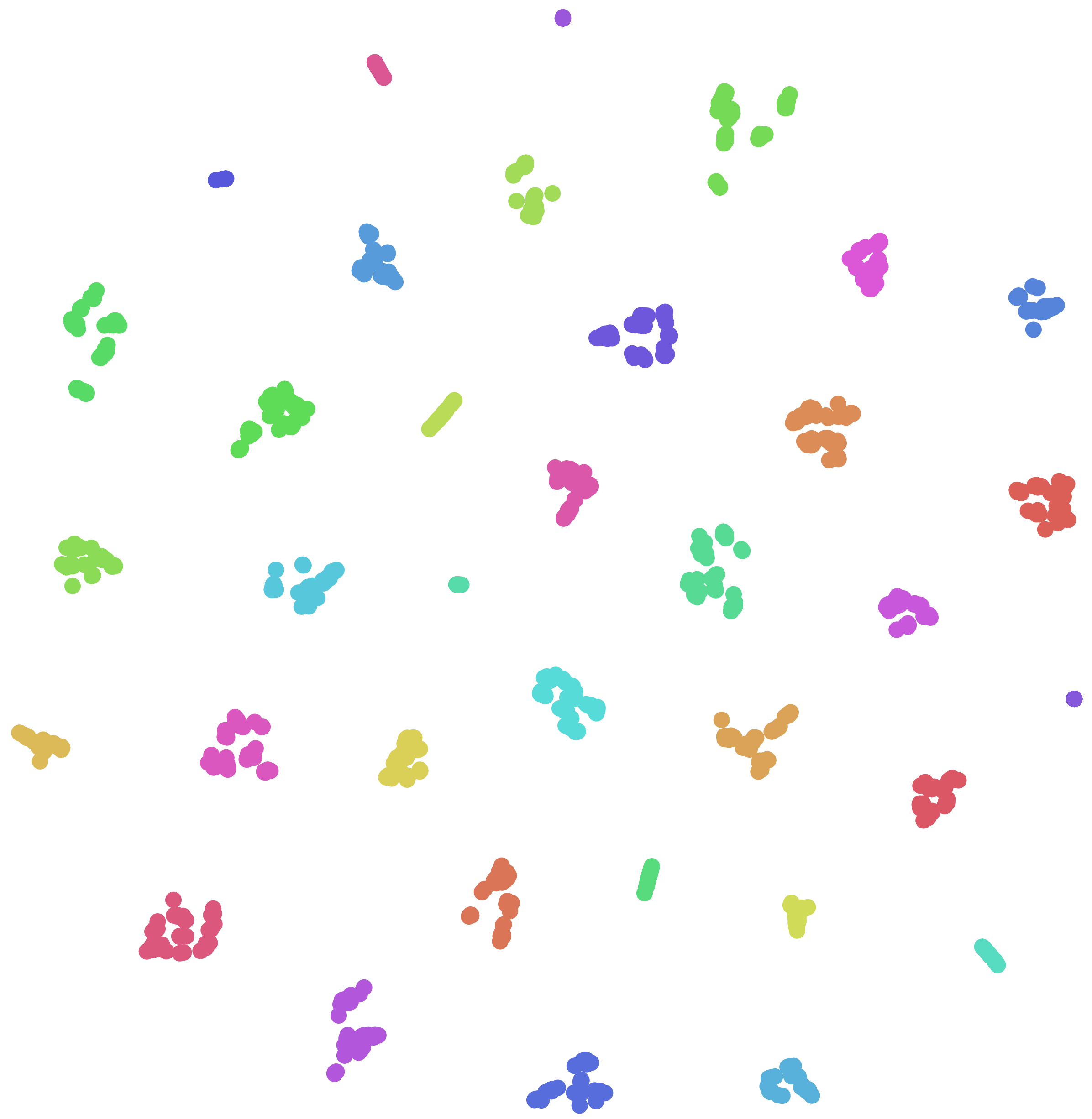


# Visualization of Latent Continuous Variables

- Clusters colored by actual lemma:



Turkish



Maltese

Carnegie Mellon University
School of Computer Science

# StructVAE: Tree-structured Latent Variable Models for Semi-supervised Semantic Parsing

Pengcheng Yin, Chunting Zhou, Junxian He, Graham Neubig

(ACL 2018)

Language Technologies Institute

NEULAB

# What About More Complicated Structure?

**Semantic Parsing:** Transducing natural language utterances (e.g., queries) into machine-executable formal meaning representations (e.g., logical form, source code)

## Domain-Specific Meaning Representations

*Show me flights from Pittsburgh to Washington*

```
lambda $0 e (and (flight $0)
     (from $0
san_Francisco:ci)
     (to $0 washington:ci))
```

lambda-calculus logical form

## General-Purpose Programming Languages

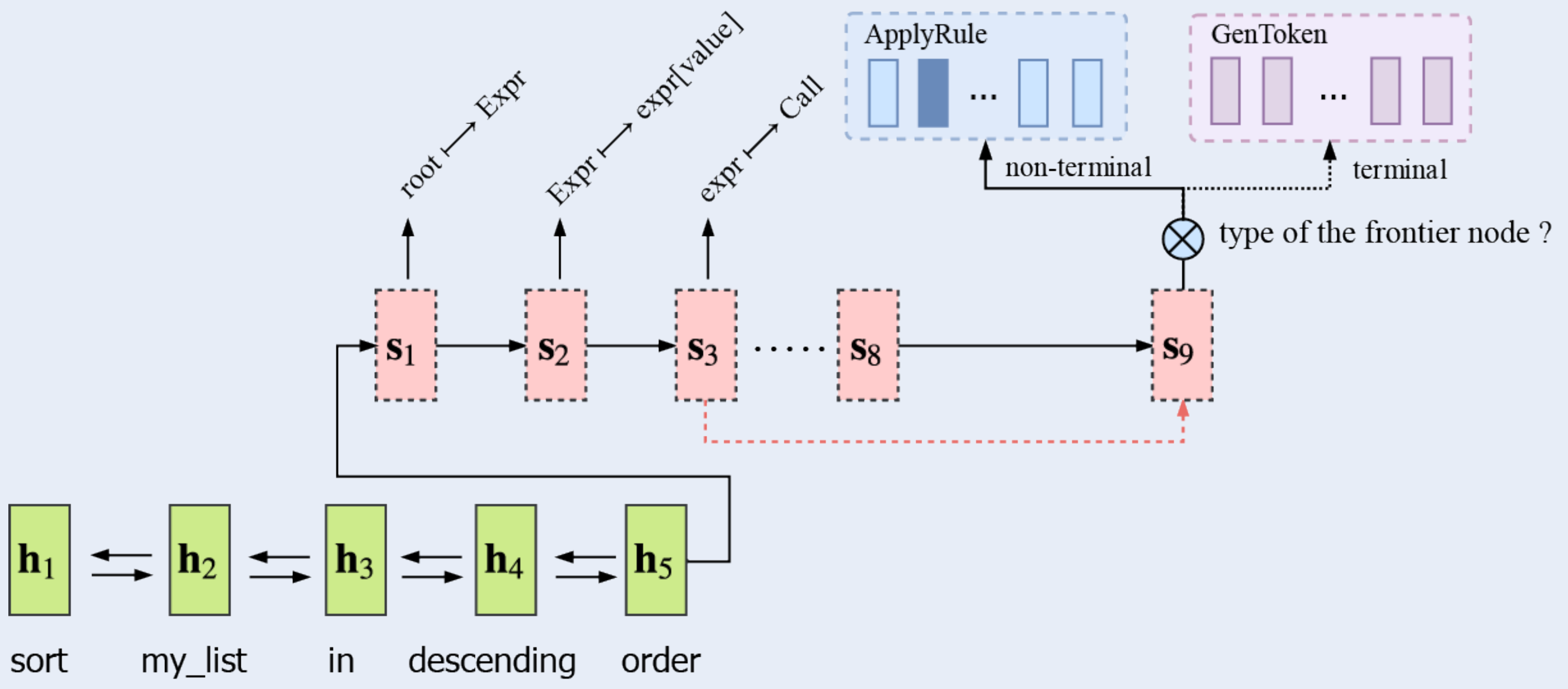
*Sort my_list in descending order*

```
sorted(my_list,
reverse=True)
```

Python

# Research Issue

## Neural Models are Data Hungry




Purely supervised neural semantic parsing models require large amounts of training data

## Data Collection is Costly

*Copy the content of file 'file.txt' to file 'file2.txt'*

```
shutil.copy('file.txt','file2.txt')
```

*Get a list of words `words` of a file 'myfile'*

```
words = open('myfile').read().split()
```

*Check if all elements in list `mylist` are the same*

```
len(set(mylist)) == 1
```

Collecting parallel training data costs  and 

Language Technologies Institute

[Yin et al., 2018] 1700 USD for 3K Python code generation examples
[Berant et al., 2013] 3000 USD for 5.7K question-to-logical form examples

# Semi-supervised Semantic Parsing

## Limited Amount of Labeled Data

*Sort my_list in descending order*

```
sorted(my_list, reverse=True)
```

*Copy the content of file 'file.txt' to file 'file2.txt'*

```
shutil.copy('file.txt',
                'file2.txt')
```

*Check if all elements in list `mylist` are the same*

```
len(set(mylist)) == 1
```

+

## Extra Unlabeled Utterances

*Get a list of words `words` of a file 'myfile'*

*Convert a list of integers into a single integer*

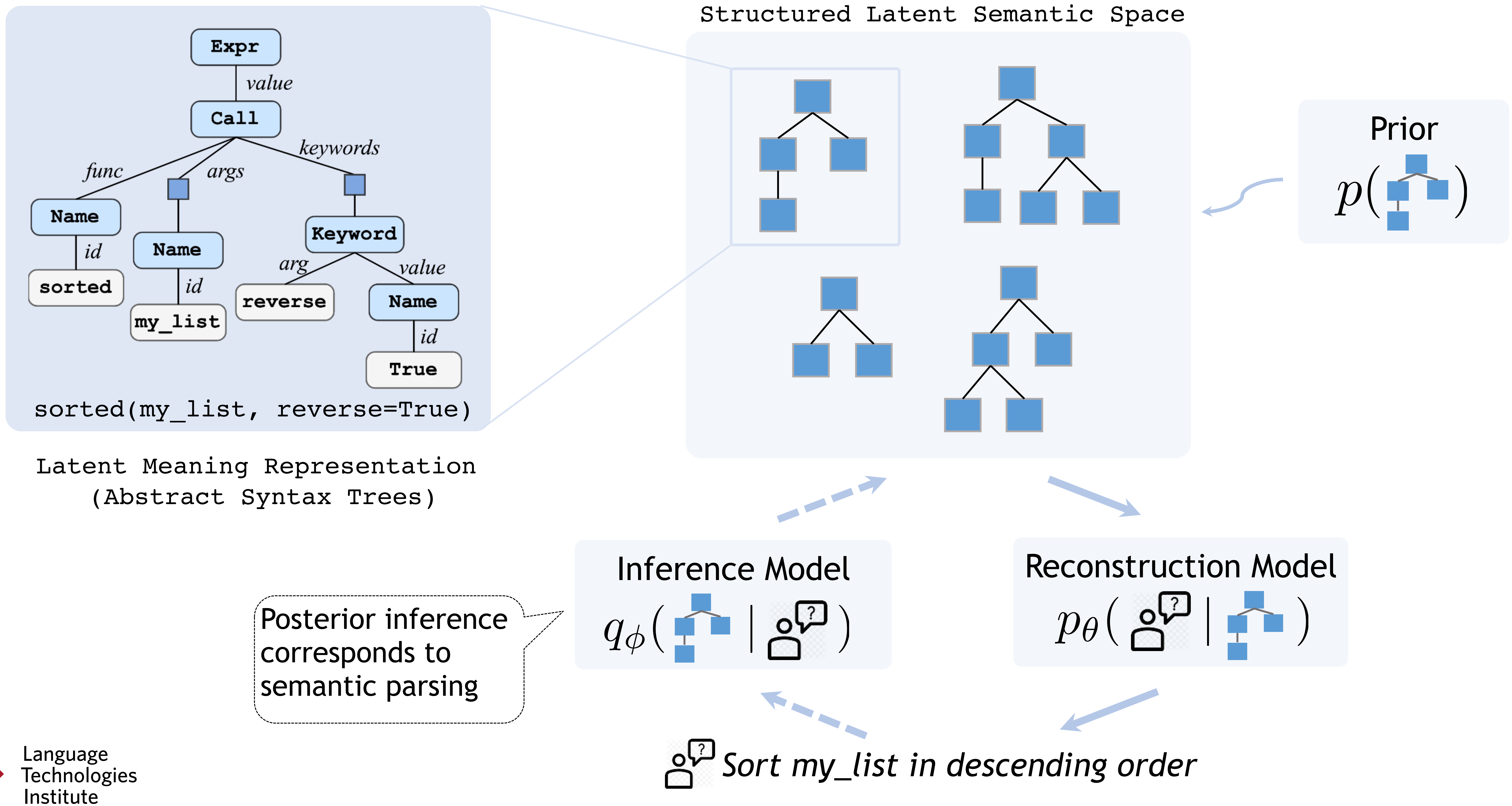
*Format a datetime object `when` to extract date only*

*Swap values in a tuple/list in list `mylist`*

*BeautifulSoup search string 'Elsie' inside tag 'a'*

*Convert string to lowercase*

# Meaning Representations as Tree-structured Latent Variables



# Semi-supervised Learning with StructVAE

Labeled Data { , }

Unlabeled Data { }

**Supervised Objective**

$\sum \log q_\phi( \cdot \mid \cdot )$

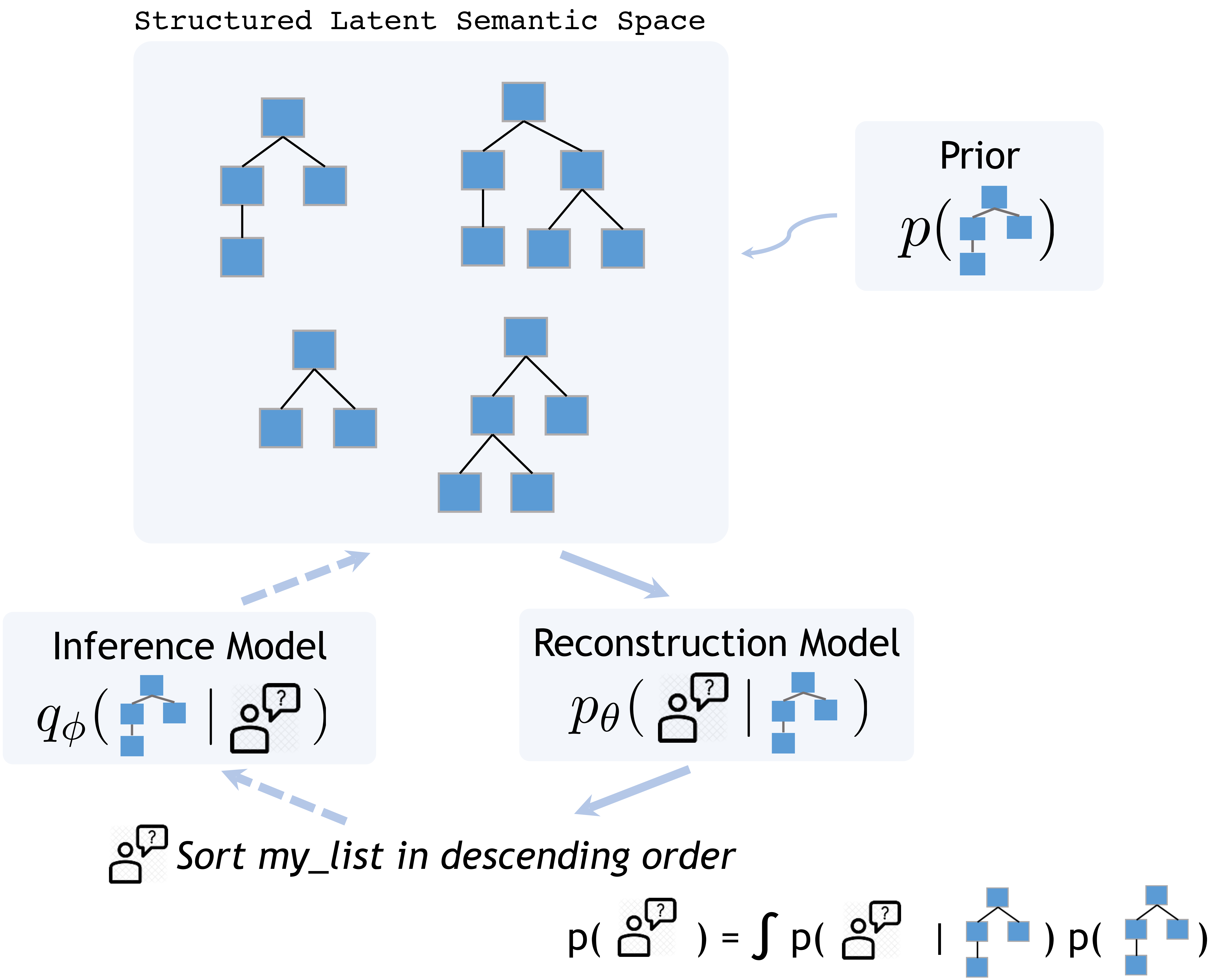
( , ) Labeled Data

+

**Unsupervised Objective**

$\sum \log p( \cdot )$

Unlabeled Data



# StructVAE: VAEs with Tree-structured Latent Variables

**Inference Model**

$q_\phi( \ | \ )$

Neural semantic parser

**Reconstruction Model**

$p_{\surd}( \ / \ )$

Neural sequence-to-sequence model

**Prior**

$p( \ )$

Neural Language Model

(use linearized trees as inputs)

**Unsupervised Objective**

$\sum \log p( \ )$

Unlabeled Data

Variational approximation of the marginal likelihood

$$\log p( \ ) \geq \sum_{'\sim q_\phi( \ | \ )} \log p_\theta( \ | \ ')$$

$$-\mathrm{KL\_Divergence}\left[q_\phi( \ | \ )||p( \ )\right]$$



[Miao and Blunsom, 2016]

# How does extra unlabeled data help learning?

Supervised Objective

$$\sum \log q_{\phi}( \quad | \quad )$$

( ) Labeled Data

$$\nabla = \sum_{\text{Training Examples}} \frac{\partial \log q_{\phi}( \quad | \quad )}{\partial \phi}$$



[Miao and Blunsom, 2016]

# How does extra unlabeled data help learning?

Unsupervised Objective

$\sum \log p( \quad )$

Unlabeled Data

$$\nabla \propto \sum_{\text{Sampled}\ ,} \quad \times \frac{\partial q_{\phi}( \quad ' \mid \quad )}{\partial \phi}$$

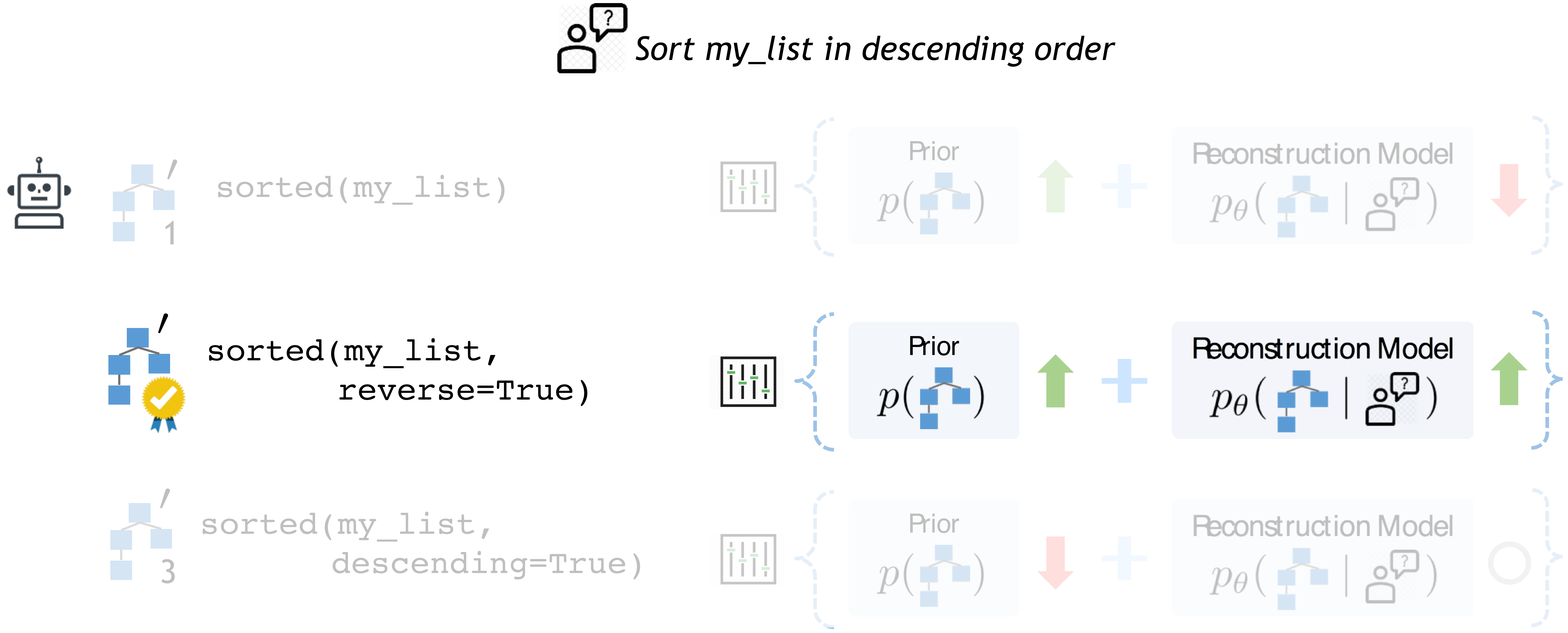
The learning signal $\approx$ { Prior $p( \quad )$ + Reconstruction Model $p_{\theta}( \quad \mid \quad )$ }

Learning signal acts as the tuning weights of gradients received by different sampled latent meaning representations from the inference model



[Miao and Blunsom, 2016]

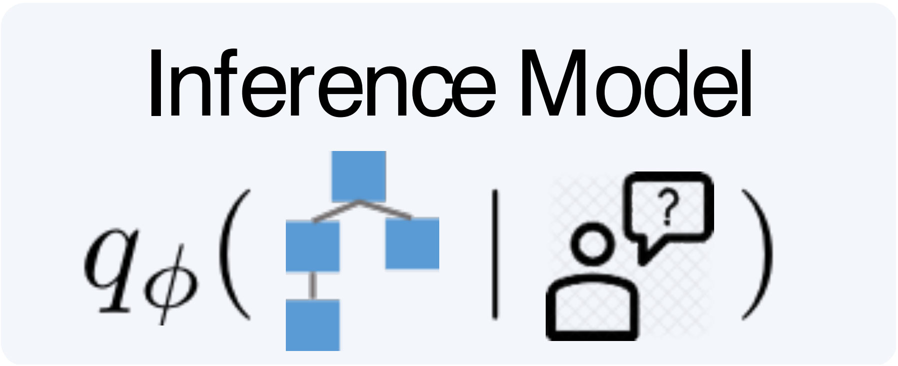
# How does extra unlabeled data help learning?




Learning favors sampled latent meaning representations that both:

- Faithfully encode the semantics of the utterance -> high reconstruction score
- Are succinct and natural -> high prior probability

# The Inference Model: a Transition-based Parser



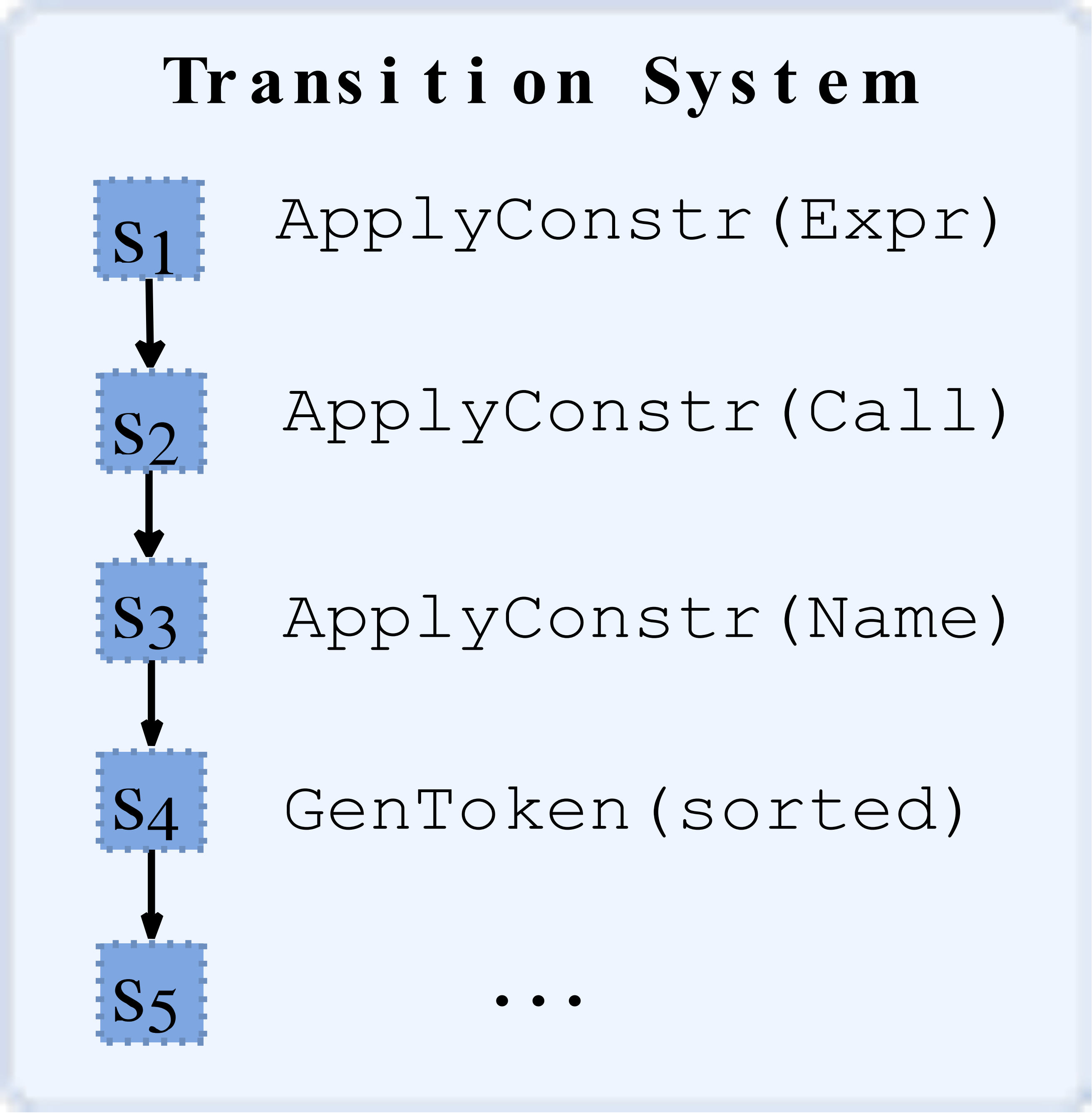

A transition-based parser that transduces natural language utterances into Abstract Syntax Trees

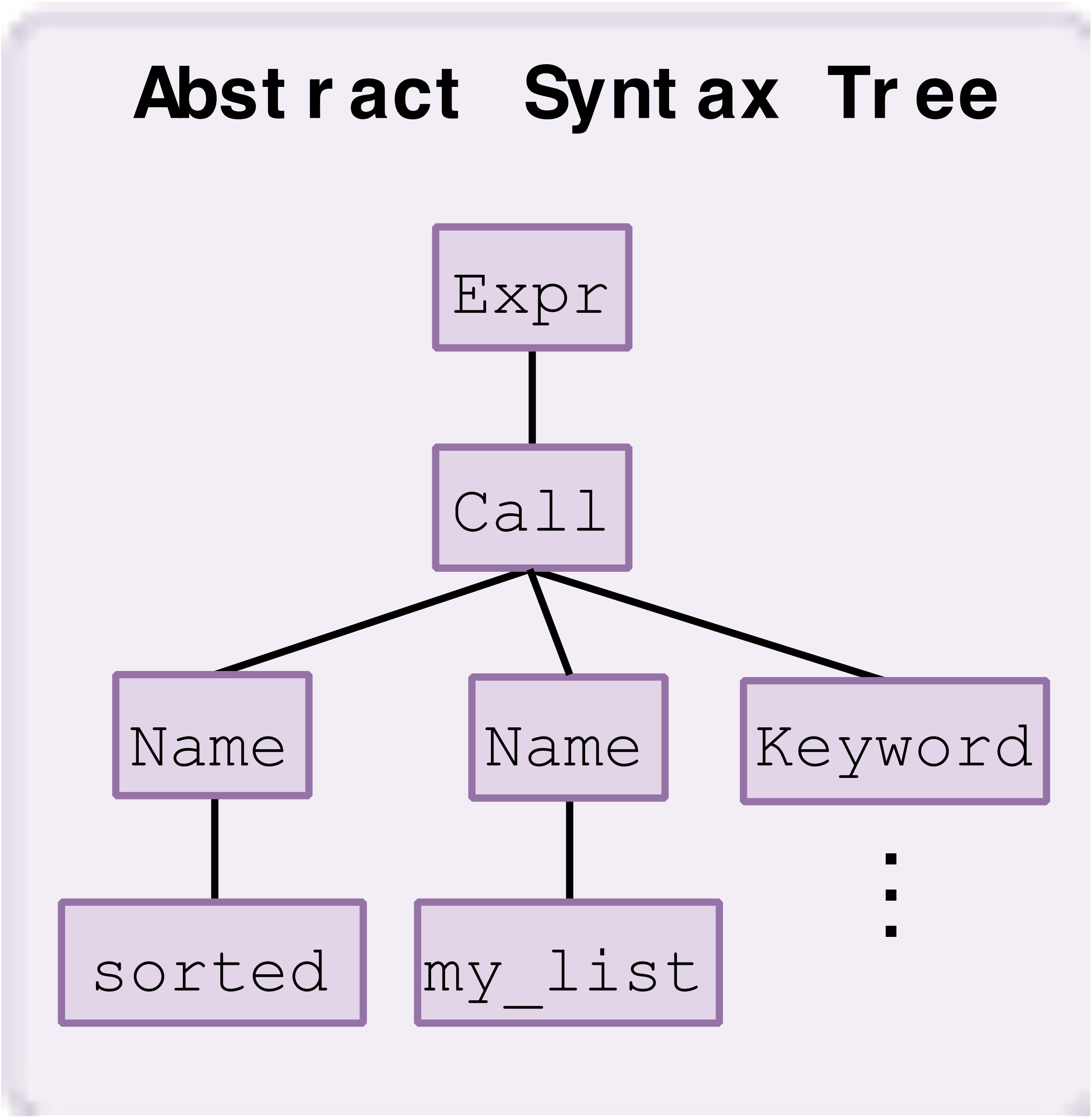
**Grammar Specification**

```
stmt ↦ FunctionDef(identifiler name,
            arguments args, stmt* body)
     | Expr(expr value)
expr ↦ Call(expr func, expr* args,
                  keyword* keywords)
     | Name(identifier id)
     | Str(string id)
```

**Input Utterance**

Sort my_list in descending order






[Yin and Neubig, 2017; Rabinovich et al. 2017]

# Datasets

## Django Python Code Generation Task

*Call the function _generator, join the result into a string, return the result*

```
return ''.join(_generator())
```

## ATIS Semantic Parsing Task

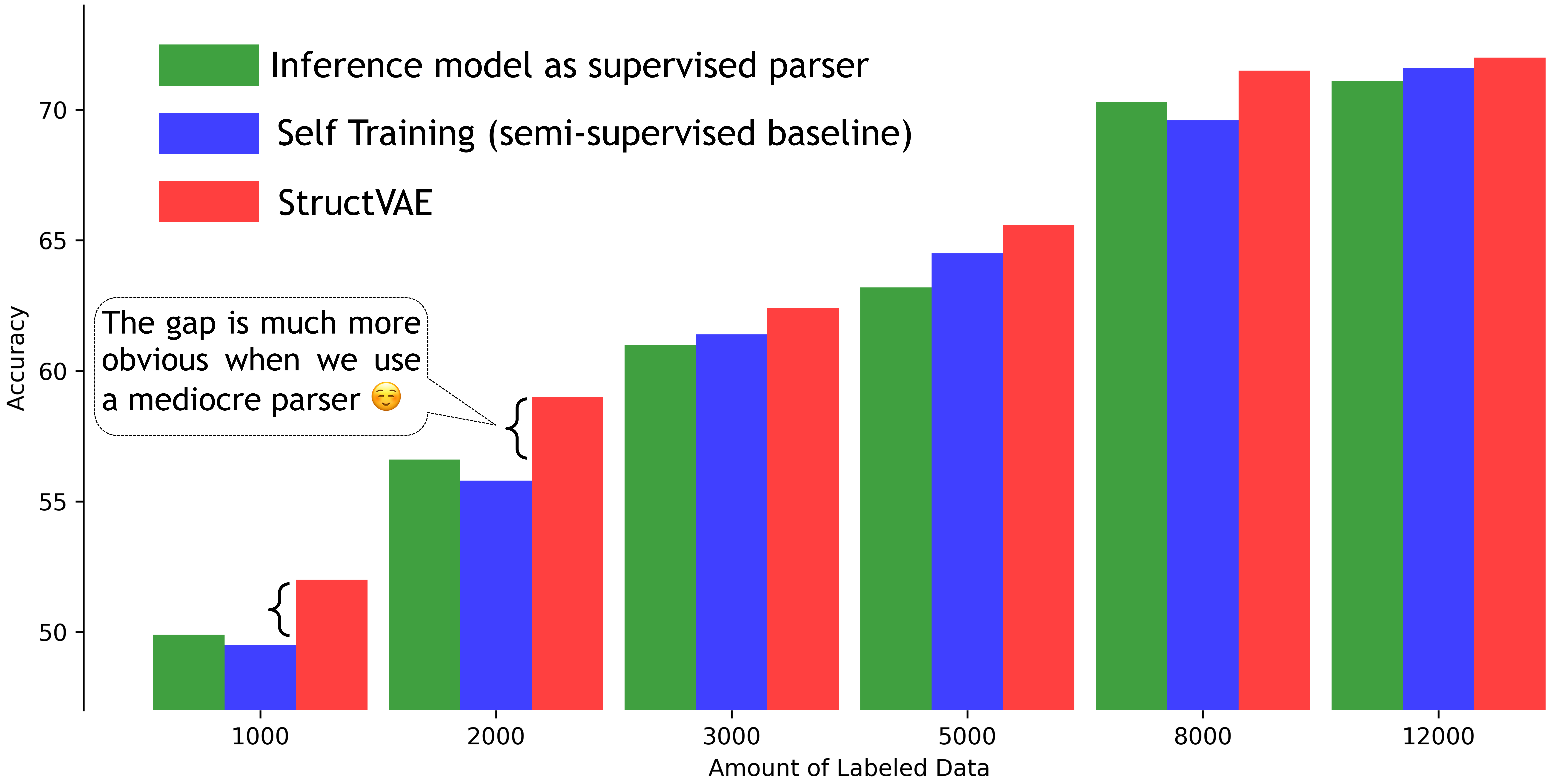
*Show me flights from San Francisco to Washington*

```
lambda $0 e
   (and (flight $0)
   (from $0 san_Francisco:ci)
   (to $0 washington:ci))
```

# Research Questions

- **RQ1** Does StructVAE outperforms purely supervised semantic parsers with extra unlabeled data?
- **RQ2** Can we get some empirical evidence about why StructVAE works?

# StructVAE v.s. Baselines



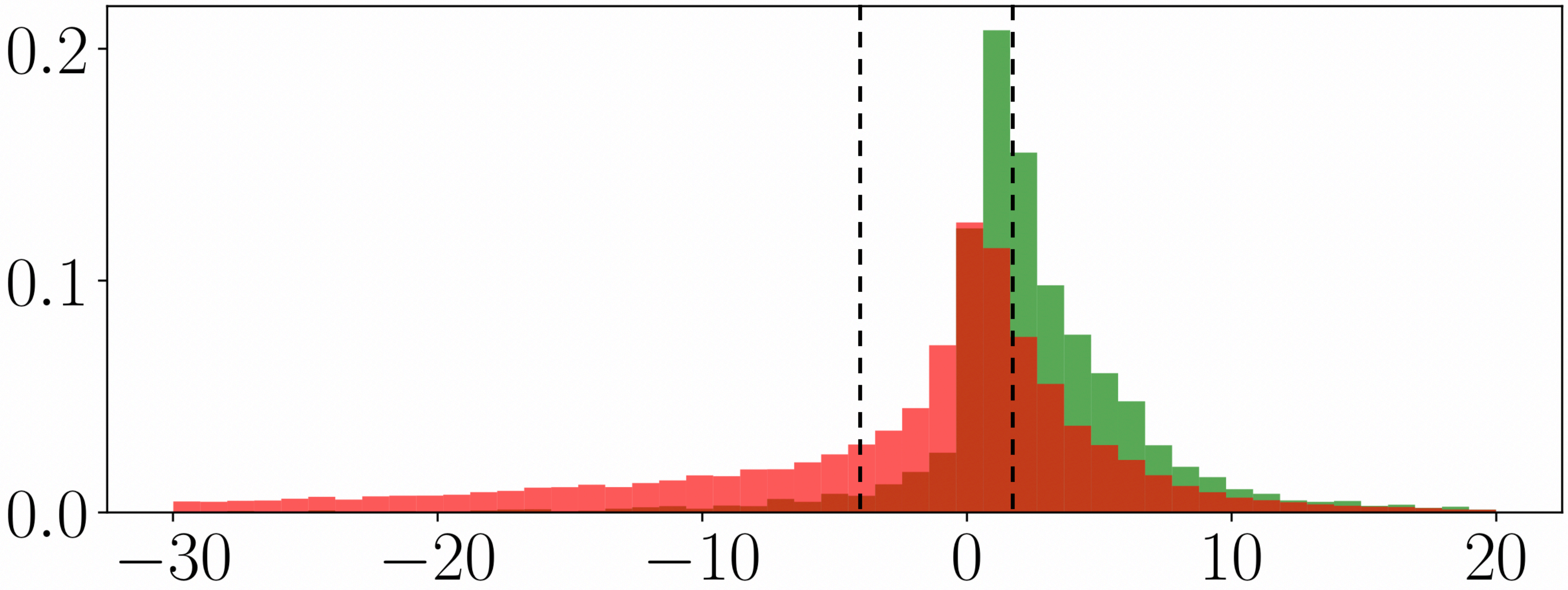

Language Technologies Institute

*all available training utterances as unlabeled data*

# Why does StructVAE work?

- For each unlabeled utterance , compute the learning signal for gold samples and other (imperfect) samples

Avg.=-5.12 Avg.=2.59




Other Samples Gold Samples

# Case Studies

*Join p and cmd into a file path, substitute it for f*

| | Parser Score $q_\phi( / )$ | Prior $p()$ | Reconstruction Score $p_\psi( / )$ | Learning Signal |
|---|---|---|---|---|
| ✔ `f = os.path.join(p, cmd)` | -1.00 | -24.33 | -2.00 | 9.14 |
| ✘ `p = path.join(p, cmd)` | -8.12 | -27.89 | -20.96 | -9.47 |

*Split string pks by ‘,’ , substitute the result for primary_keys*

| | Parser Score $q_\phi( / )$ | Prior $p()$ | Reconstruction Score $p_\psi( / )$ | Learning Signal |
|---|---|---|---|---|
| ✔ `primary_keys = pks.split(',')` | -2.38 | -10.24 | -11.39 | 2.05 |
| ✘ `primary_keys = pks.split + ','` | -1.83 | -20.41 | -14.87 | -2.60 |

Carnegie Mellon University
School of Computer Science

# Unsupervised Learning of Syntactic Structure w/ Invertible Neural Projections

Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick

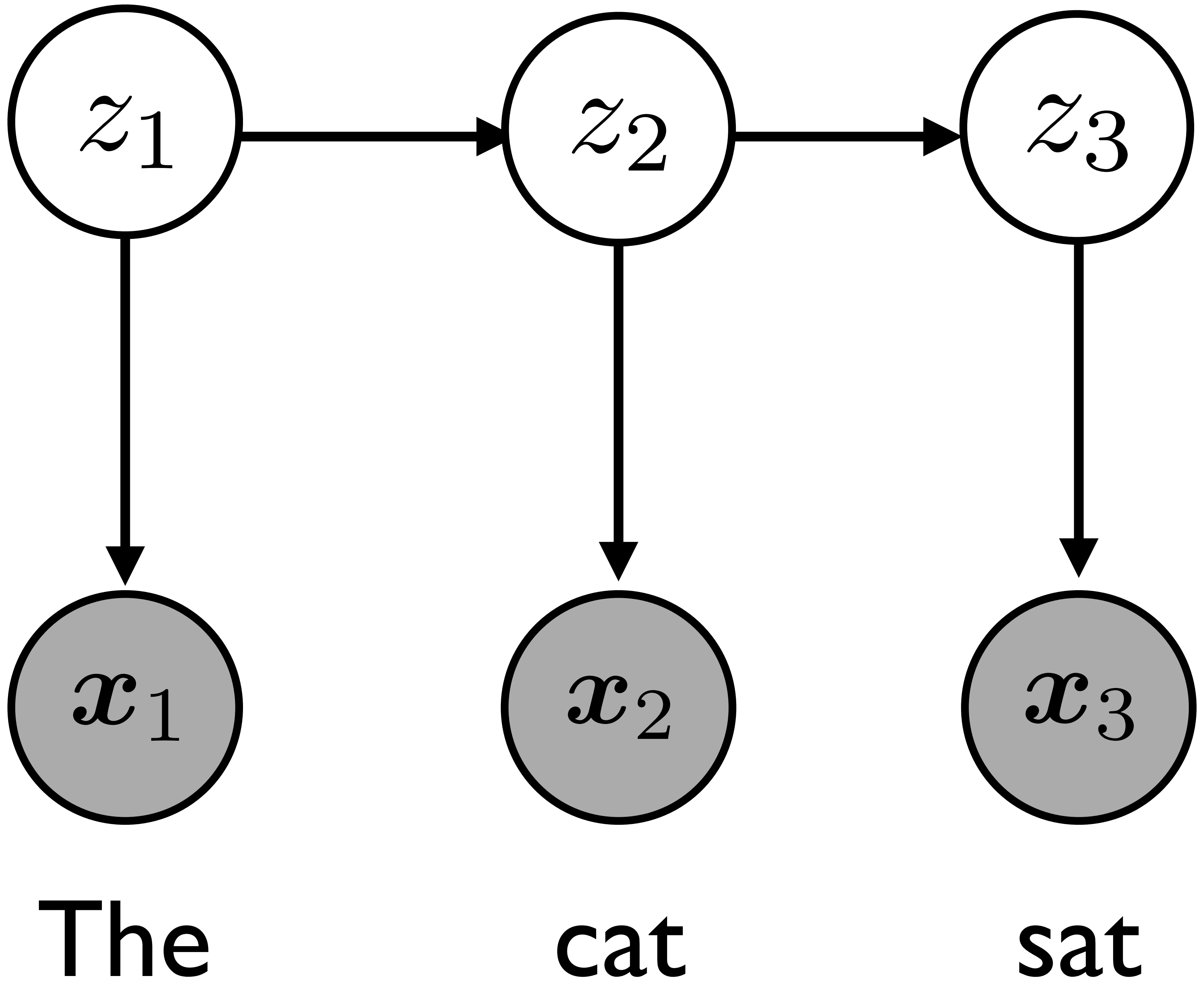
(EMNLP 2018)

Language Technologies Institute

NEULAB

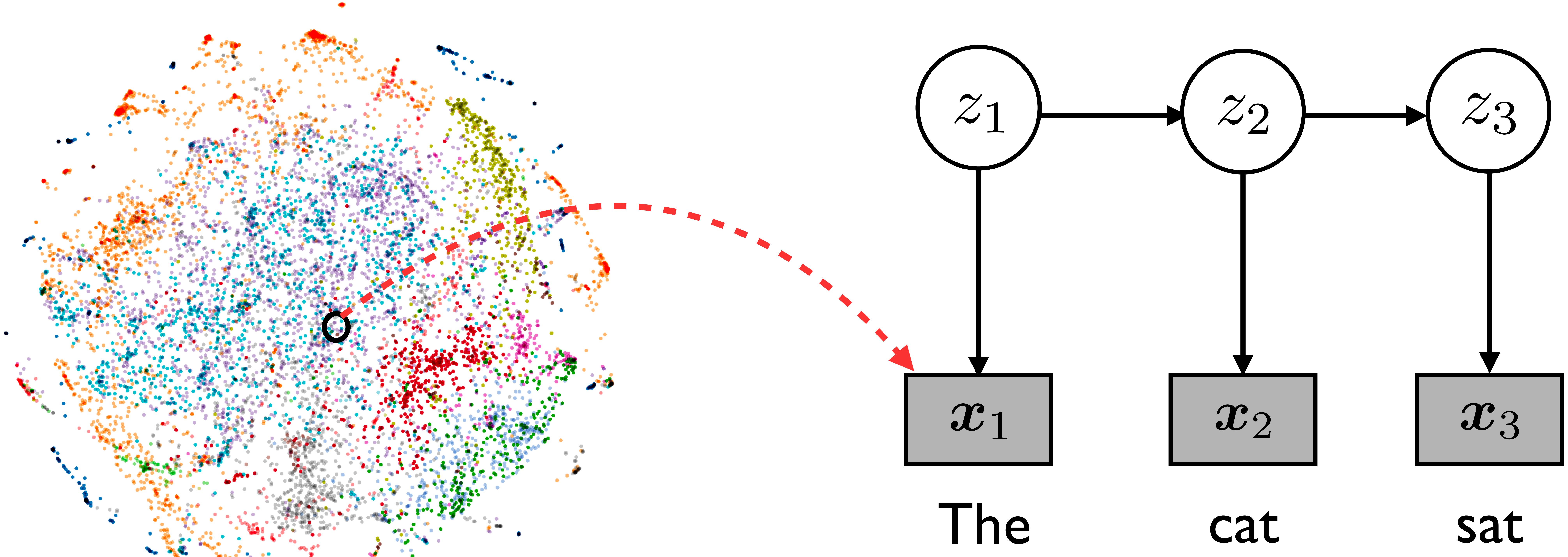
Language Technologies Institute

# HMM for Part-of-Speech Induction



Language Technologies Institute

# Gaussian HMM for POS Induction




$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \Sigma_{z_i})$$

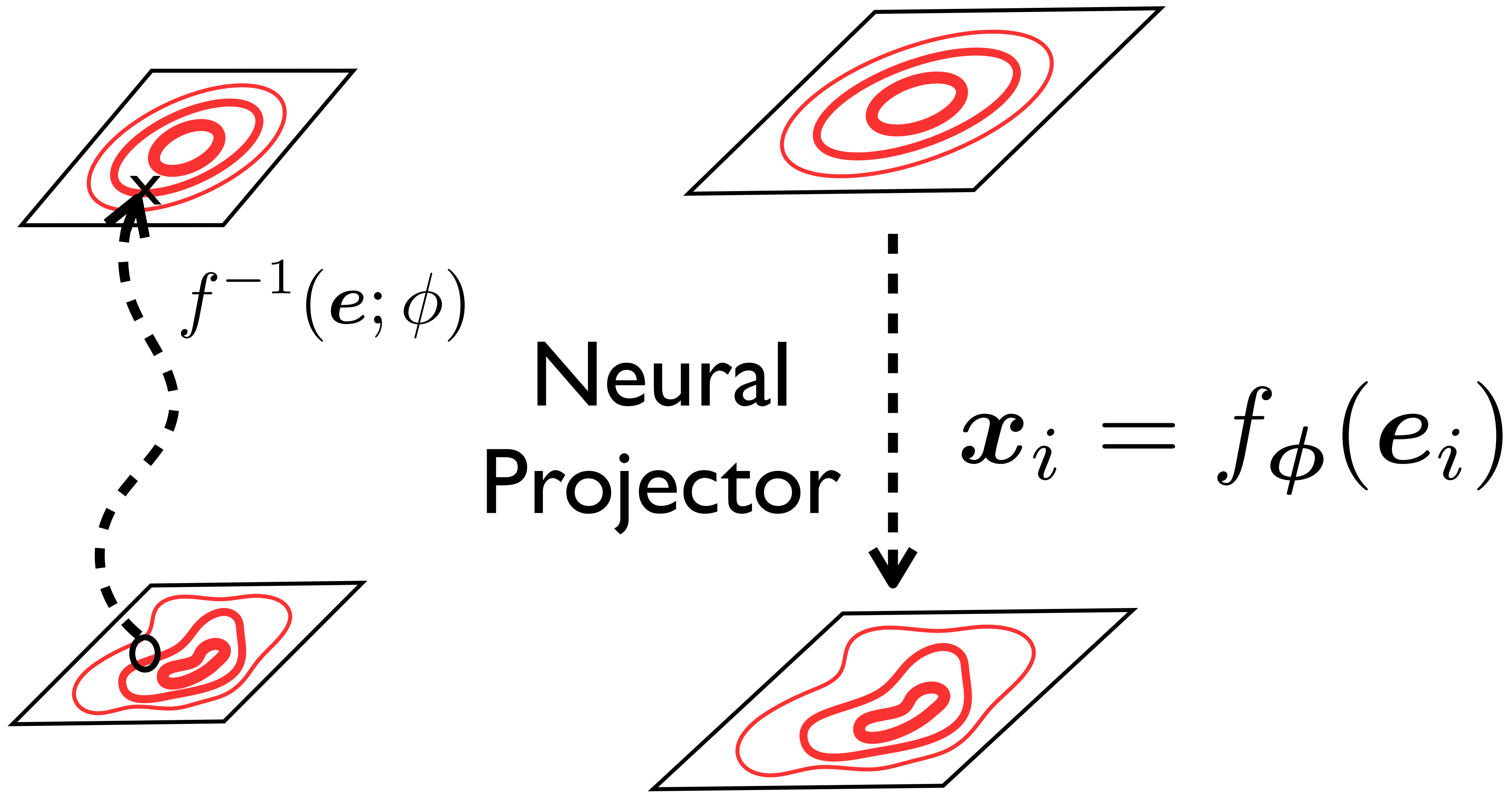
[Lin et al. 2015]

Language Technologies Institute

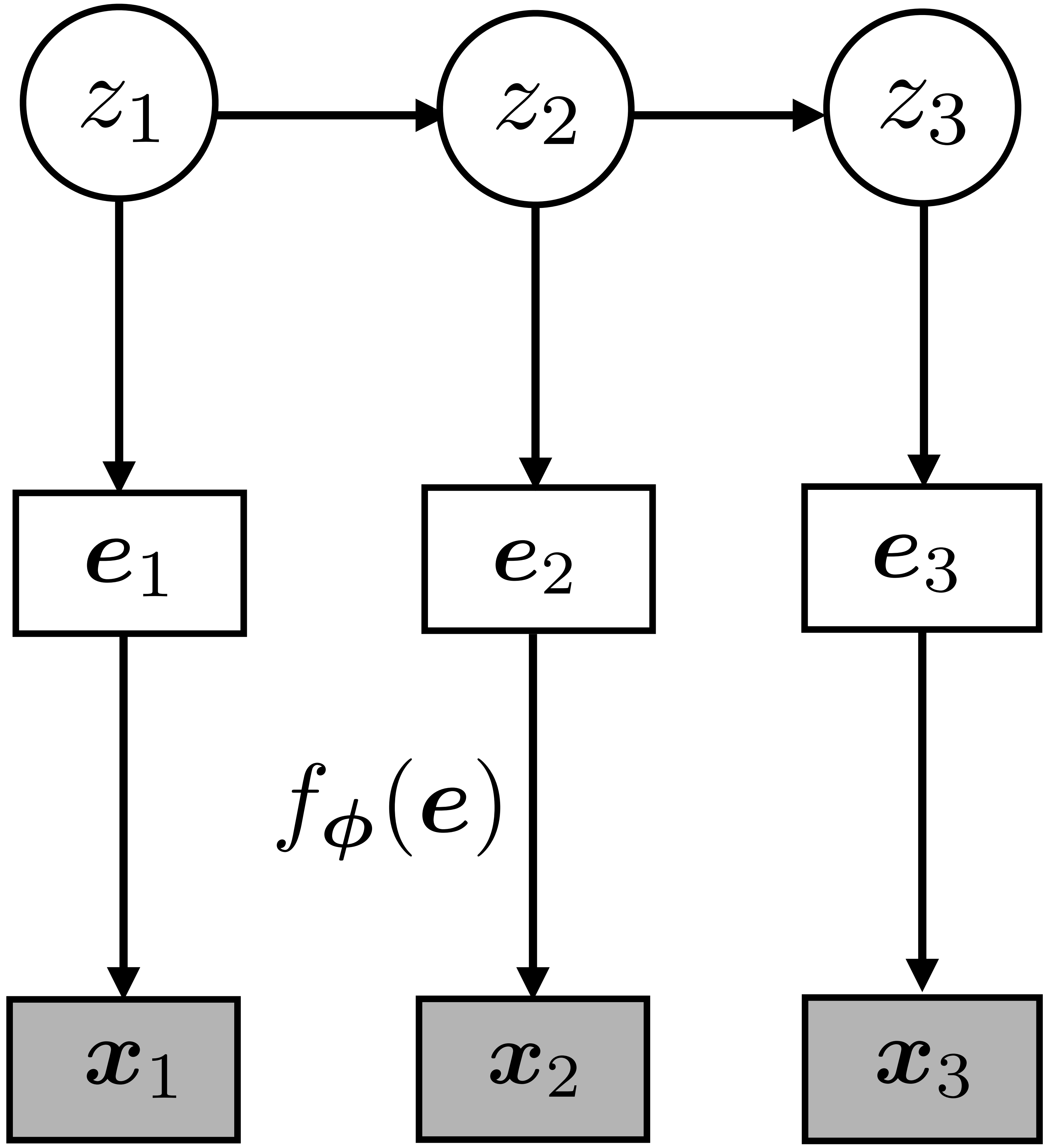
# Latent Embeddings w/ Neural Projection

$z_i \sim$ Markov Structure

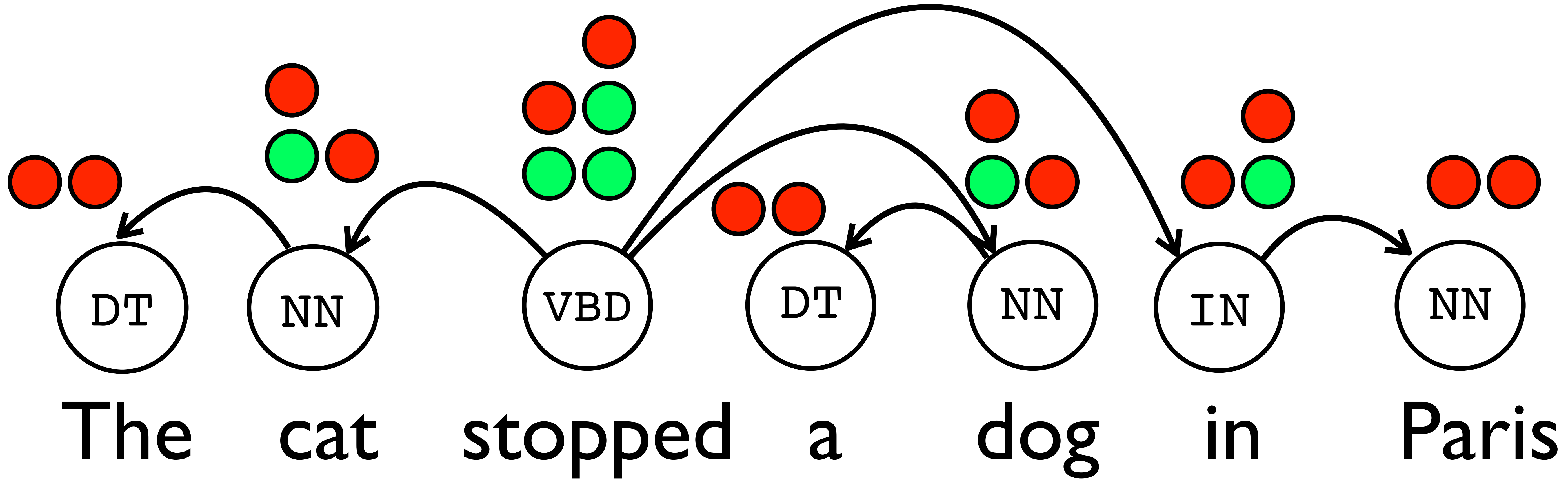
$\boldsymbol{e}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})$




$\boldsymbol{x}_i \sim$ Point mass at $f_\phi(\boldsymbol{e}_i)$



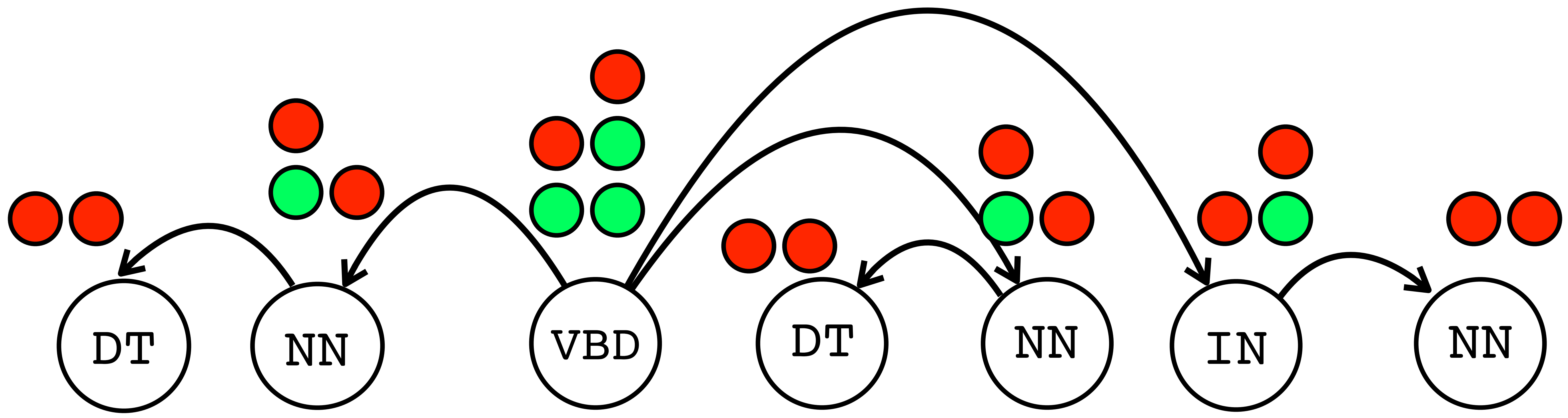
# Dependency Model with Valence




[Klein and Manning 2004]

Language Technologies Institute

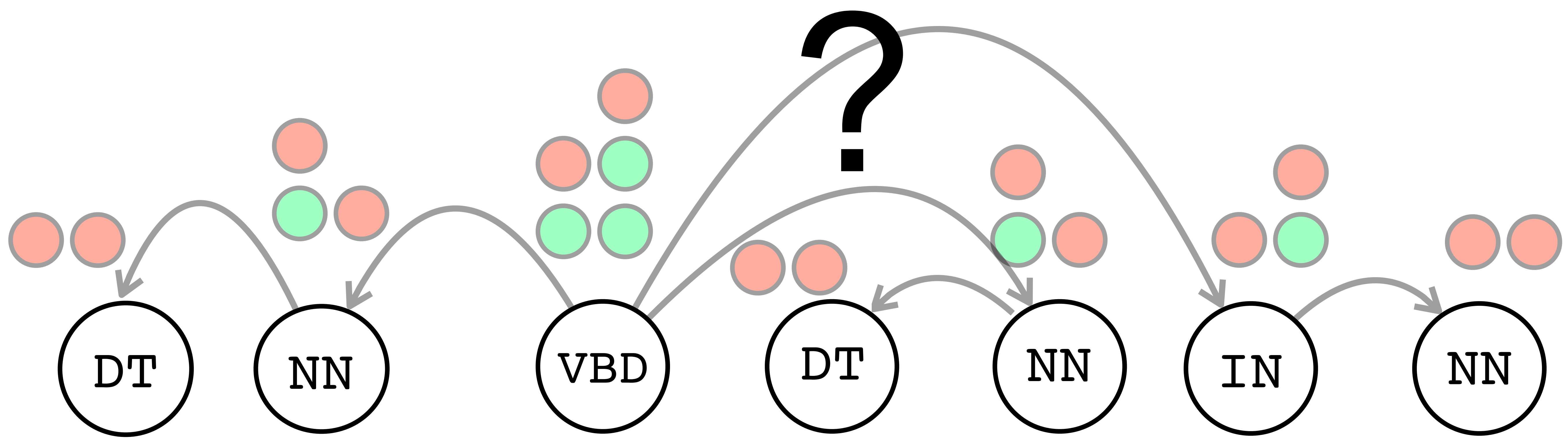
# Dependency Model with Valence



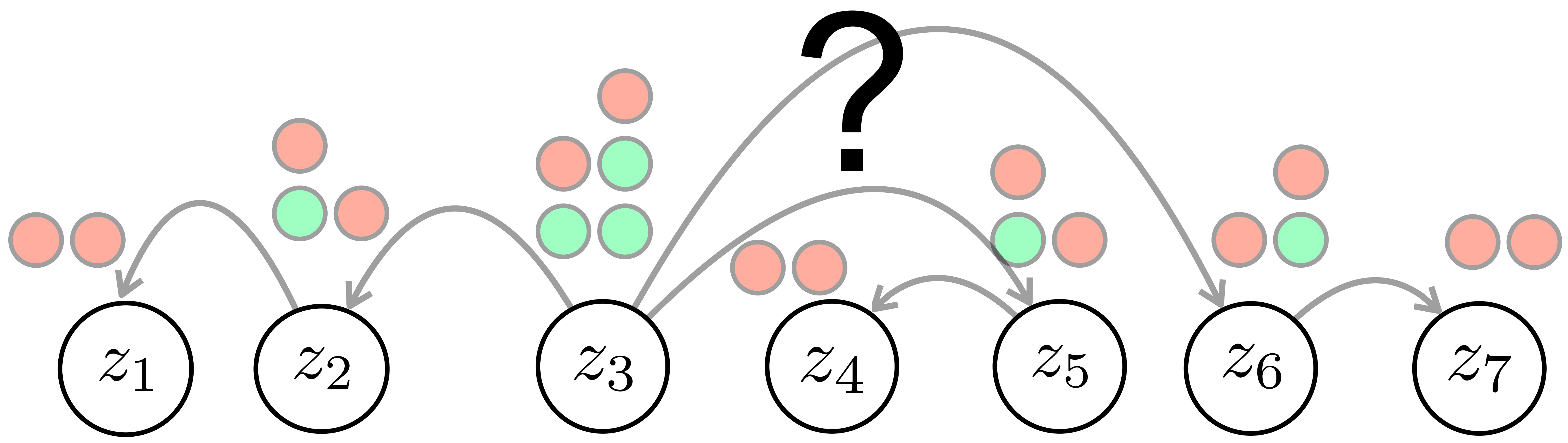

[Klein and Manning 2004]

Language Technologies Institute

# Dependency Parse Induction from POS

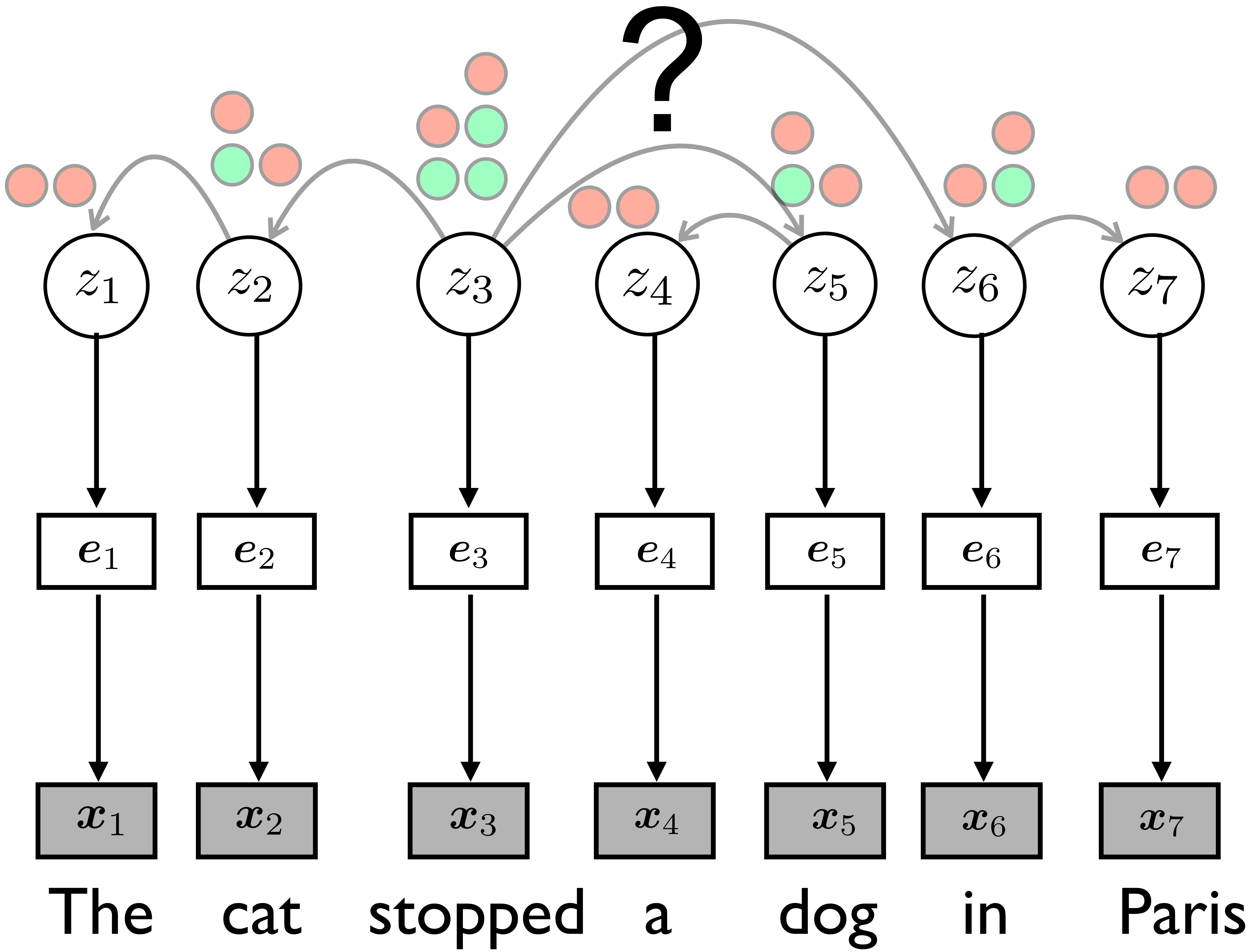


# Grammar Induction from Raw Text



Language Technologies Institute

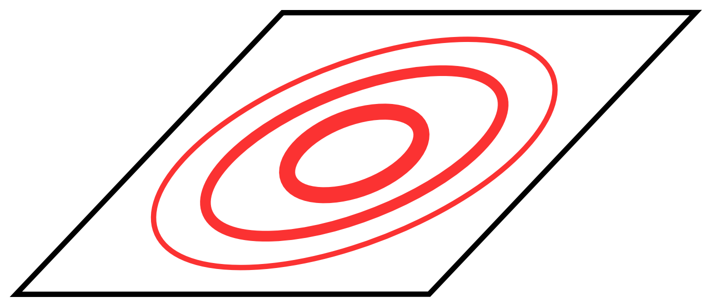
# Grammar Induction from Raw Text



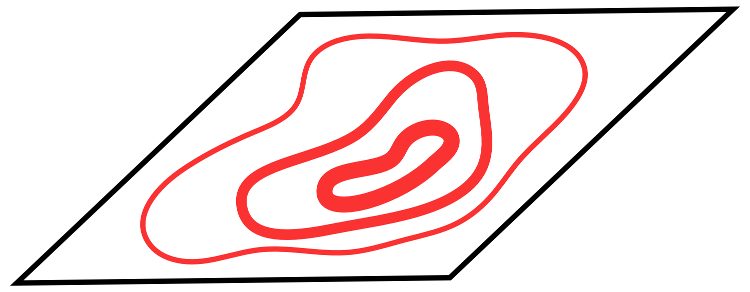
# Latent Embeddings w/ Neural Projection

$z_i \sim$ Syntax Model

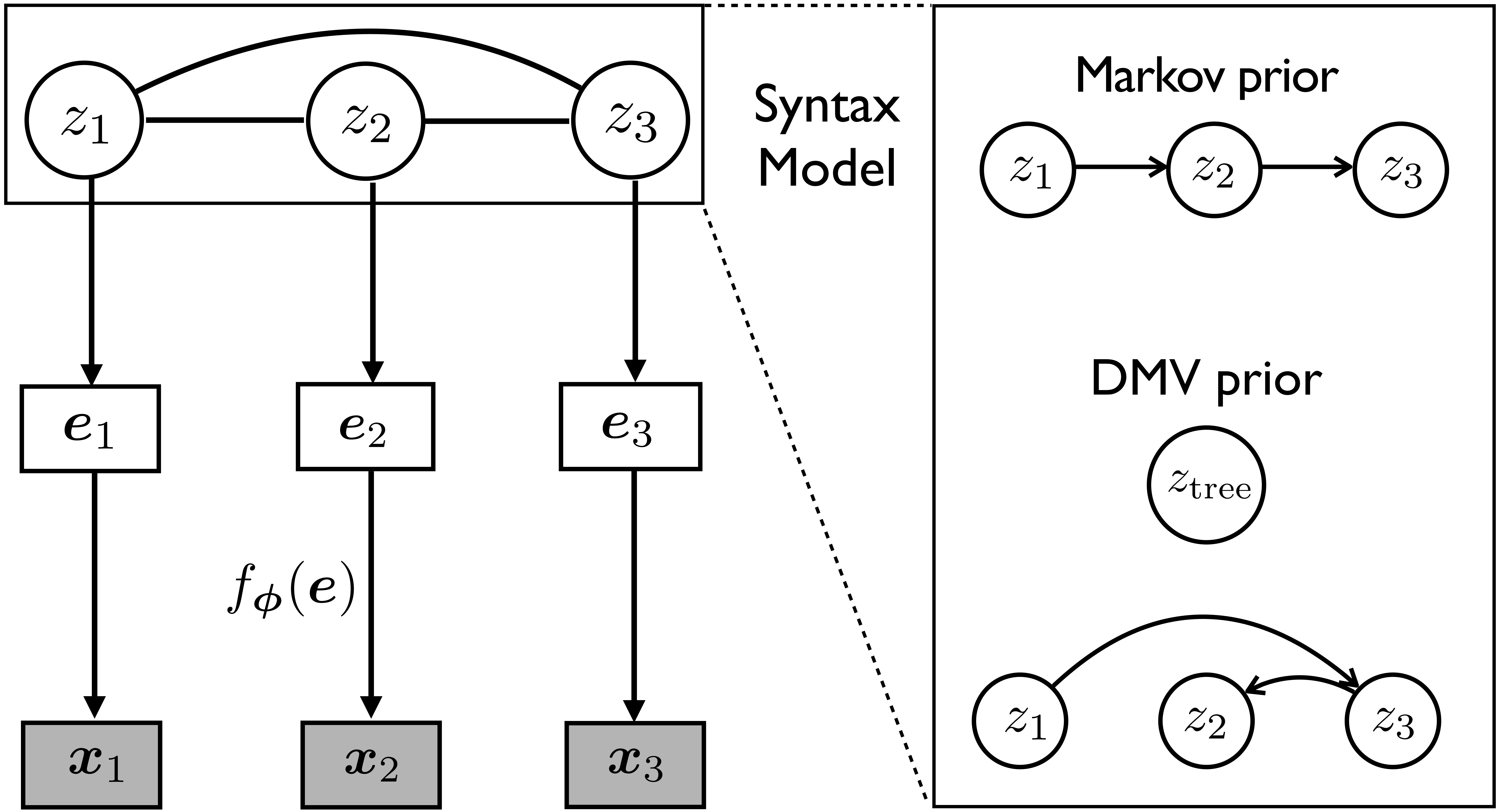
$\boldsymbol{e}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})$



Neural Projector $\boldsymbol{x}_i = f_{\phi}(\boldsymbol{e}_i)$

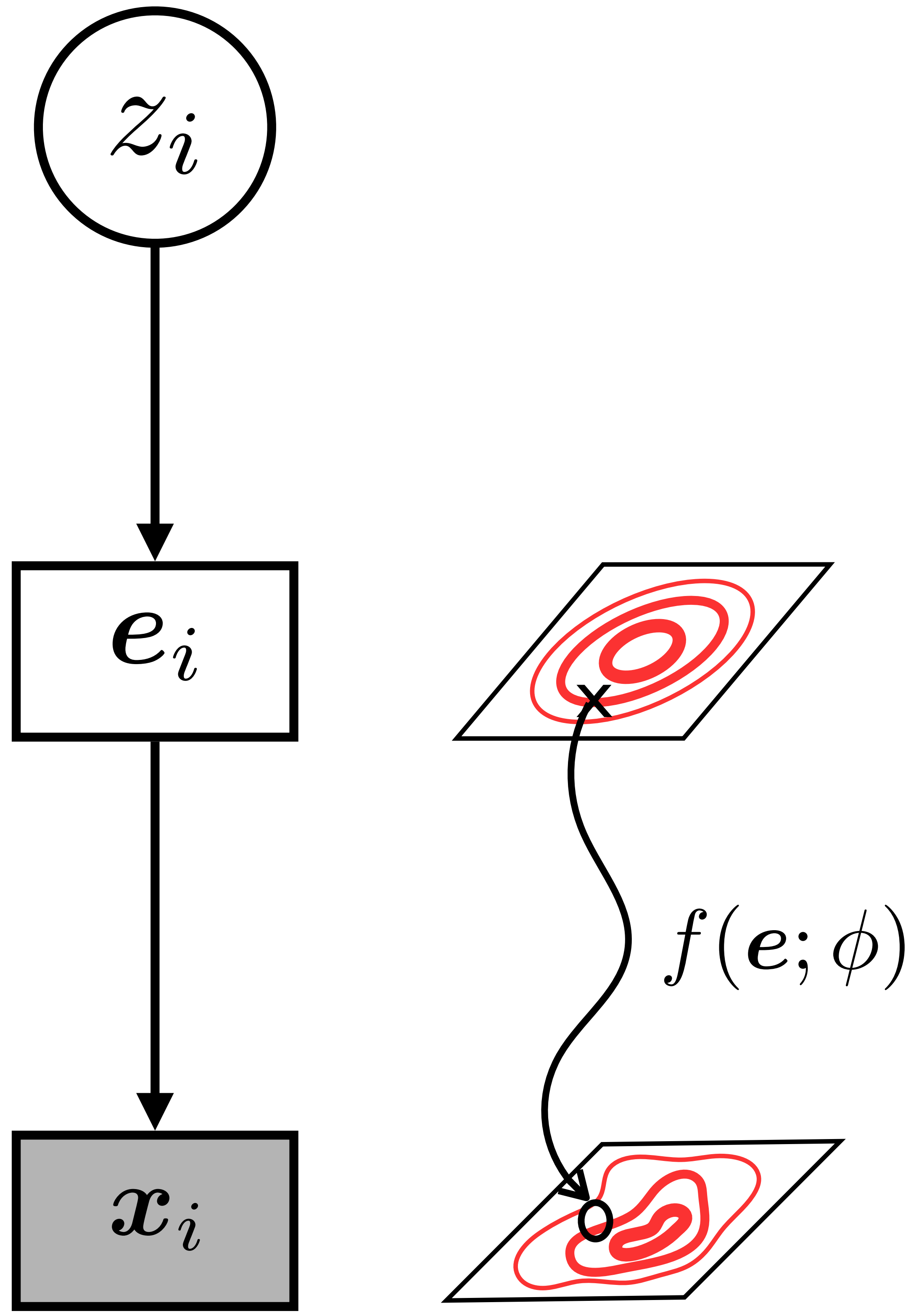


$\boldsymbol{x}_i \sim$ Point mass at $f_{\phi}(\boldsymbol{e}_i)$



Language Technologies Institute

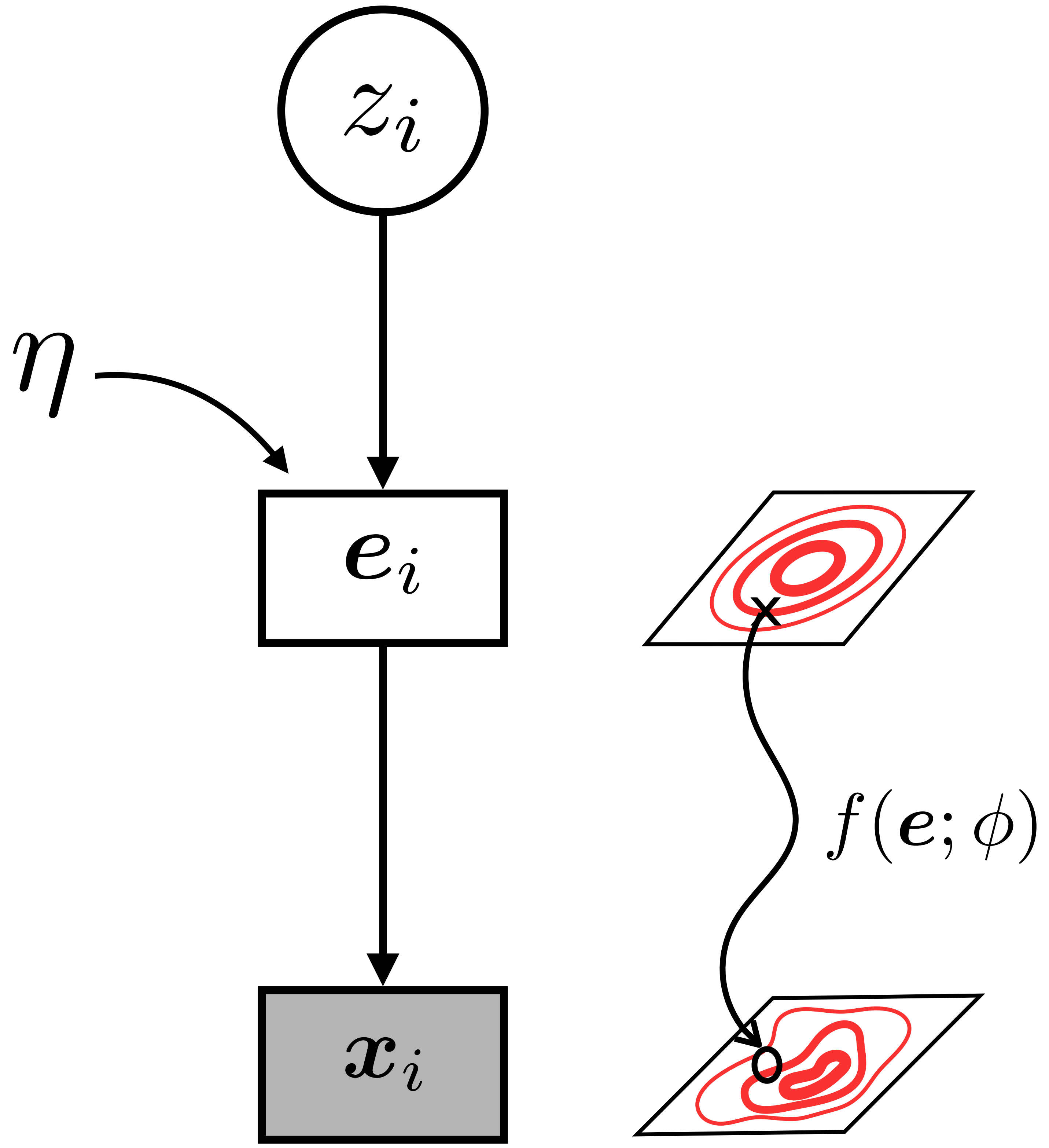
# Learning and Inference




$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$

Language Technologies Institute

# Learning and Inference

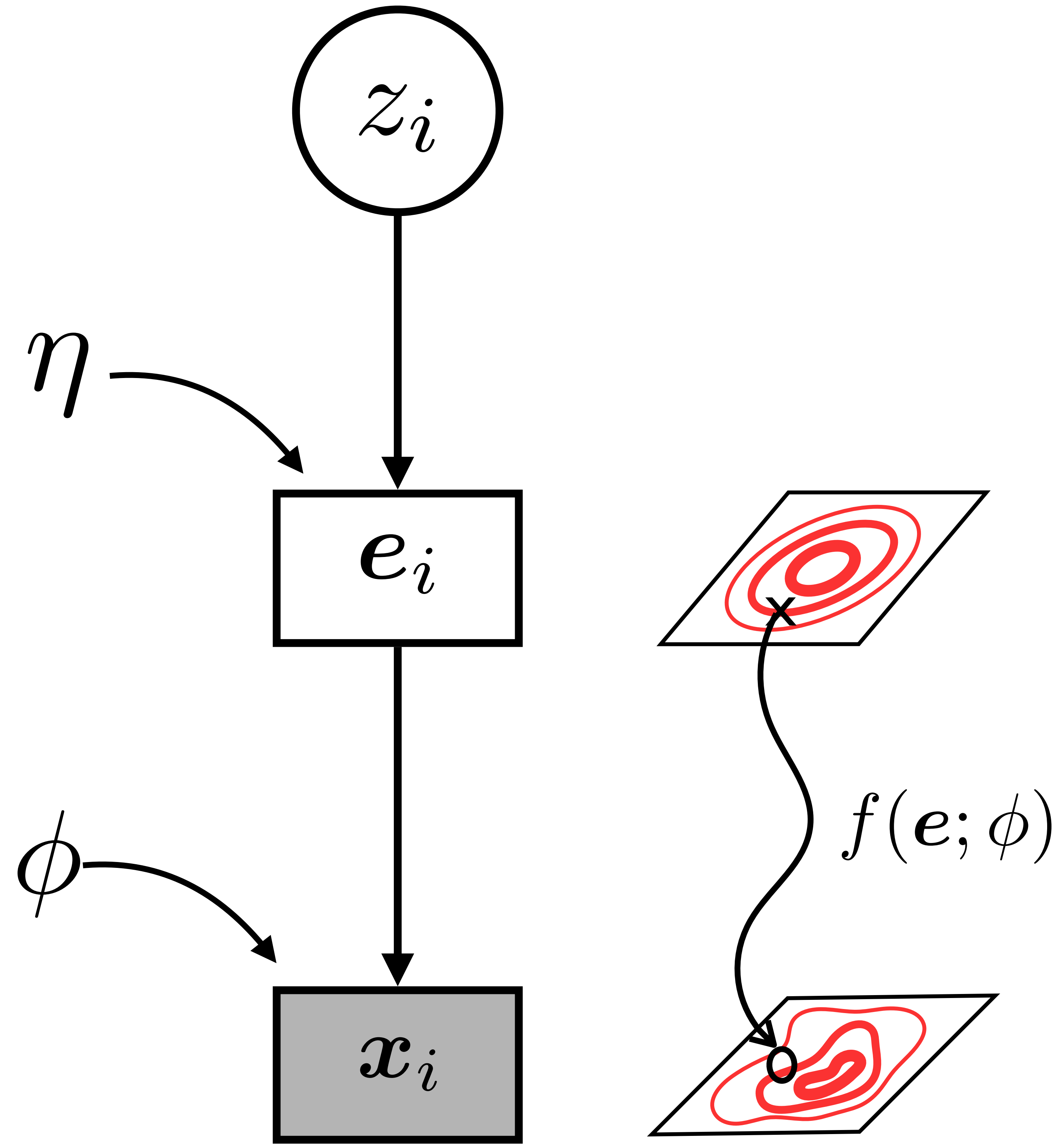



$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$

Gaussian embedding parameters

Language Technologies Institute

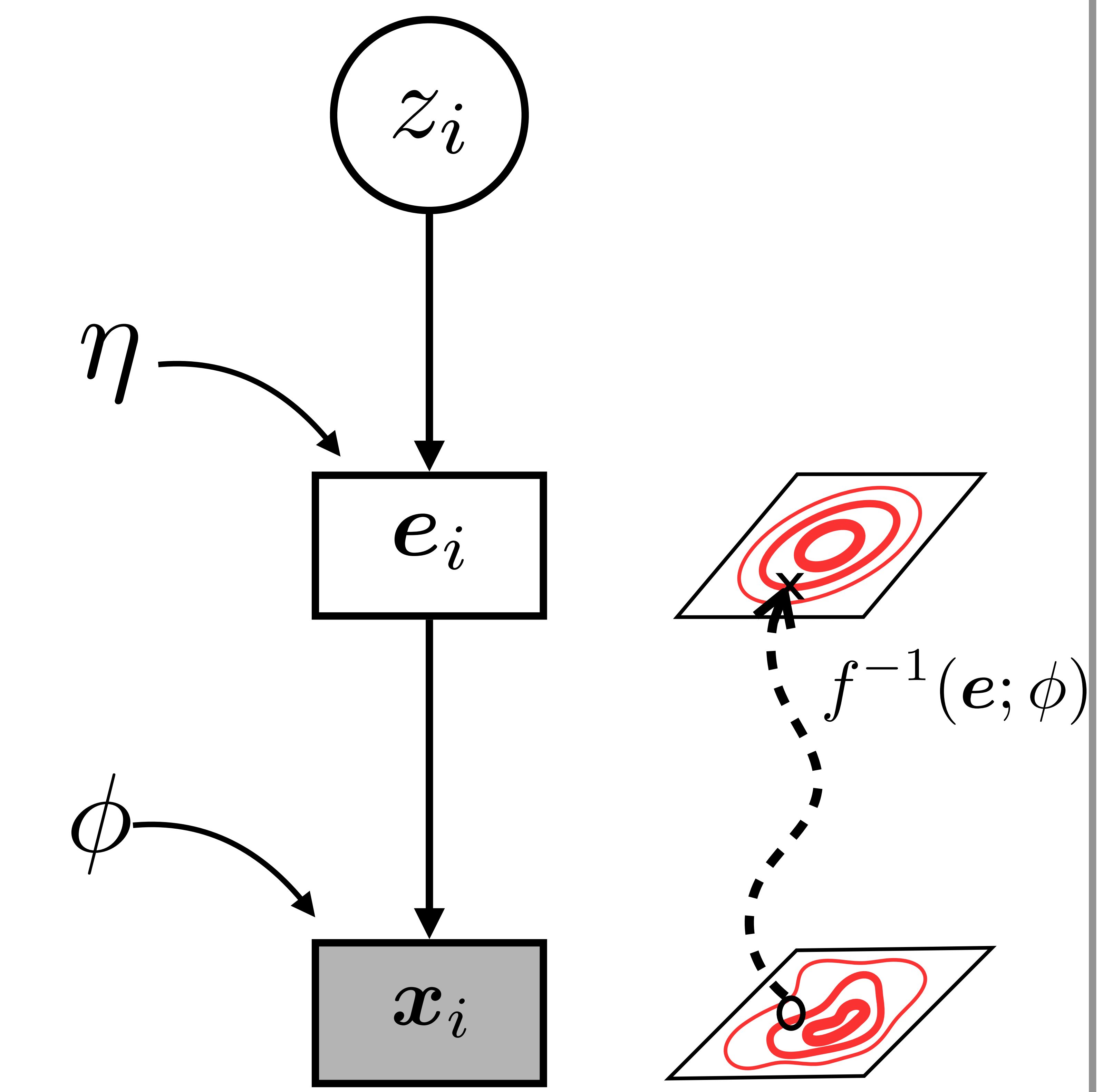
# Learning and Inference




$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$

Projection parameters

# Learning and Inference

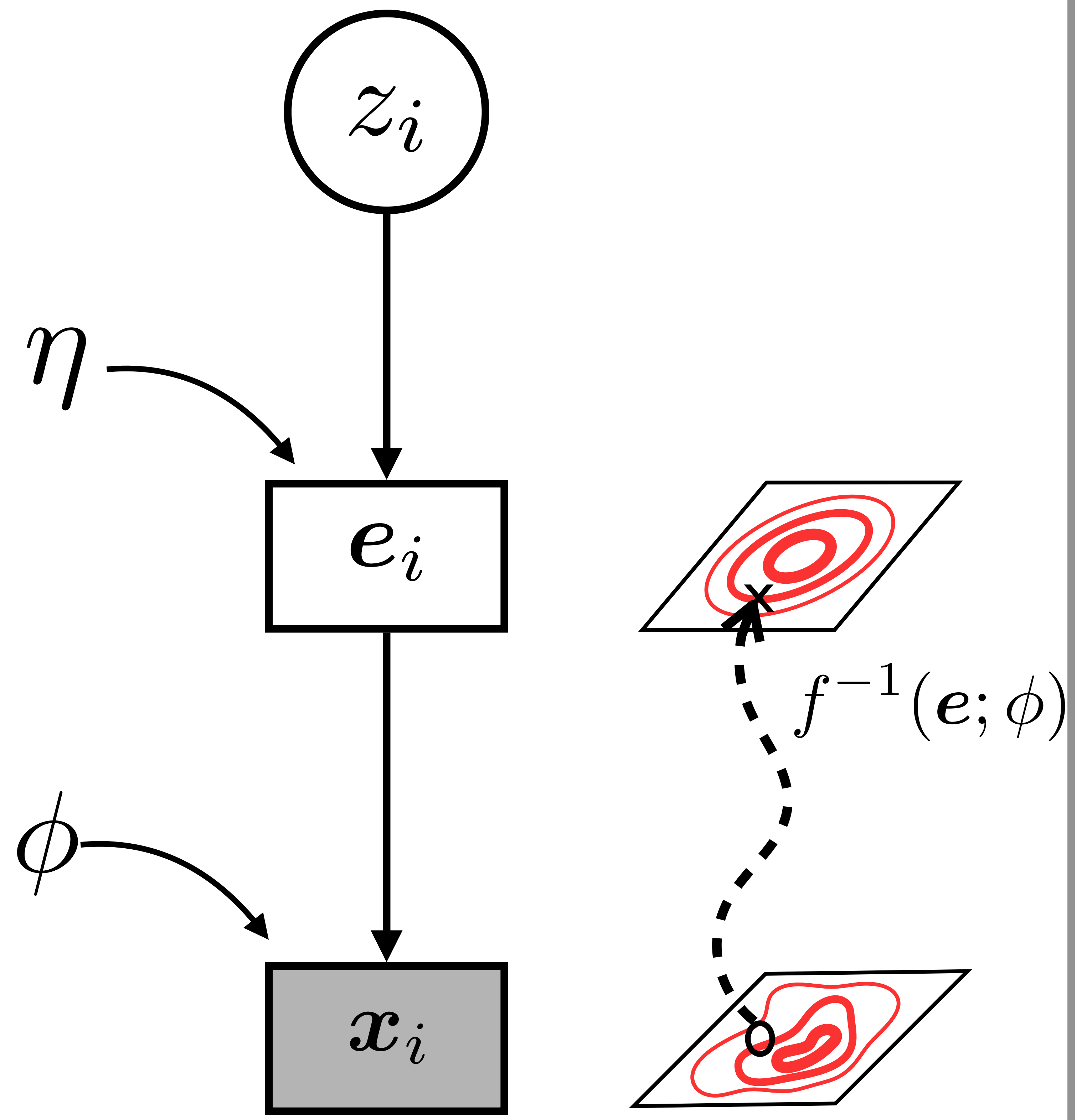



dim(x) = dim(e) and $f$ is invertible

$$p(\boldsymbol{x}_i|z_i;\eta,\phi)$$

$$= p(f_\phi^{-1}(\boldsymbol{x}_i)|z_i;\eta)\left|\det\frac{\partial f^{-1}}{\partial \boldsymbol{x}_i}\right|$$

# Learning and Inference



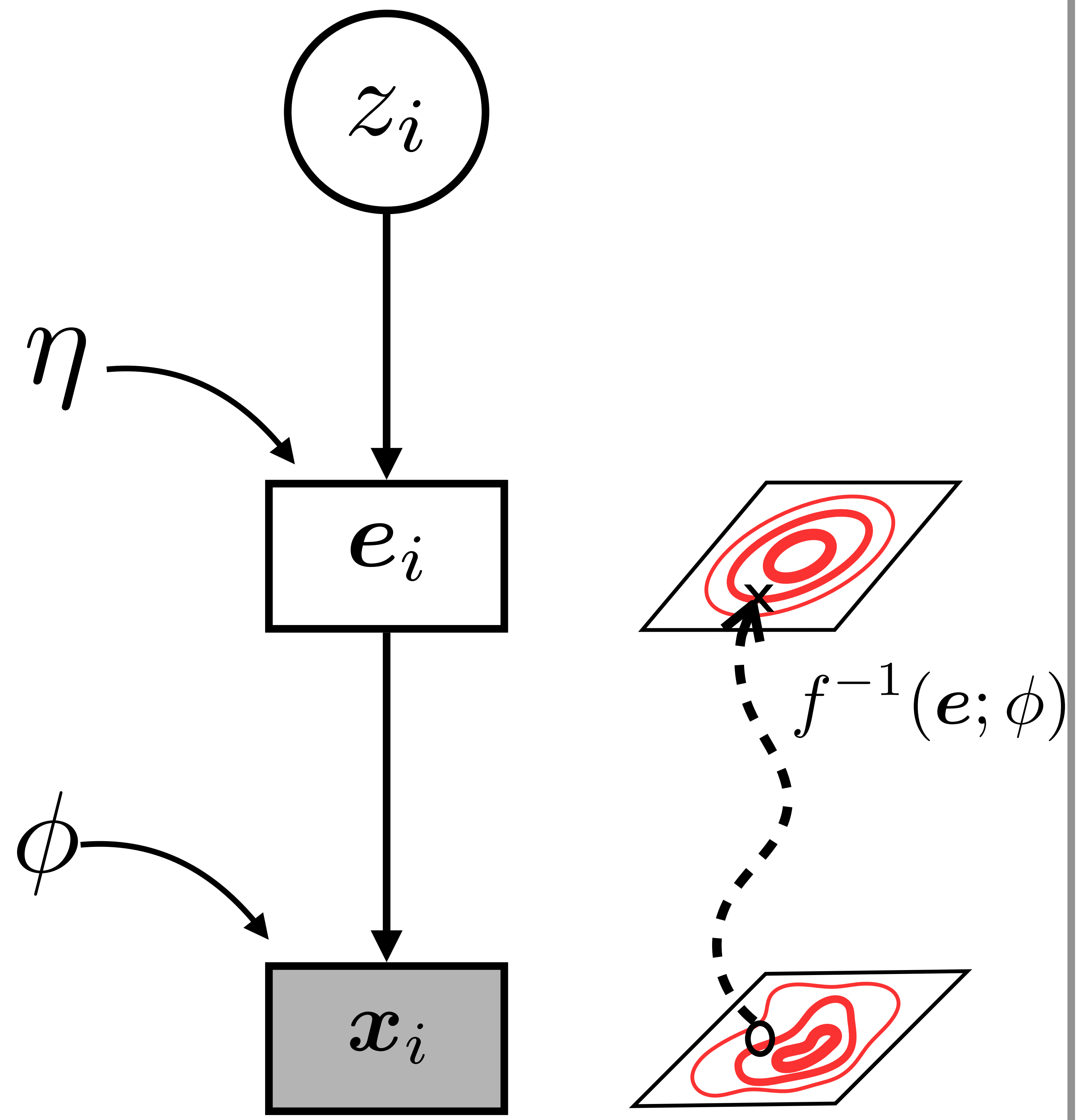

dim(x) = dim(e) and $f$ is invertible

$$p(\boldsymbol{x}_i|z_i;\eta,\phi)$$

$$= p(f_\phi^{-1}(\boldsymbol{x}_i)|z_i;\eta)\left|\det\frac{\partial f^{-1}}{\partial \boldsymbol{x}_i}\right|$$

Determinant of Jacobian matrix

# Learning and Inference




dim(x) = dim(e) and $f$ is invertible

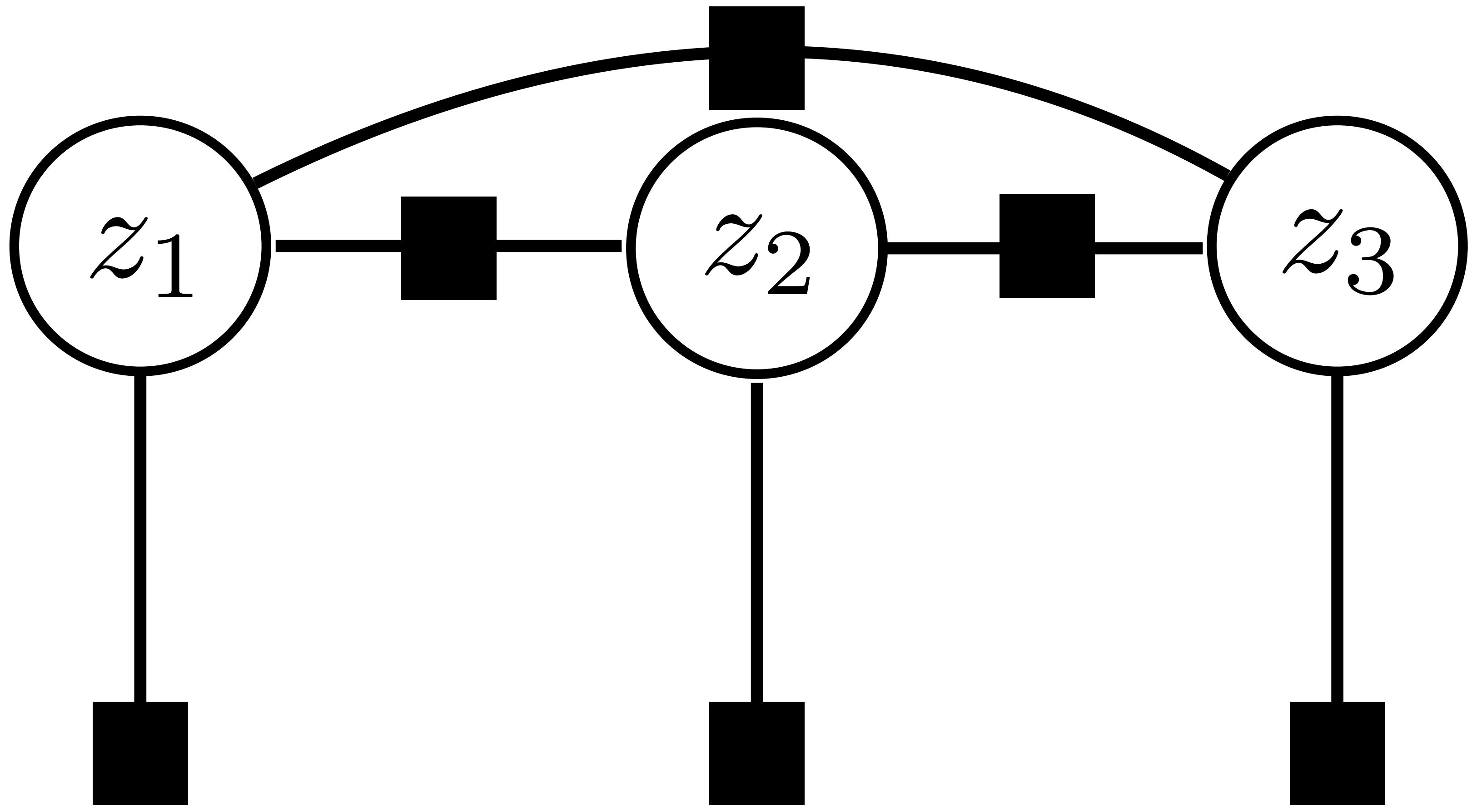
$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$

$$= p(f_\phi^{-1}(\boldsymbol{x}_i) | z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial \boldsymbol{x}_i} \right|$$

Gaussian distribution

Determinant of Jacobian matrix

# Learning and Inference



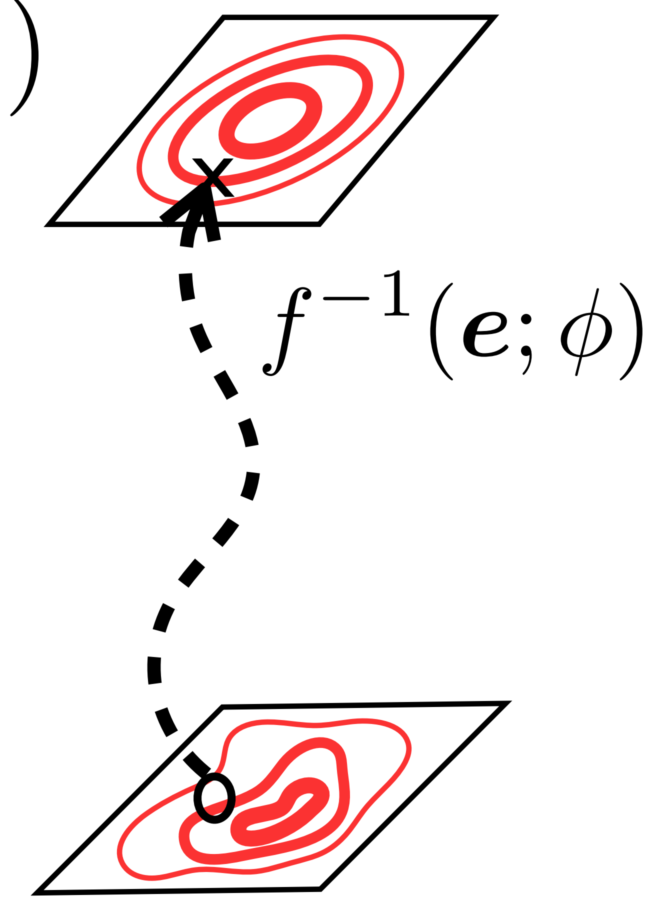

$$\boxed{p(f_\phi^{-1}(\boldsymbol{x}_i)|z_i;\eta)\left|\det\frac{\partial f^{-1}}{\partial \boldsymbol{x}_i}\right|}$$

Example of Markov prior

$$\log p(\boldsymbol{x}) = \log p_{\mathrm{GHMM}}(f_\phi^{-1}(\boldsymbol{x}))$$

$$+\sum \log\left|\det\frac{\partial f_\phi^{-1}}{\partial \boldsymbol{x}_i}\right|$$

$-\infty$ when $f$ is not invertible



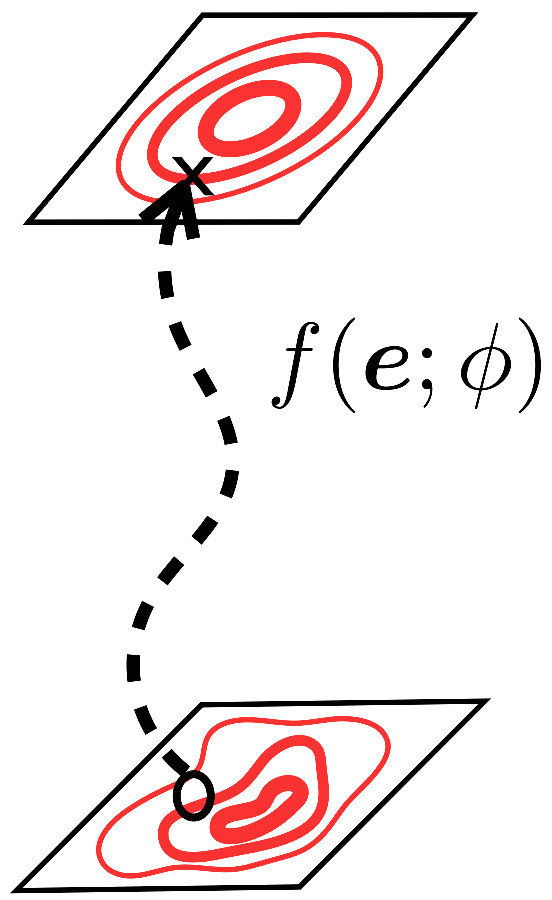
Language Technologies Institute

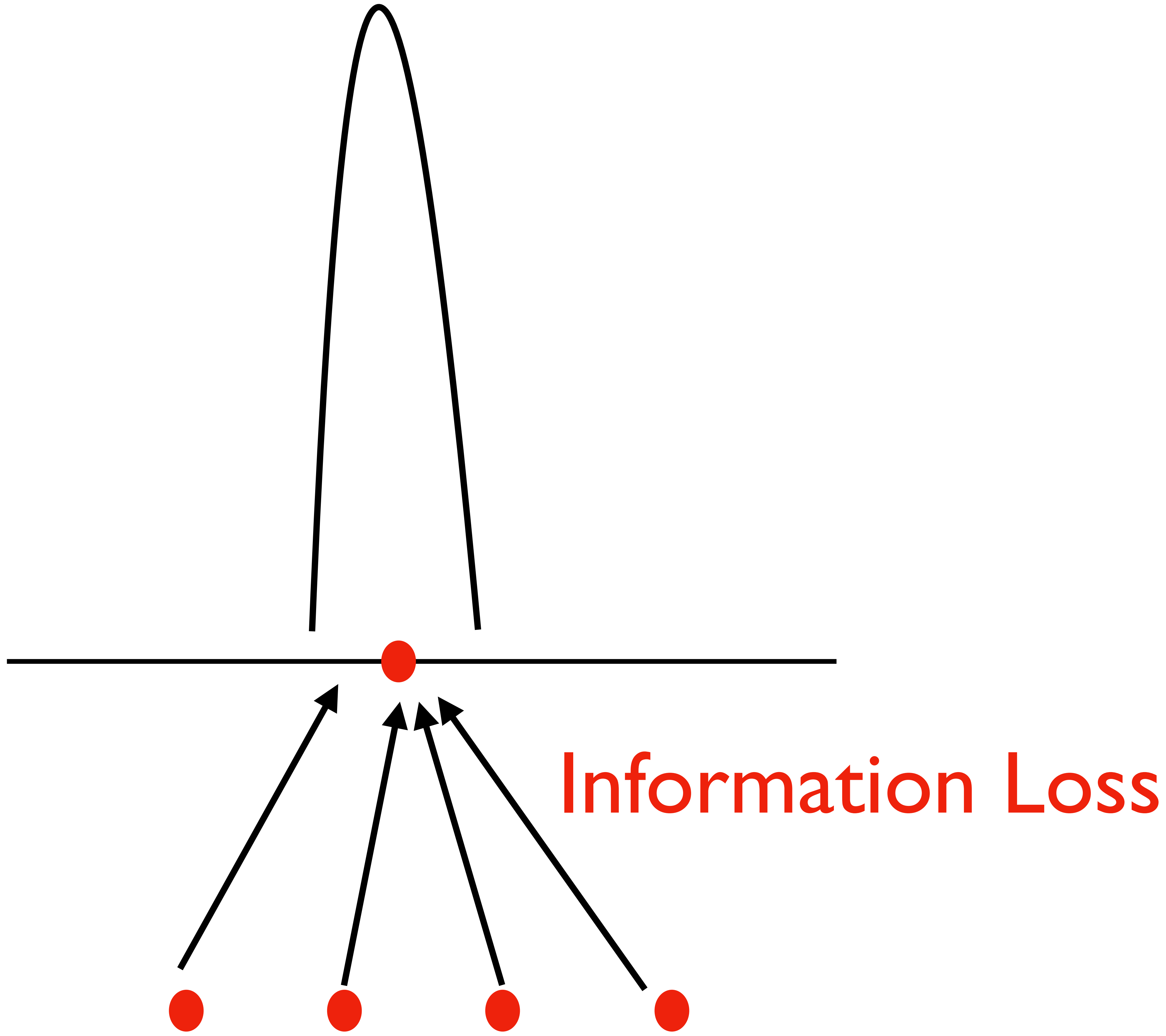
# Why Invertible

Example of Markov prior

$$\max \ \log p_{\mathrm{GHMM}}(f_\phi(\boldsymbol{x}))$$

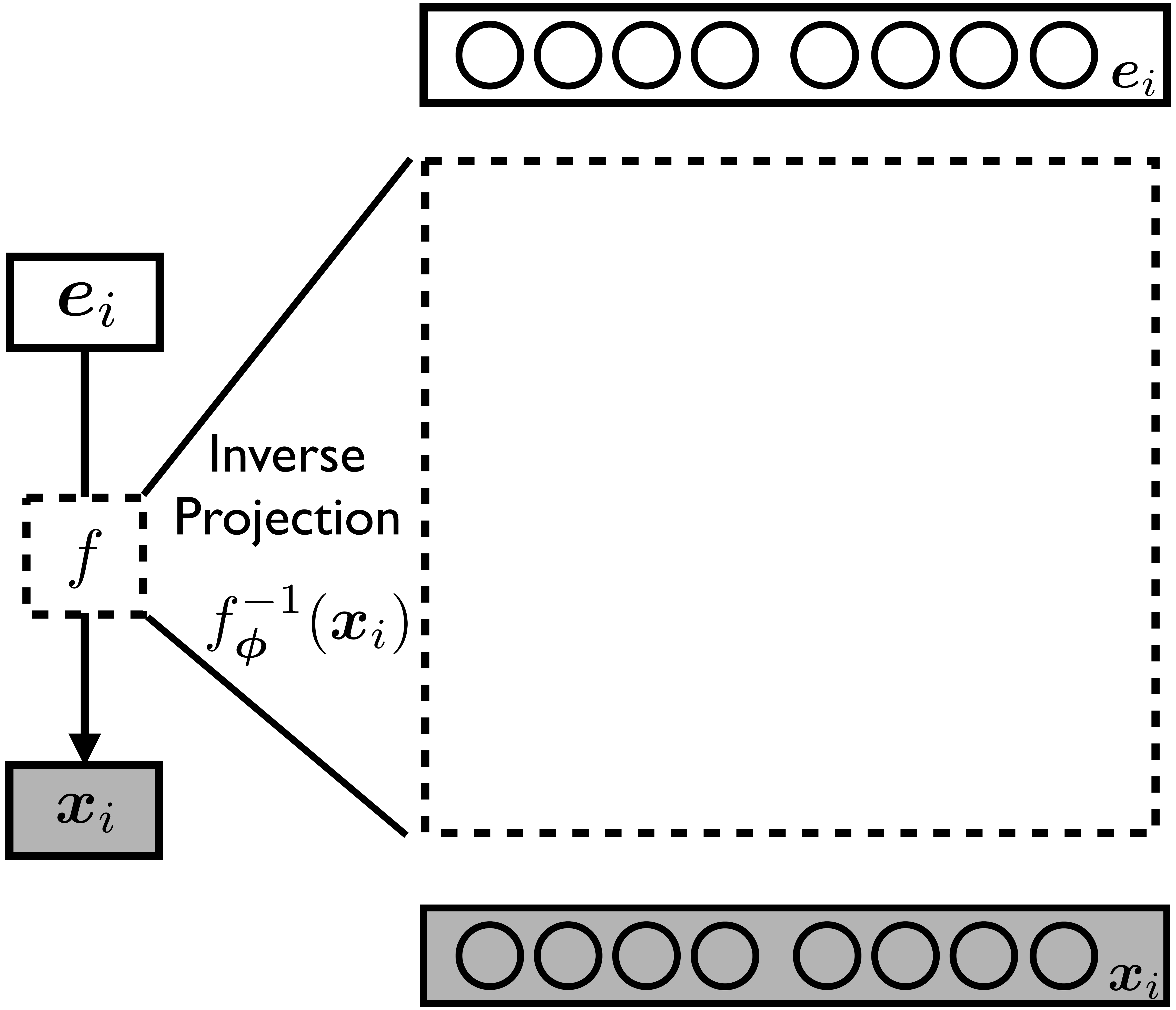
?



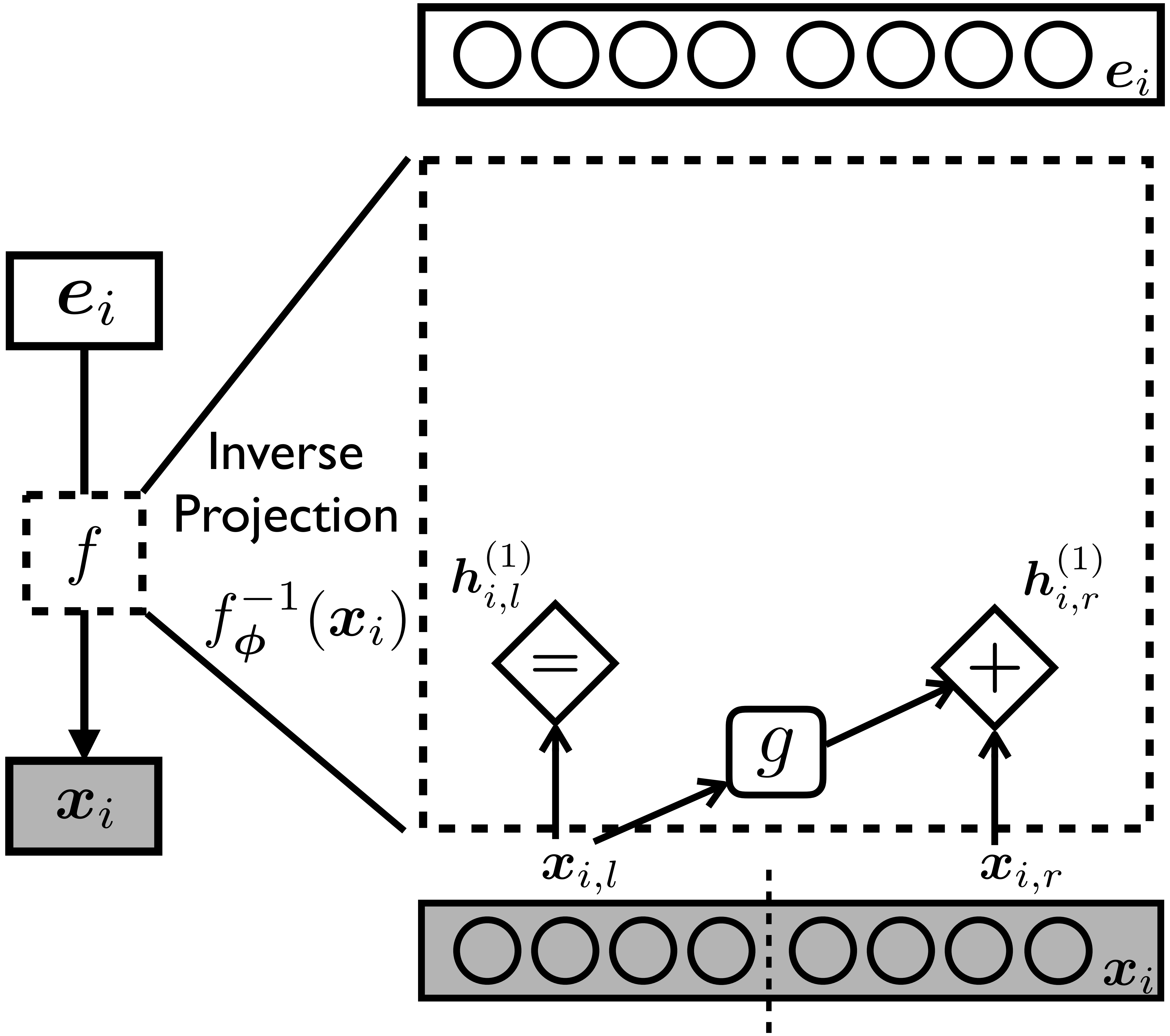
Language Technologies Institute

# Learning with Inverse Projection



Language Technologies Institute

# Learning with Inverse Projection



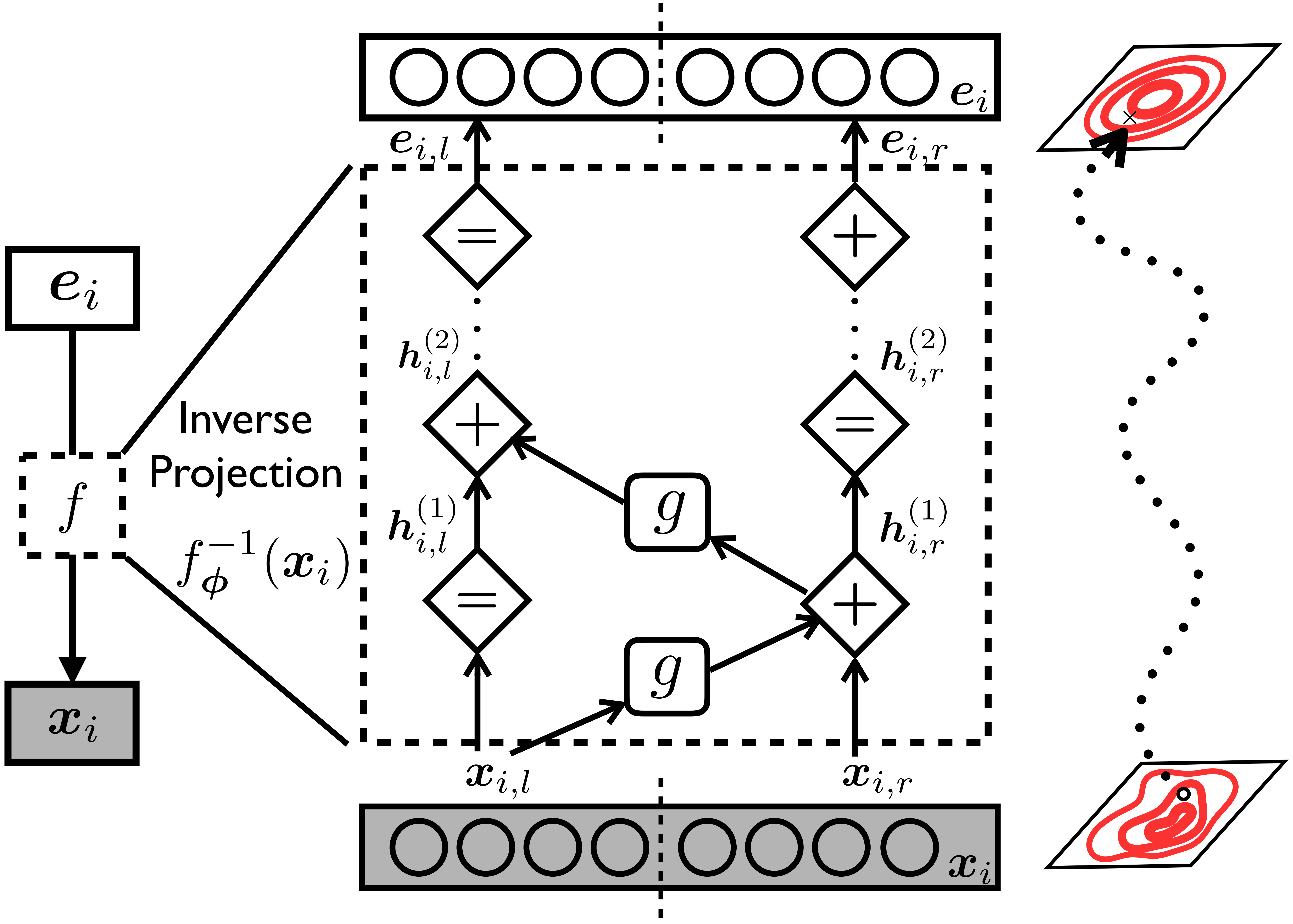

$$\boldsymbol{h}_{i,l}^{(1)} = \boldsymbol{x}_{i,l}$$

$$\boldsymbol{h}_{i,r}^{(1)} = \boldsymbol{x}_{i,r} + g(\boldsymbol{x}_{i,l})$$

$$\begin{bmatrix} 1 & & & & & & \\ & 1 & & & 0 & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & \ddots & & & 1 & & \\ & & & & & & 1 \end{bmatrix}$$

[Dinh et al. 2014]

# Learning with Inverse Projection




$$h_{i,l}^{(1)} = x_{i,l}$$

$$h_{i,r}^{(1)} = x_{i,r} + g(x_{i,l})$$

$$\begin{bmatrix} 1 & & & & & \\ & 1 & & 0 & & \\ & & 1 & & & \\ & & & 1 & & \\ & \ddots & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

[Dinh et al. 2014]

Language Technologies Institute

# Experiments

- Dataset: English Penn Treebank
- POS tagging

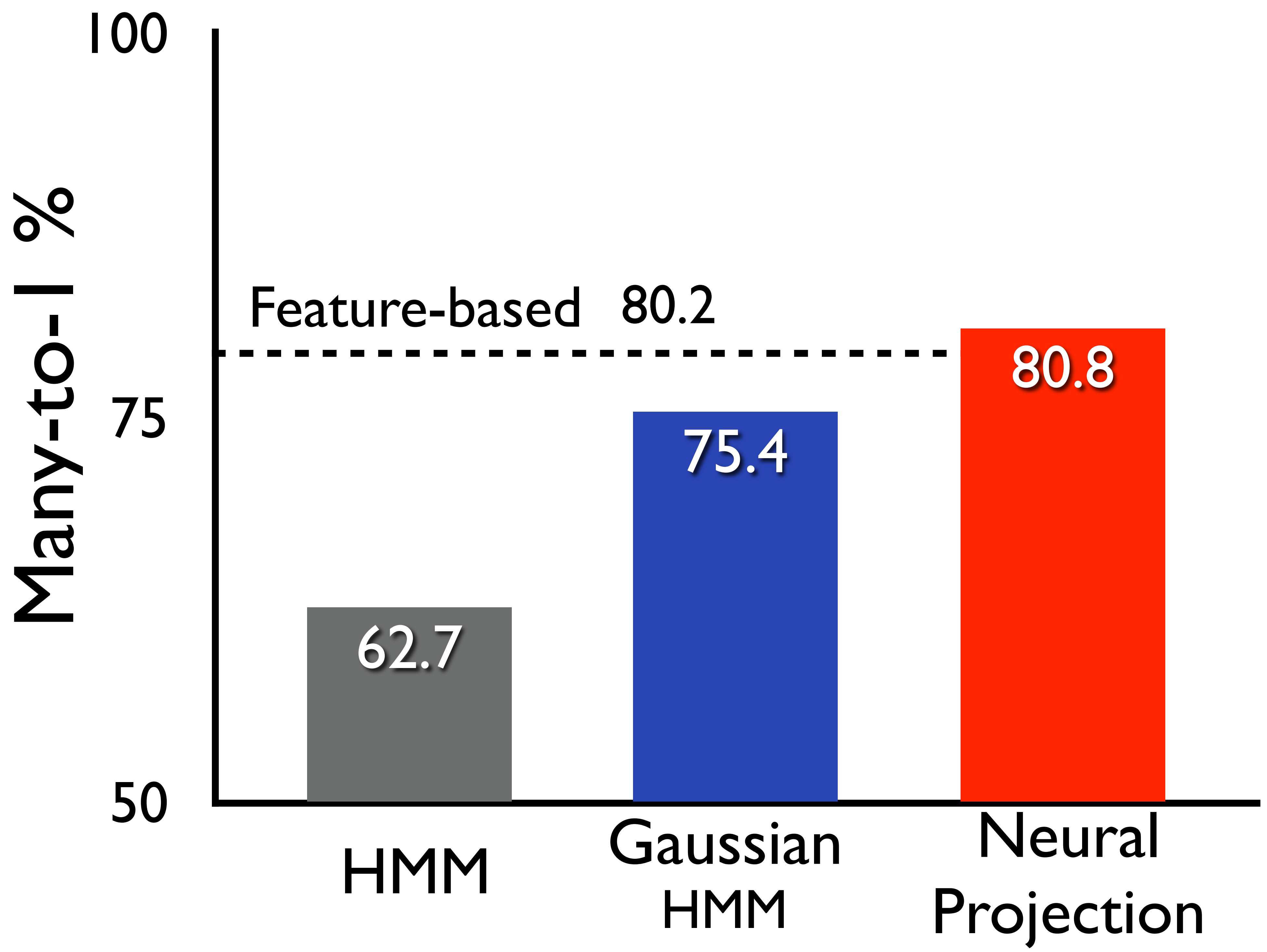
  Trained and tested on whole PTB
- Grammar induction

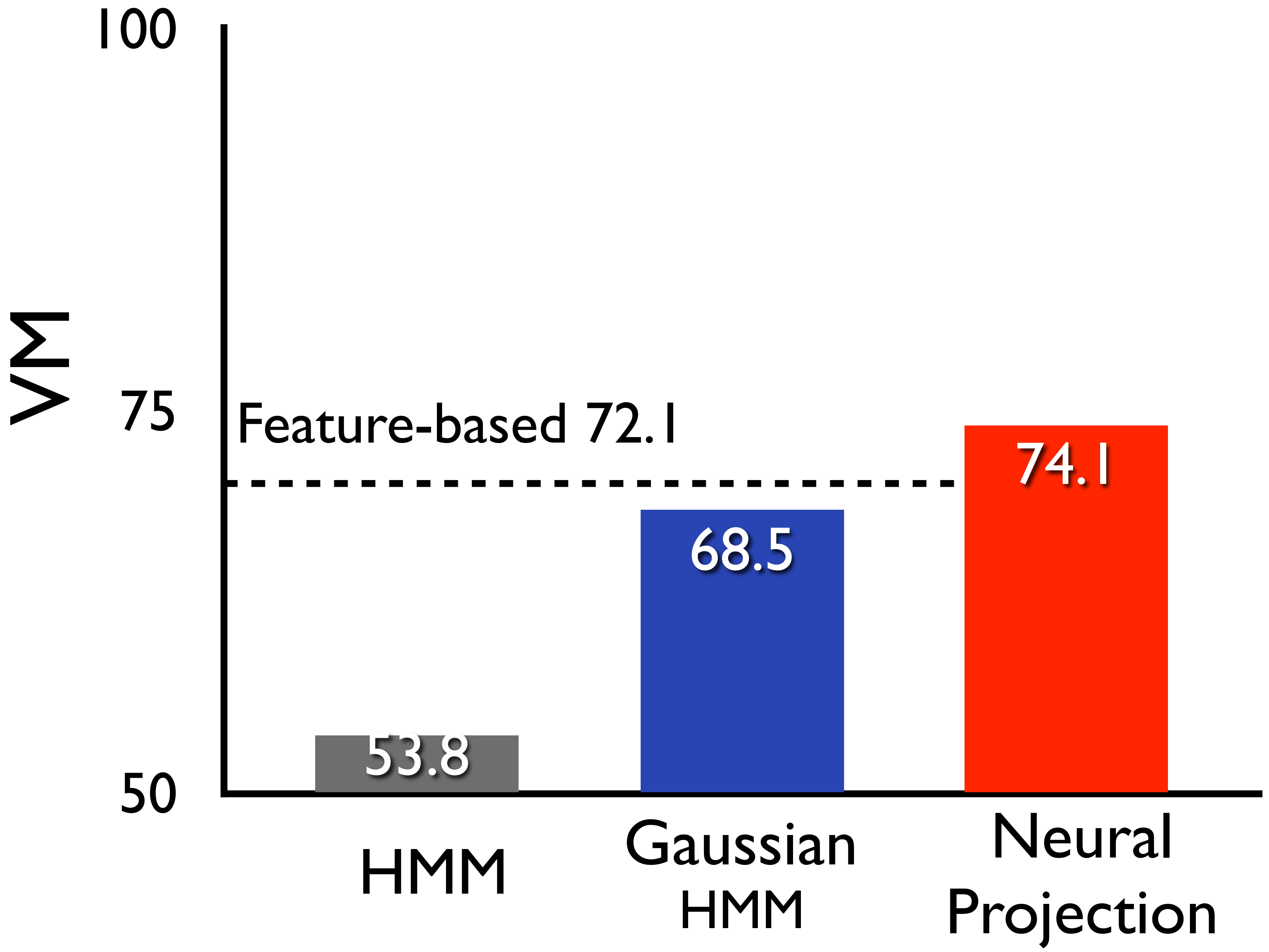
  Trained on sentences of length <= 10 in section 2-21

  Tested on sentences in section 23

Language Technologies Institute

# Part-of-speech Induction

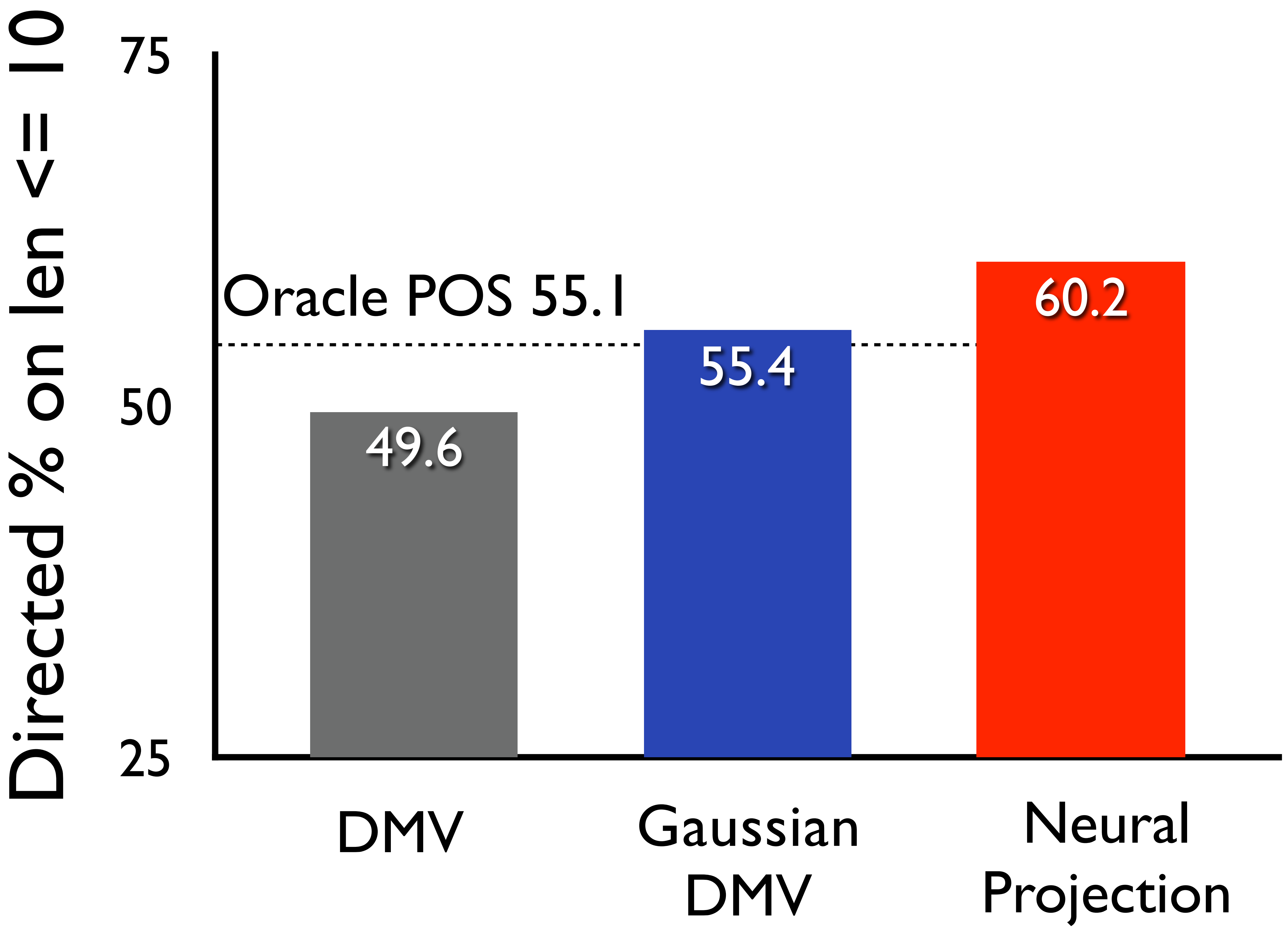


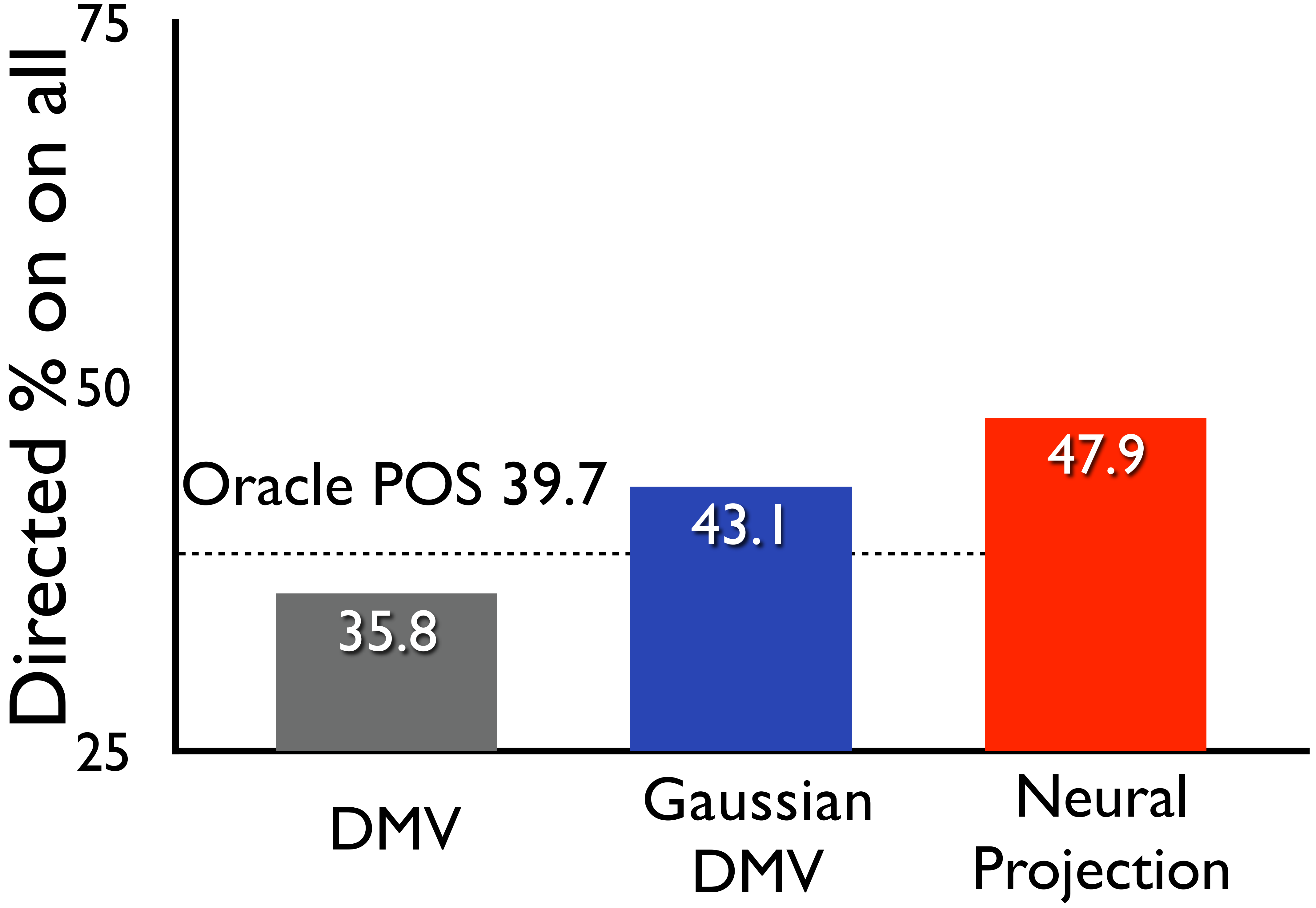




## Outperform feature-based SOTA

Language Technologies Institute

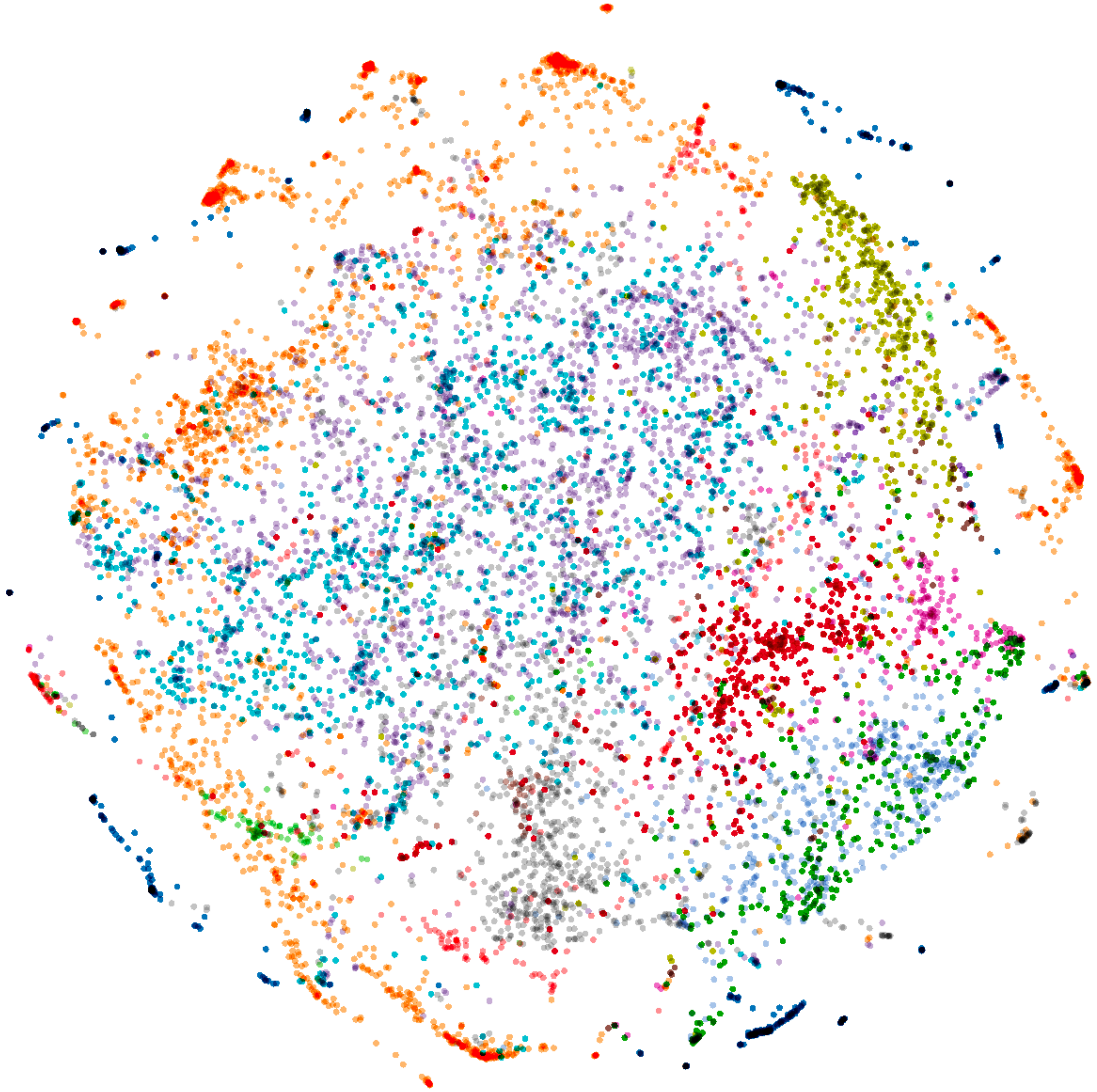
# Dependency Parse Induction



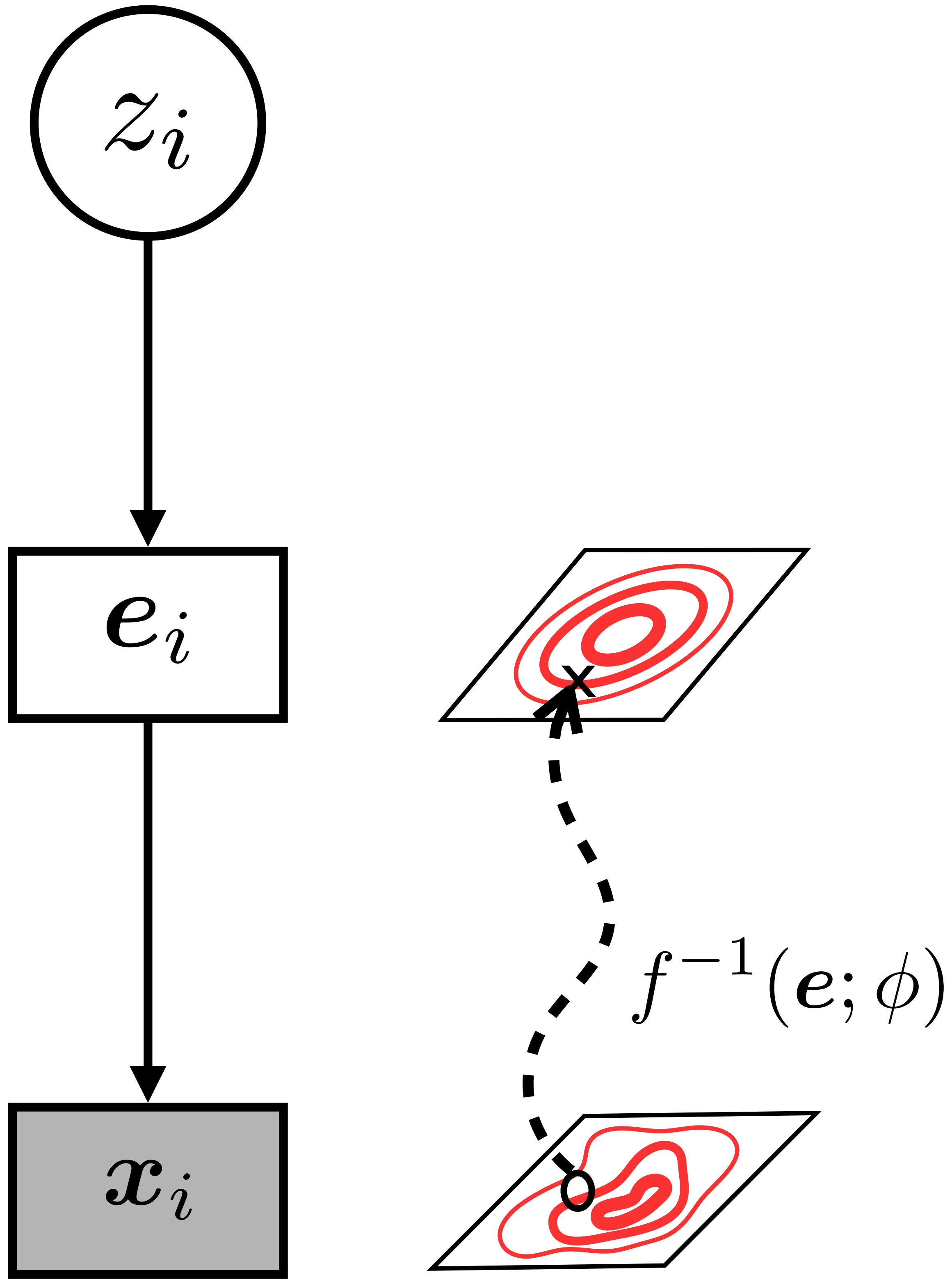
# Original Embedding Space

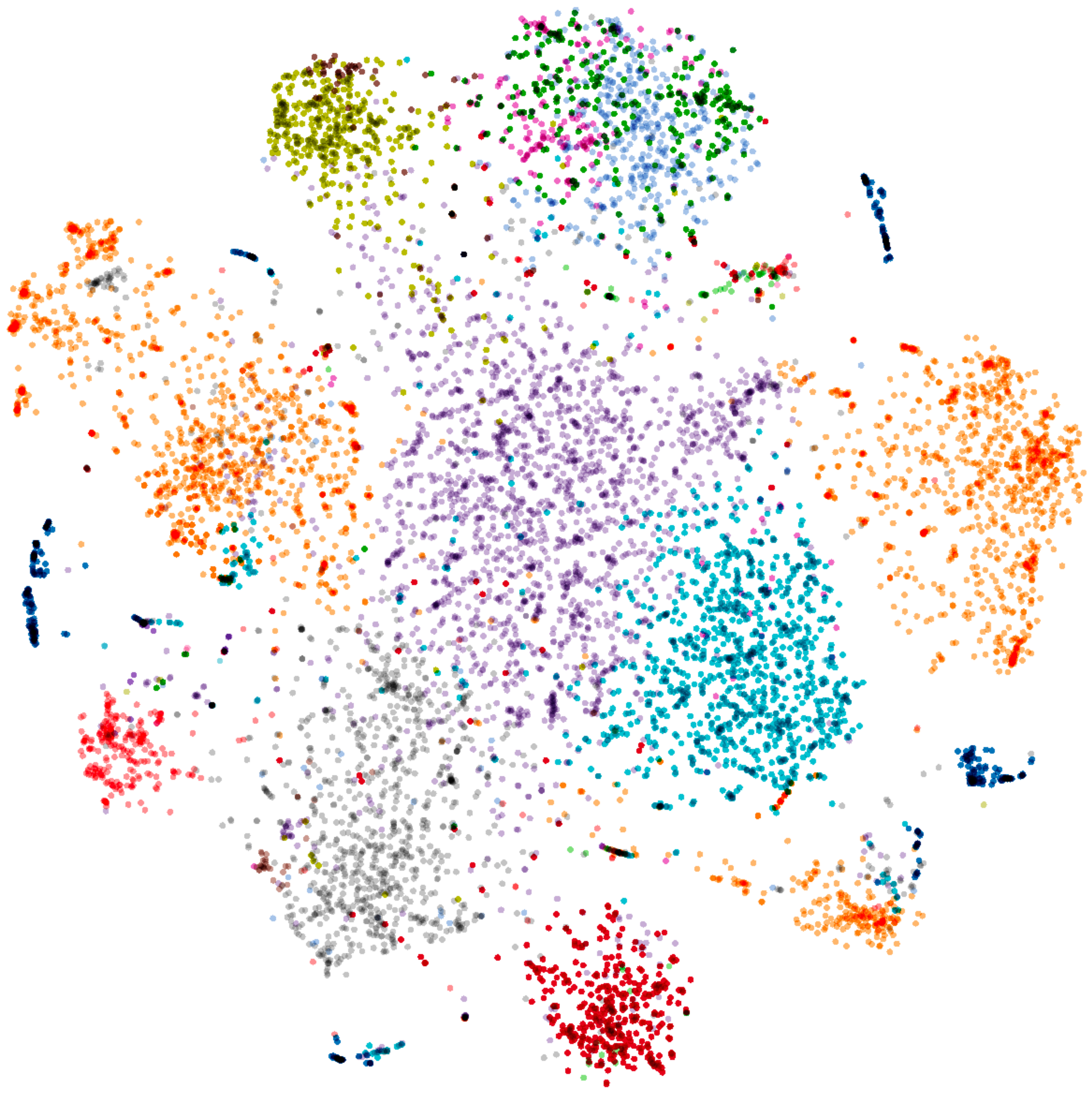
- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number



Language Technologies Institute

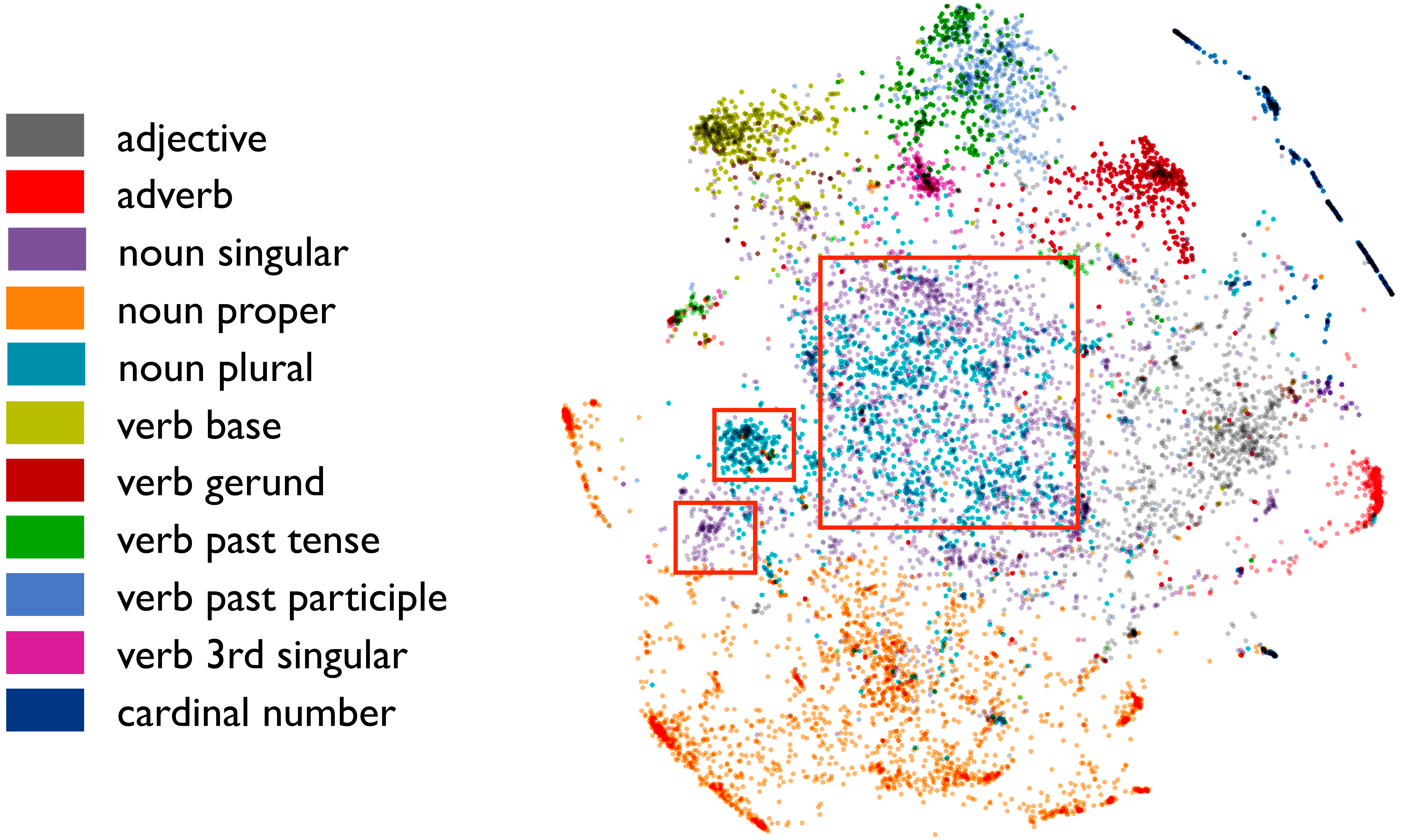
# Projected Embedding Space w/ Markov Prior



Language Technologies Institute

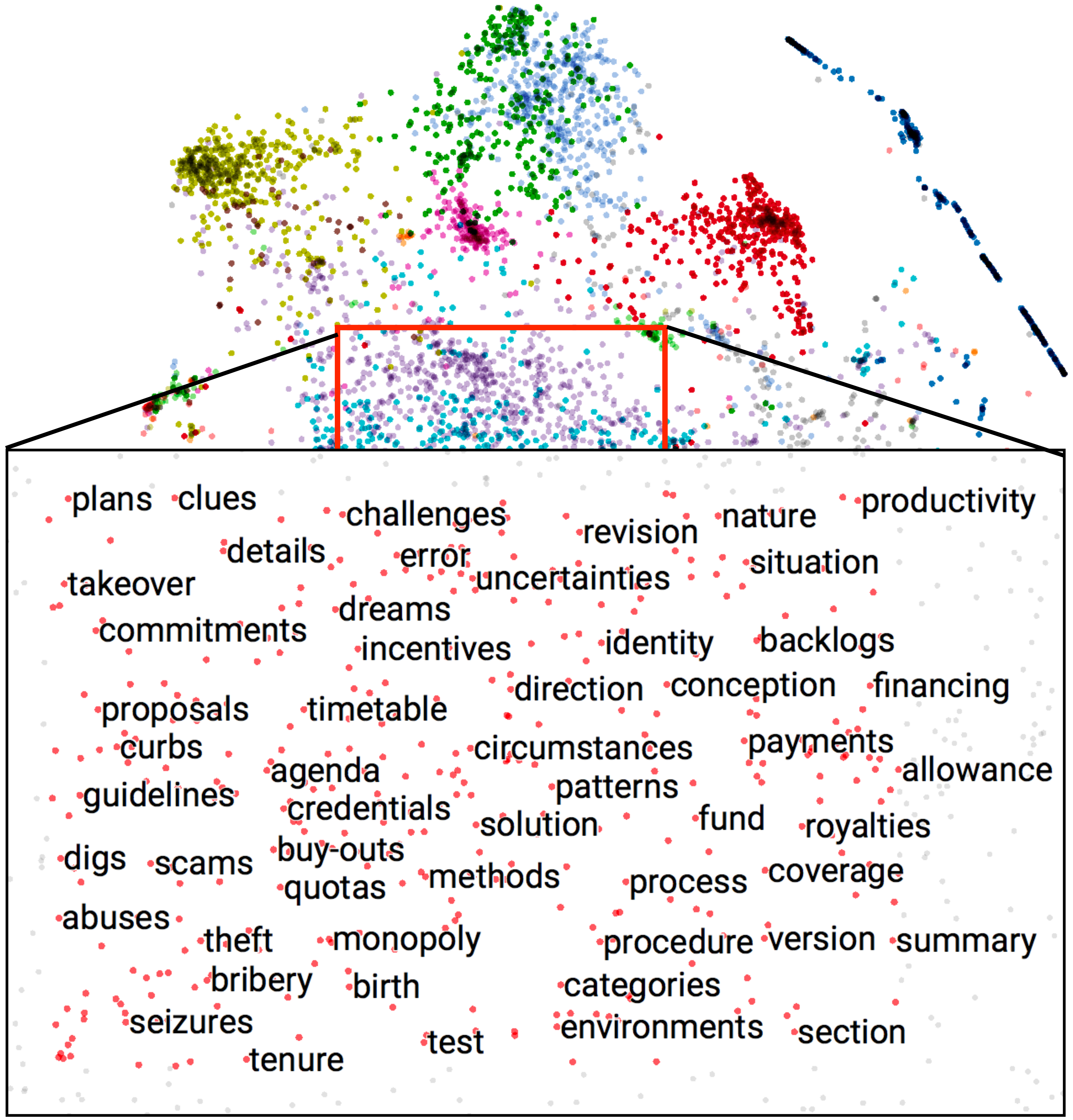
# Projected Embedding Space w/ DMV Prior



Language Technologies Institute

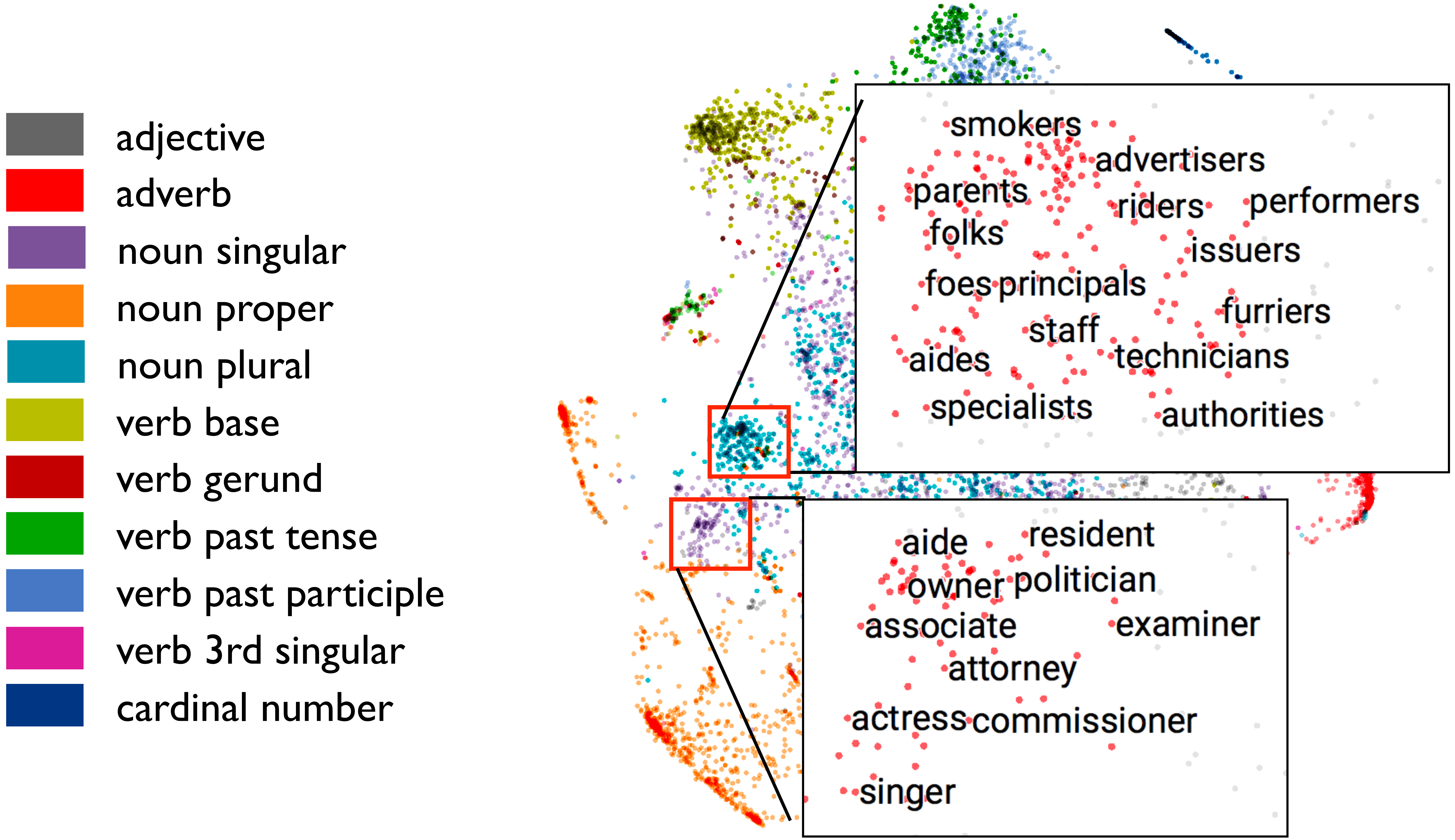
# Projected Embedding Space w/ DMV Prior

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number



Language Technologies Institute

# Projected Embedding Space w/ DMV Prior



# Conclusion

Language Technologies Institute

# Learning with Latent Linguistic Structure

- How can we harness the power of neural networks?
  - NN-based learning on top of latent structured representations
- How can we learn on unlabeled data?
  - Structured variational auto-encoders for semi-supervised learning
  - Structured priors and invertible transformations for unsupervised learning