Novel Inference, Training and Decoding Methods over Translation Forests

Zhifei Li

Center for Language and Speech Processing

Computer Science Department

Johns Hopkins University

Advisor: Sanjeev Khudanpur

Co-advisor: Jason Eisner

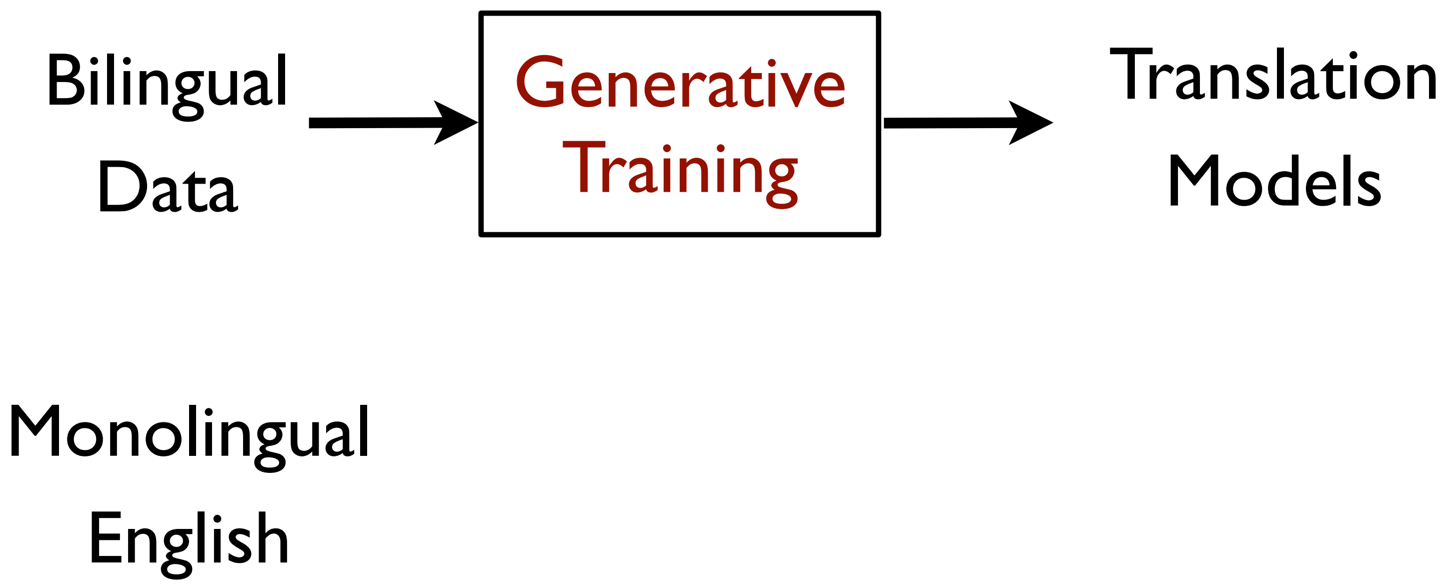
Statistical Machine Translation Pipeline

Statistical Machine Translation Pipeline

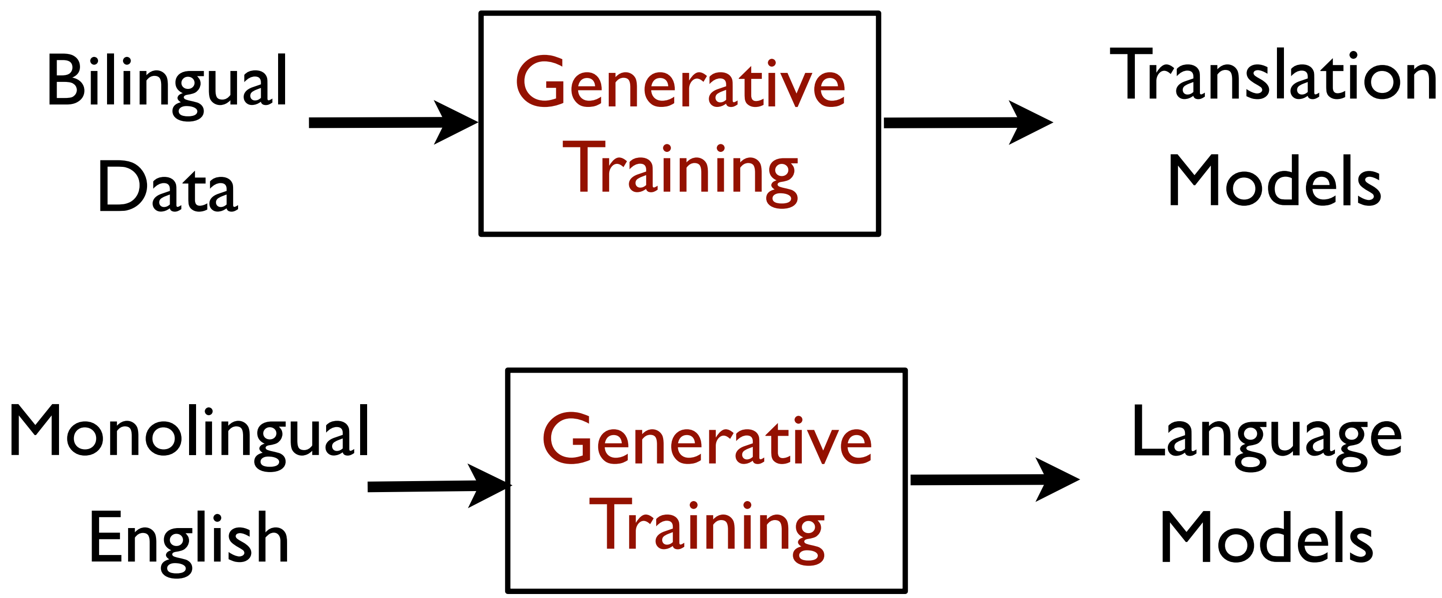
Bilingual
Data

Statistical Machine Translation Pipeline

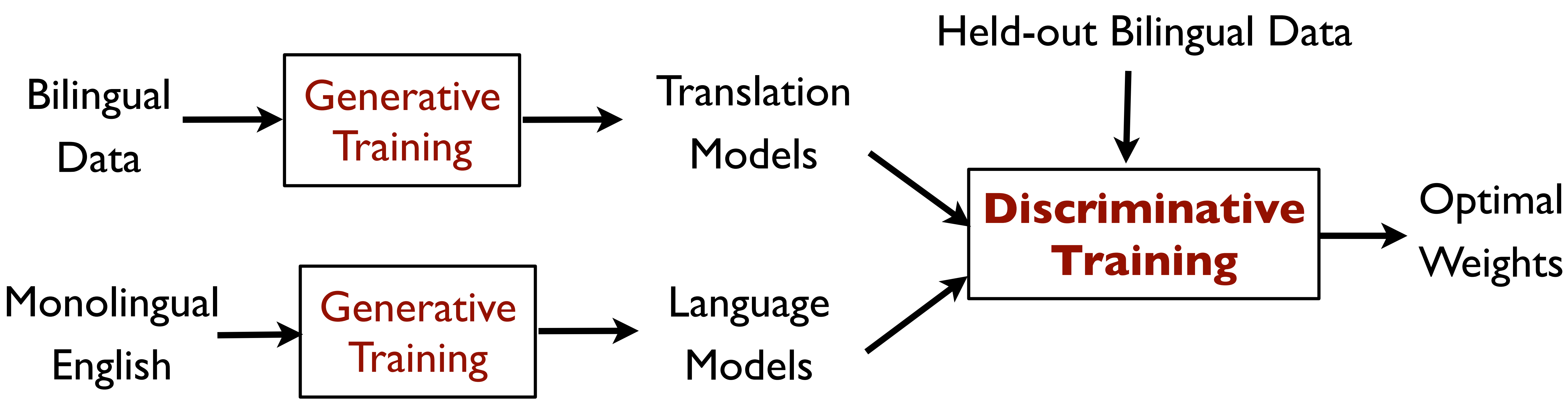
Statistical Machine Translation Pipeline



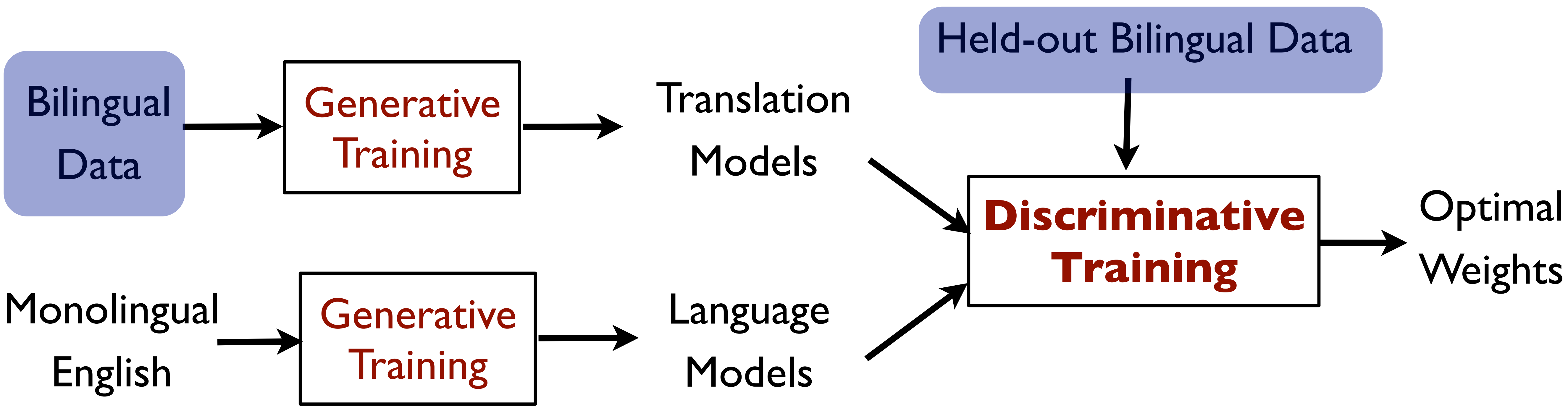
Statistical Machine Translation Pipeline



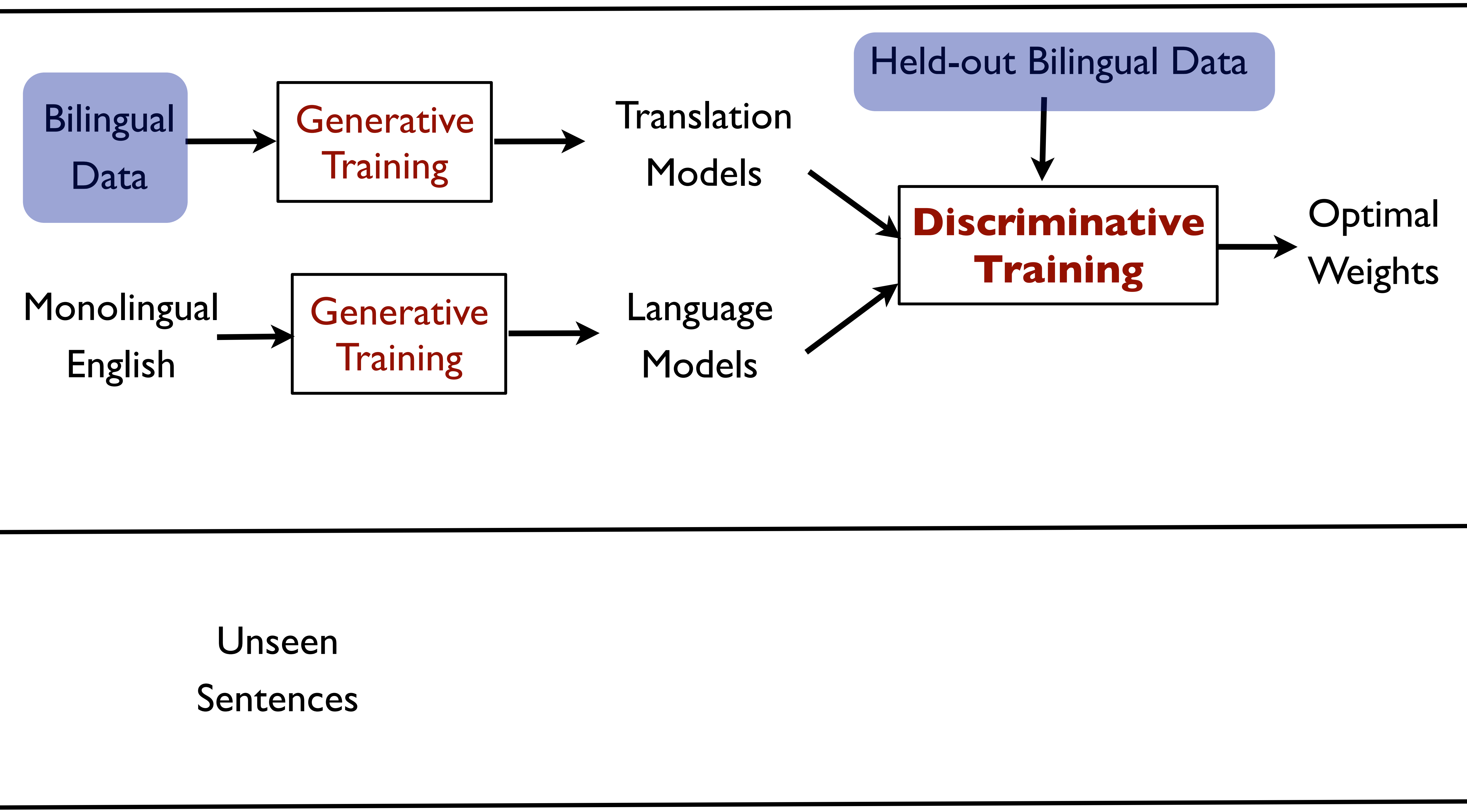
Statistical Machine Translation Pipeline



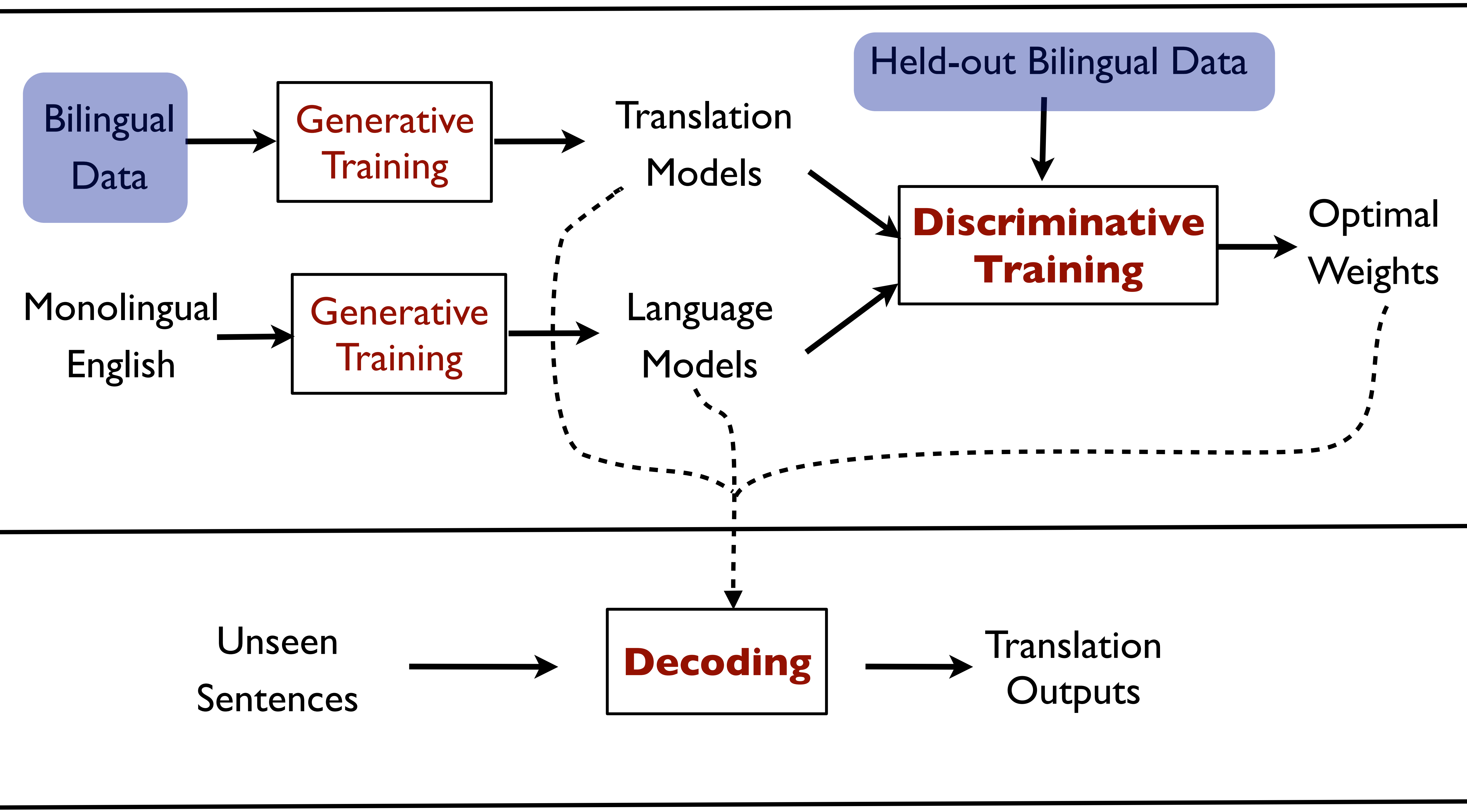
Statistical Machine Translation Pipeline



Statistical Machine Translation Pipeline



Statistical Machine Translation Pipeline



Training a Translation Model

Training a Translation Model

垫子　上　的　猫

dianzi shang de mao

Training a Translation Model

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

Training a Translation Model

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

Training a Translation Model

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$X \rightarrow \langle$ dianzi shang , the mat $\rangle$

Training a Translation Model

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$$X \rightarrow \langle \text{ dianzi shang , the mat } \rangle$$

$$X \rightarrow \langle \text{ mao , a cat } \rangle$$

Training a Translation Model

垫子 上 的 猫

dianzi shang de

on the mat

$$X \rightarrow \langle\ \text{dianzi shang},\ \text{the mat}\ \rangle$$
$$X \rightarrow \langle\ \text{mao},\ \text{a cat}\ \rangle$$

Training a Translation Model

垫子 上 的 猫

dianzi shang de X_0

X_0 on the mat

$X \rightarrow \langle$ dianzi shang , the mat $\rangle$

$X \rightarrow \langle$ mao , a cat $\rangle$

Training a Translation Model

垫子 上 的 猫

dianzi shang de X_0

X_0 on the mat

$$X \rightarrow \langle\ \text{dianzi shang}\ ,\ \text{the mat}\ \rangle$$

$$X \rightarrow \langle\ \text{mao}\ ,\ \text{a cat}\ \rangle$$

$$X \rightarrow \langle\ \text{dianzi shang de}\ X_0\ ,\ X_0\ \text{on the mat}\ \rangle$$

Training a Translation Model

垫子 上 的 猫

X_0 de mao

a cat on X_0

$$X \rightarrow \langle \text{ dianzi shang , the mat } \rangle$$

$$X \rightarrow \langle \text{ mao , a cat } \rangle$$

$$X \rightarrow \langle \text{ dianzi shang de } X_0 \text{ , } X_0 \text{ on the mat } \rangle$$

Training a Translation Model

垫子 上 的 猫

X_0 de mao

a cat on X_0

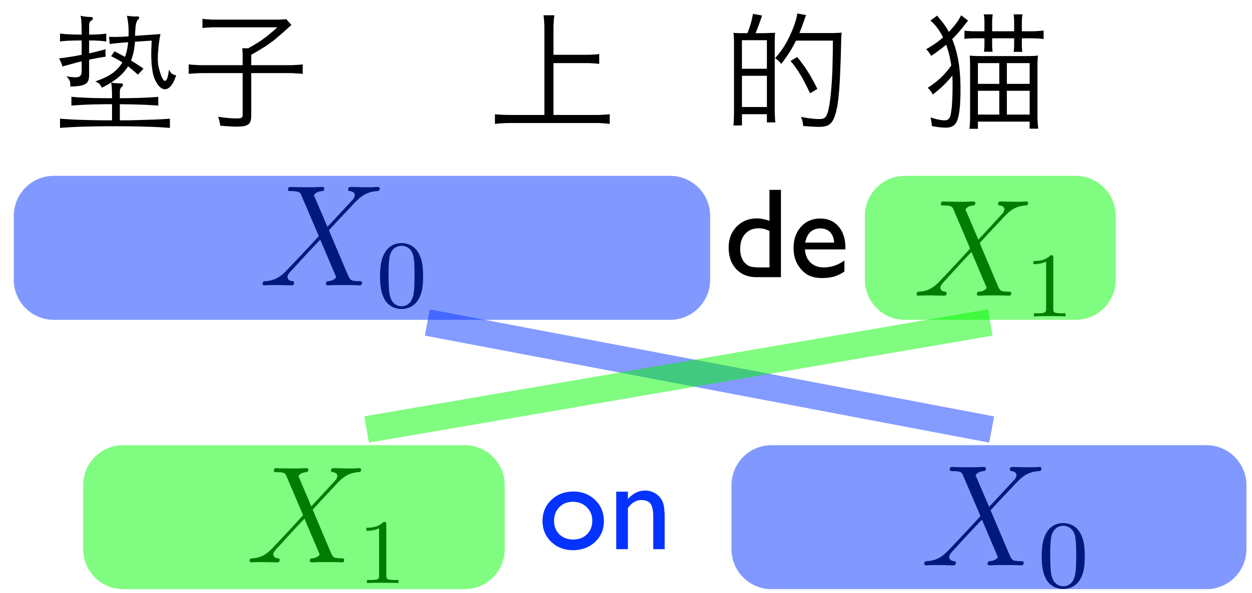
$$X \rightarrow \langle \text{ dianzi shang , the mat } \rangle$$

$$X \rightarrow \langle \text{ mao , a cat } \rangle$$

$$X \rightarrow \langle \text{ dianzi shang de } X_0 \text{ , } X_0 \text{ on the mat } \rangle$$

$$X \rightarrow \langle X_0 \text{ de mao , a cat on } X_0 \rangle$$

Training a Translation Model



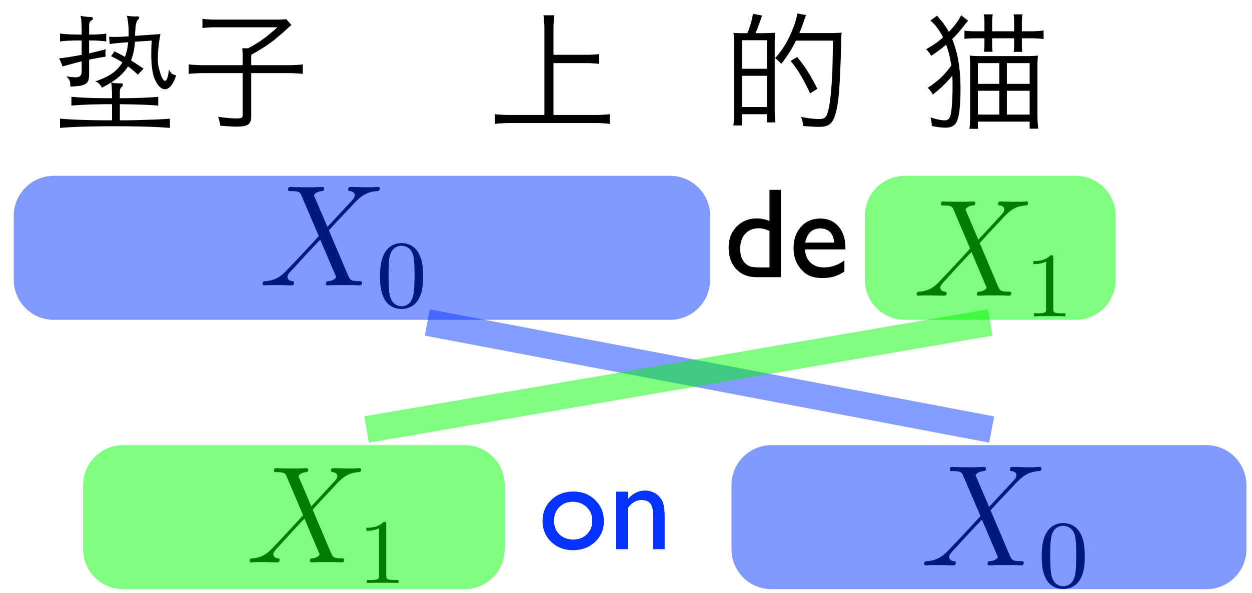
$$X \rightarrow \langle \text{ dianzi shang , the mat } \rangle$$

$$X \rightarrow \langle \text{ mao , a cat } \rangle$$

$$X \rightarrow \langle \text{ dianzi shang de } X_0 \text{ , } X_0 \text{ on the mat } \rangle$$

$$X \rightarrow \langle X_0 \text{ de mao , a cat on } X_0 \rangle$$

Training a Translation Model



$$X \rightarrow \langle \text{ dianzi shang , the mat } \rangle$$

$$X \rightarrow \langle \text{ mao , a cat } \rangle$$

$$X \rightarrow \langle \text{ dianzi shang de } X_0 \text{ , } X_0 \text{ on the mat } \rangle$$

$$X \rightarrow \langle X_0 \text{ de mao , a cat on } X_0 \rangle$$

$$X \rightarrow \langle X_0 \text{ de } X_1 \text{ , } X_1 \text{ on } X_0 \rangle$$

Decoding a Test Sentence

Decoding a Test Sentence

Decoding a Test Sentence

垫子　上　的　狗

Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

X	$\rightarrow$	$\langle$ dianzi shang , the mat $\rangle$
X	$\rightarrow$	$\langle$ gou , the dog $\rangle$
X	$\rightarrow$	$\langle X_0$ de X_1 , X_1 on $X_0 \rangle$
S	$\rightarrow$	$\langle X_0 , X_0 \rangle$

Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

X	$\rightarrow$	$\langle$ dianzi shang , the mat $\rangle$
X	$\rightarrow$	$\langle$ gou , the dog $\rangle$
X	$\rightarrow$	$\langle X_0$ de X_1 , X_1 on $X_0 \rangle$
S	$\rightarrow$	$\langle X_0 , X_0 \rangle$

Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

X	$\rightarrow$	$\langle$ dianzi shang , the mat $\rangle$
X	$\rightarrow$	$\langle$ gou , the dog $\rangle$
X	$\rightarrow$	$\langle X_0$ de X_1 , X_1 on $X_0 \rangle$
S	$\rightarrow$	$\langle X_0 , X_0 \rangle$

Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

X	$\rightarrow$	$\langle$ dianzi shang , the mat $\rangle$
X	$\rightarrow$	$\langle$ gou , the dog $\rangle$
X	$\rightarrow$	$\langle X_0$ de X_1 , X_1 on $X_0 \rangle$
S	$\rightarrow$	$\langle X_0 , X_0 \rangle$

dianzi shang de gou

Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

$$X \rightarrow \langle \text{ dianzi shang }, \text{ the mat } \rangle$$
$$X \rightarrow \langle \text{ gou }, \text{ the dog } \rangle$$
$$X \rightarrow \langle X_0 \text{ de } X_1 , X_1 \text{ on } X_0 \rangle$$
$$S \rightarrow \langle X_0 , X_0 \rangle$$

$\mathbf{X} \rightarrow \langle \textbf{dianzi shang}, \textbf{the mat} \rangle$

dianzi shang de gou

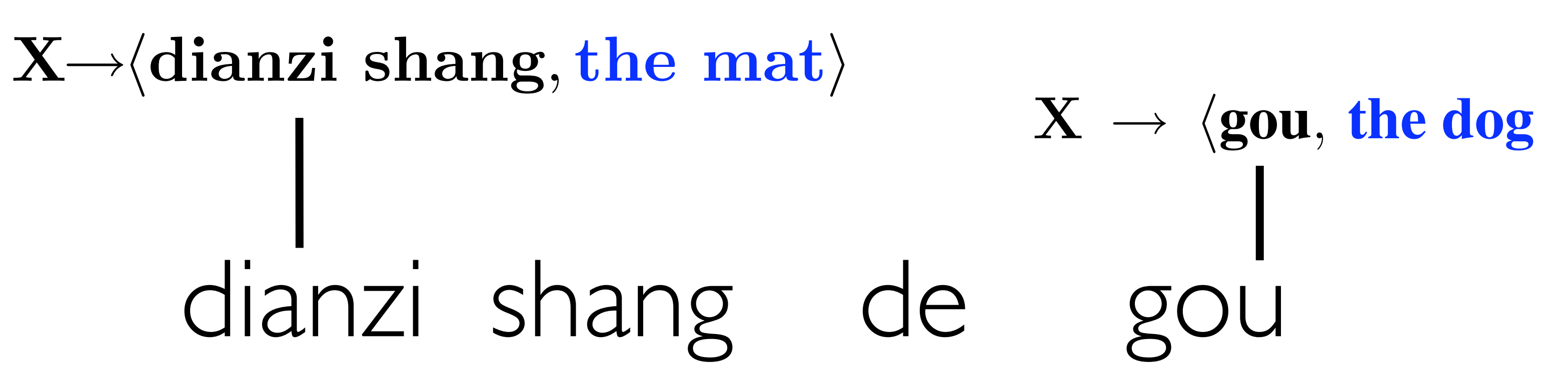
Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

$$X \rightarrow \langle \text{ dianzi shang }, \text{ the mat } \rangle$$
$$X \rightarrow \langle \text{ gou }, \text{ the dog } \rangle$$
$$X \rightarrow \langle X_0 \text{ de } X_1 , X_1 \text{ on } X_0 \rangle$$
$$S \rightarrow \langle X_0 , X_0 \rangle$$



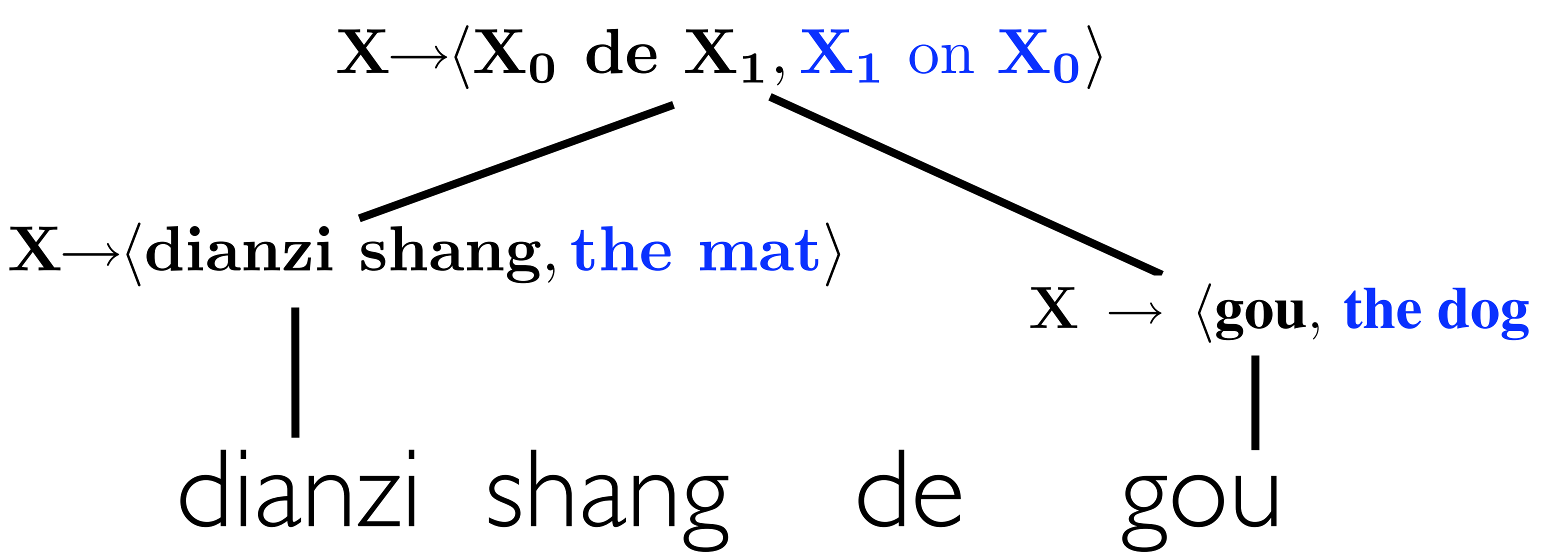
Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

$$X \rightarrow \langle \text{ dianzi shang }, \text{ the mat } \rangle$$
$$X \rightarrow \langle \text{ gou }, \text{ the dog } \rangle$$
$$X \rightarrow \langle X_0 \text{ de } X_1 , X_1 \text{ on } X_0 \rangle$$
$$S \rightarrow \langle X_0 , X_0 \rangle$$



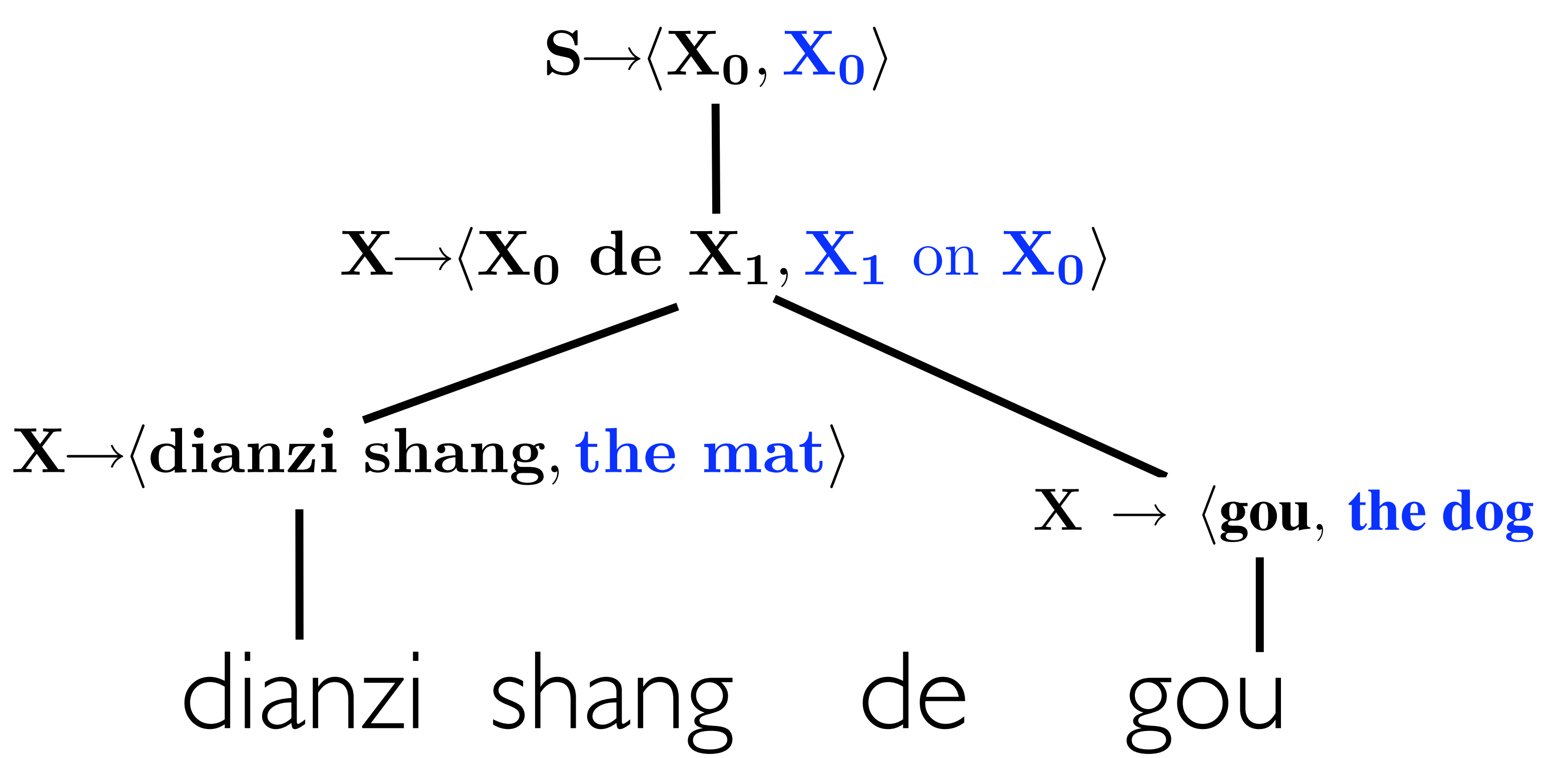
Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

$$X \rightarrow \langle \text{ dianzi shang }, \text{ the mat } \rangle$$
$$X \rightarrow \langle \text{ gou }, \text{ the dog } \rangle$$
$$X \rightarrow \langle X_0 \text{ de } X_1 , X_1 \text{ on } X_0 \rangle$$
$$S \rightarrow \langle X_0 , X_0 \rangle$$



Decoding a Test Sentence

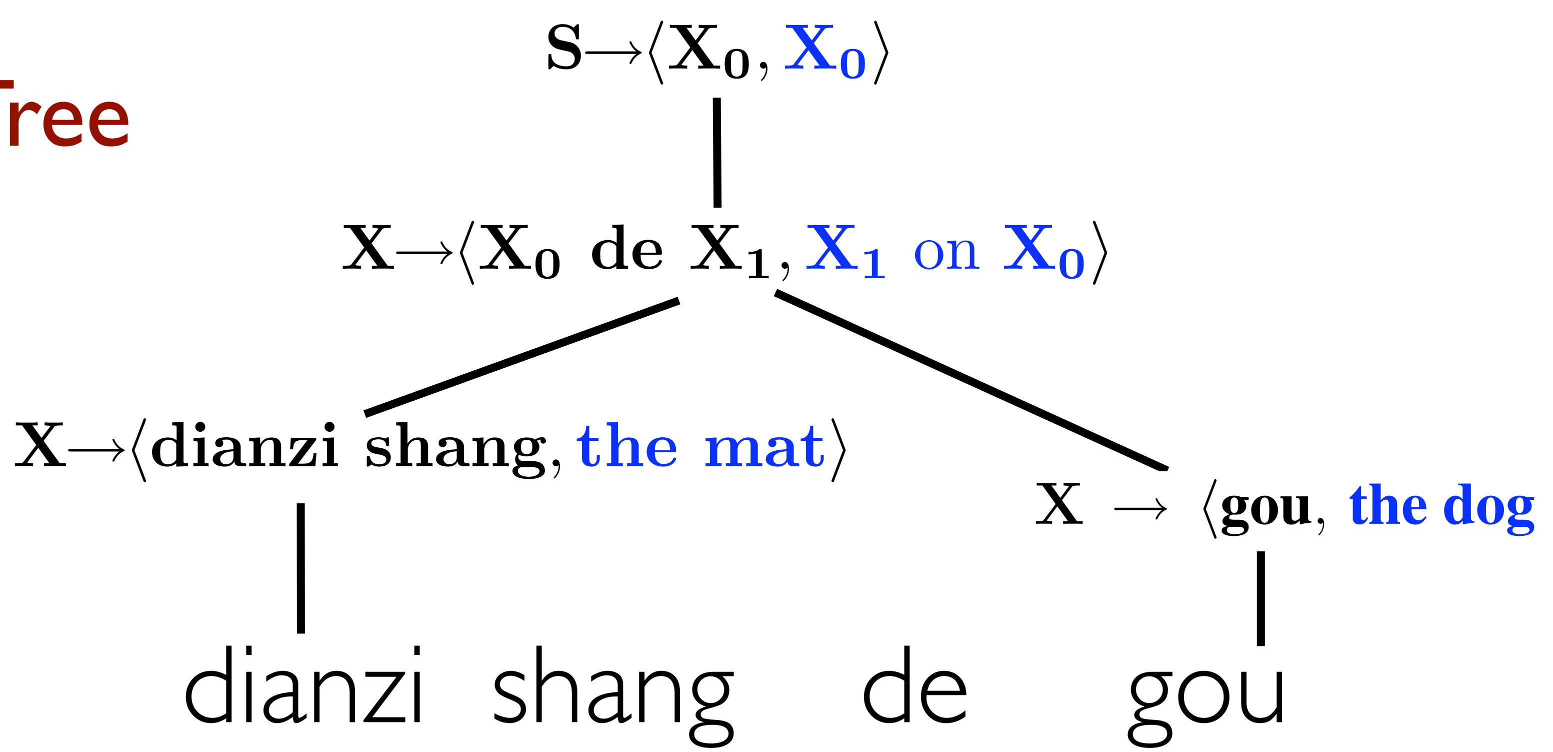
垫子 上 的 狗

dianzi shang de gou

the dog on the mat

$$X \rightarrow \langle \text{ dianzi shang }, \text{ the mat } \rangle$$
$$X \rightarrow \langle \text{ gou }, \text{ the dog } \rangle$$
$$X \rightarrow \langle X_0 \text{ de } X_1 , X_1 \text{ on } X_0 \rangle$$
$$S \rightarrow \langle X_0 , X_0 \rangle$$

Derivation Tree



Decoding a Test Sentence

垫子 上 的 狗

dianzi shang de gou

the dog on the mat

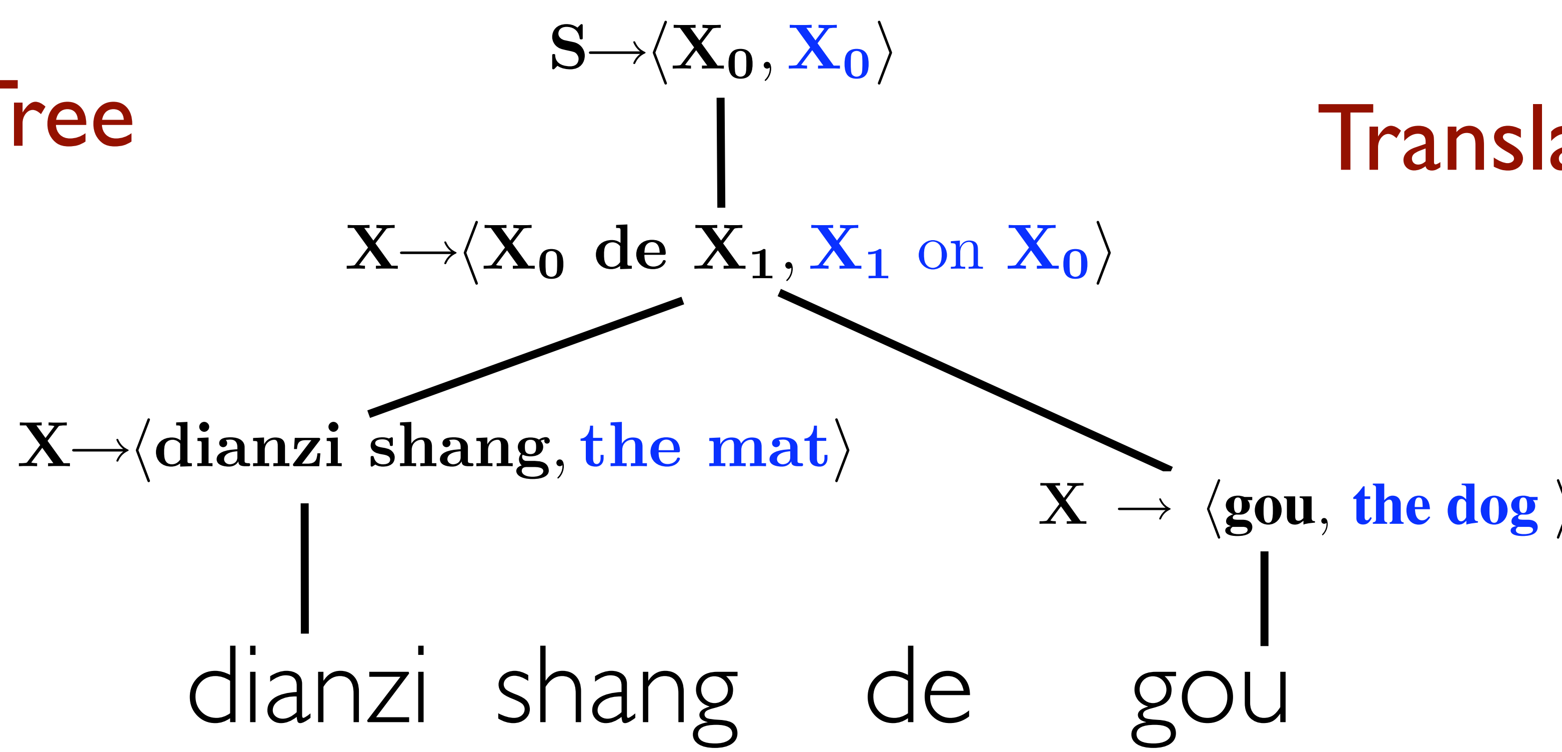
$X \rightarrow \langle$ dianzi shang , the mat $\rangle$

$X \rightarrow \langle$ gou , the dog $\rangle$

$X \rightarrow \langle X_0$ de X_1 , X_1 on $X_0 \rangle$

$S \rightarrow \langle X_0 , X_0 \rangle$

Derivation Tree



Translation is easy?

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_1}\ \mathrm{on}\ \mathbf{X_0} \rangle$$

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_1}\ \mathrm{on}\ \mathbf{X_0} \rangle$$

---

zhongguo de shoudu

capital of China

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_1}\ \mathrm{on}\ \mathbf{X_0} \rangle$$

---

zhongguo de shoudu

capital of China

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$$\mathbf{X} \rightarrow \langle \mathbf{X_0} \ \mathbf{de} \ \mathbf{X_1}, \mathbf{X_1} \ \text{on} \ \mathbf{X_0} \rangle$$

---

zhongguo de shoudu

capital of China

$$\mathbf{X} \rightarrow \langle \mathbf{X_0} \ \mathbf{de} \ \mathbf{X_1}, \mathbf{X_1} \ \text{of} \ \mathbf{X_0} \rangle$$

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_1}\ \text{on}\ \mathbf{X_0} \rangle$

zhongguo de shoudu

capital of China

$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_1}\ \text{of}\ \mathbf{X_0} \rangle$

wo de mao

my cat

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$$\mathbf{X} \rightarrow \langle \mathbf{X}_0 \ \mathbf{de} \ \mathbf{X}_1, \mathbf{X}_1 \ \text{on} \ \mathbf{X}_0 \rangle$$

---

zhongguo de shoudu

capital of China

$$\mathbf{X} \rightarrow \langle \mathbf{X}_0 \ \mathbf{de} \ \mathbf{X}_1, \mathbf{X}_1 \ \text{of} \ \mathbf{X}_0 \rangle$$

---

wo de mao

my cat

$$\mathbf{X} \rightarrow \langle \mathbf{X}_0 \ \mathbf{de} \ \mathbf{X}_1, \mathbf{X}_0 \ \mathbf{X}_1 \rangle$$

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$$X \rightarrow \langle X_0 \text{ de } X_1, X_1 \text{ on } X_0 \rangle$$

---

zhongguo de shoudu

capital of China

$$X \rightarrow \langle X_0 \text{ de } X_1, X_1 \text{ of } X_0 \rangle$$

---

wo de mao

my cat

$$X \rightarrow \langle X_0 \text{ de } X_1, X_0\ X_1 \rangle$$

---

zhifei de mao

zhifei 's cat

Translation Ambiguity

垫子 上 的 猫

dianzi shang de mao

a cat on the mat

$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_1}\ \text{on}\ \mathbf{X_0} \rangle$

---

zhongguo de shoudu

capital of China

$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_1}\ \text{of}\ \mathbf{X_0} \rangle$

---

wo de mao

my cat

$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_0}\ \mathbf{X_1} \rangle$

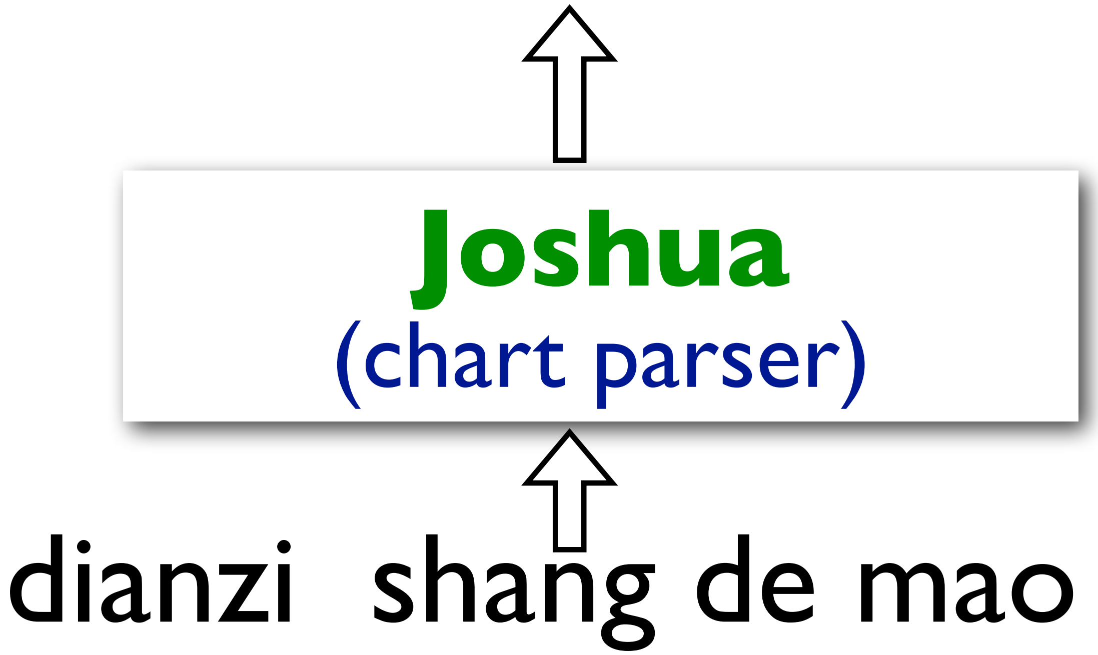
---

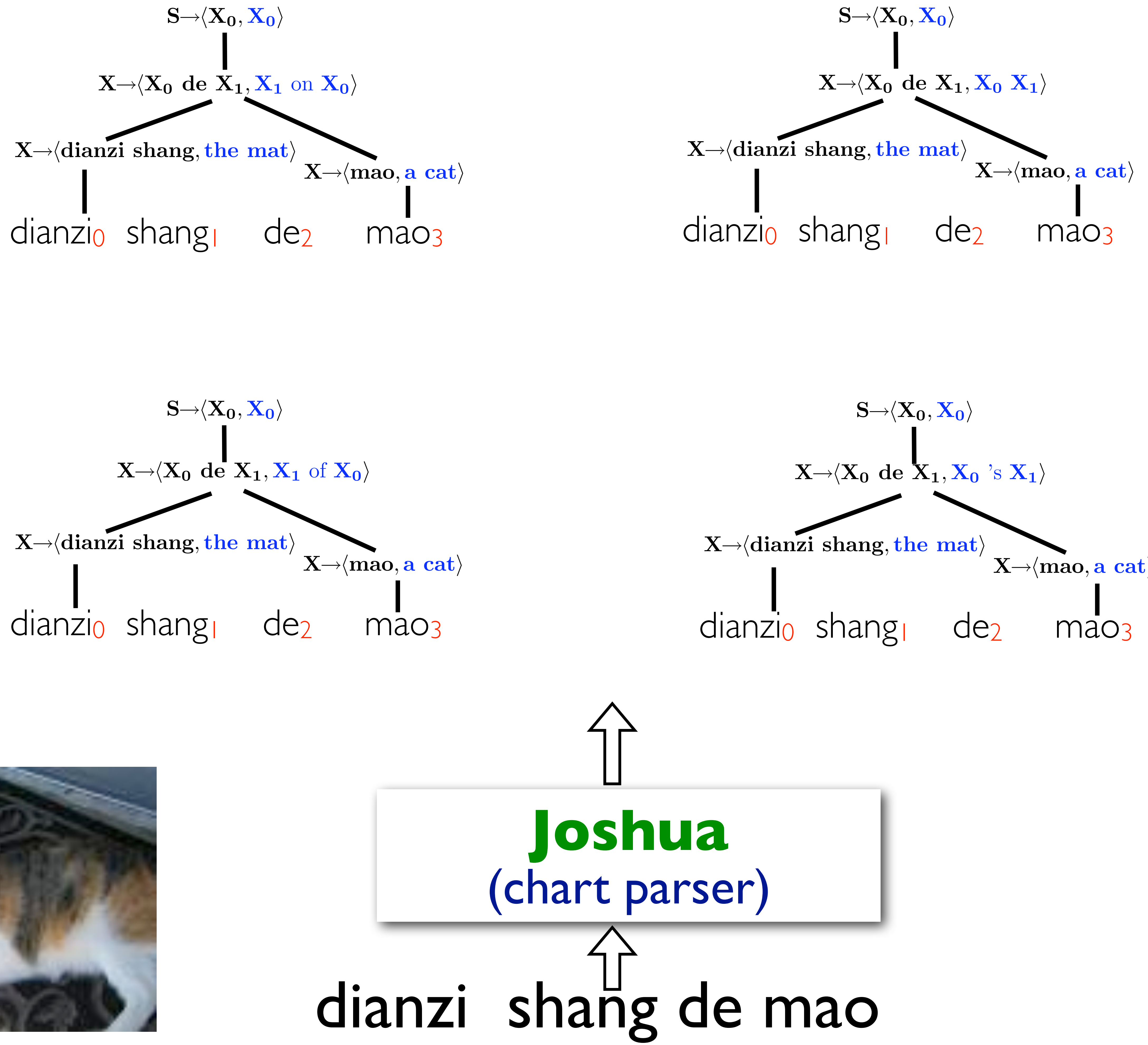
zhifei de mao

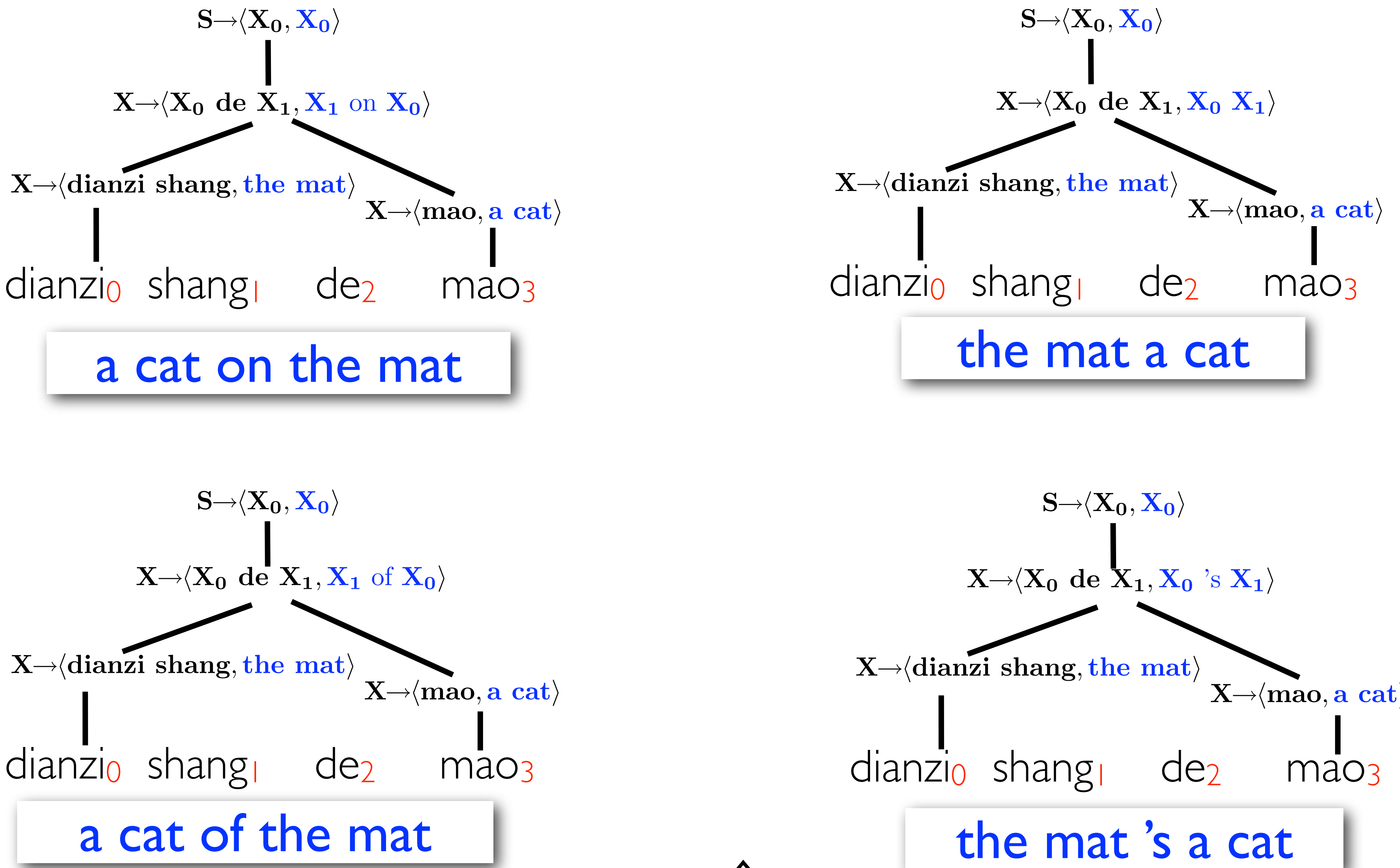
zhifei 's cat

$\mathbf{X} \rightarrow \langle \mathbf{X_0}\ \mathbf{de}\ \mathbf{X_1}, \mathbf{X_0}\ \text{'s}\ \mathbf{X_1} \rangle$

dianzi shang de mao


Joshua
(chart parser)
dianzi shang de mao

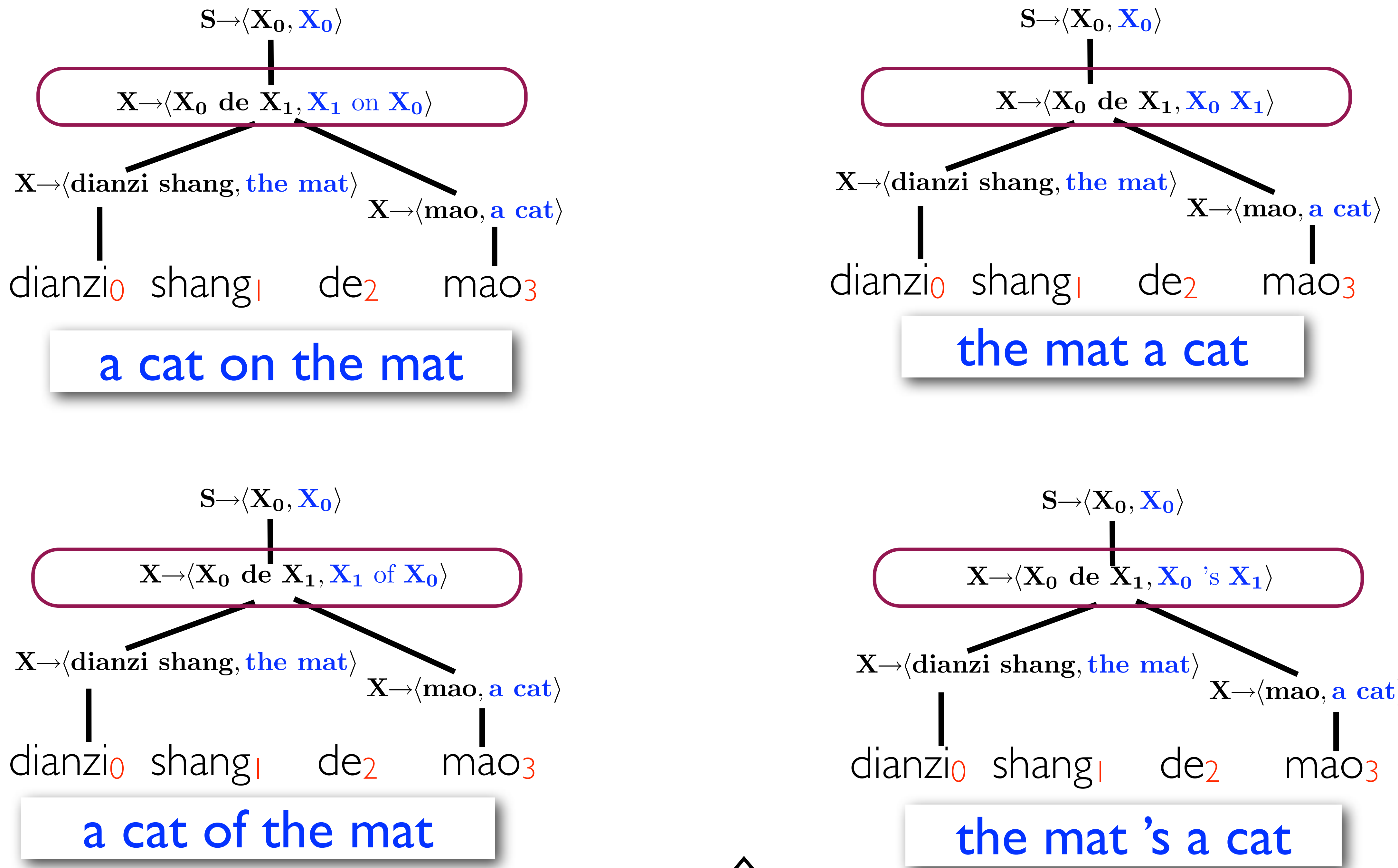

S→⟨X0, X0⟩
X→⟨X0 de X1, X1 on X0⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
S→⟨X0, X0⟩
X→⟨X0 de X1, X0 X1⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
S→⟨X0, X0⟩
X→⟨X0 de X1, X1 of X0⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
S→⟨X0, X0⟩
X→⟨X0 de X1, X0 's X1⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
Joshua
(chart parser)
dianzi shang de mao



Joshua

(chart parser)

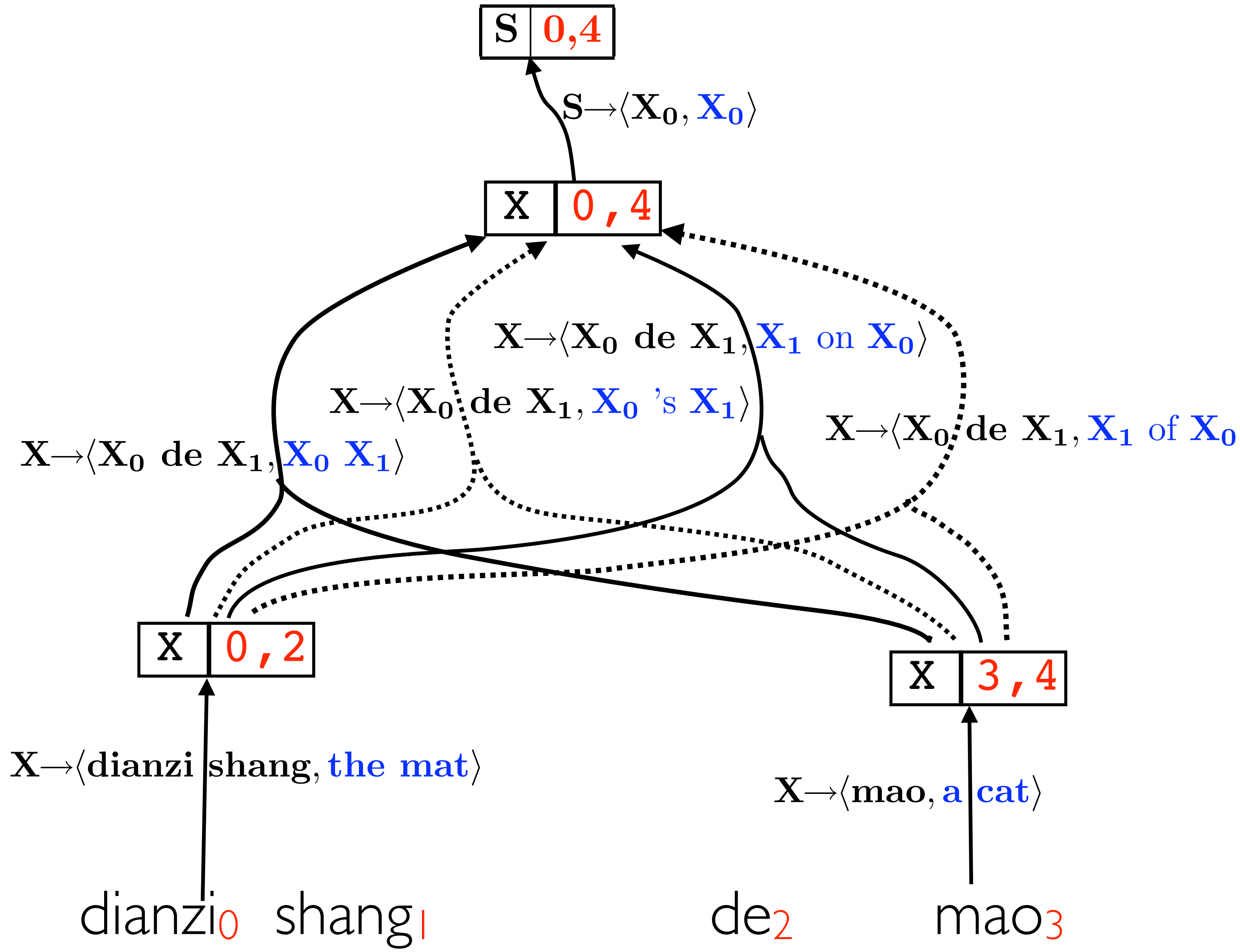
dianzi shang de mao



Joshua
(chart parser)

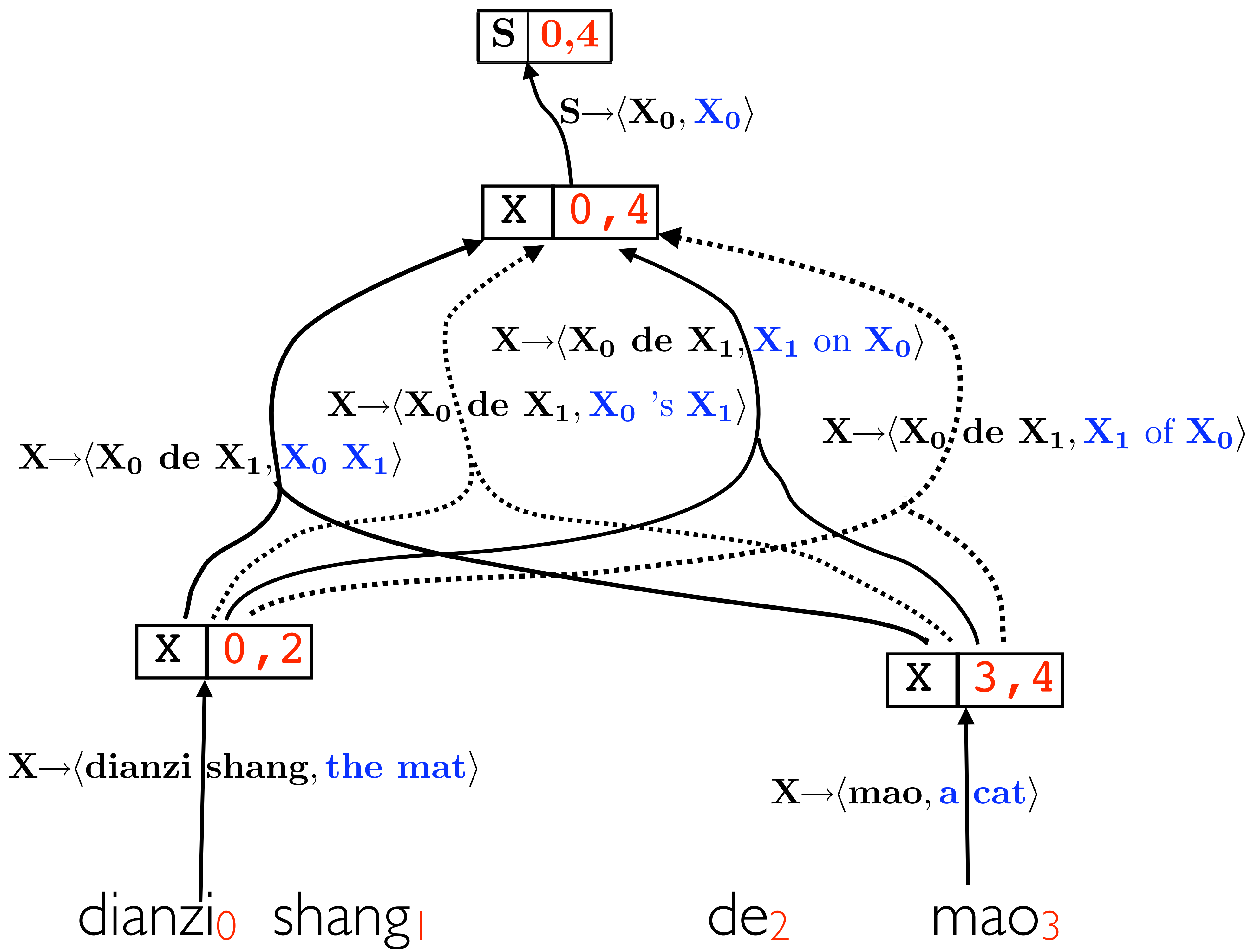
dianzi shang de mao

hypergraph

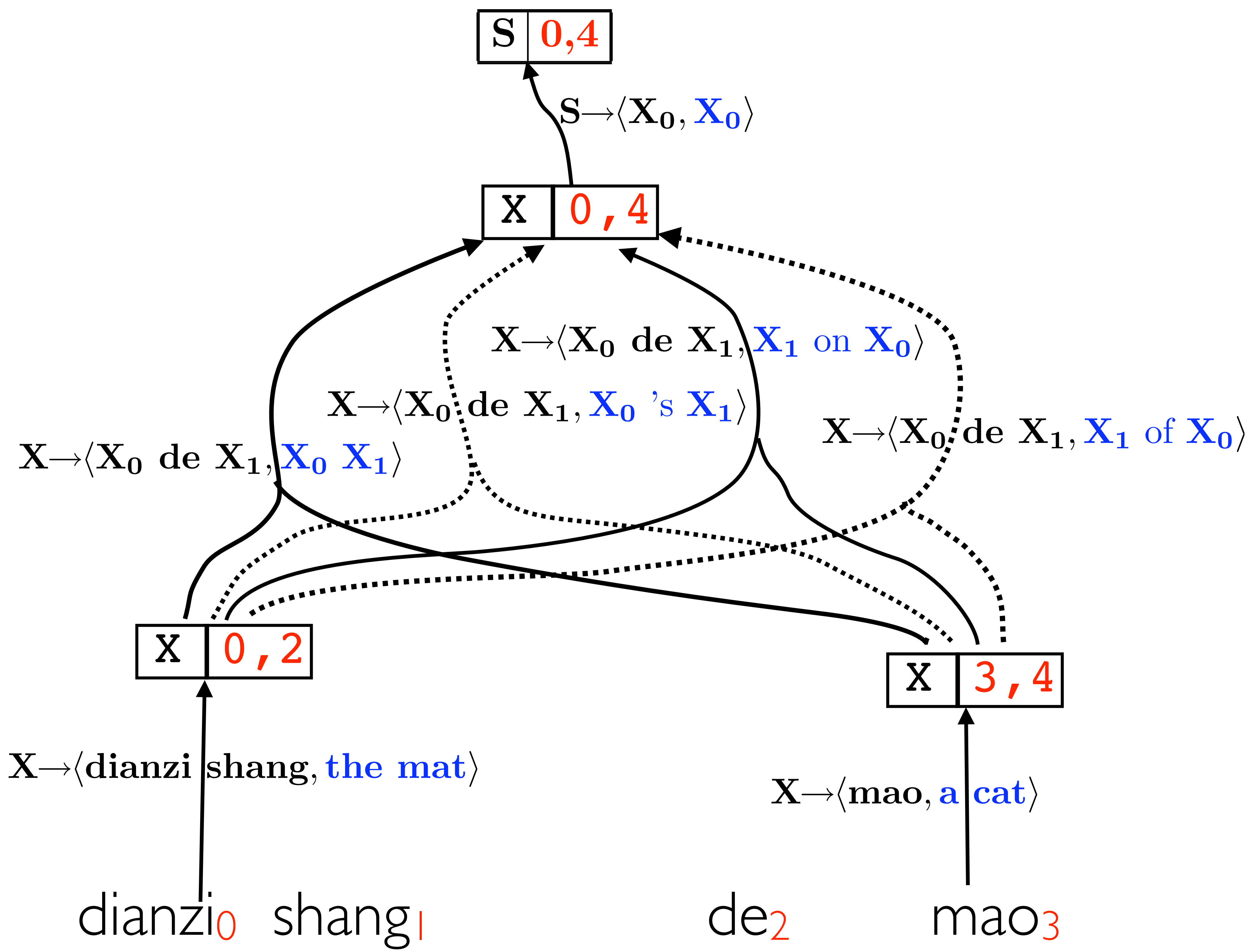


Joshua
(chart parser)

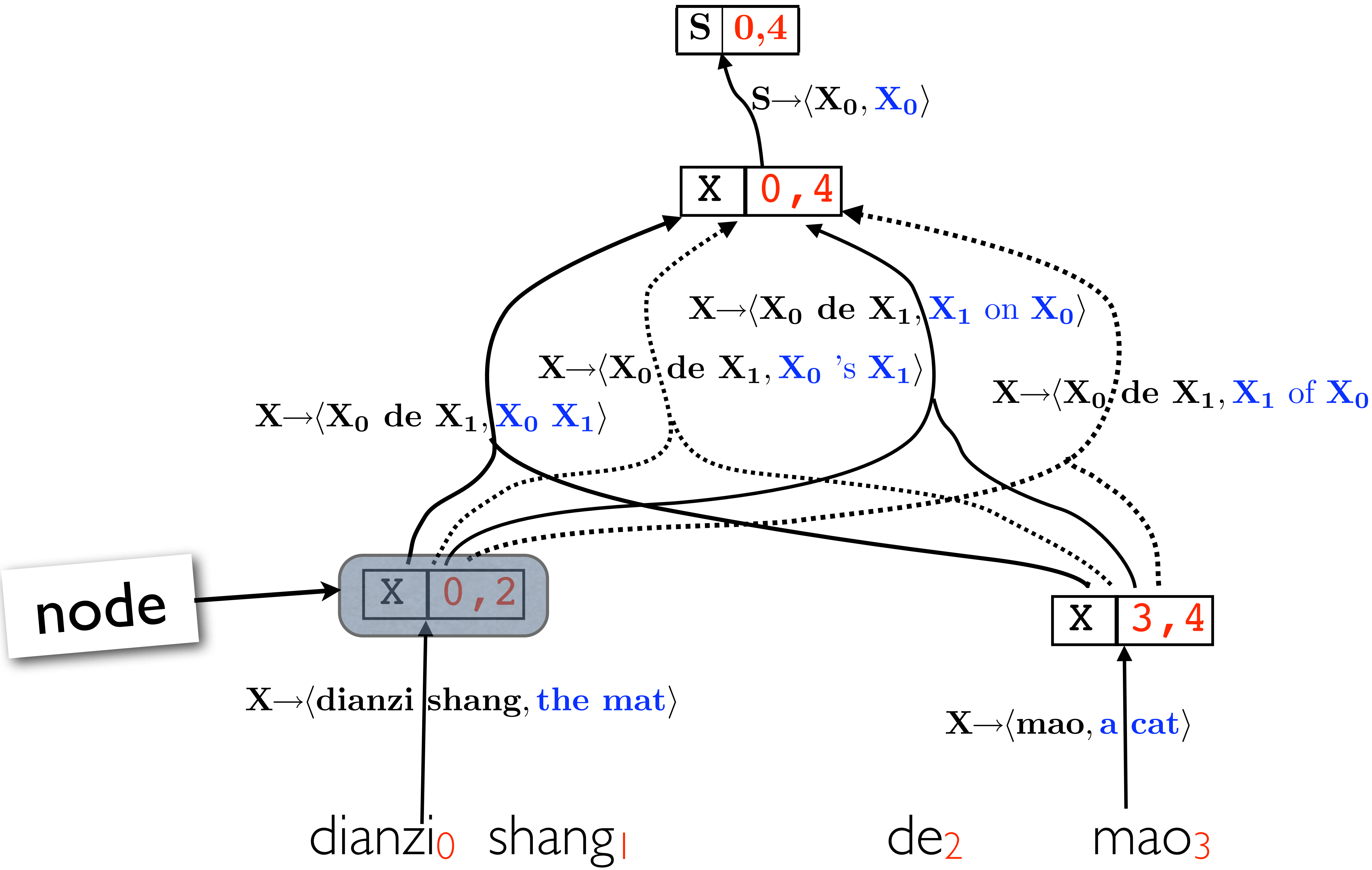
dianzi shang de mao


S 0,4
S→⟨X0, X0⟩
X 0,4
X→⟨X0 de X1, X1 on X0⟩
X→⟨X0 de X1, X0 's X1⟩
X→⟨X0 de X1, X0 X1⟩
X→⟨X0 de X1, X1 of X0⟩
X 0,2
X 3,4
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3

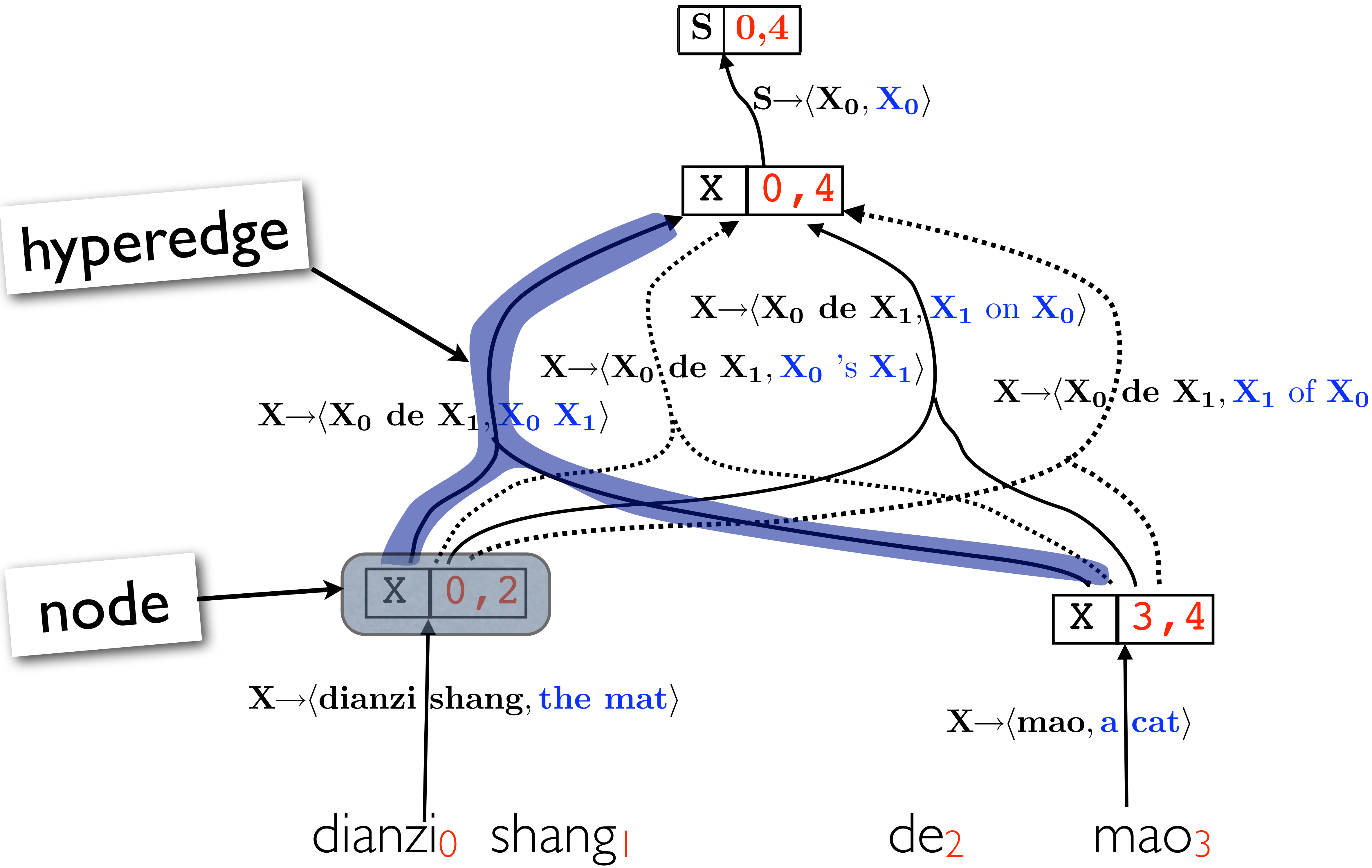
A hypergraph is a compact data structure to encode **exponentially many trees**.



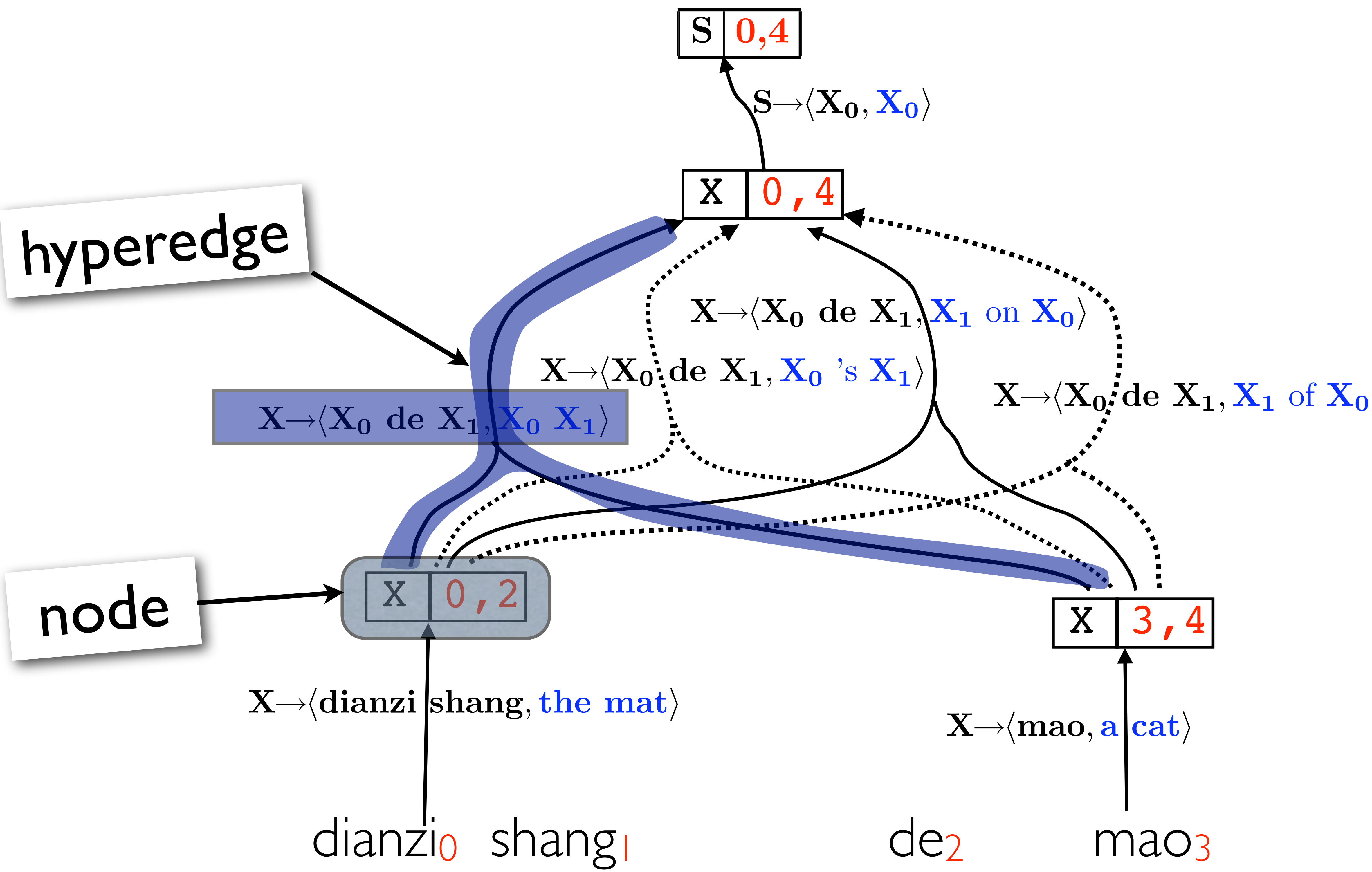
A hypergraph is a compact data structure to encode **exponentially many trees**.



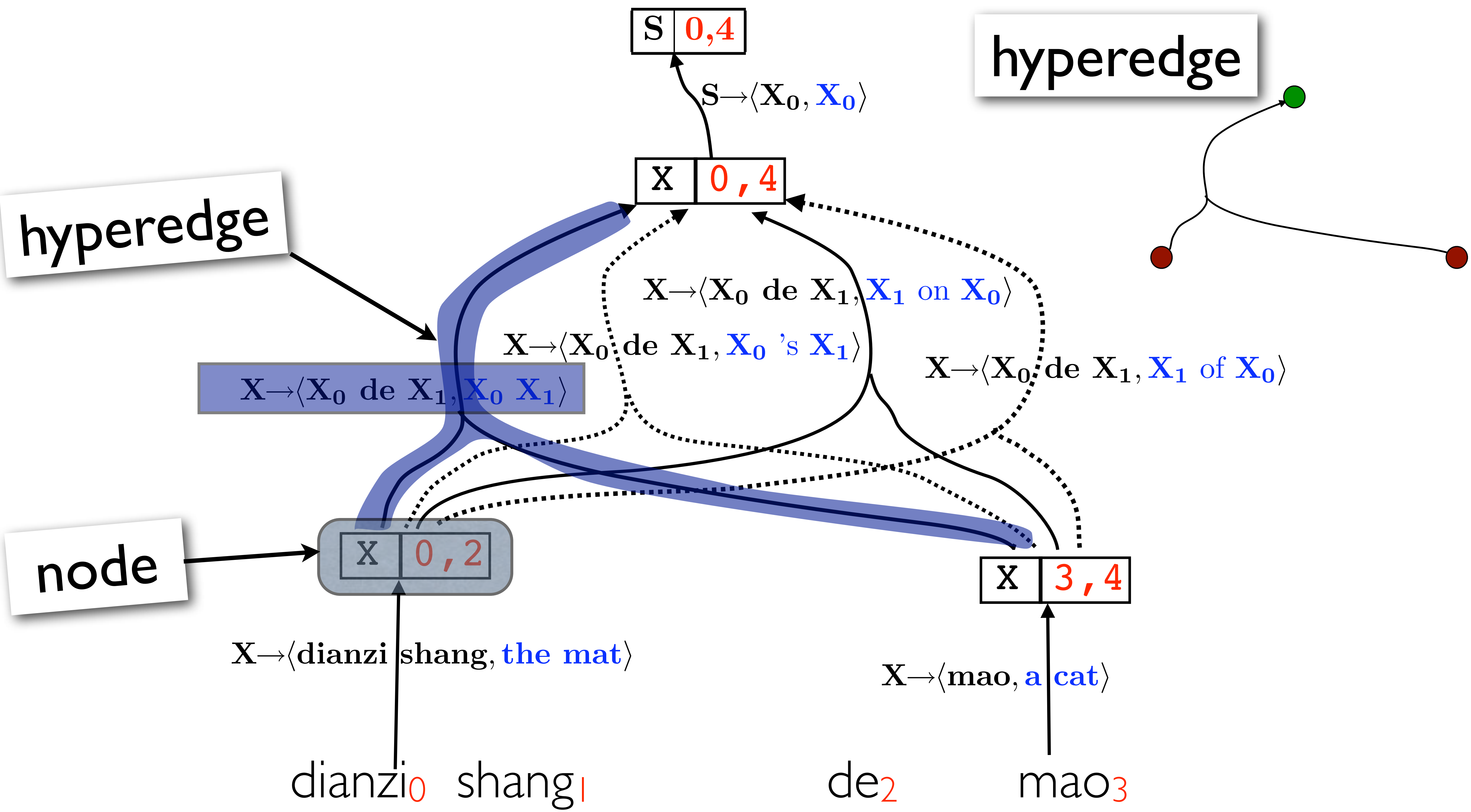
A hypergraph is a compact data structure to encode **exponentially many trees**.



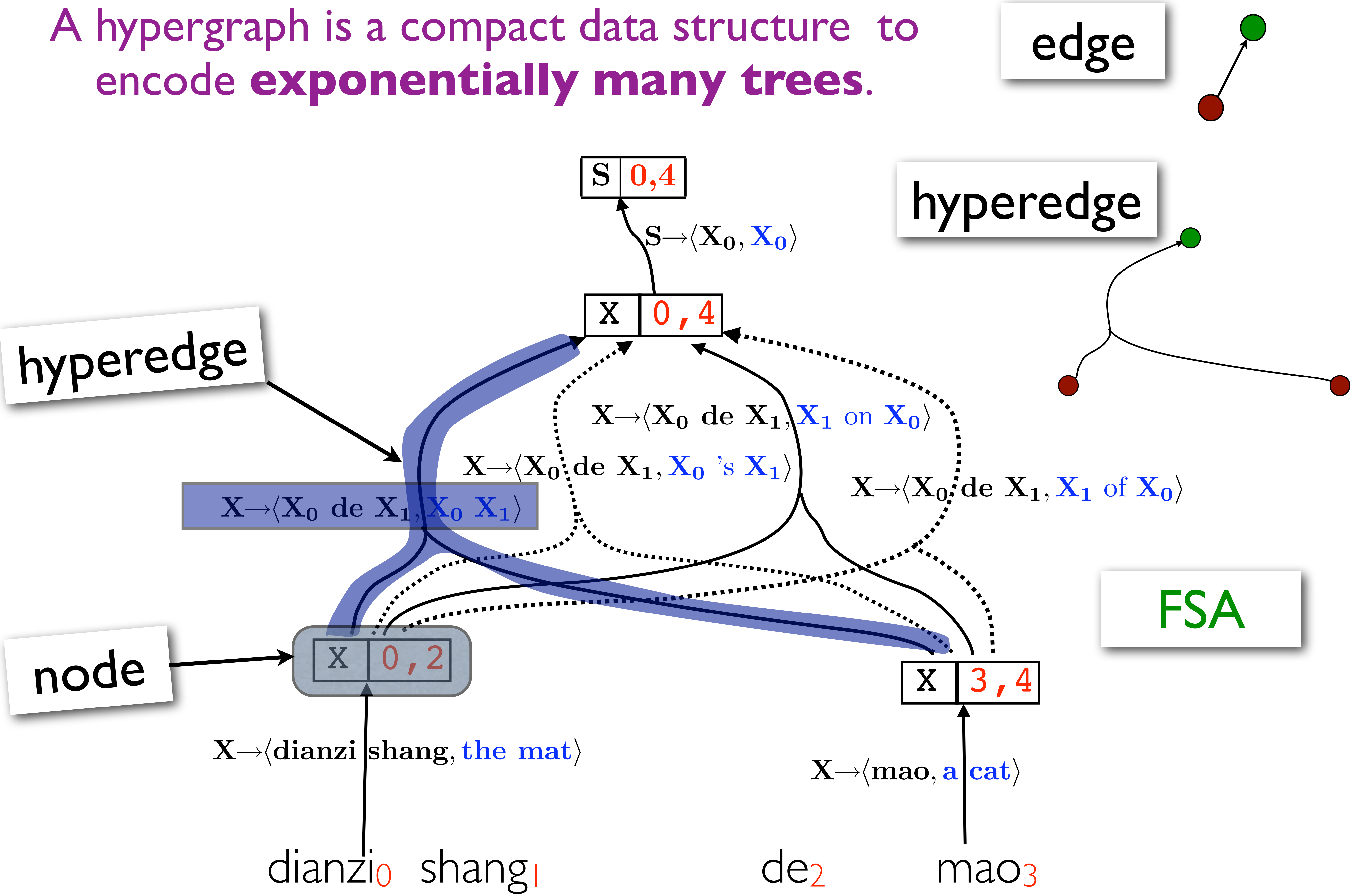
A hypergraph is a compact data structure to encode **exponentially many trees**.



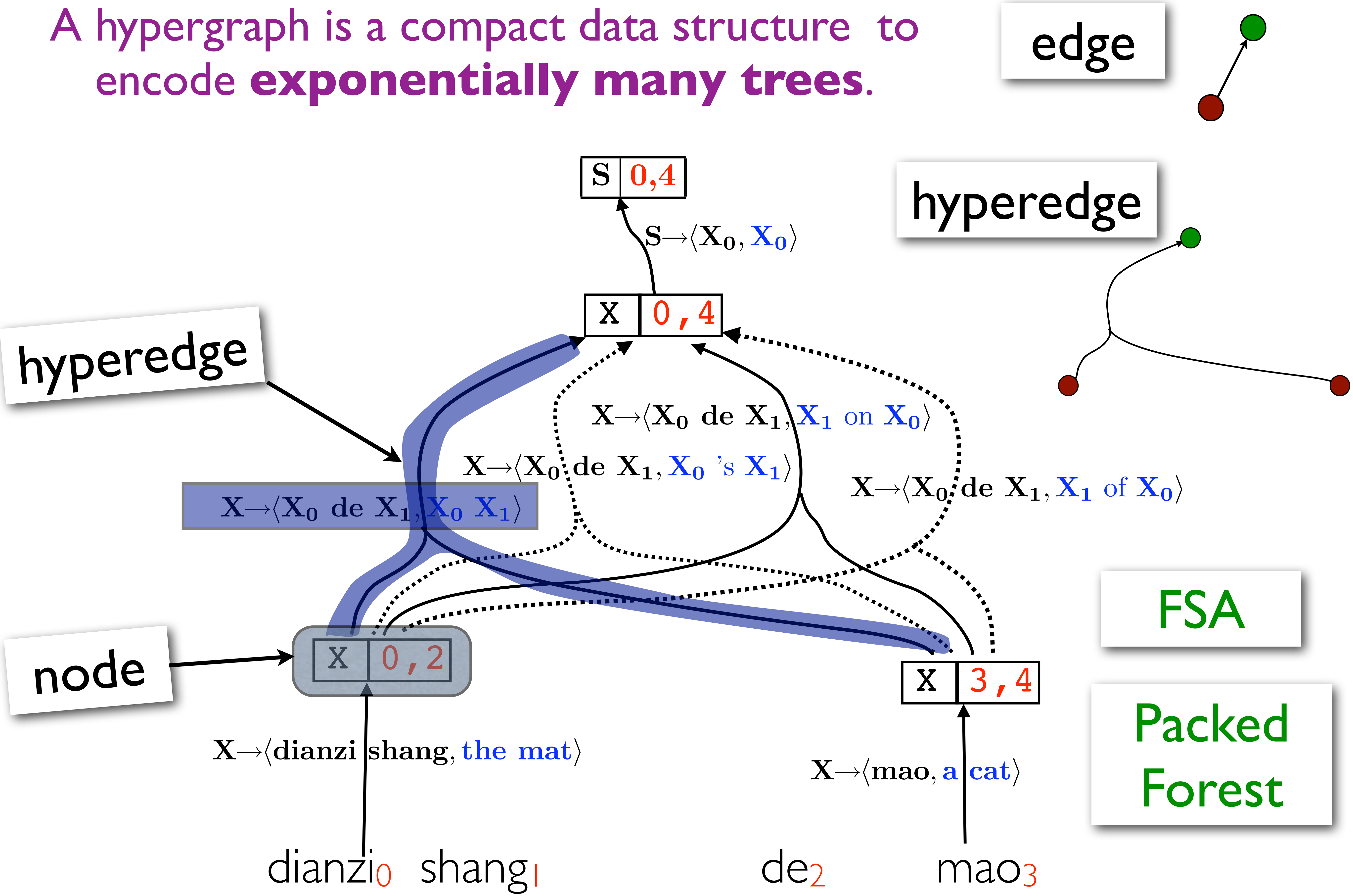
A hypergraph is a compact data structure to encode **exponentially many trees**.



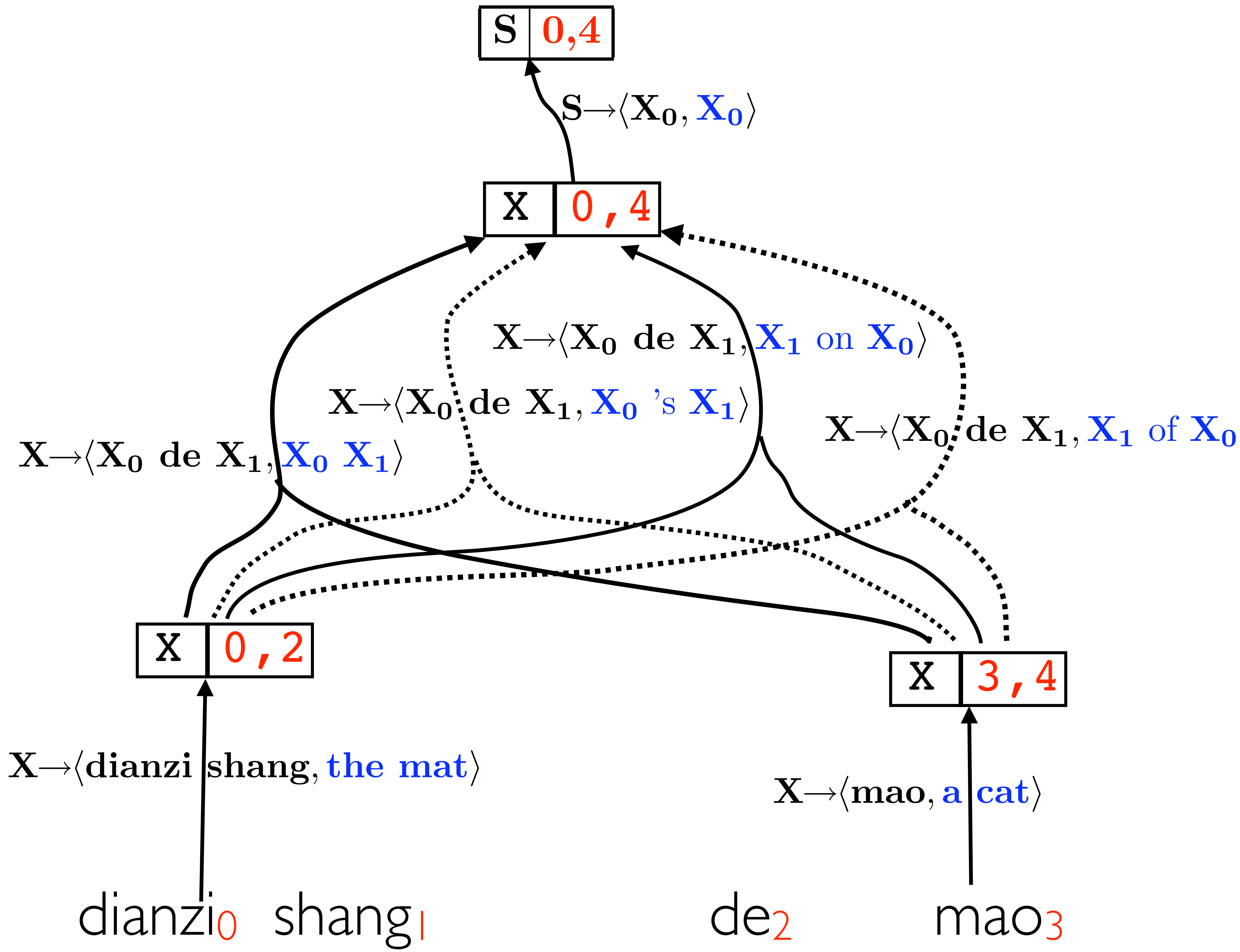
A hypergraph is a compact data structure to encode **exponentially many trees**.



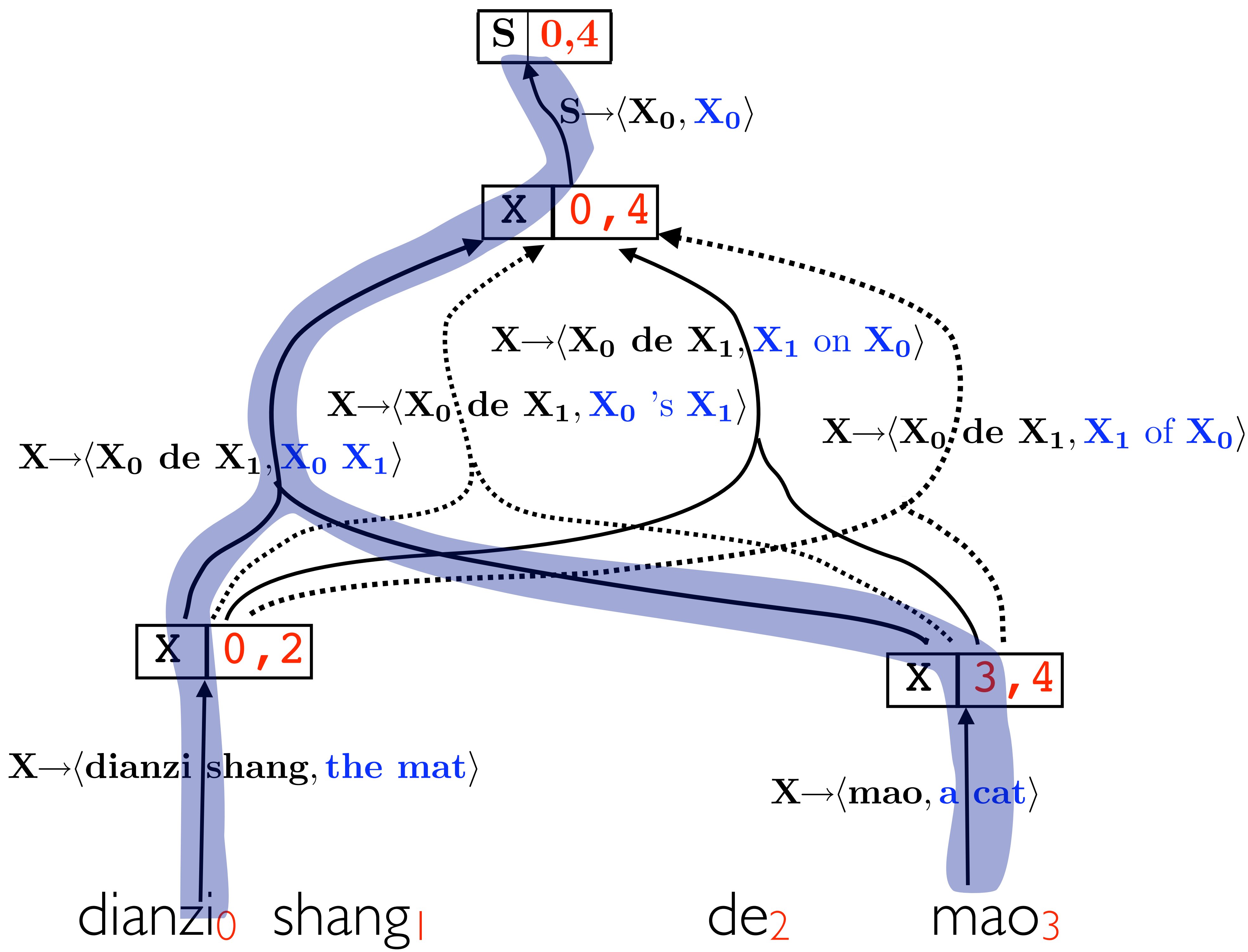
A hypergraph is a compact data structure to encode **exponentially many trees**.



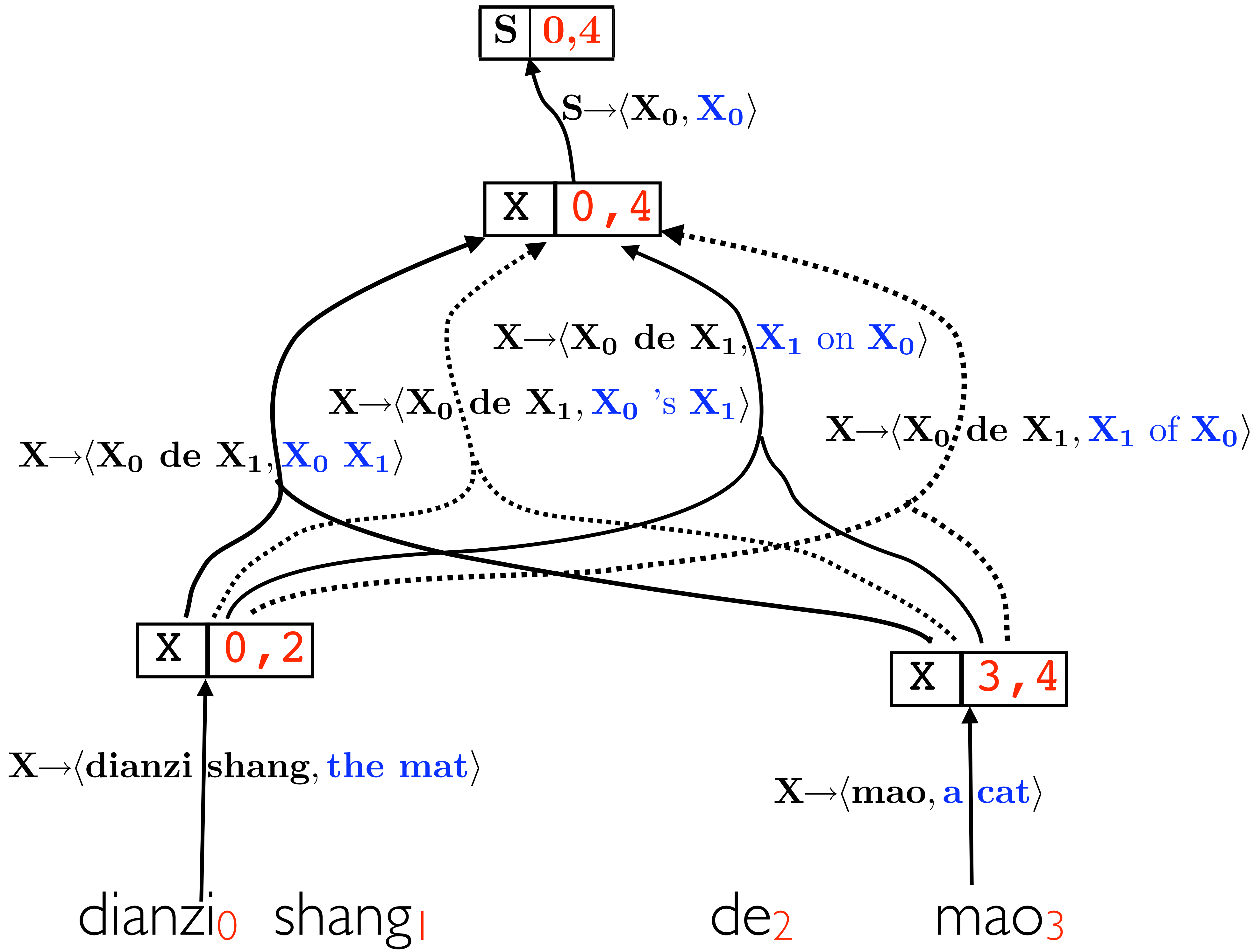
A hypergraph is a compact data structure to encode **exponentially many trees**.



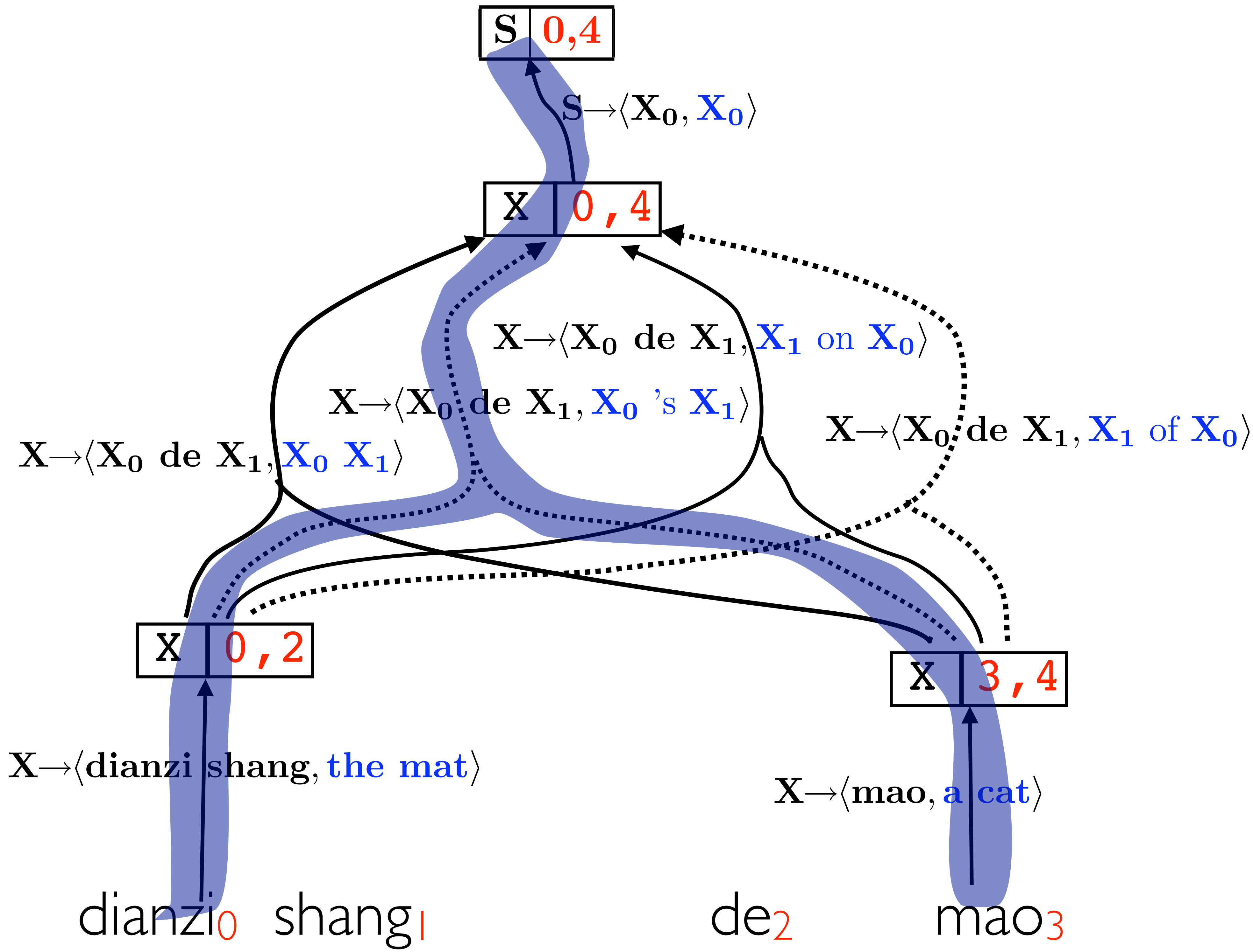
A hypergraph is a compact data structure to encode **exponentially many trees**.



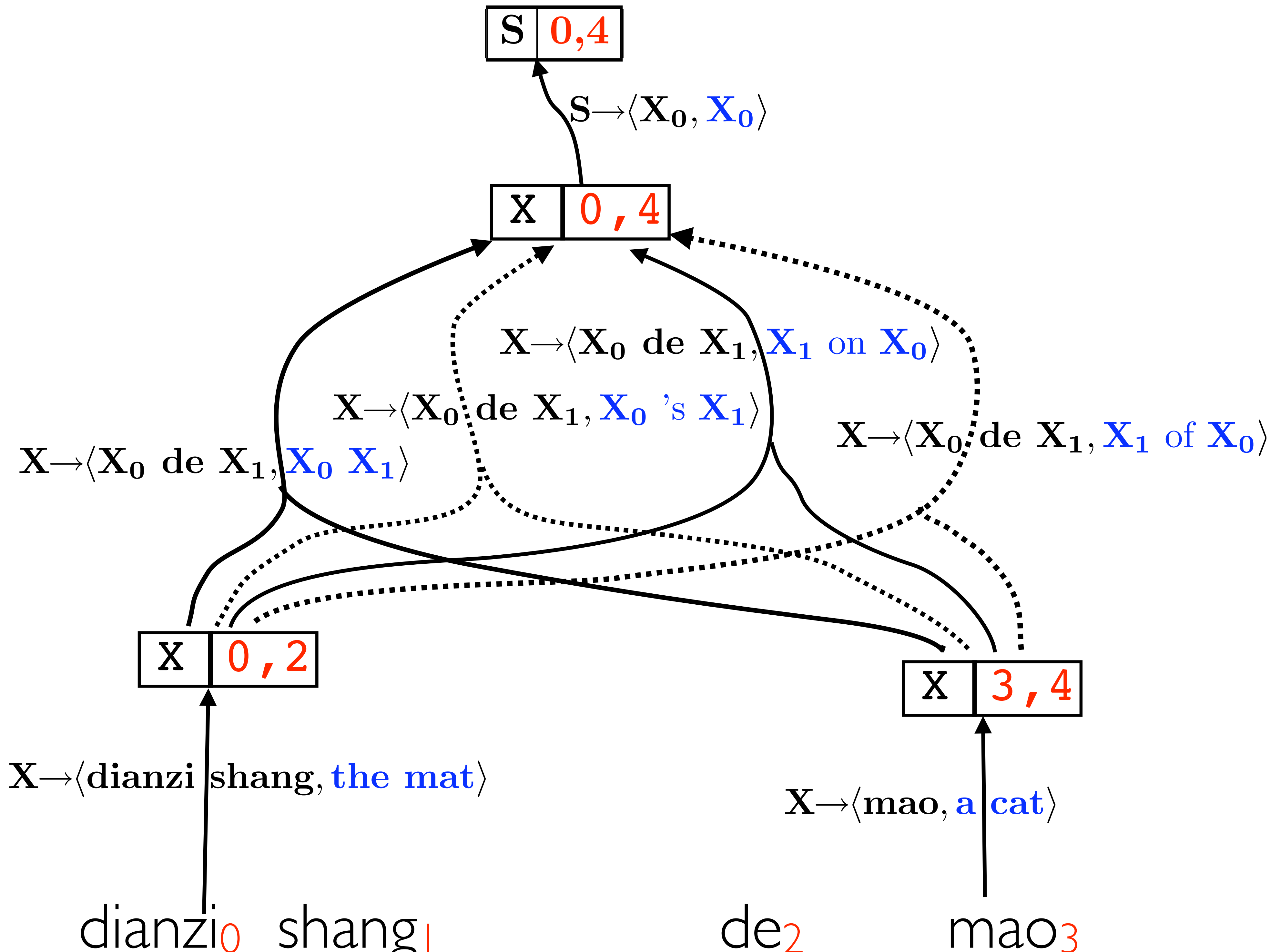
A hypergraph is a compact data structure to encode **exponentially many trees**.



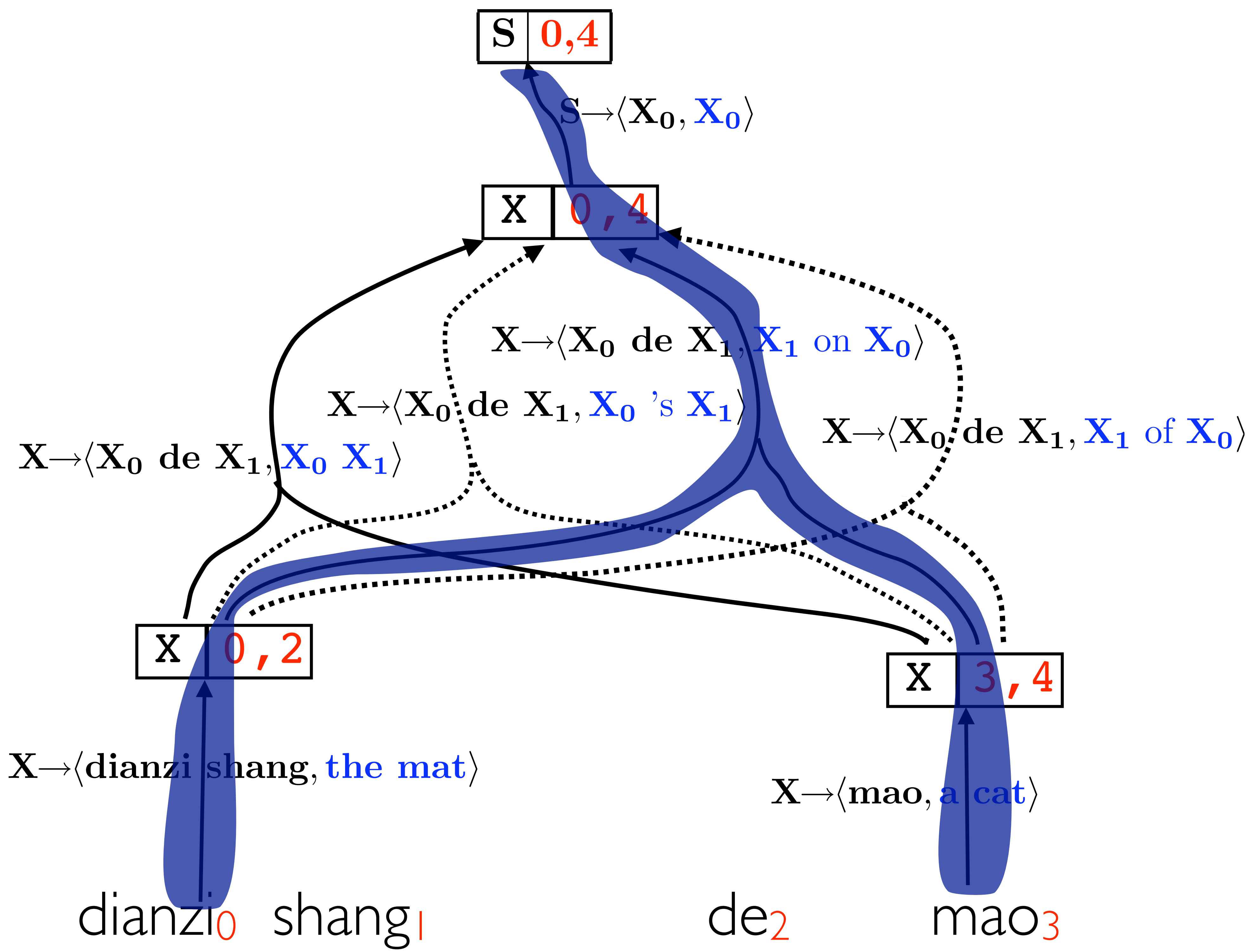
A hypergraph is a compact data structure to encode **exponentially many trees**.



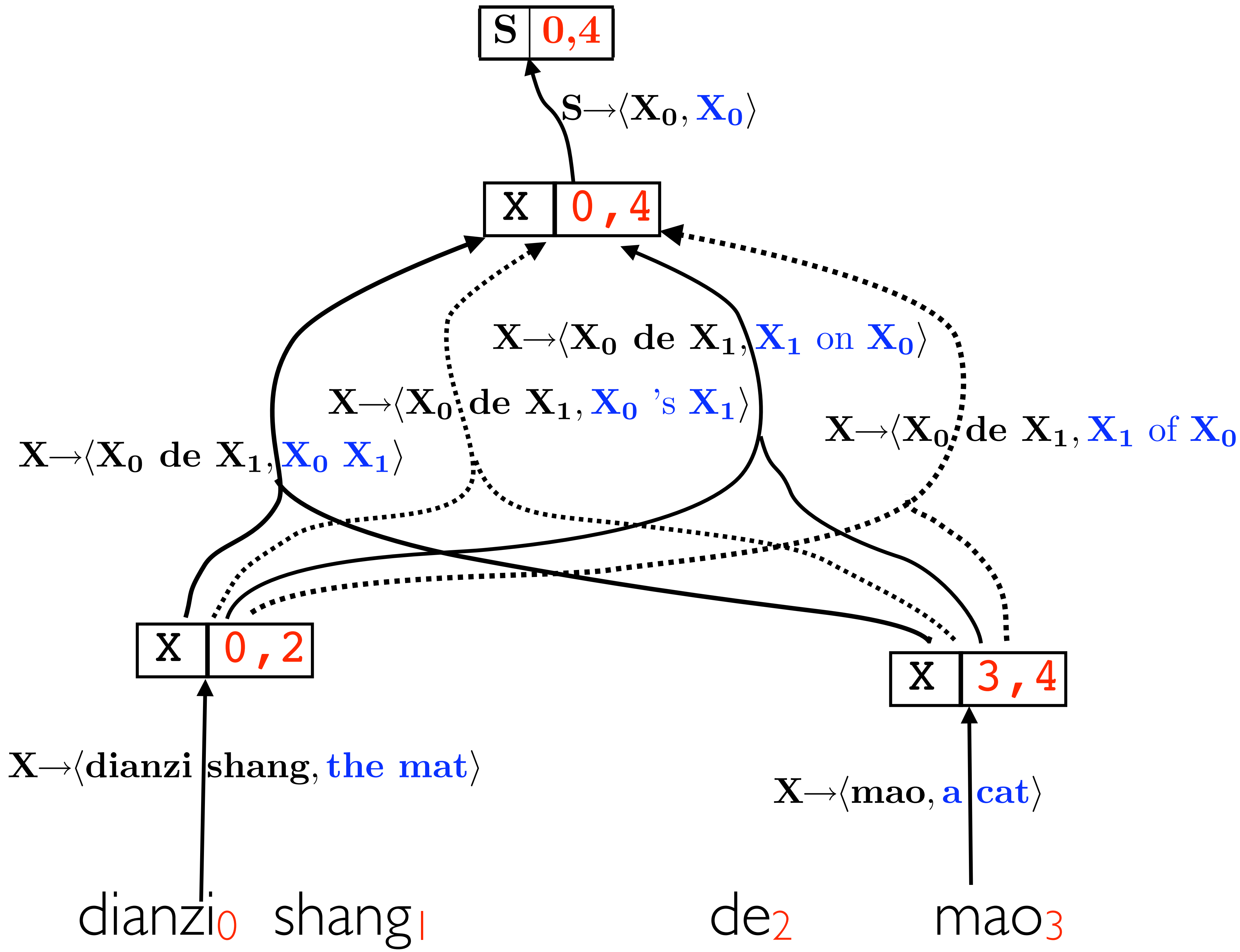
A hypergraph is a compact data structure to encode **exponentially many trees**.



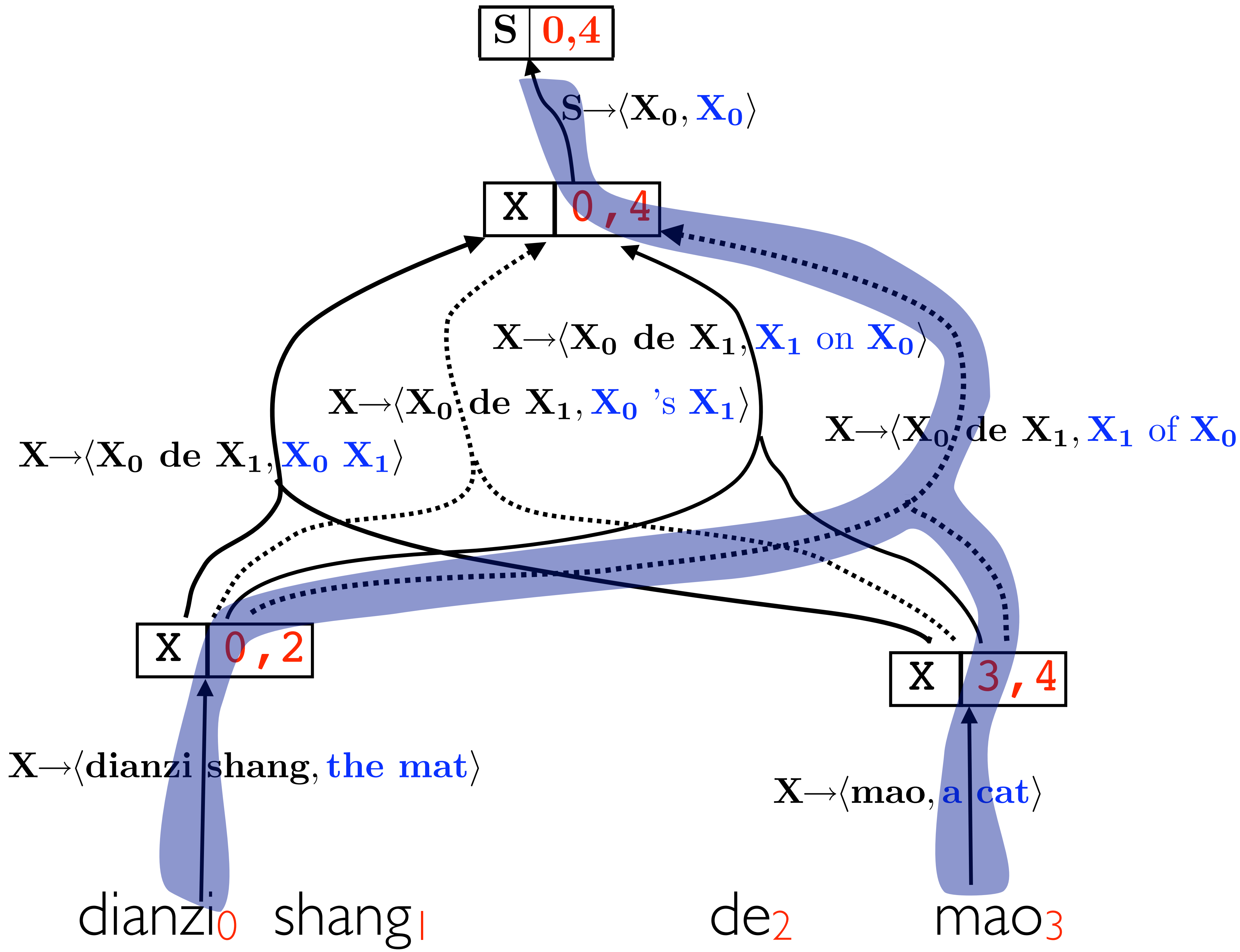
A hypergraph is a compact data structure to encode **exponentially many trees**.



A hypergraph is a compact data structure to encode **exponentially many trees**.

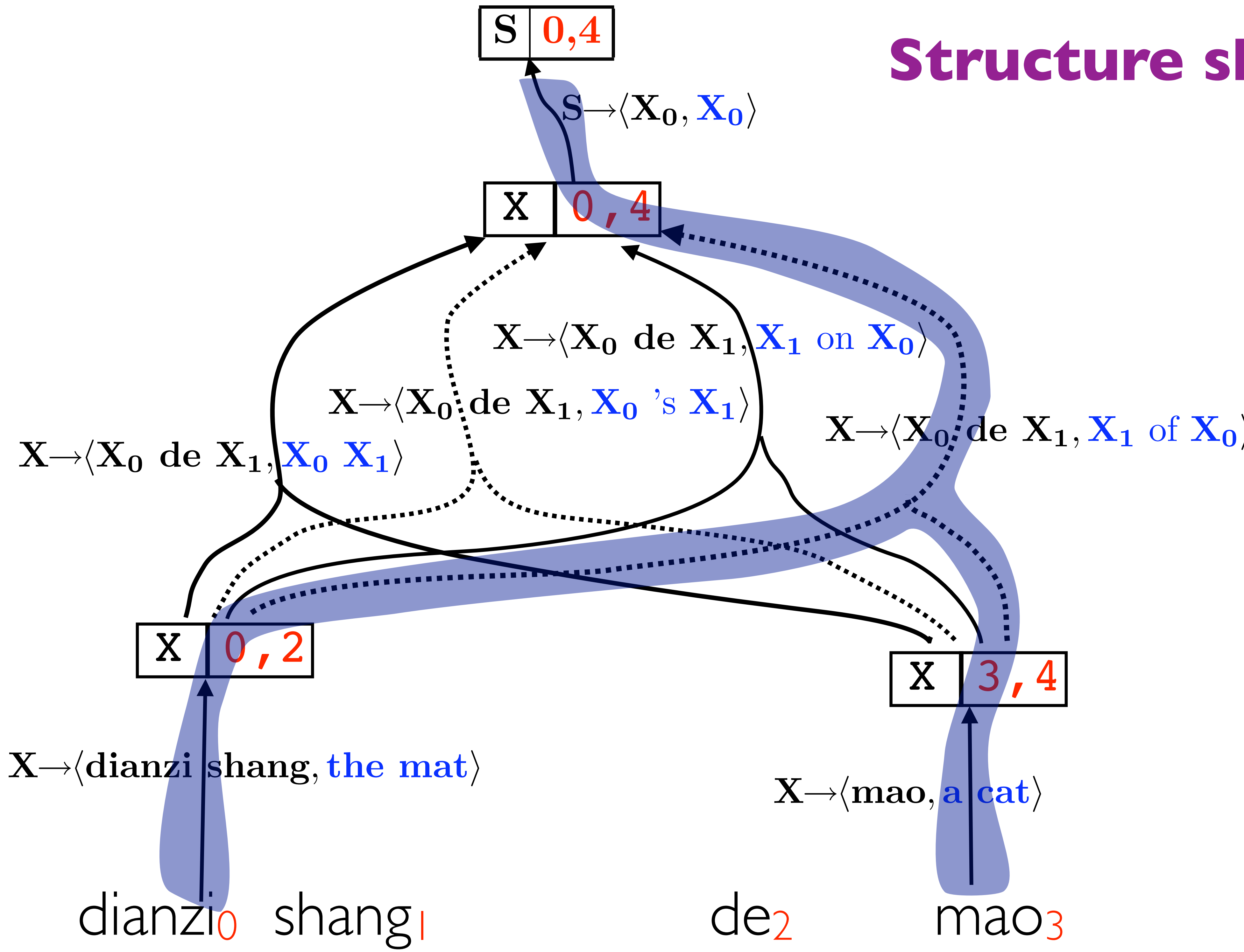


A hypergraph is a compact data structure to encode **exponentially many trees**.



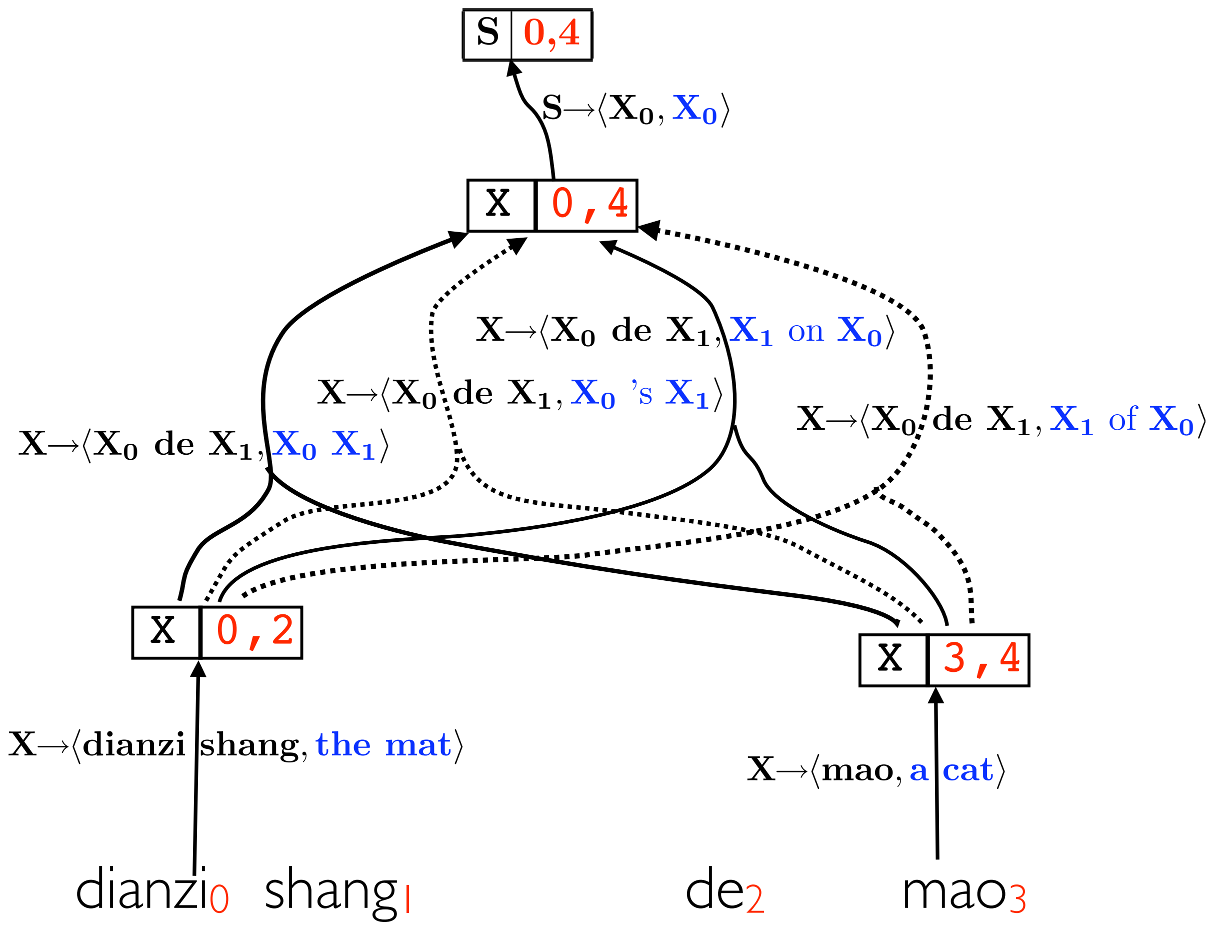
A hypergraph is a compact data structure to encode **exponentially many trees**.

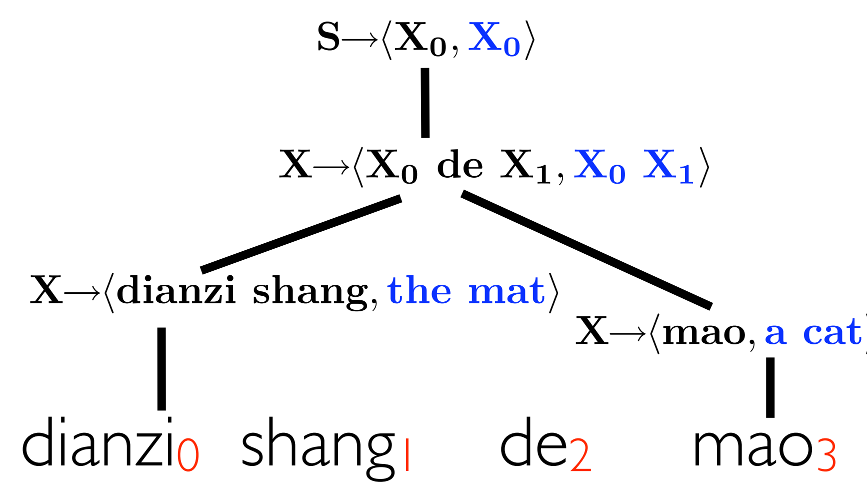
Structure sharing

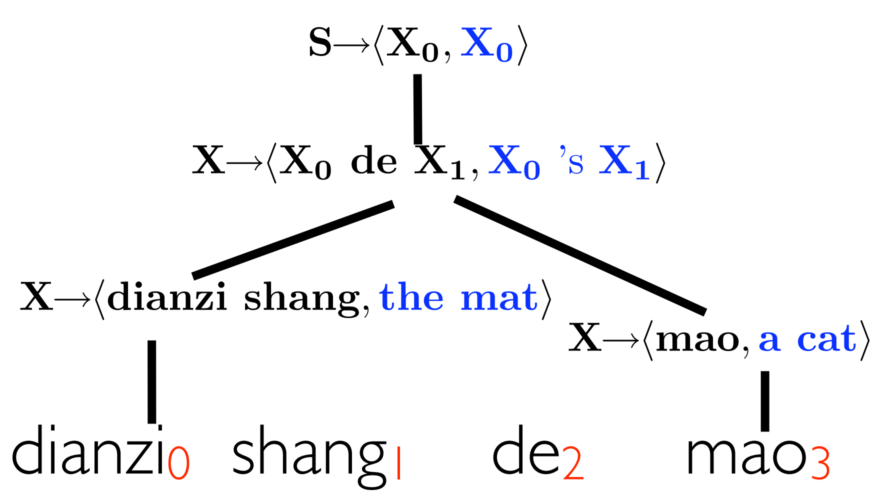


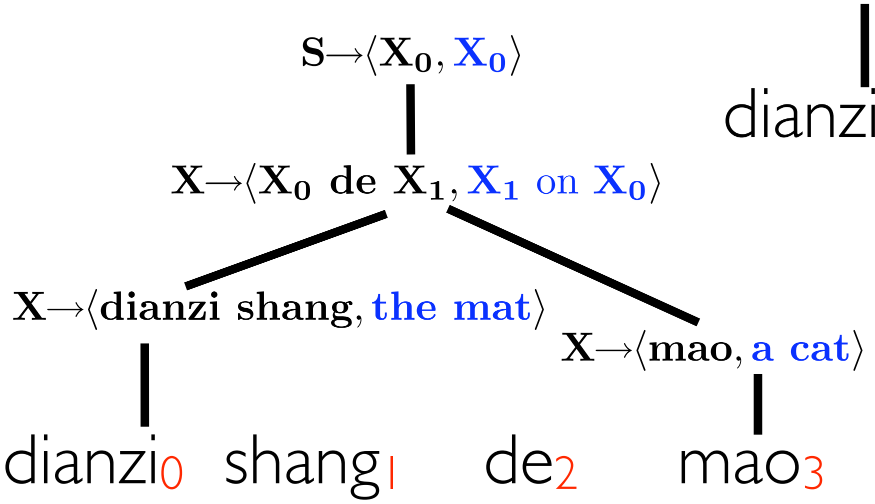
Why Hypergraphs?

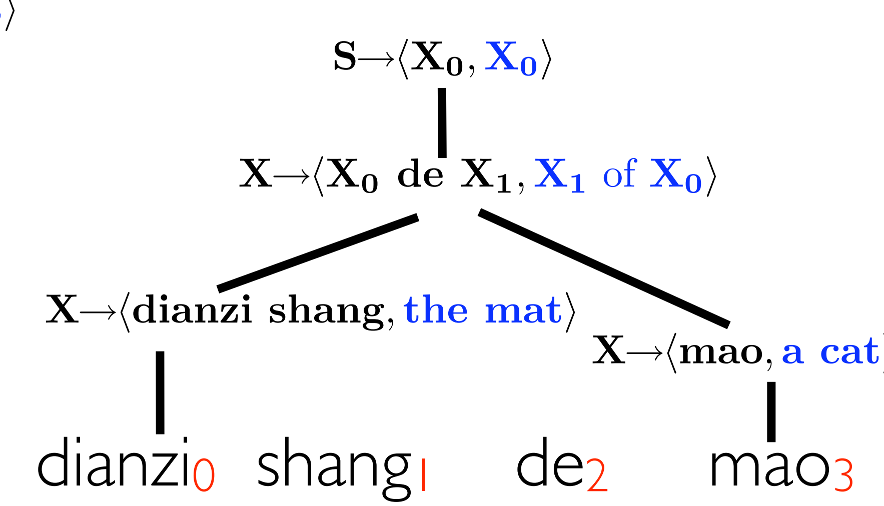
- Contains a much larger hypothesis space than a k-best list
- General compact data structure
  - special cases include
    - finite state machine (e.g., lattice),
    - and/or graph
    - packed forest
  - can be used for speech, parsing, tree-based MT systems, and many more


S 0,4
S→⟨X0, X0⟩
X 0,4
X→⟨X0 de X1, X1 on X0⟩
X→⟨X0 de X1, X0 's X1⟩
X→⟨X0 de X1, X1 of X0⟩
X→⟨X0 de X1, X0 X1⟩
X 0,2
X 3,4
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3

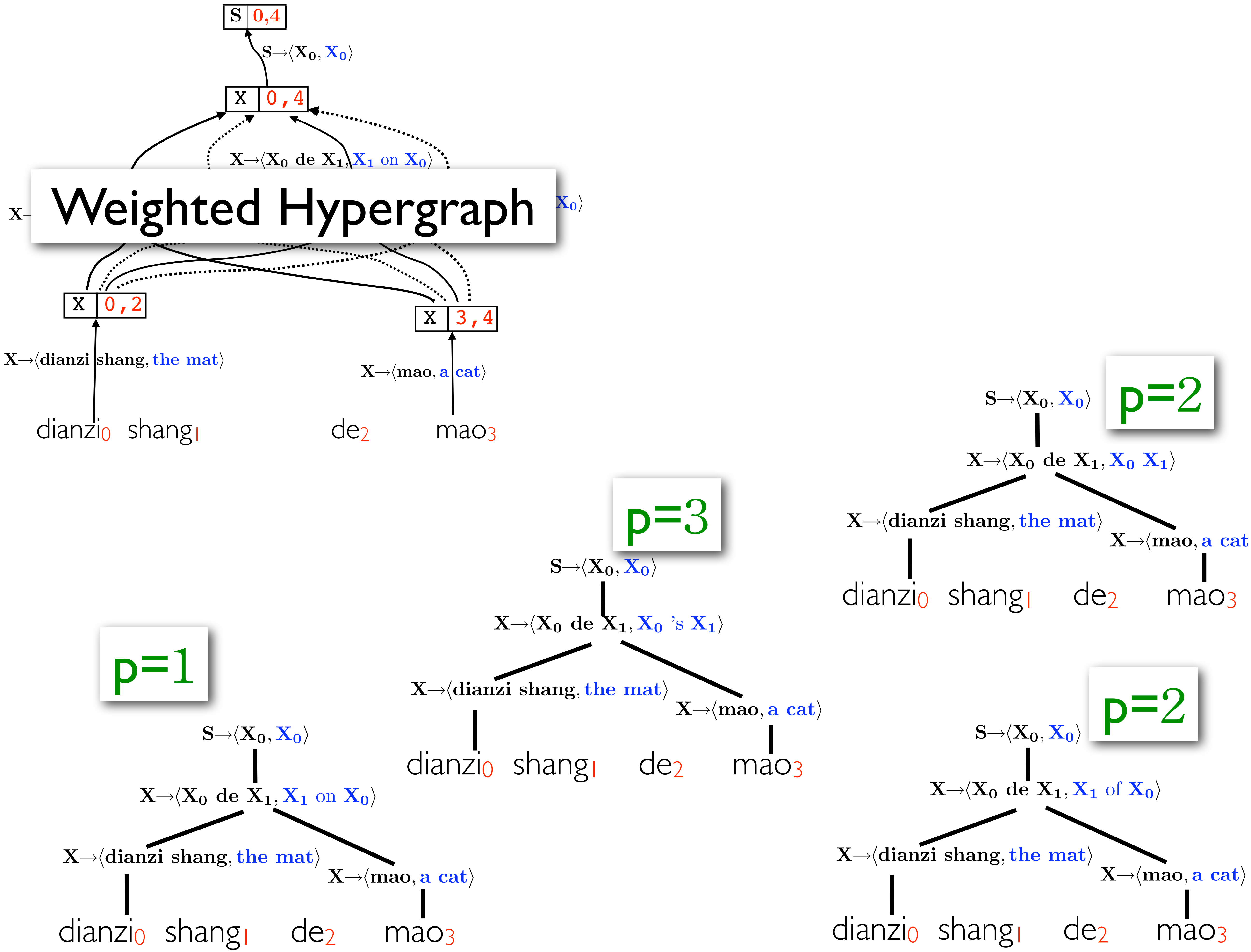

S→⟨X0, X0⟩
X→⟨X0 de X1, X0 X1⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3


S→⟨X0, X0⟩
X→⟨X0 de X1, X0 's X1⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3

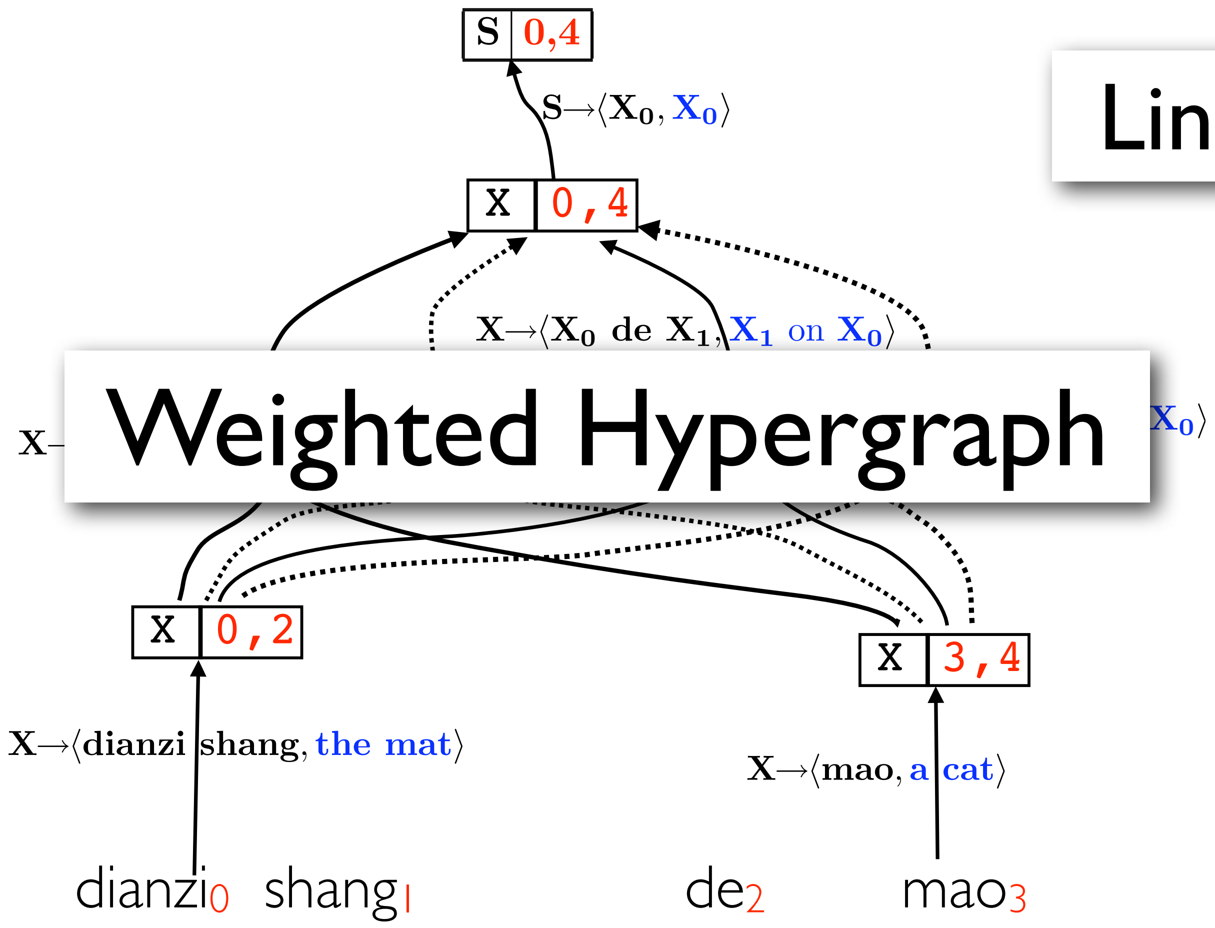

S→⟨X0, X0⟩
X→⟨X0 de X1, X1 on X0⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3


S→⟨X0, X0⟩
X→⟨X0 de X1, X1 of X0⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3

Weighted Hypergraph

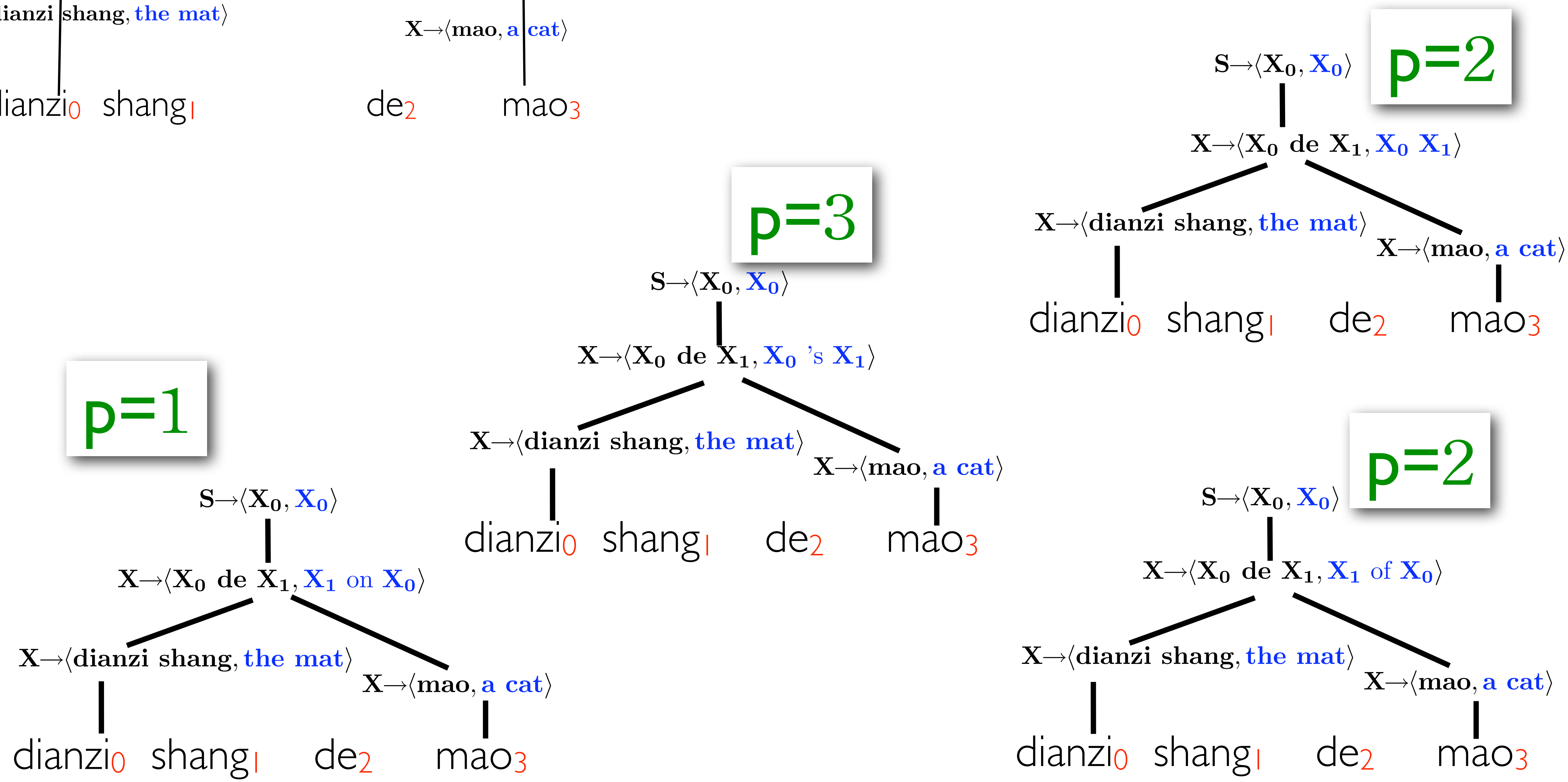


Weighted Hypergraph

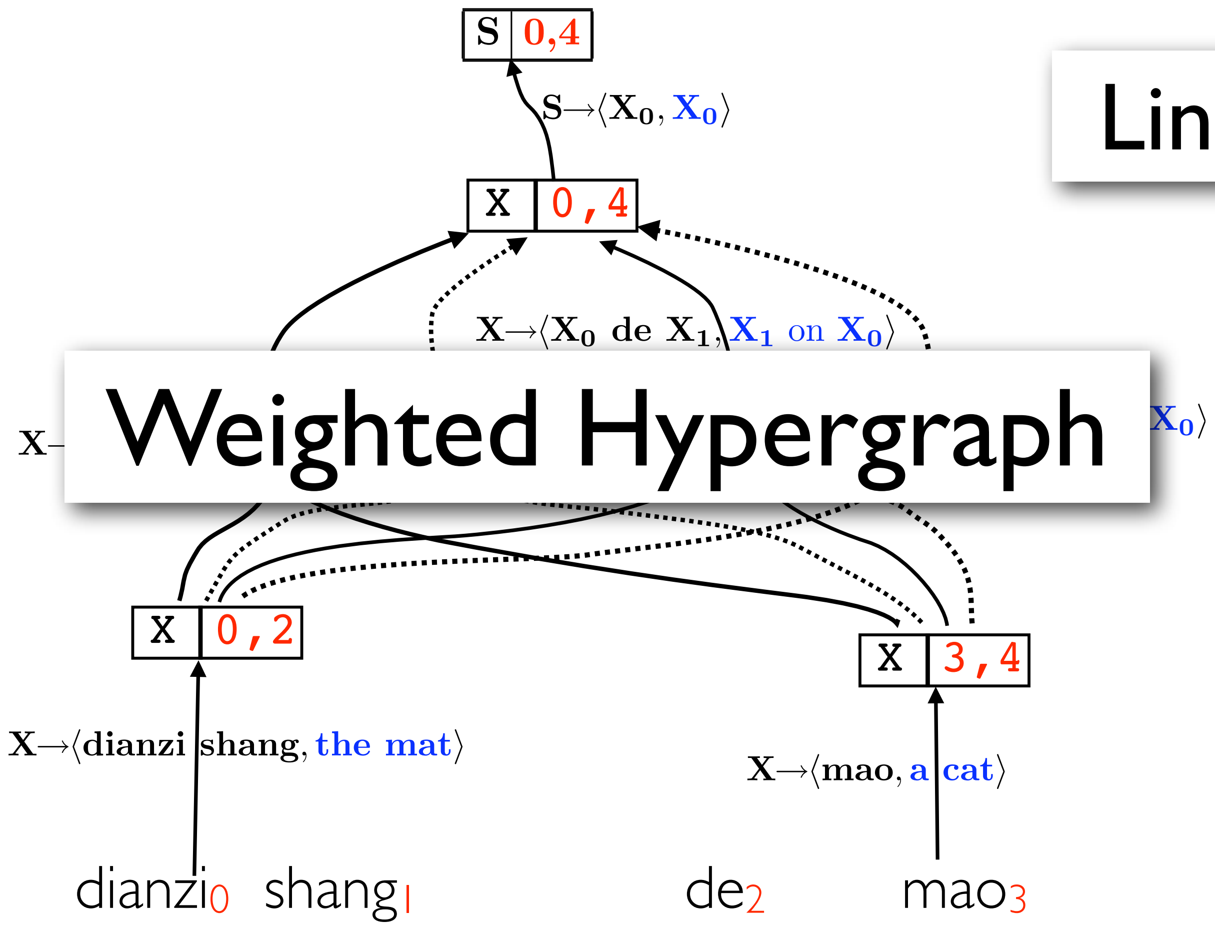


Linear model:

$$p(d \mid x) = \theta \cdot \Phi(d, x)$$



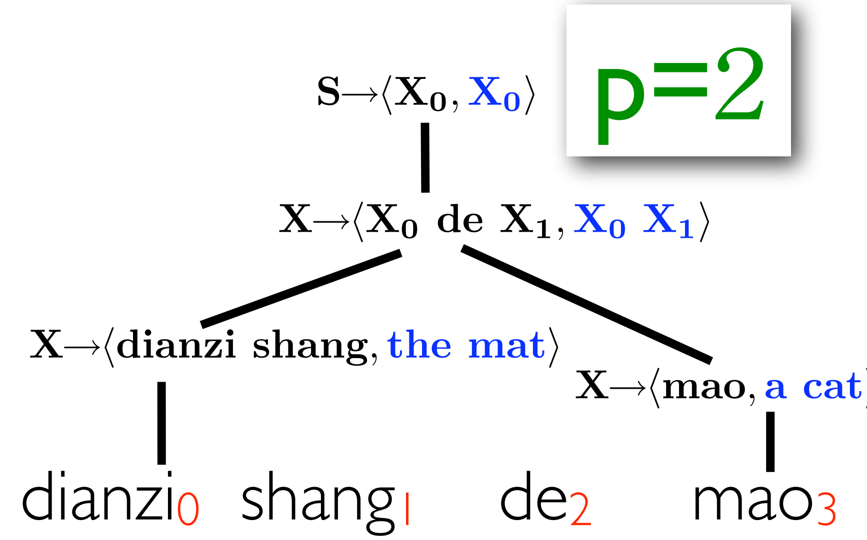
Weighted Hypergraph

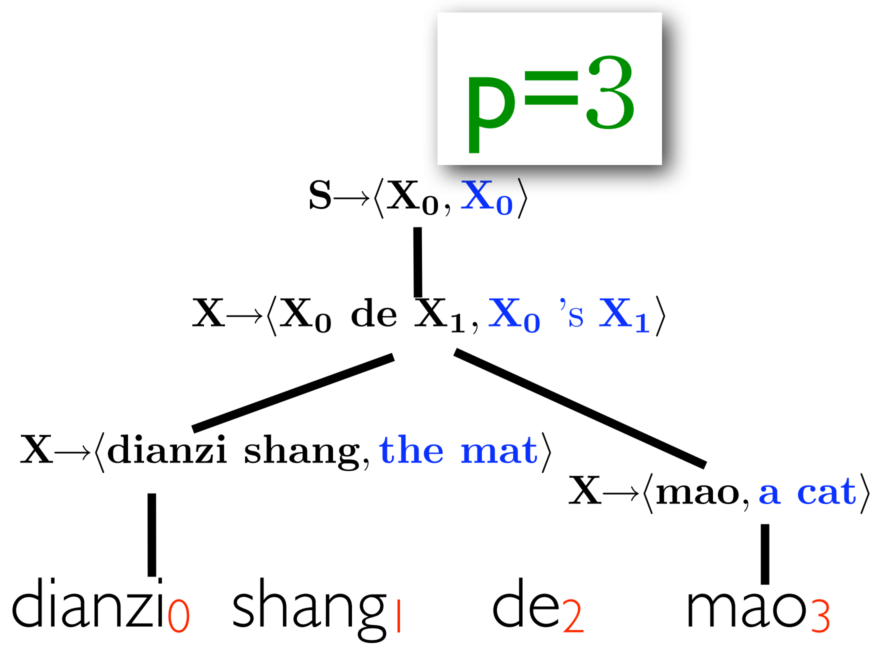


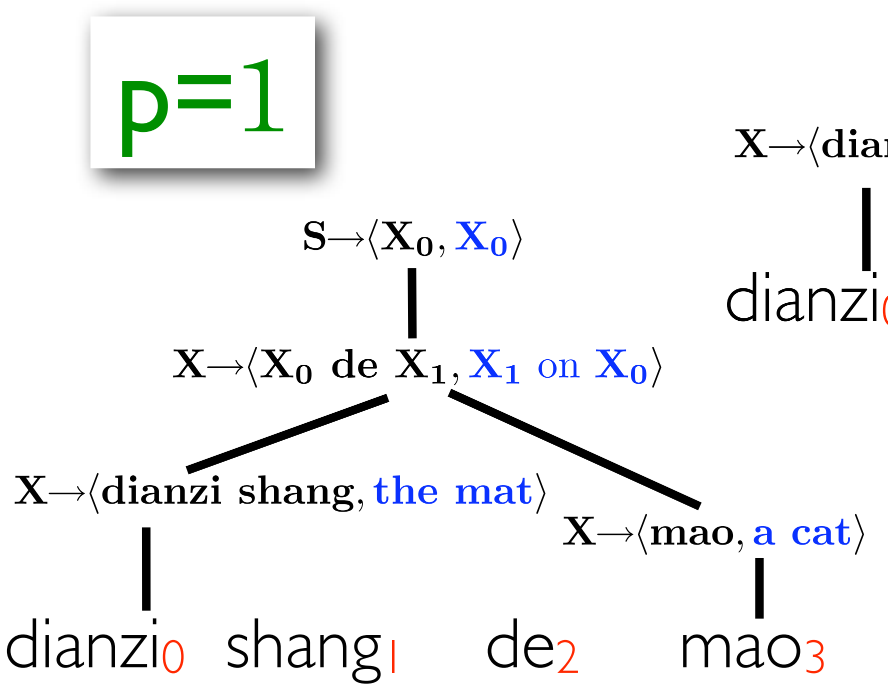
Linear model:

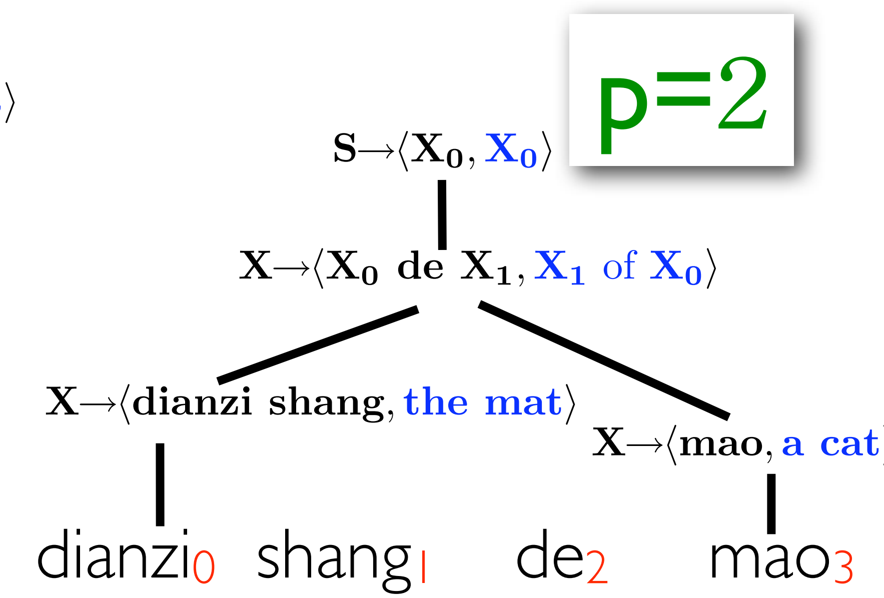
foreign input

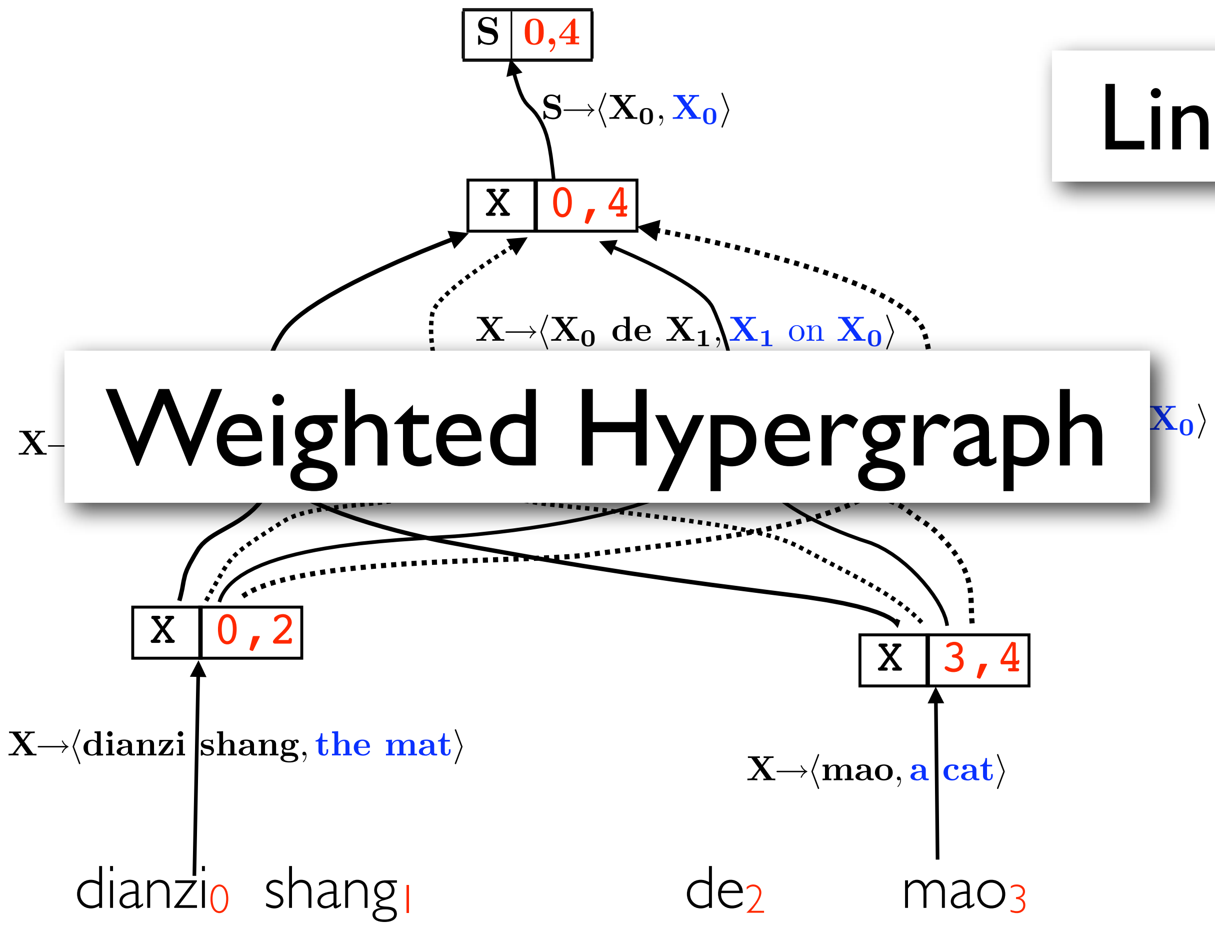
$$p(d \mid x) = \theta \cdot \Phi(d, x)$$











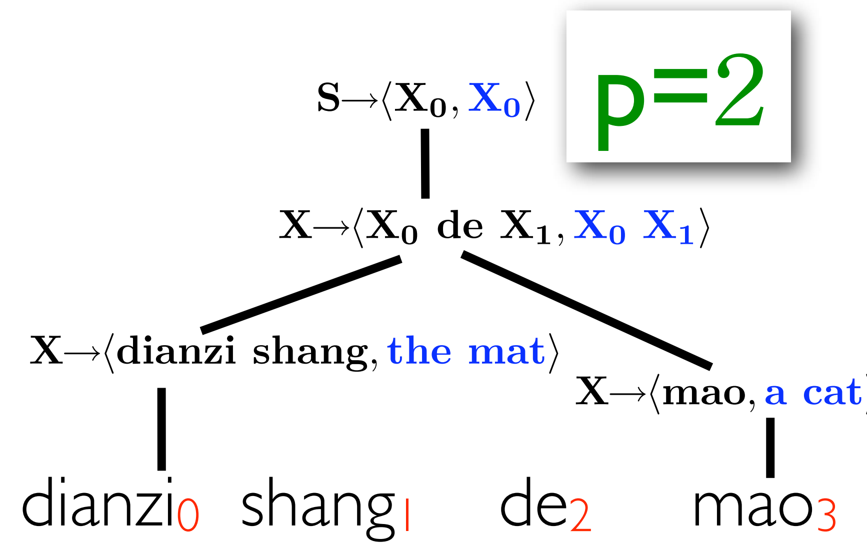
Weighted Hypergraph

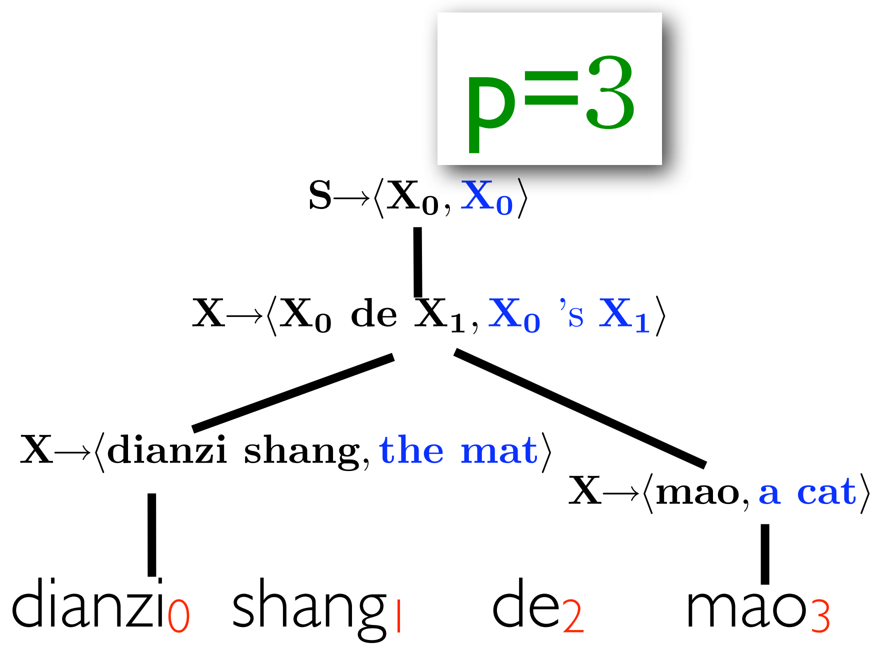
Linear model:

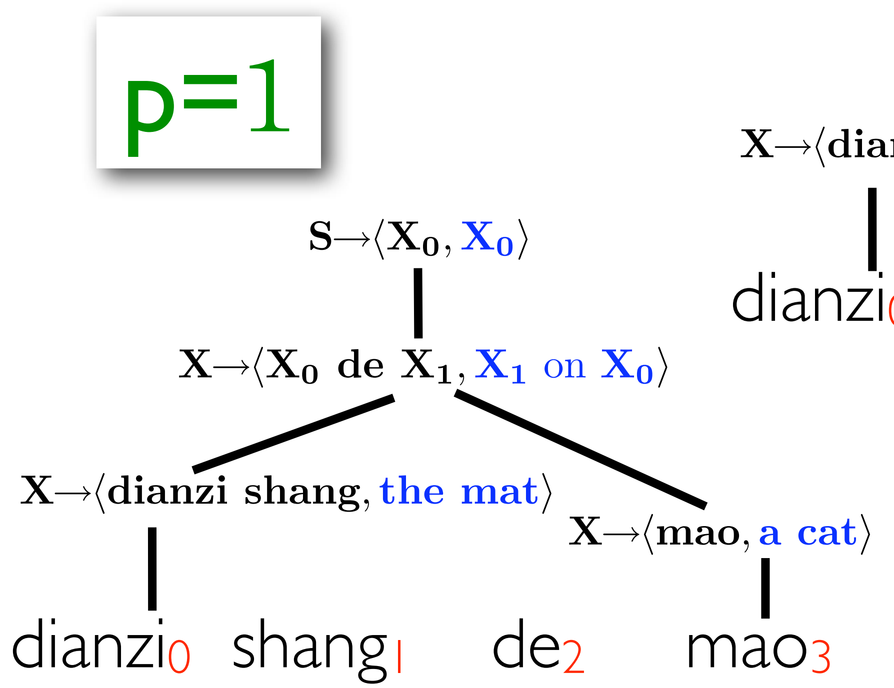
derivation

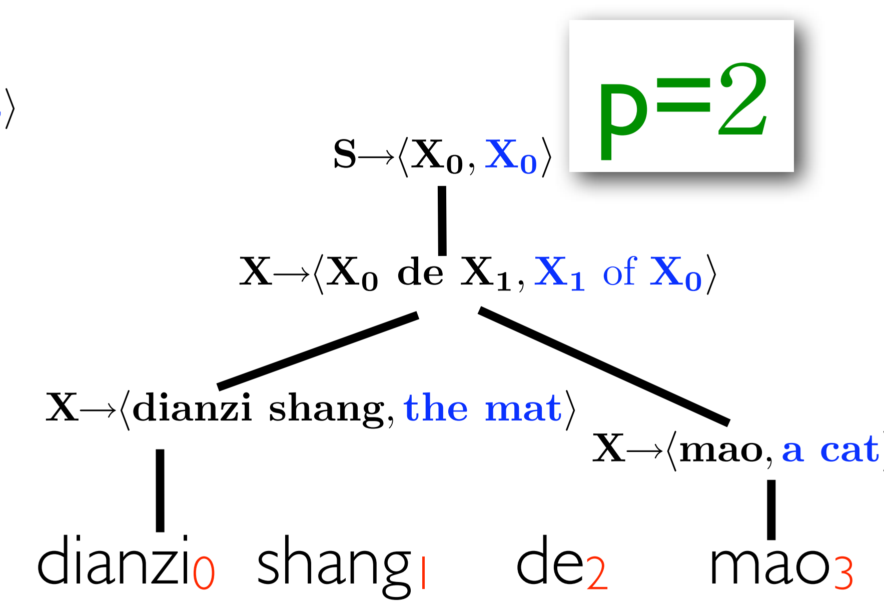
foreign input

$$p(d \mid x) = \theta \cdot \Phi(d, x)$$

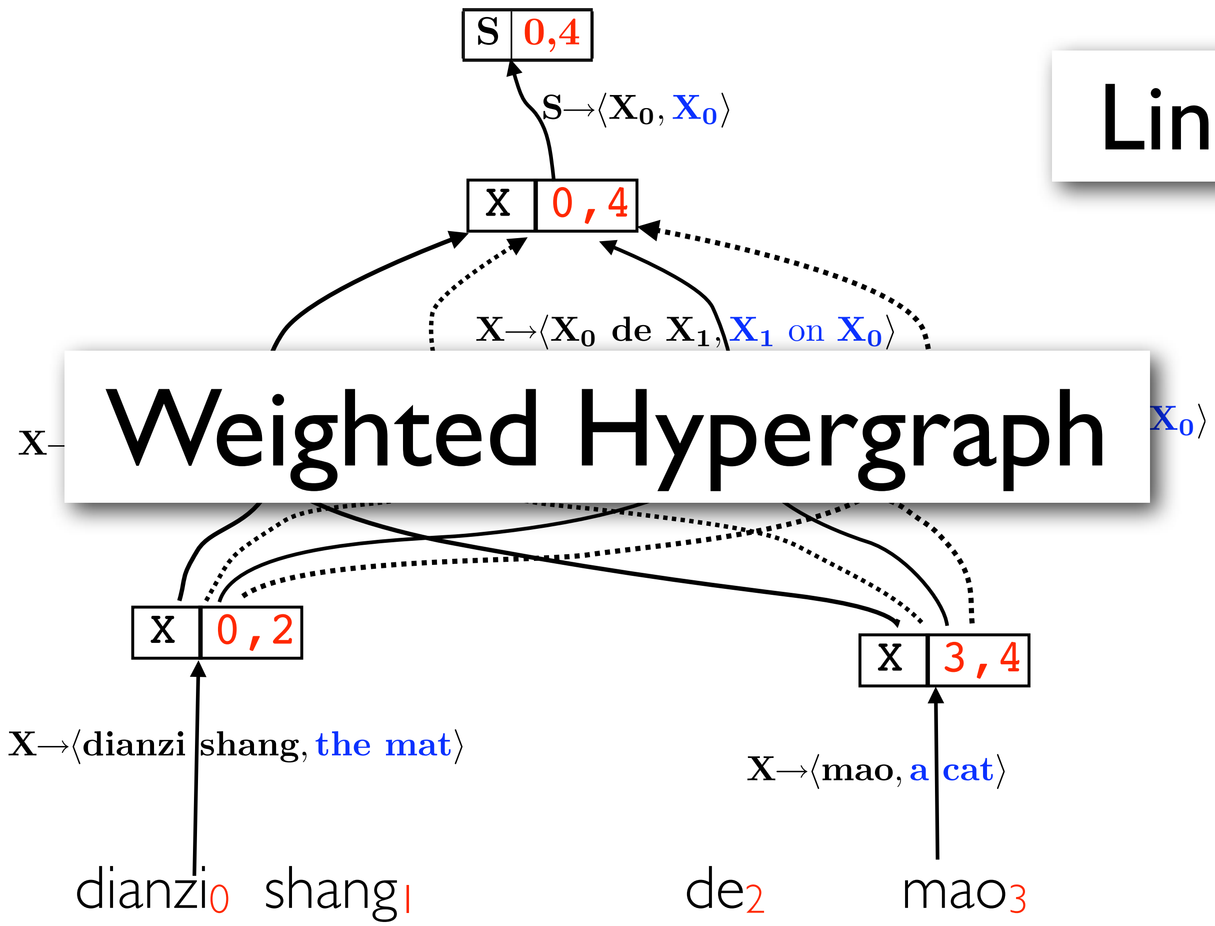








Weighted Hypergraph



Linear model:

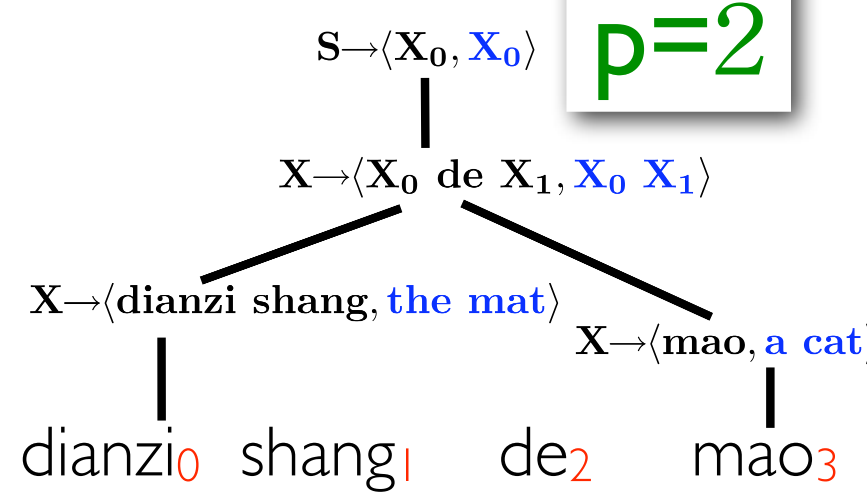
$$p(d \mid x) = \theta \cdot \Phi(d, x)$$

derivation

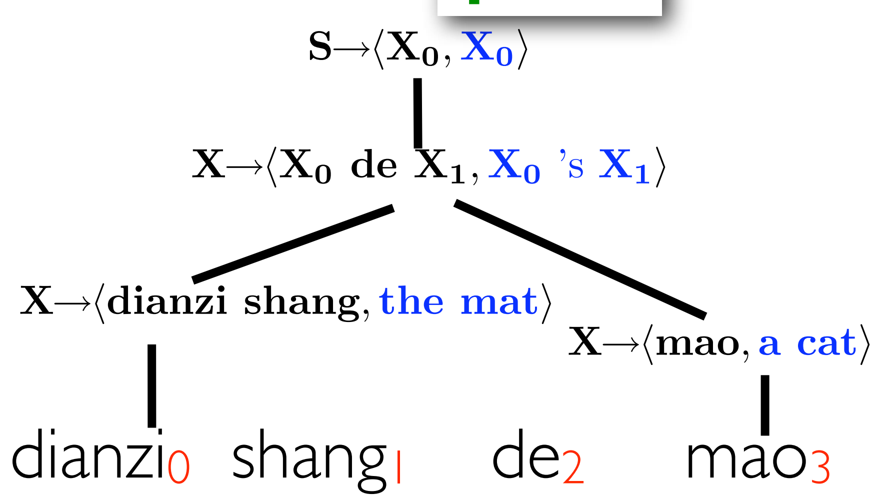
foreign input

features

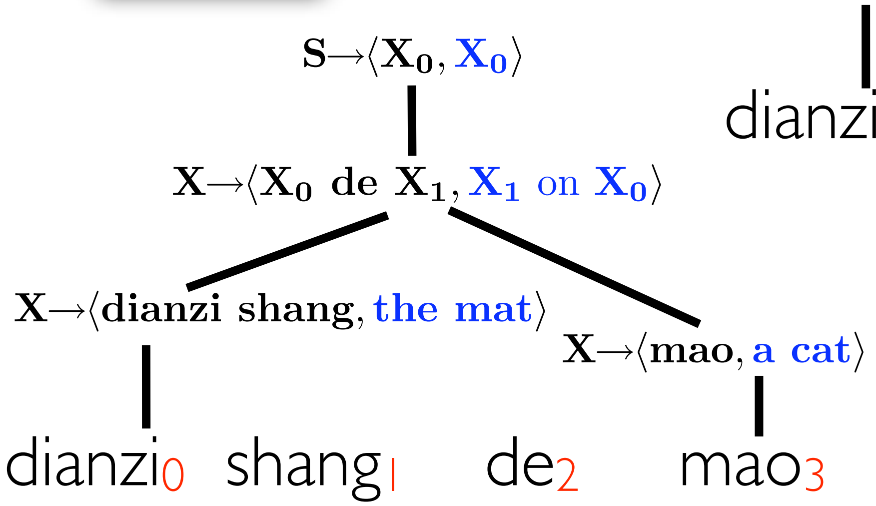
p=2



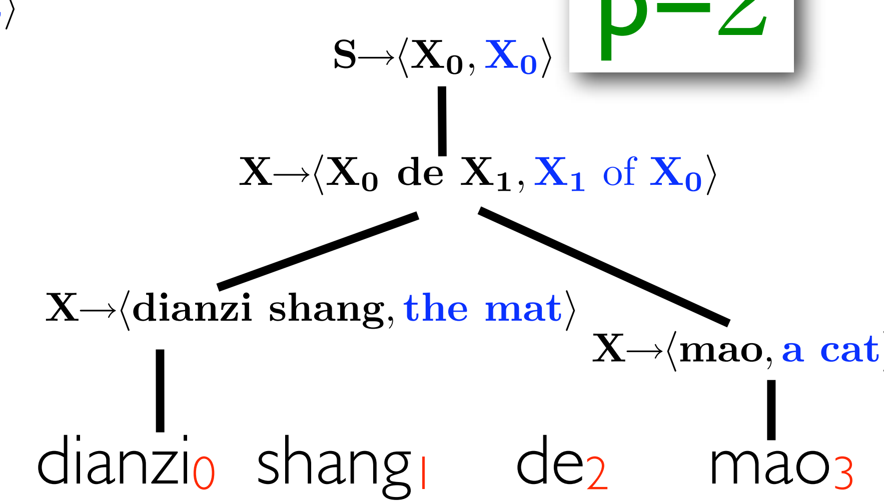
p=3



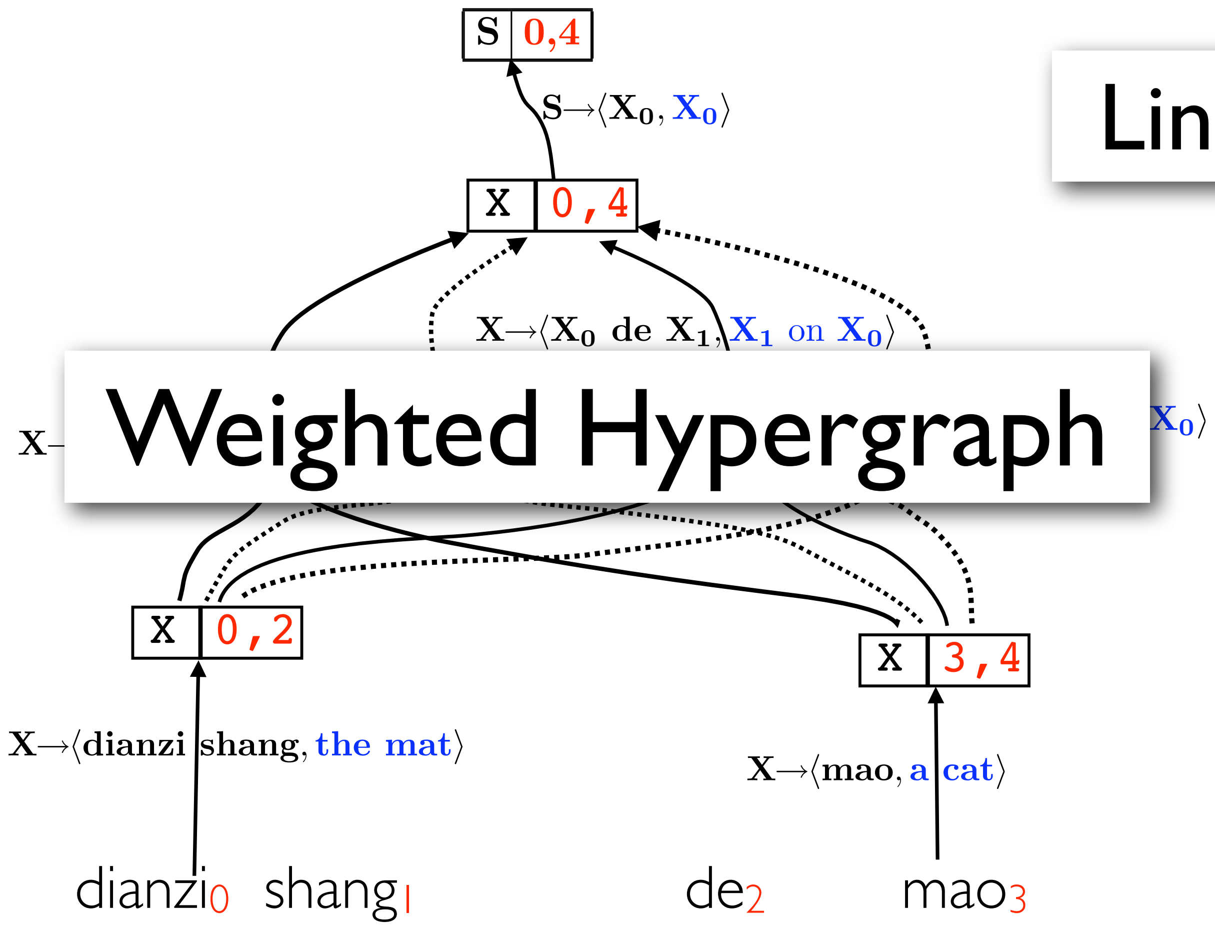
p=1



p=2



Weighted Hypergraph



Linear model:

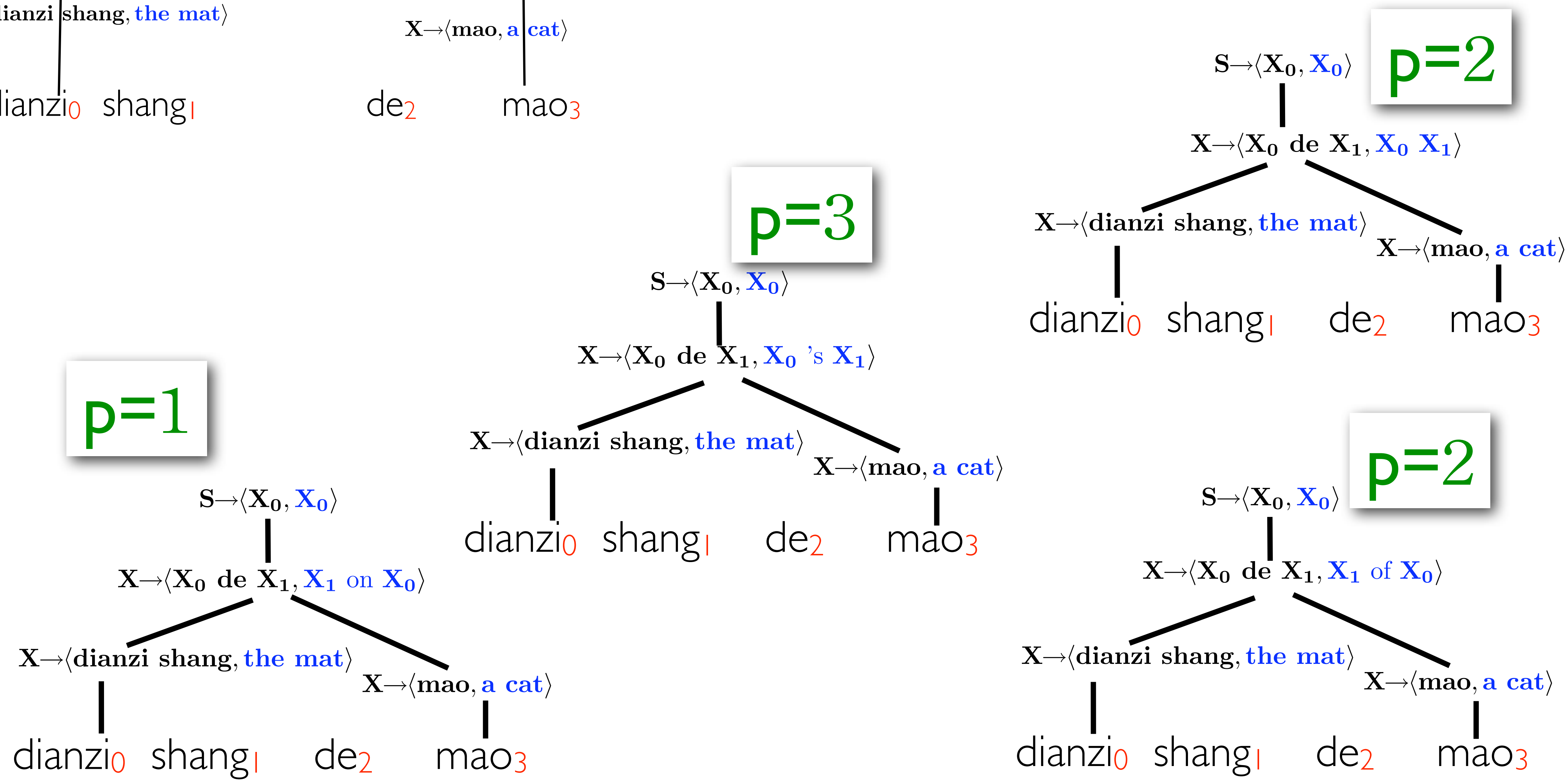
$$p(d \mid x) = \theta \cdot \Phi(d, x)$$

derivation

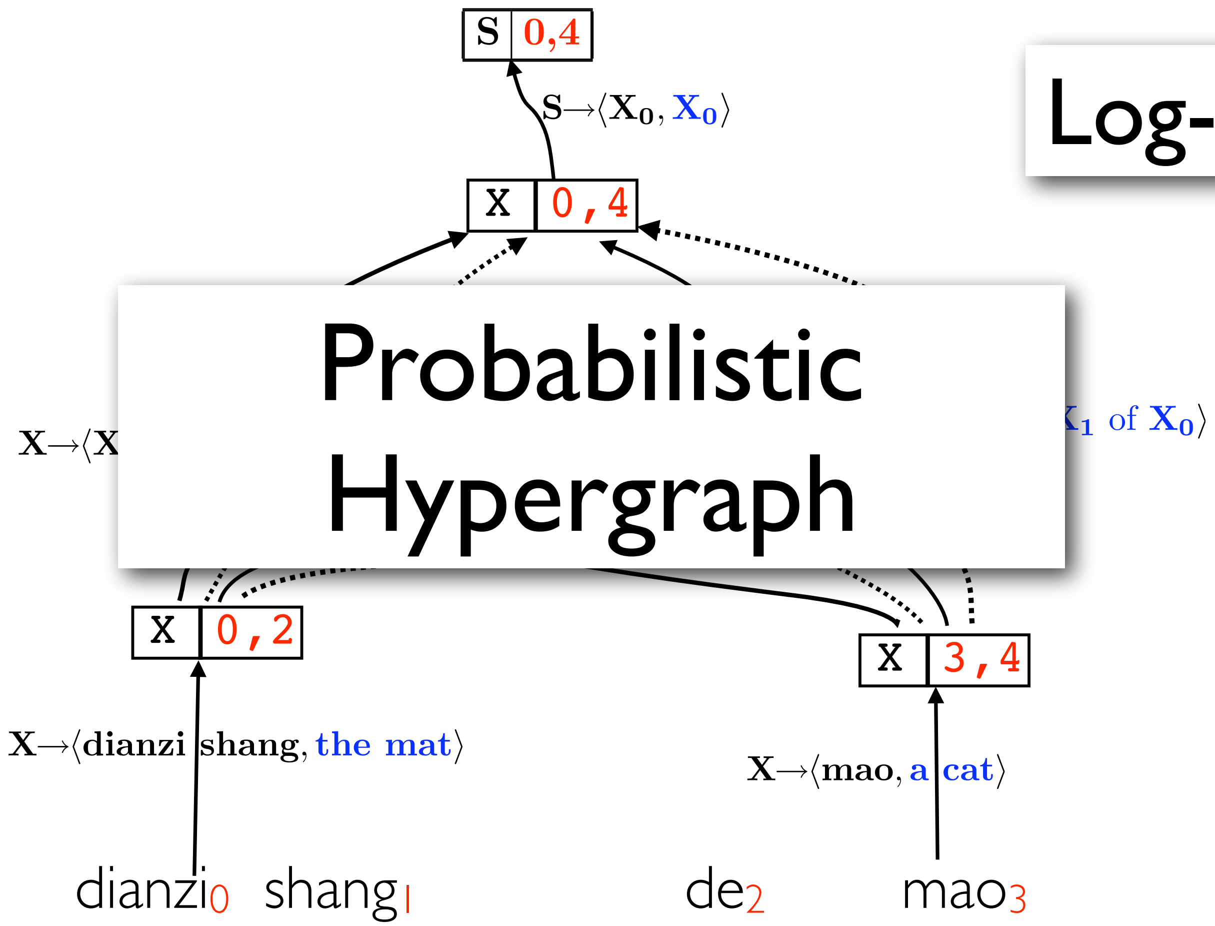
foreign input

weights

features



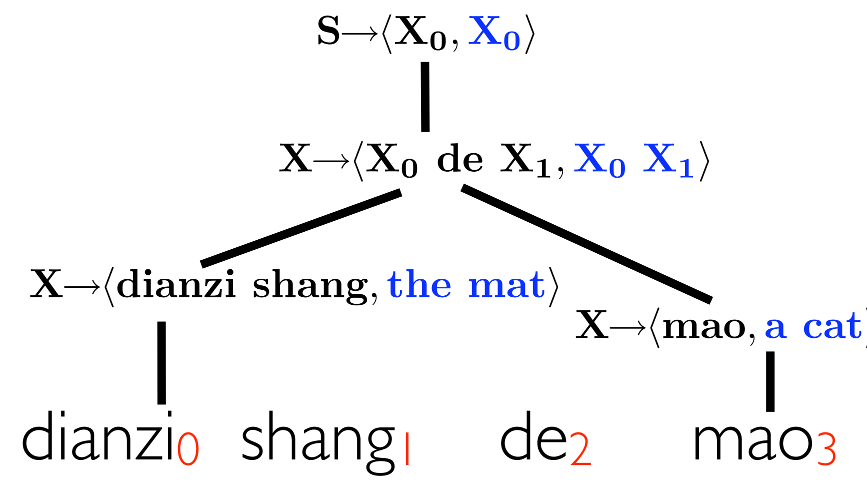
Probabilistic Hypergraph

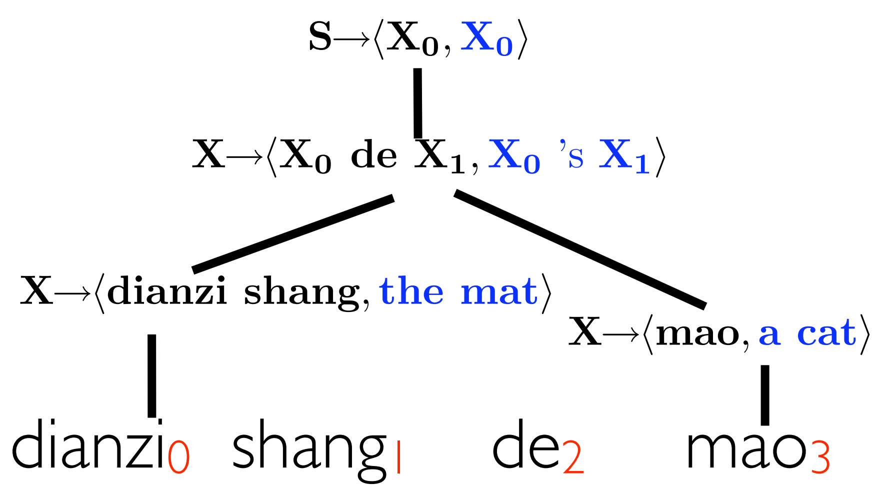


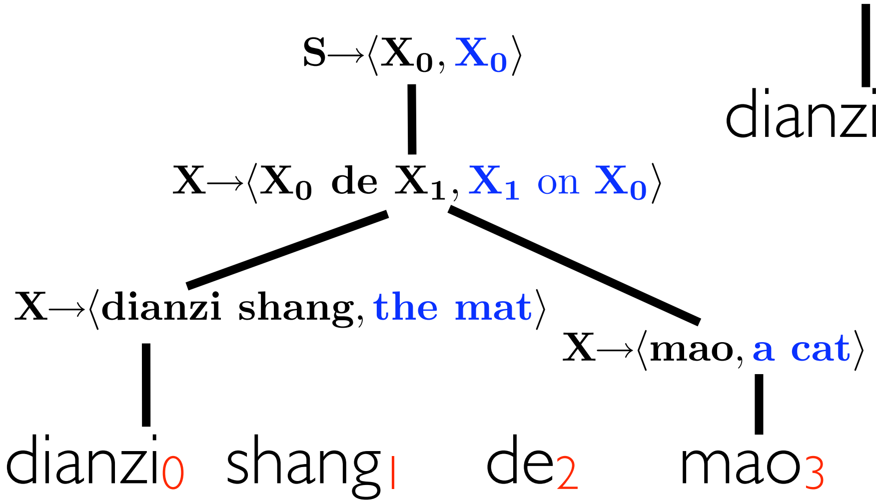
Log-linear model:

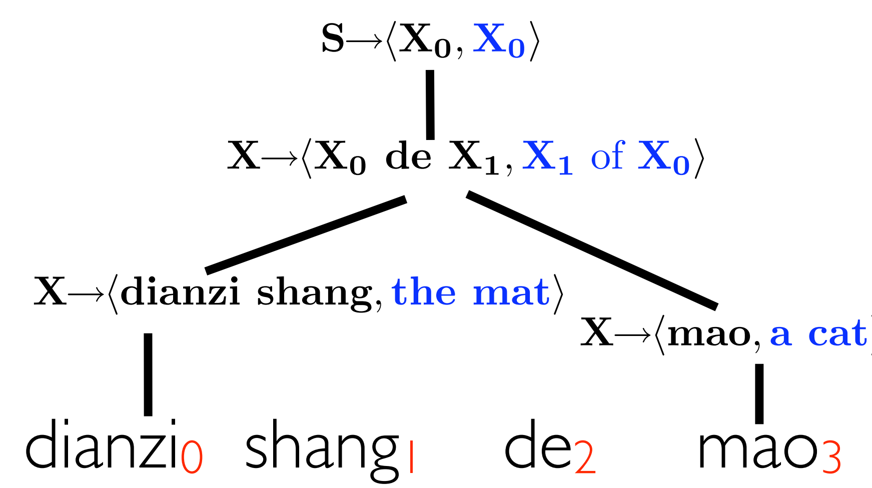
$$p(d \mid x) = \frac{e^{\theta \cdot \Phi(d,x)}}{Z(x)}$$

Z=2+1+3+2=8

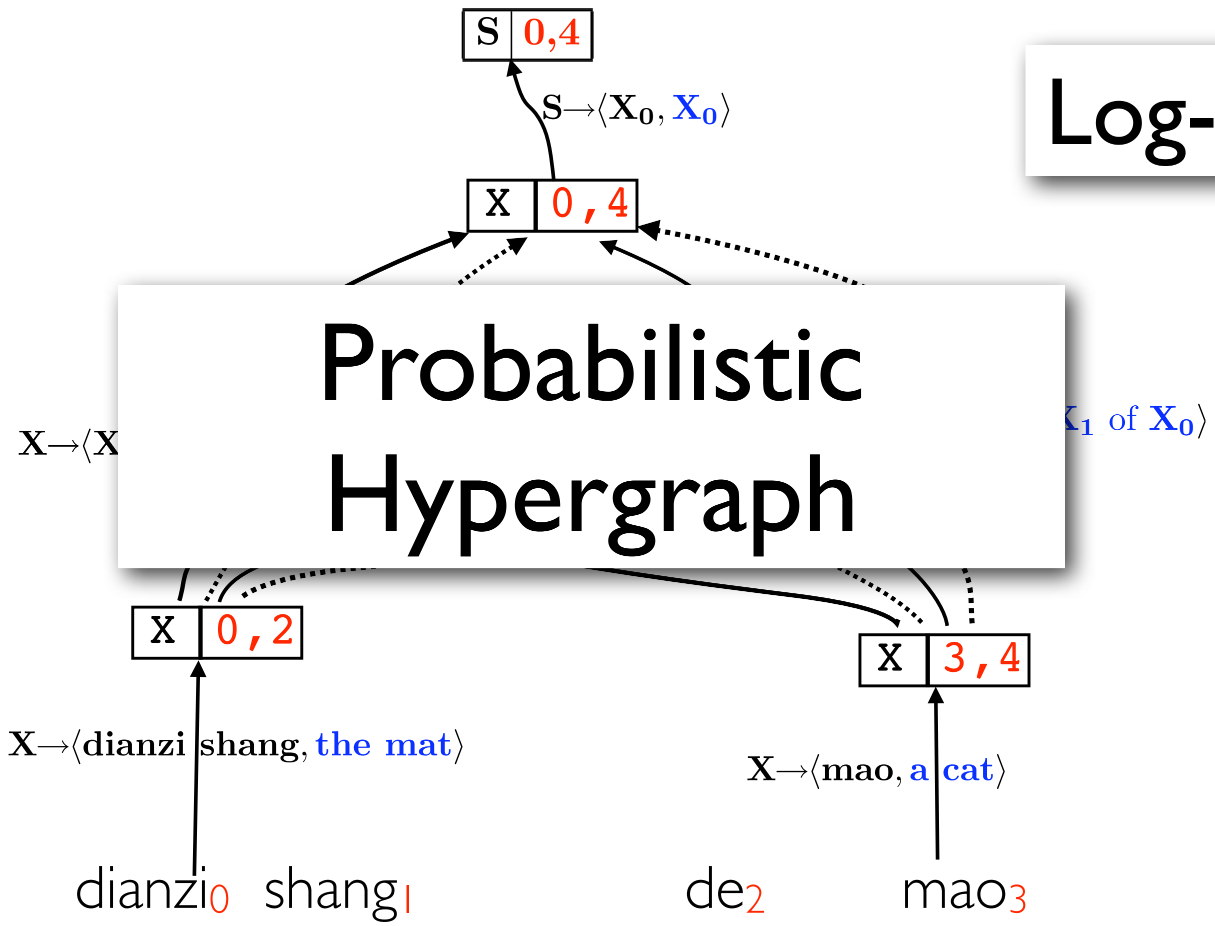








Probabilistic Hypergraph

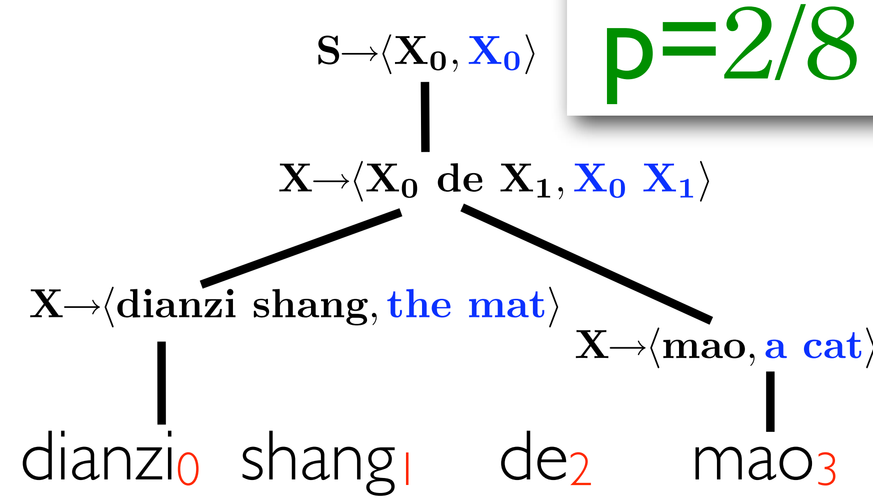


Log-linear model:

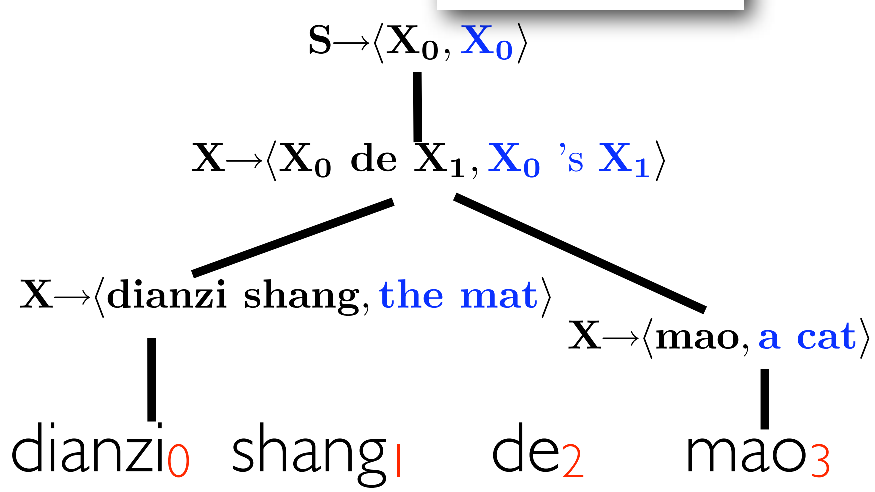
$$p(d \mid x) = \frac{e^{\theta \cdot \Phi(d,x)}}{Z(x)}$$

Z=2+1+3+2=8

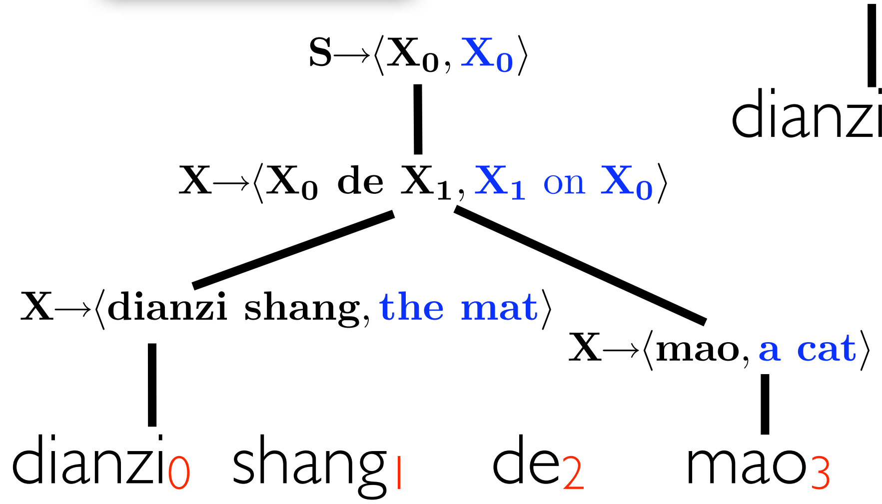
p=2/8



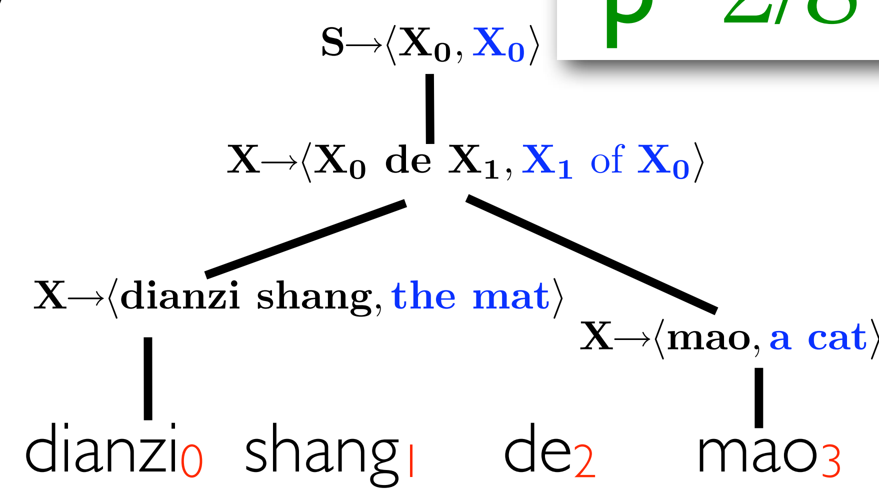
p=3/8



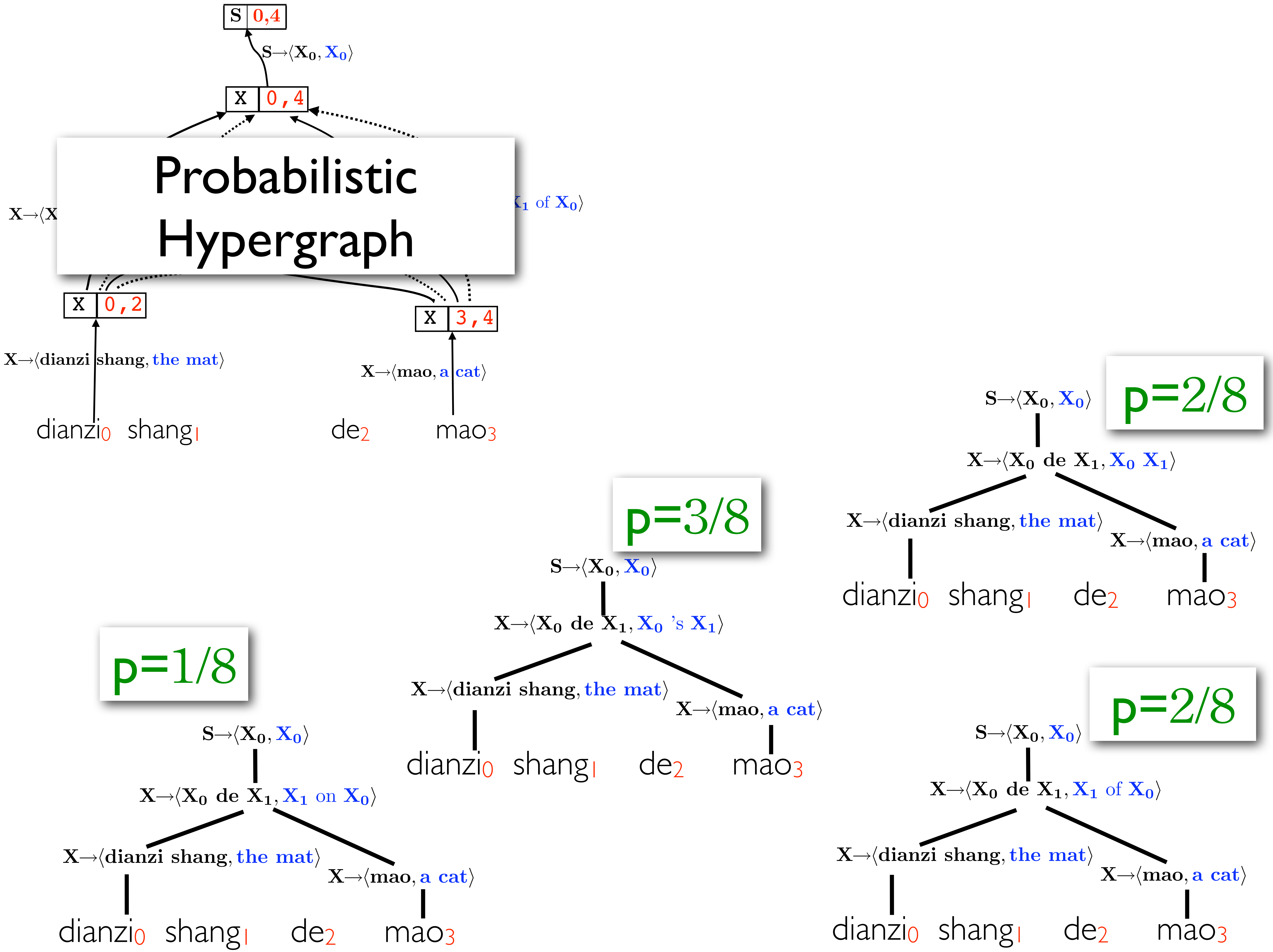
p=1/8



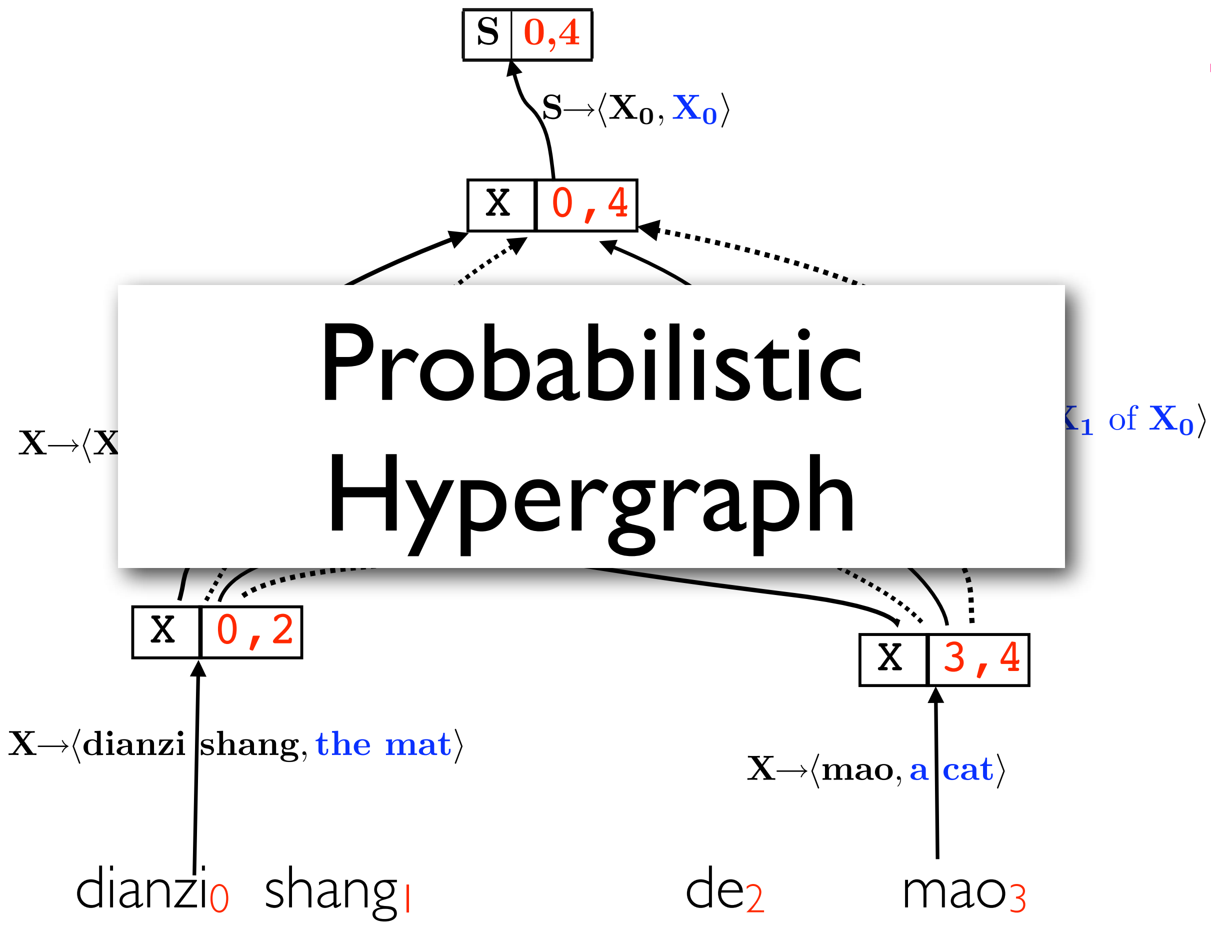
p=2/8

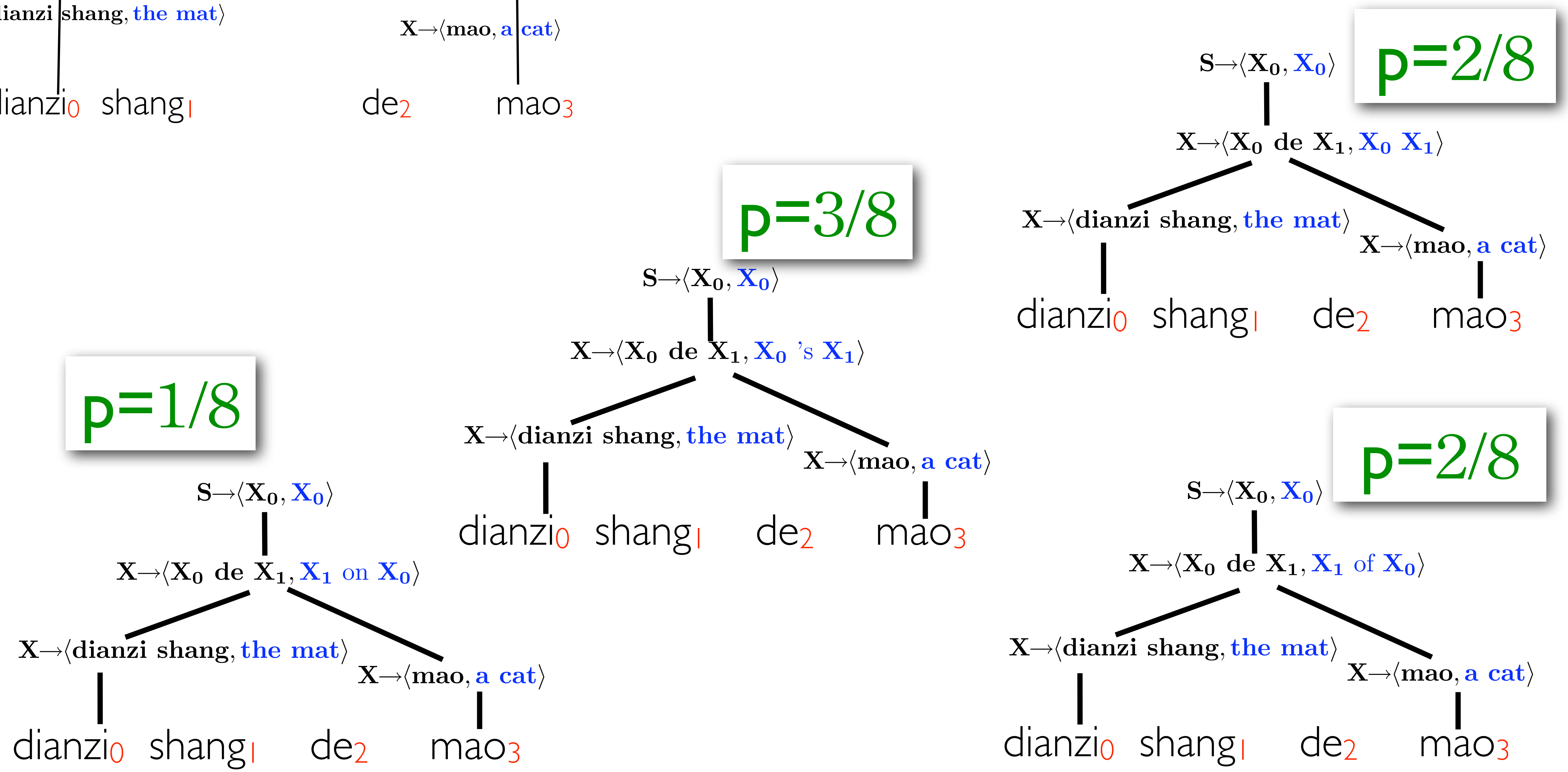


Probabilistic Hypergraph



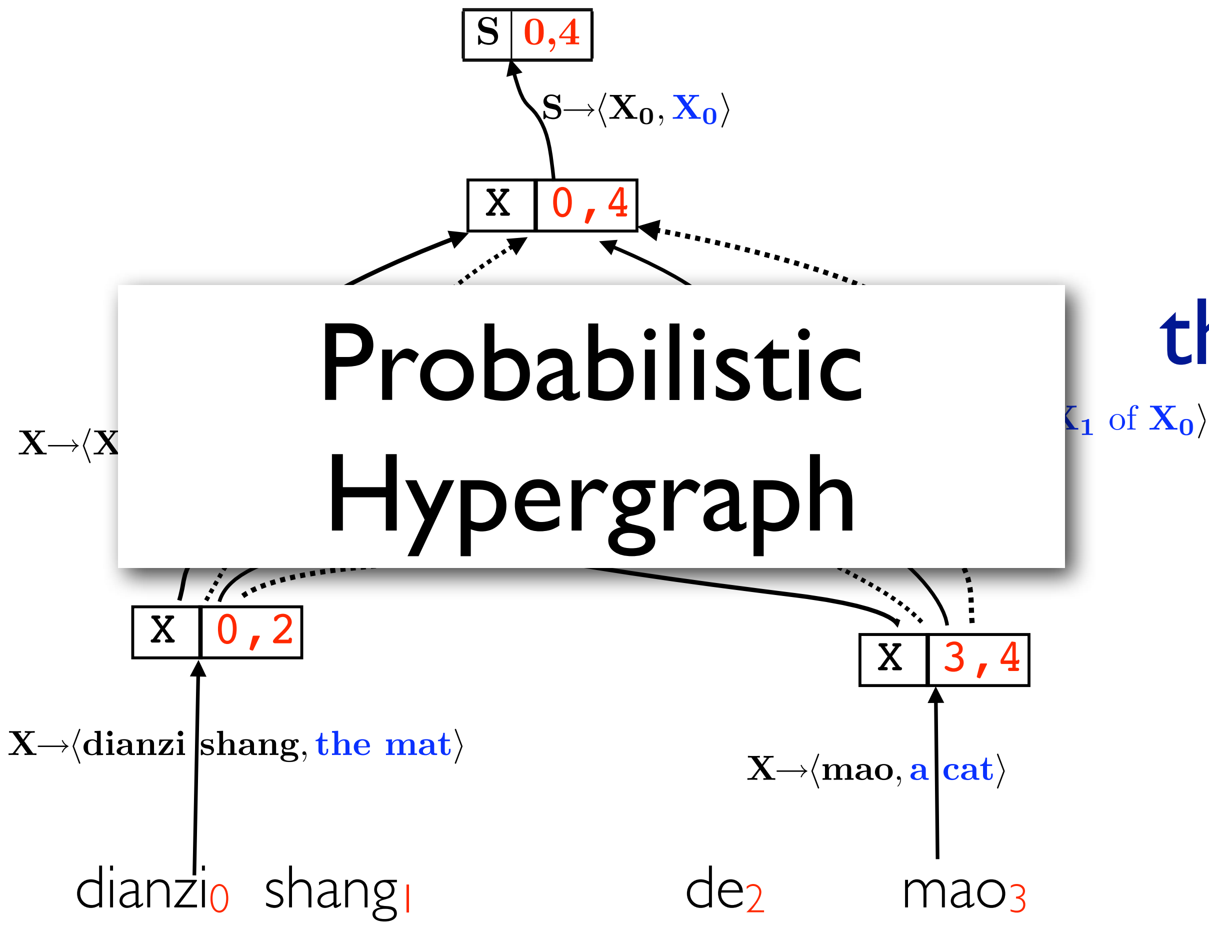
The hypergraph defines a probability distribution over **trees**!

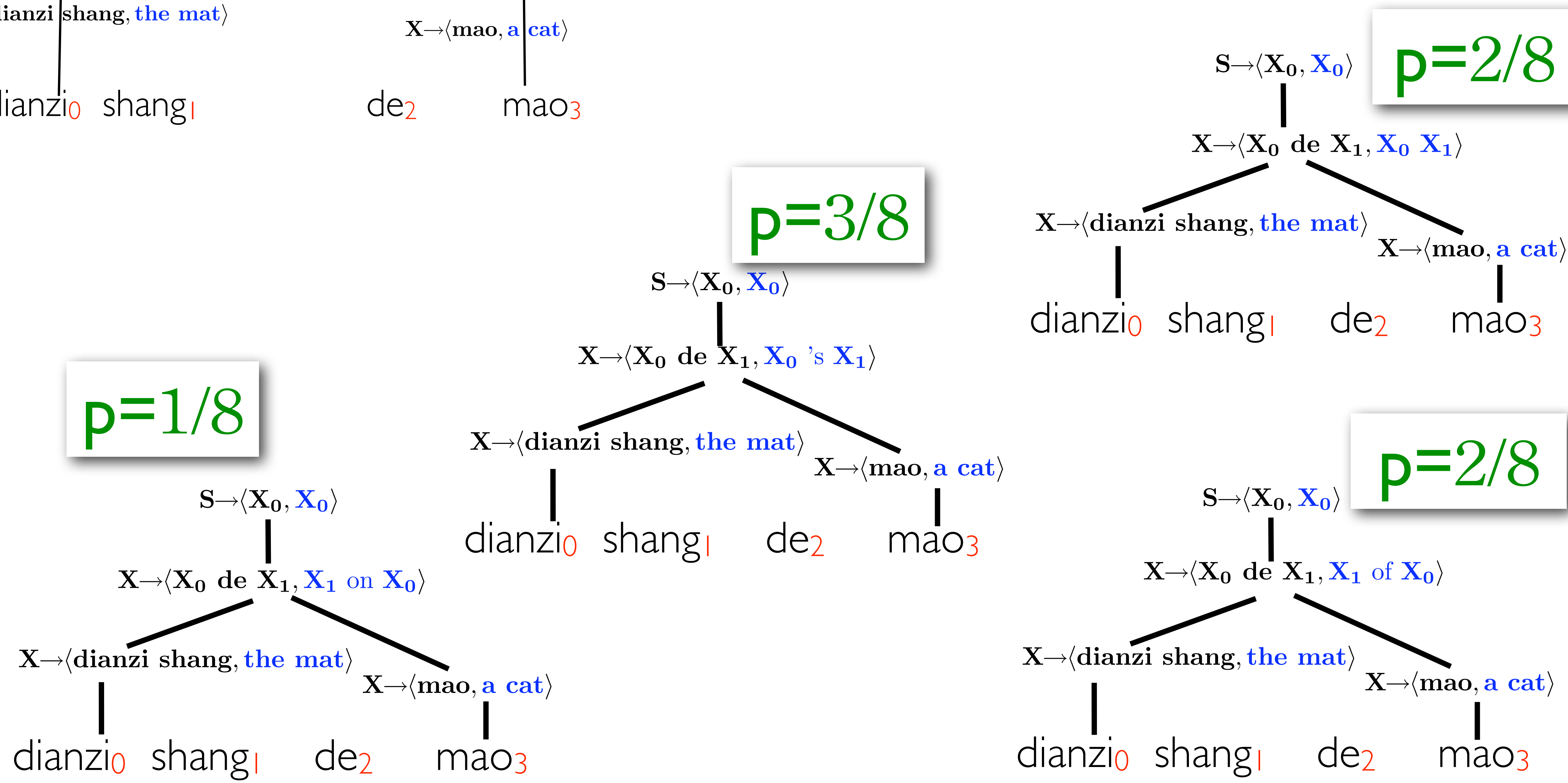


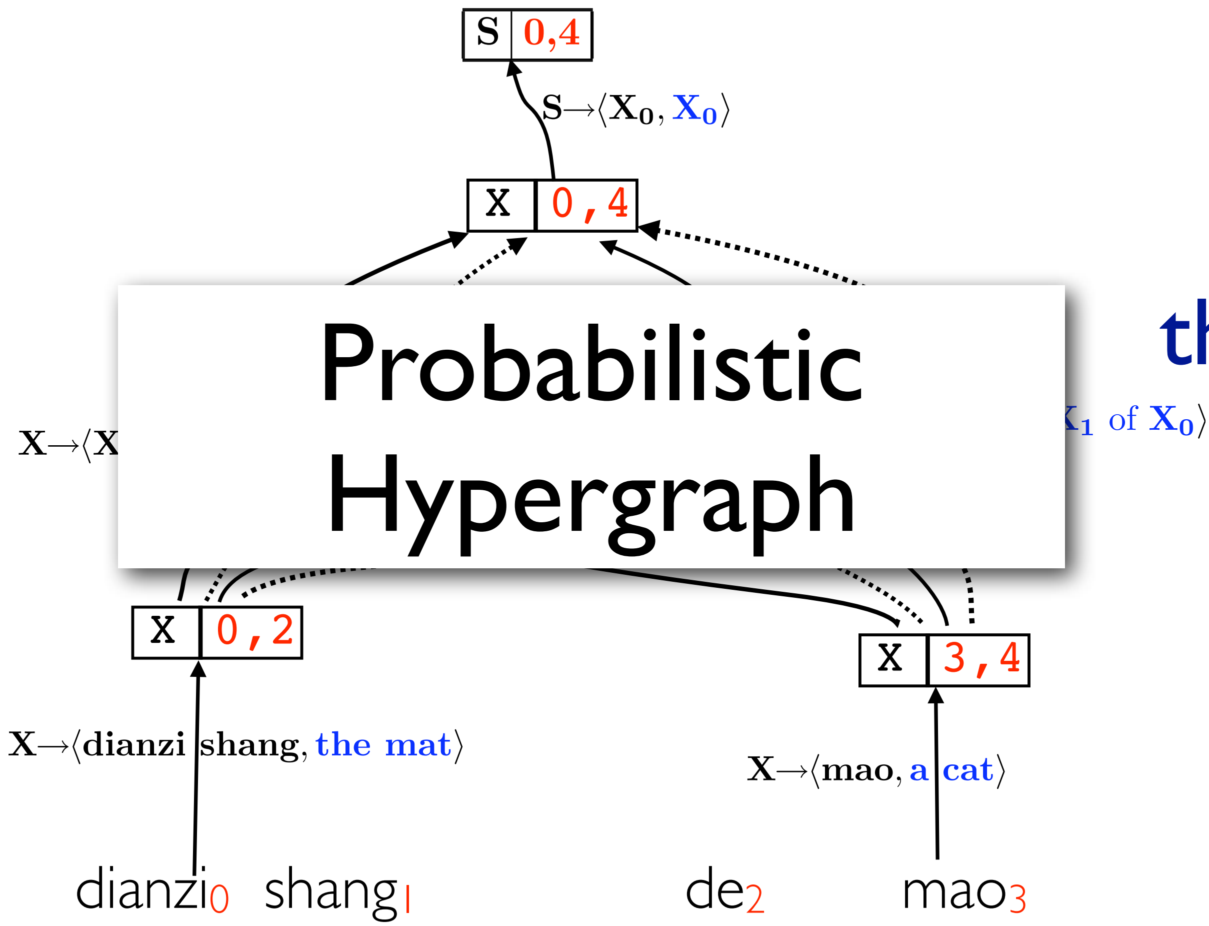


The hypergraph defines a probability distribution over **trees**!

the distribution is parameterized by Θ



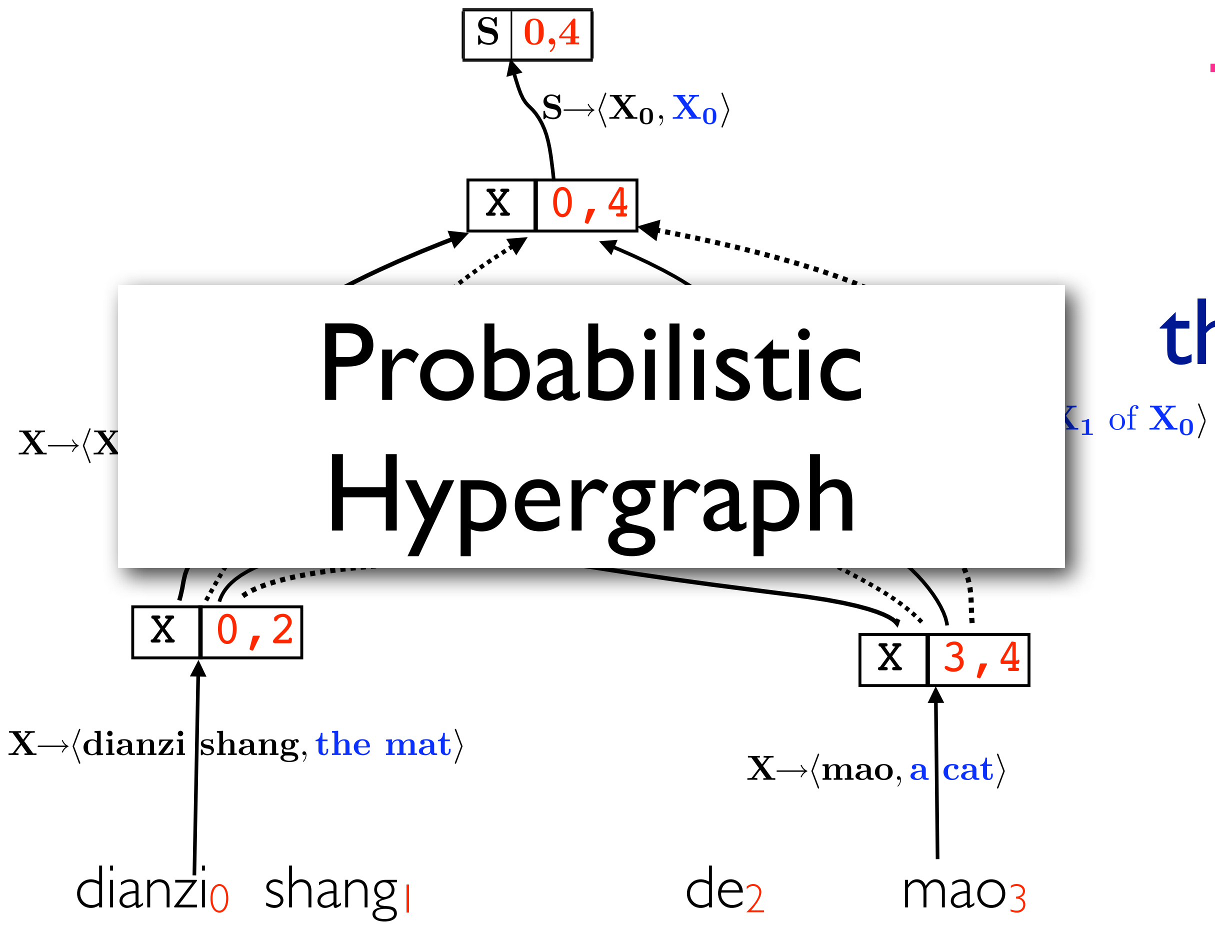




The hypergraph defines a probability distribution over **trees**!

the distribution is parameterized by Θ

Probabilistic Hypergraph

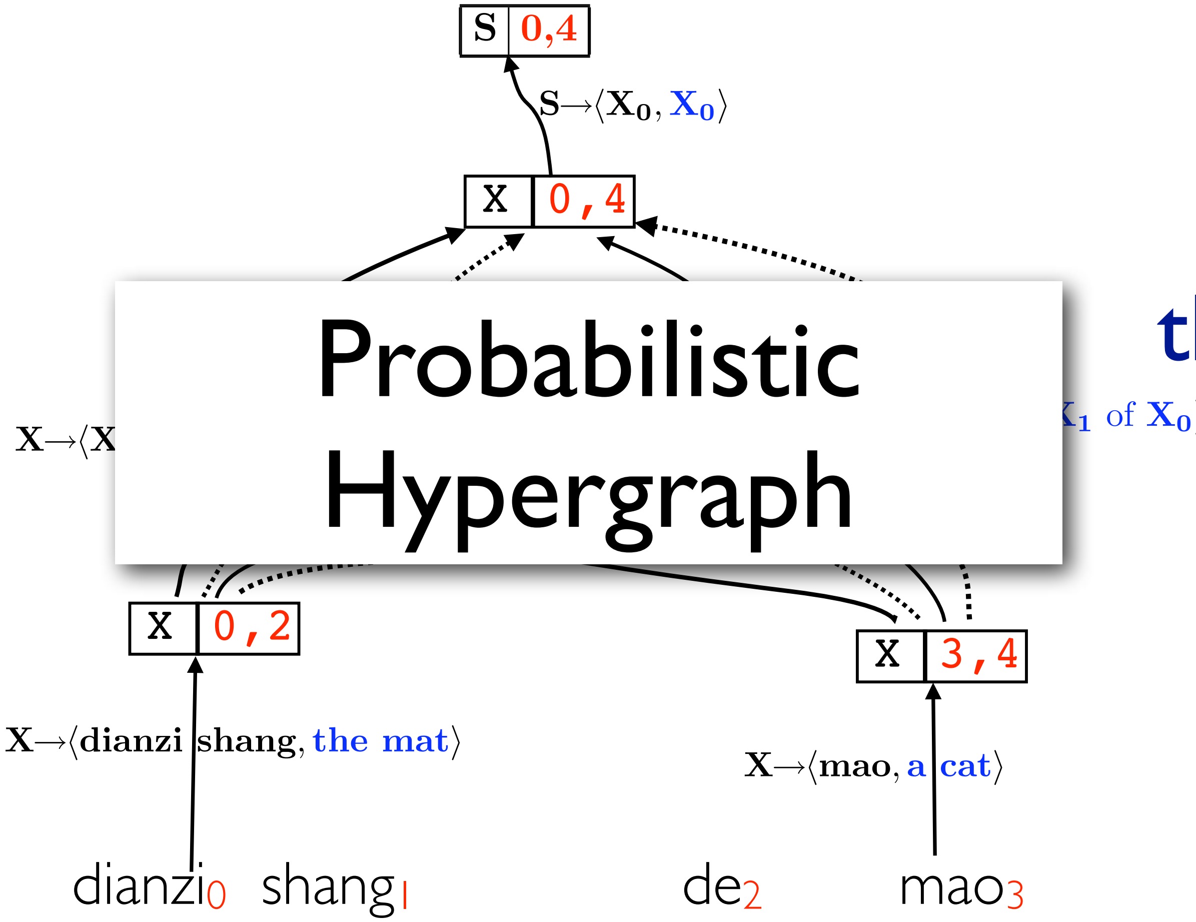


The hypergraph defines a probability distribution over **trees**!

the distribution is parameterized by Θ

Which translation do we present to a user?

Decoding

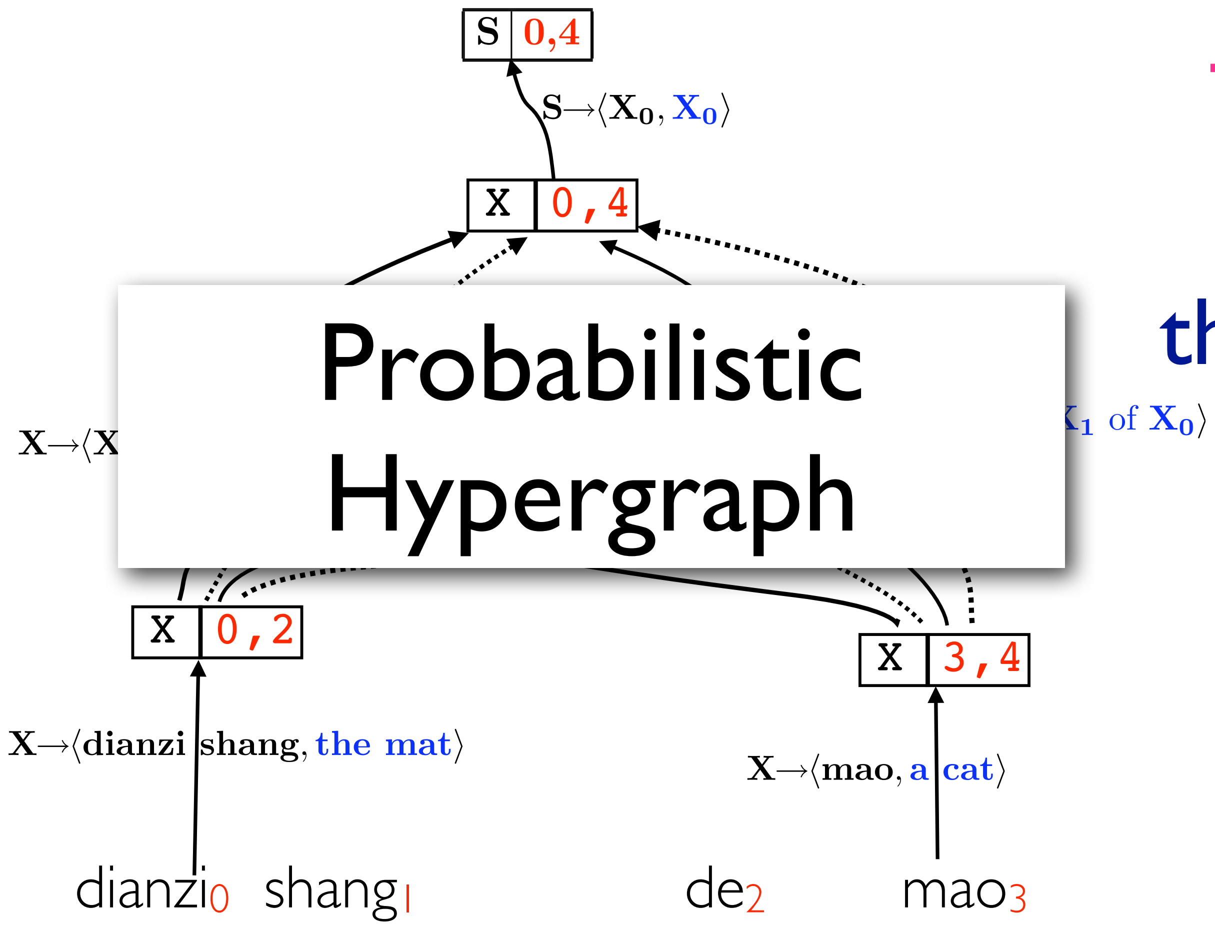


The hypergraph defines a probability distribution over **trees**!

the distribution is parameterized by Θ

Which translation do we present to a user? Decoding

How do we set the parameters Θ? Training



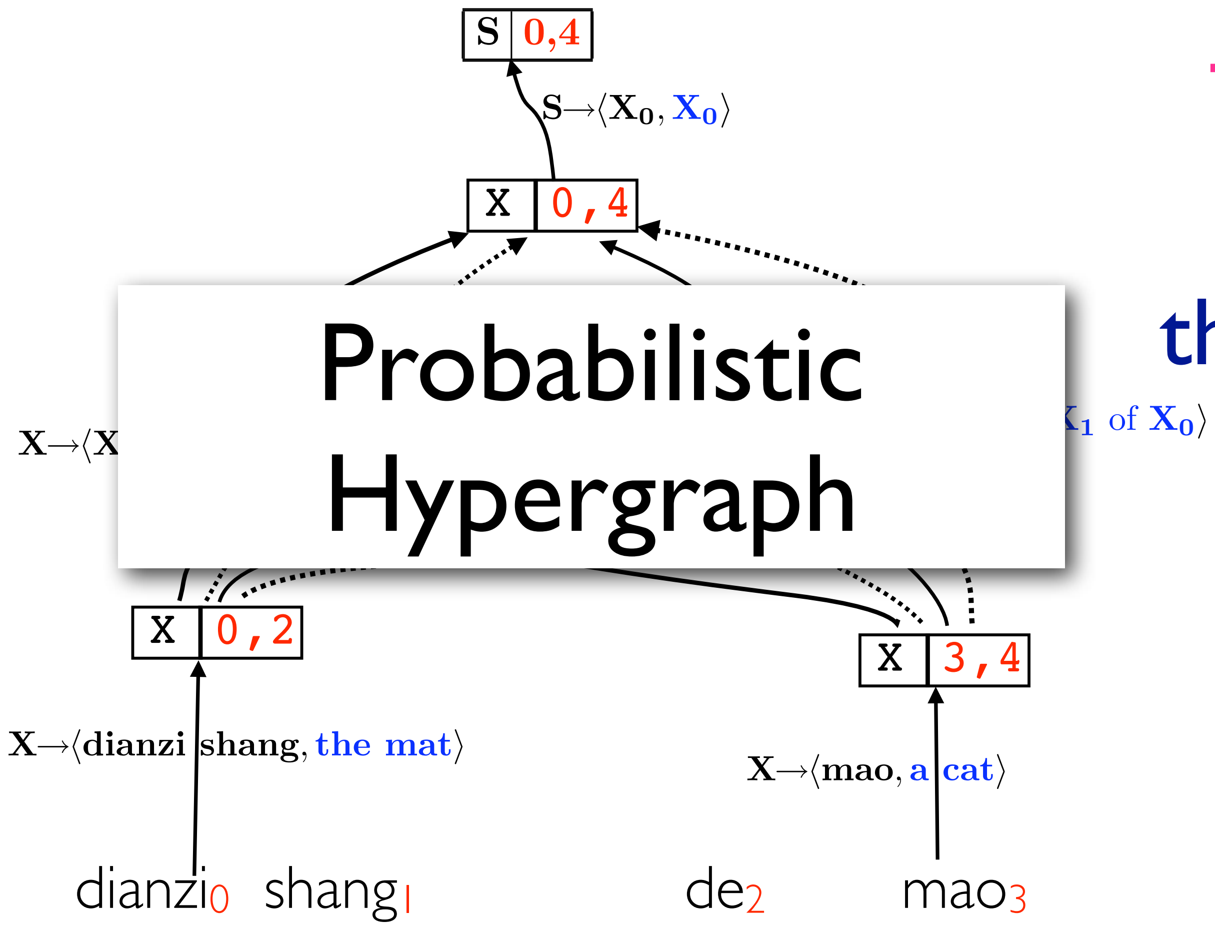
The hypergraph defines a probability distribution over **trees**!

the distribution is parameterized by Θ

Which translation do we present to a user? Decoding

How do we set the parameters Θ? Training

What atomic operations do we need to perform? Atomic Inference



The hypergraph defines a probability distribution over **trees**!

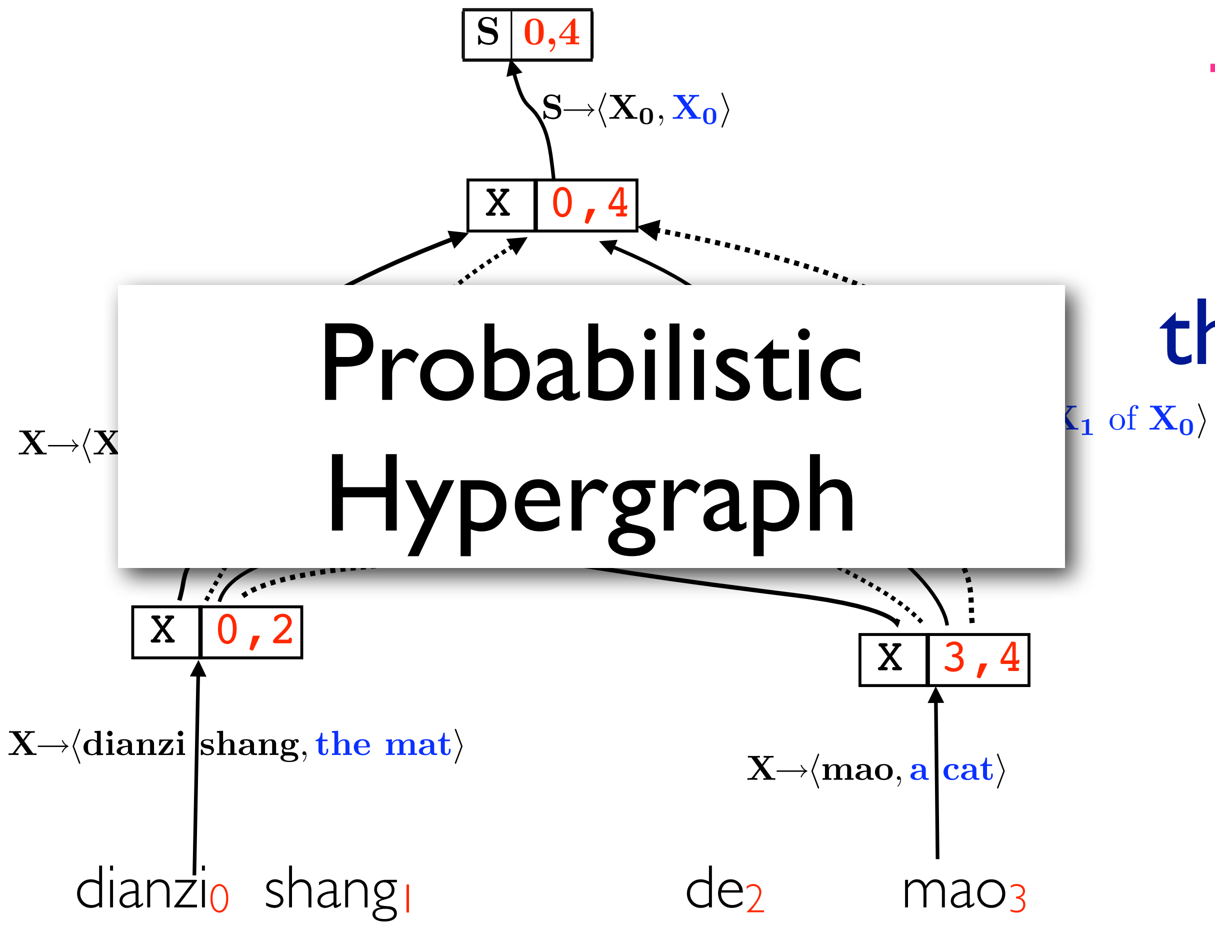
the distribution is parameterized by Θ

training (e.g., mert)	**decoding** (e.g., mbr)
atomic inference operations (e.g., finding one-best, k-best or expectation, inference can be *exact* or *approximate*)	

Which translation do we present to a user? Decoding

How do we set the parameters Θ? Training

What atomic operations do we need to perform? Atomic Inference



The hypergraph defines a probability distribution over **trees**!

the distribution is parameterized by Θ

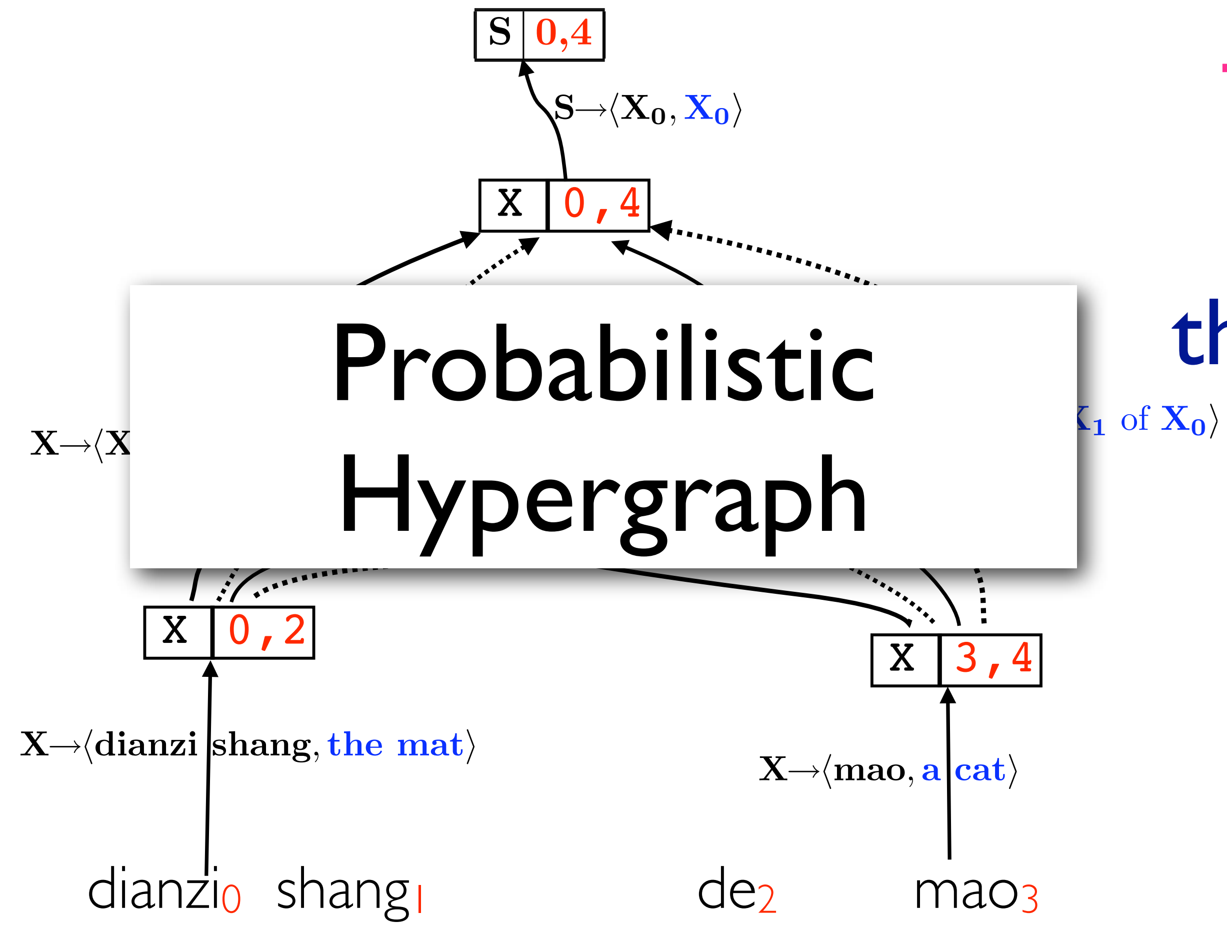
training (e.g., mert)	**decoding** (e.g., mbr)
atomic inference operations (e.g., finding one-best, k-best or expectation, inference can be *exact* or *approximate*)	

Which translation do we present to a user? Decoding

How do we set the parameters Θ? Training

What atomic operations do we need to perform? Atomic Inference

Why are the problems difficult?



The hypergraph defines a probability distribution over **trees**!

the distribution is parameterized by Θ

training (e.g., mert)	**decoding** (e.g., mbr)
atomic inference operations (e.g., finding one-best, k-best or expectation, inference can be *exact* or *approximate*)	

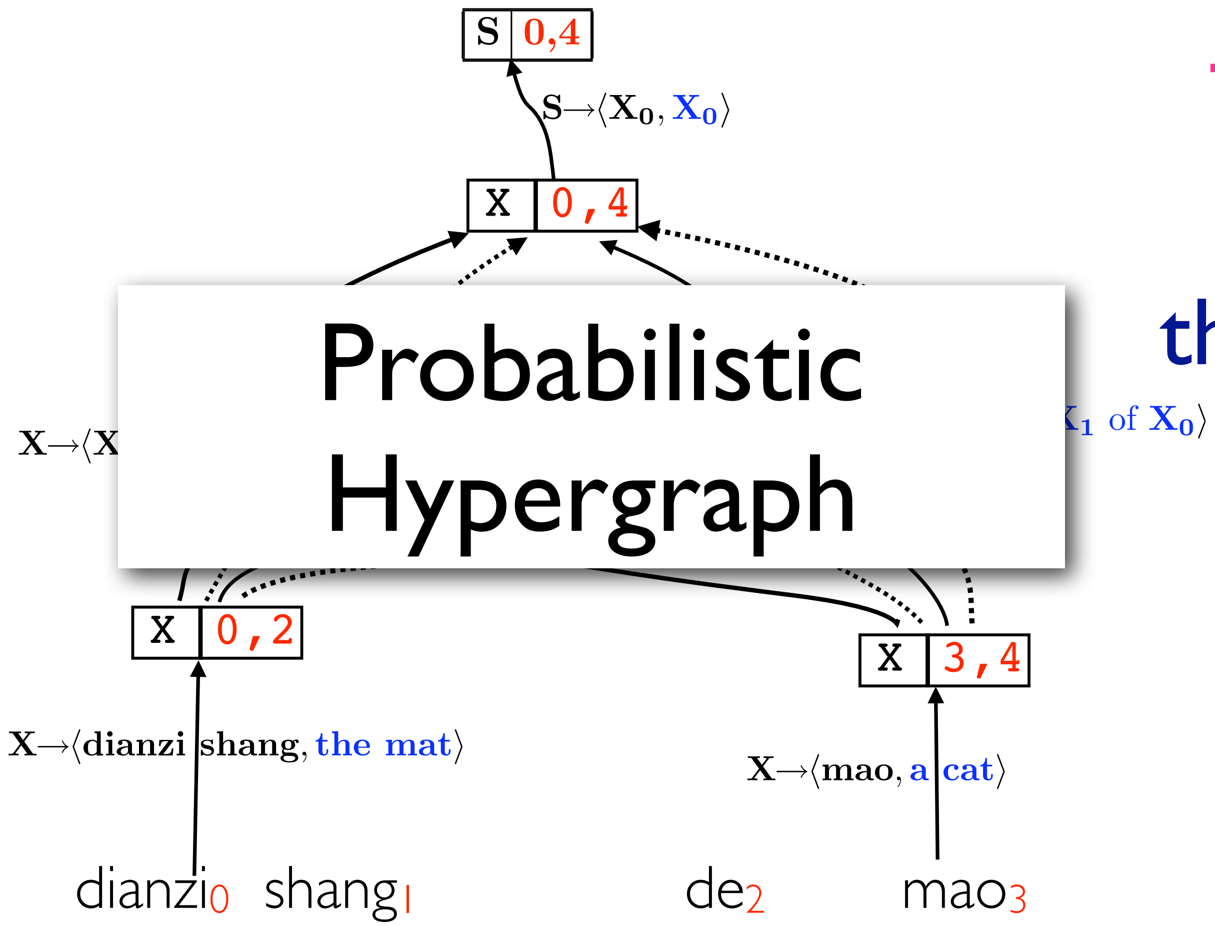
Which translation do we present to a user? Decoding

How do we set the parameters Θ? Training

What atomic operations do we need to perform? Atomic Inference

Why are the problems difficult?

- brute-force will be too slow as there are exponentially many trees, so require sophisticated dynamic programs



The hypergraph defines a probability distribution over **trees**!

the distribution is parameterized by Θ

training (e.g., mert)	decoding (e.g., mbr)
atomic inference operations (e.g., finding one-best, k-best or expectation, inference can be *exact* or *approximate*)	

Which translation do we present to a user? Decoding

How do we set the parameters Θ? Training

What atomic operations do we need to perform? Atomic Inference

Why are the problems difficult?

- brute-force will be too slow as there are exponentially many trees, so require sophisticated dynamic programs
- sometimes intractable, require approximations

Inference, Training and Decoding on Hypergraphs

- Atomic Inference
  - finding one-best derivations

Graph	Topological	Best-first		
		no heuristic	with heuristic	with hierarchy
FSA	Viterbi	Dijkstra	A*	HA*
Hypergraph	CYK	Knuth	Klein and Manning	Generalized A^*

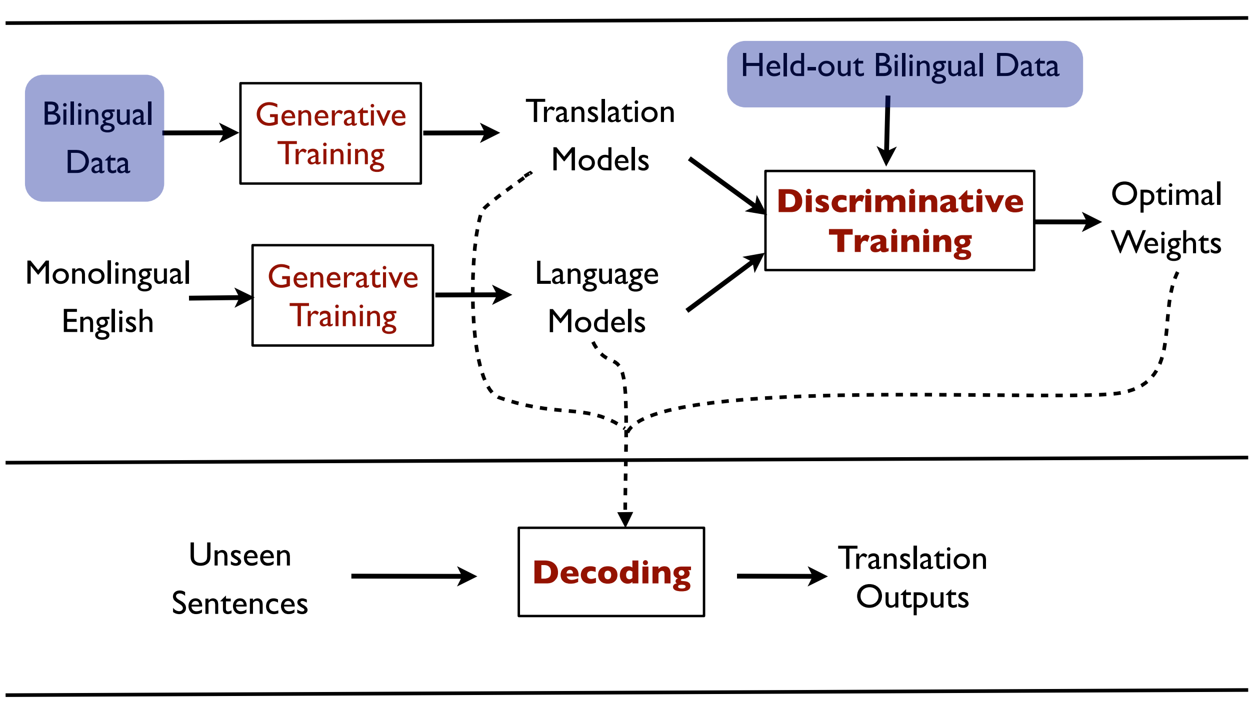
  - finding k-best derivations
  - computing expectations (e.g., of features)
- Training
  - Perceptron, conditional random field (CRF), minimum error rate training (MERT), minimum risk, and MIRA
- Decoding
  - Viterbi decoding, maximum a posterior (MAP) decoding, and minimum Bayes risk (MBR) decoding

Outline

- Hypergraph as Hypothesis Space
- Unsupervised Discriminative Training
  - minimum imputed risk
  - contrastive language model estimation
- Variational Decoding
- First- and Second-order Expectation Semirings

Outline

- Hypergraph as Hypothesis Space
- Unsupervised Discriminative Training
  - minimum imputed risk
  - contrastive language model estimation
- Variational Decoding
- First- and Second-order Expectation Semirings

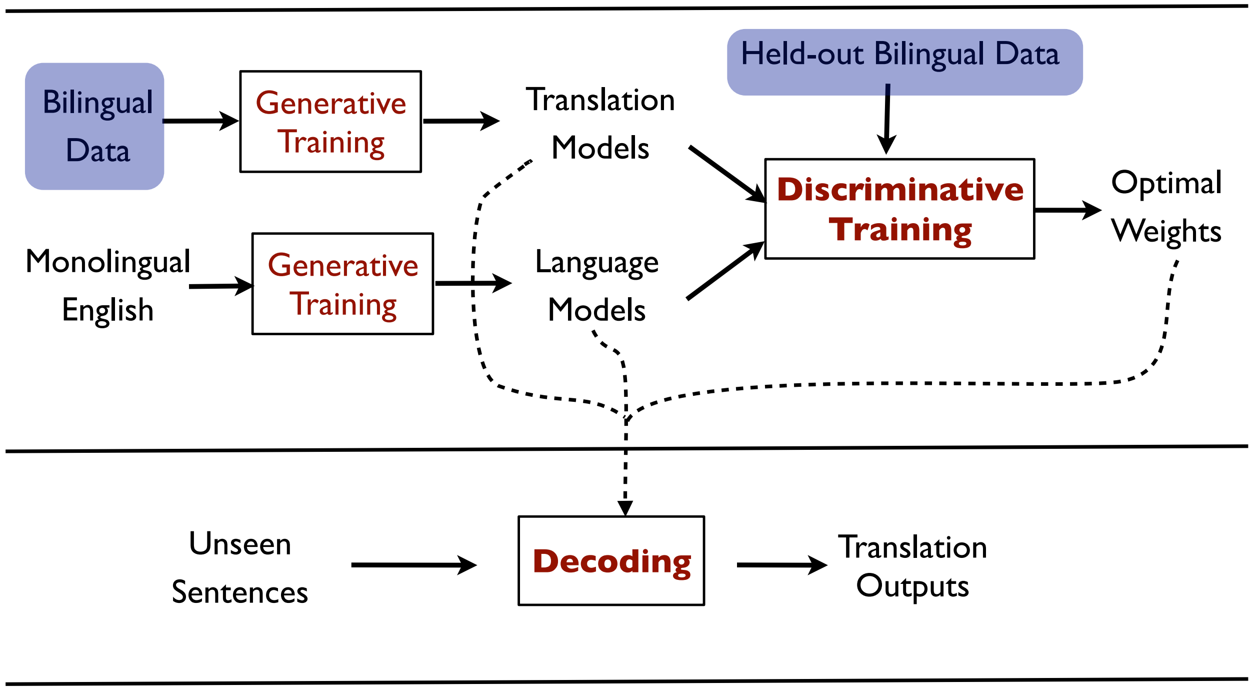


Outline

- Hypergraph as Hypothesis Space
- Unsupervised Discriminative Training
  - minimum imputed risk
  - contrastive language model estimation

main focus

- Variational Decoding
- First- and Second-order Expectation Semirings

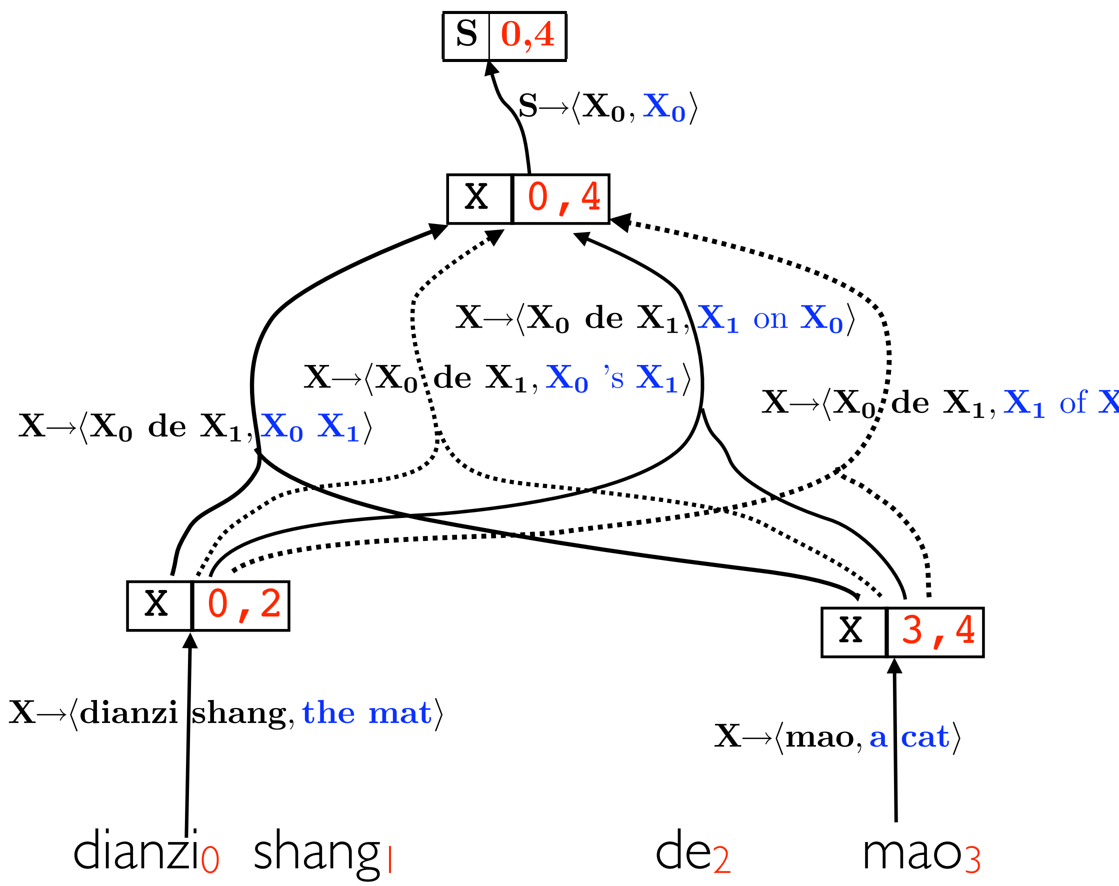


Training Setup

- Each **training example** consists of
  - a foreign sentence (from which a hypergraph is generated to represent many possible translations)
  - a reference translation

x: dianzi shang de mao

y: a cat on the mat



- **Training**
  - adjust the parameters Θ so that the reference translation is preferred by the model

Supervised: Minimum Empirical Risk

Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_\theta \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

empirical distribution

Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_\theta \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

empirical distribution

Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_\theta \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

empirical distribution

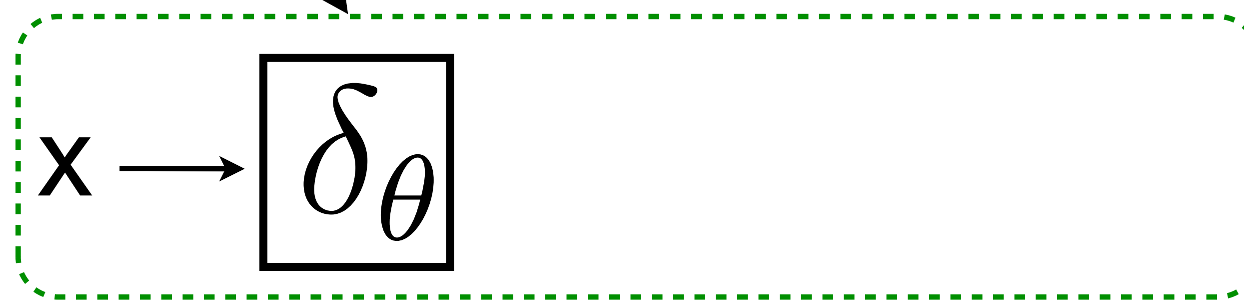
X

Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

empirical distribution



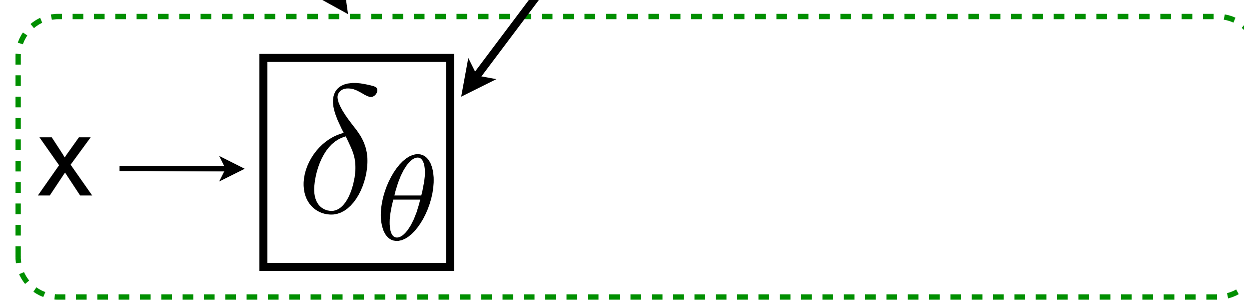
Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

MT decoder

empirical distribution



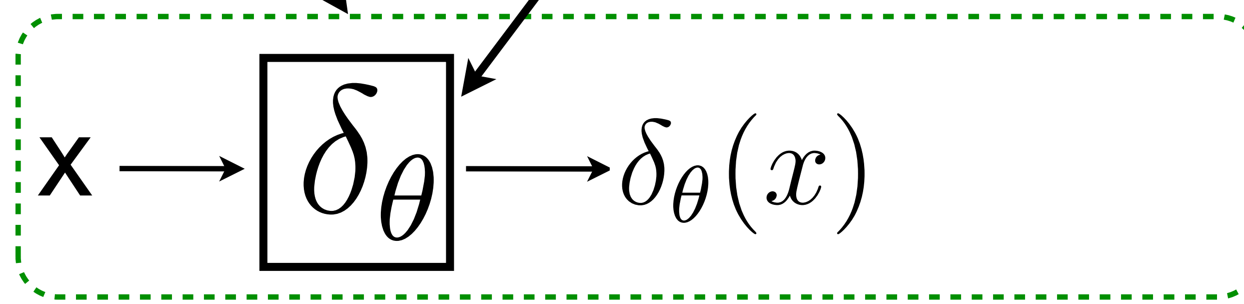
Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

MT decoder

empirical distribution



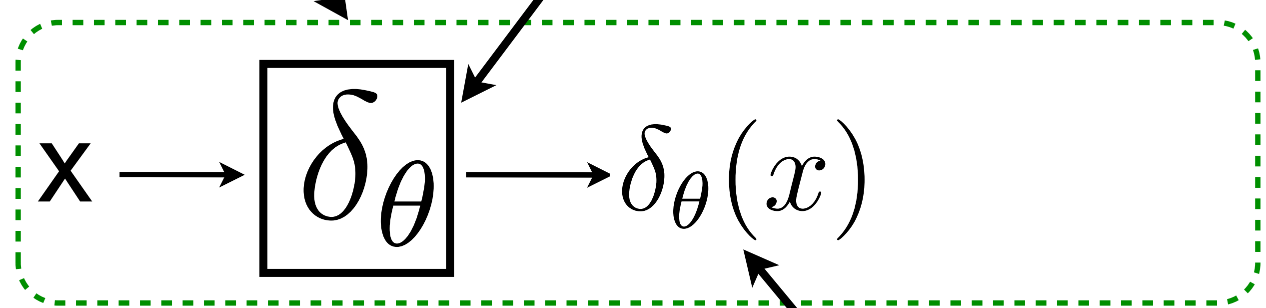
Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

empirical distribution

MT decoder



MT output

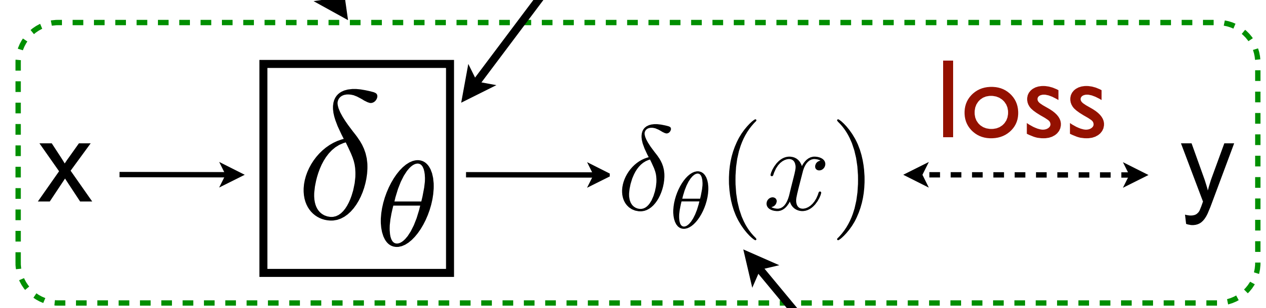
Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

empirical distribution

MT decoder

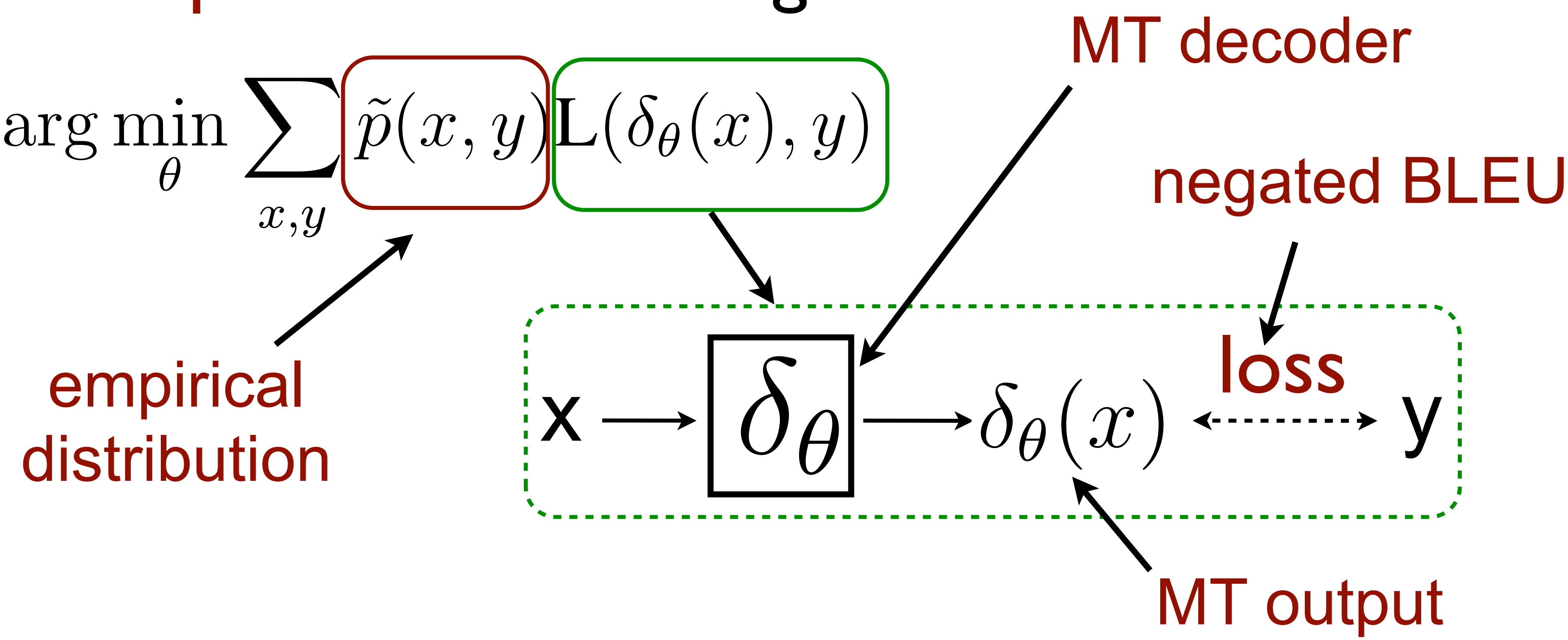


MT output

Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

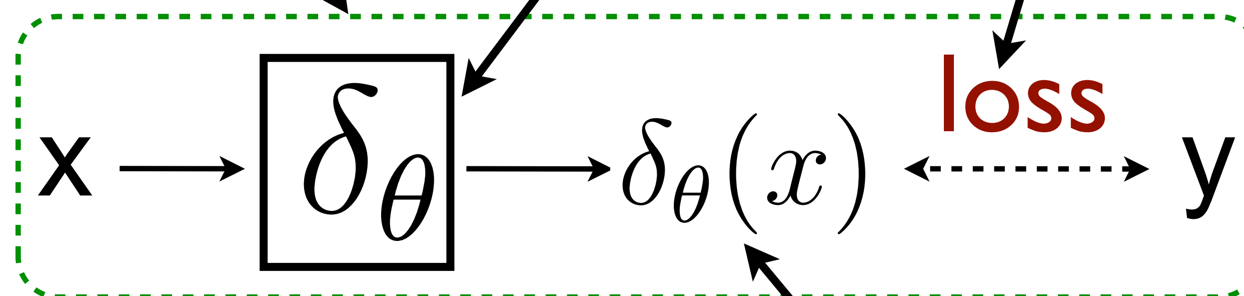
$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$



Supervised: Minimum Empirical Risk

- Minimum Empirical Risk Training

$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$



- Uniform Empirical Distribution

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

Supervised: Minimum Empirical Risk

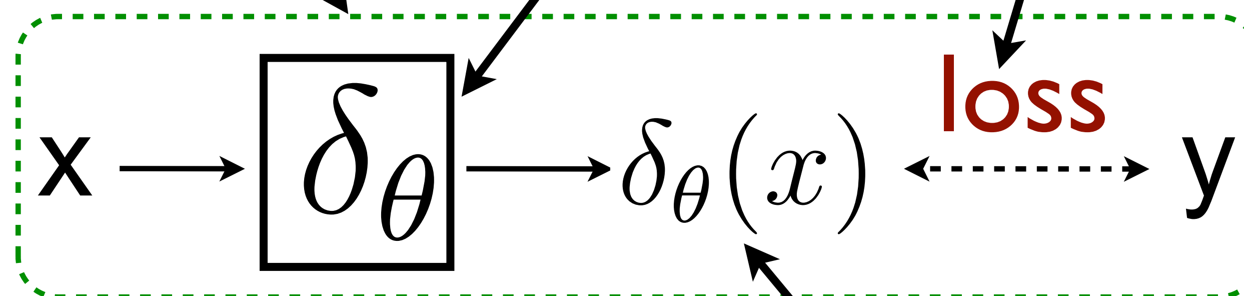
- Minimum Empirical Risk Training

$$\theta^* = \arg\min_{\theta} \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

empirical distribution

MT decoder

negated BLEU



MT output

- Uniform Empirical Distribution

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- MERT
- CRF
- Peceptron

Supervised: Minimum Empirical Risk

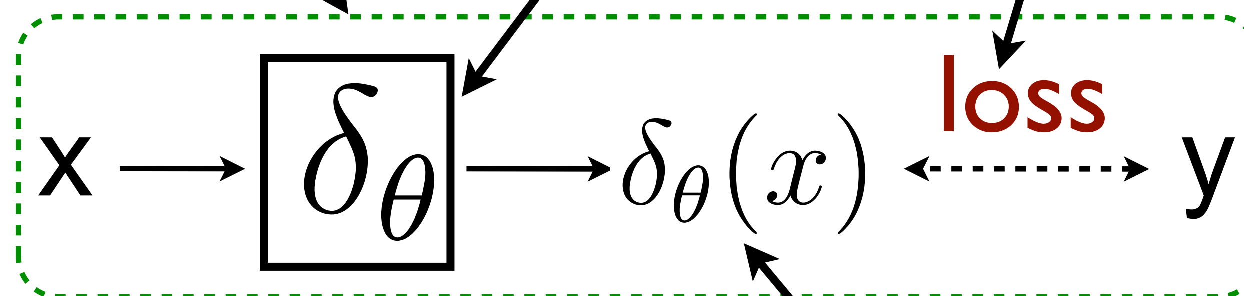
- Minimum Empirical Risk Training

$$\theta^* = \arg\min_\theta \sum_{x,y} \tilde{p}(x,y) \mathrm{L}(\delta_\theta(x), y)$$

MT decoder

negated BLEU

empirical distribution



MT output

- Uniform Empirical Distribution

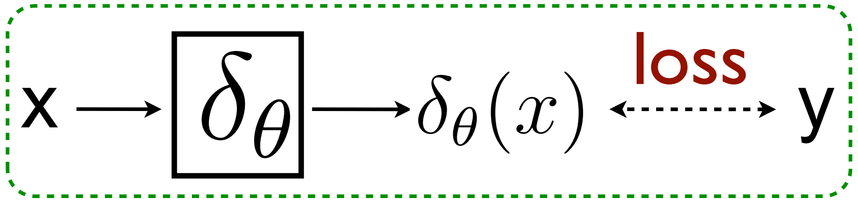
$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- MERT
- CRF
- Peceptron

What if the input x is missing?

Unsupervised: Minimum Imputed Risk

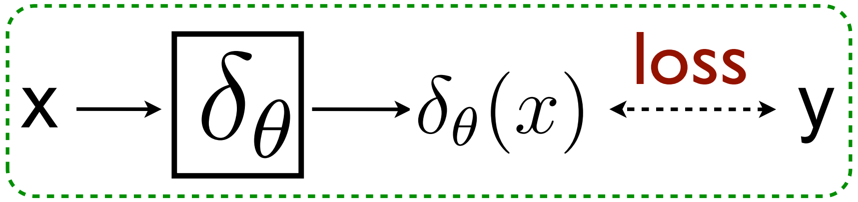
- Minimum Empirical Risk Training



$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

Unsupervised: Minimum Imputed Risk

- Minimum Empirical Risk Training



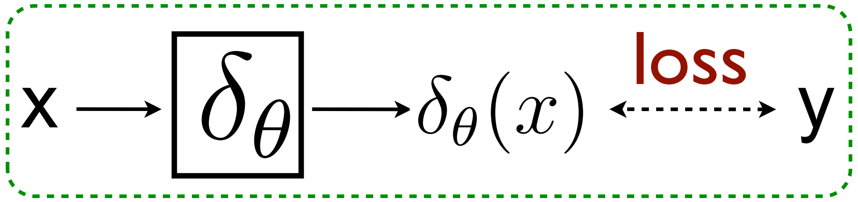
$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- Minimum Imputed Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_{x} p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

Unsupervised: Minimum Imputed Risk

- Minimum Empirical Risk Training



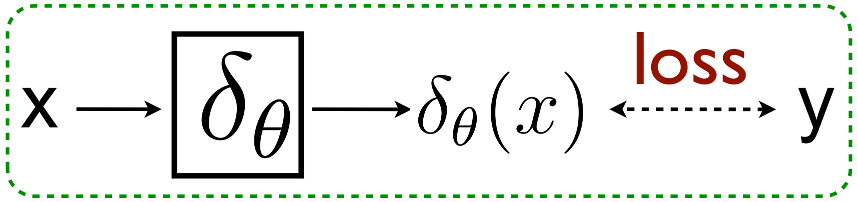
$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- Minimum Imputed Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_{x} p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

Unsupervised: Minimum Imputed Risk

- Minimum Empirical Risk Training



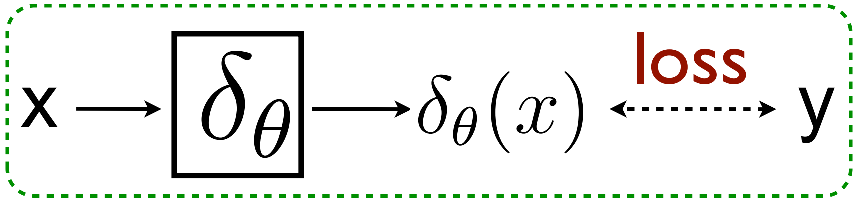
$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- Minimum Imputed Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

Unsupervised: Minimum Imputed Risk

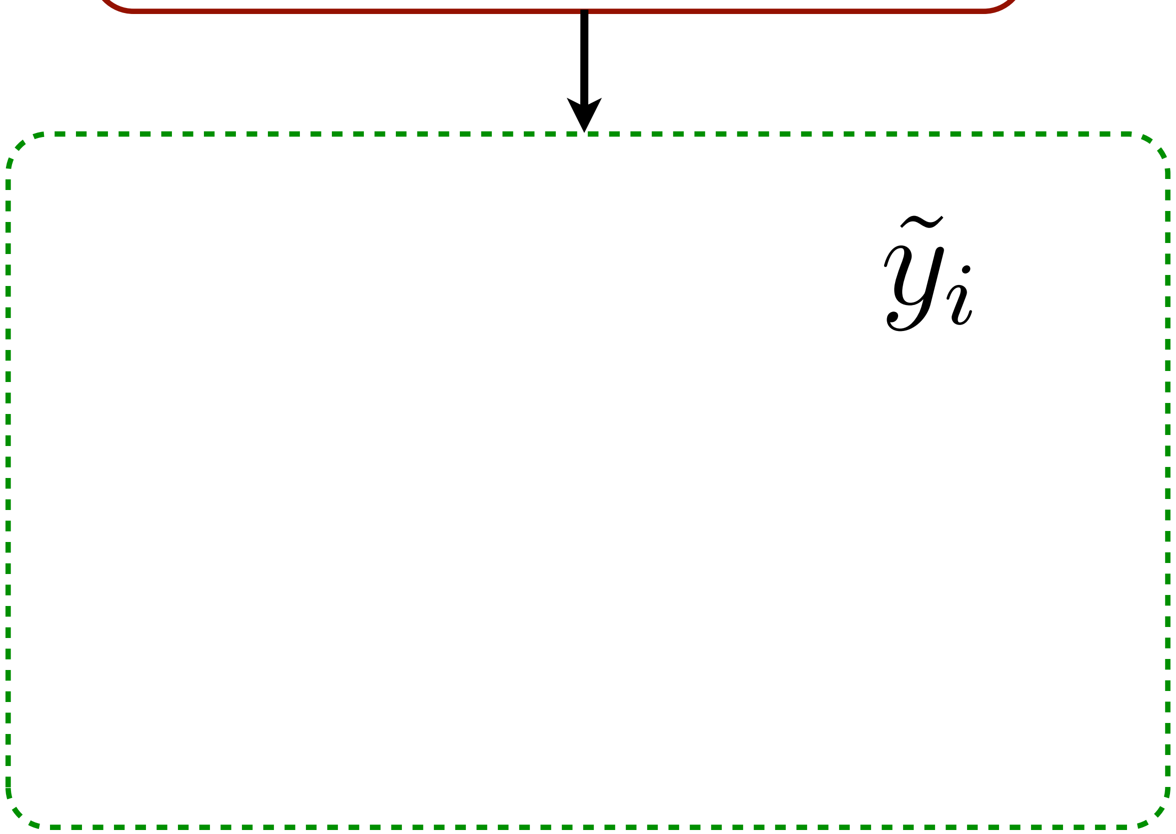
- Minimum Empirical Risk Training



$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

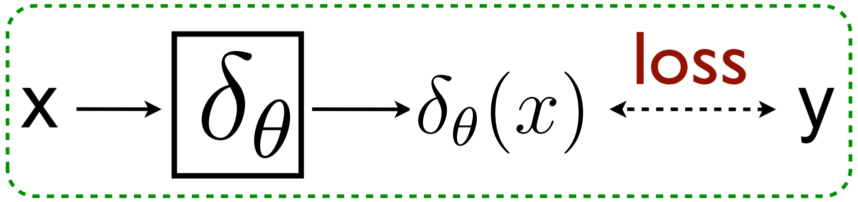
- Minimum Imputed Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$



Unsupervised: Minimum Imputed Risk

- Minimum Empirical Risk Training

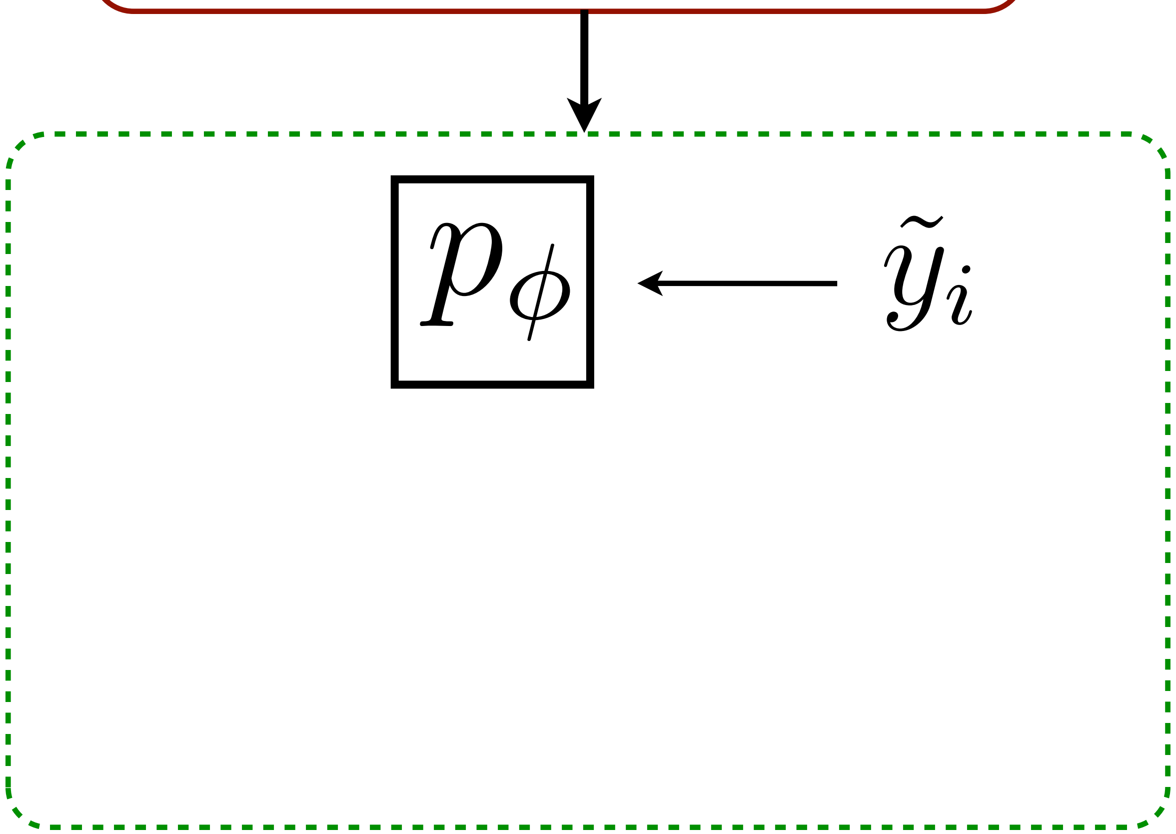


$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- Minimum Imputed Risk Training

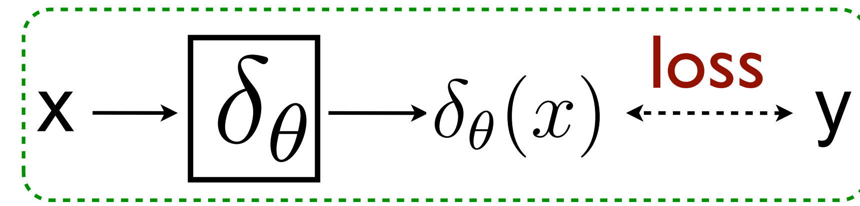
$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

p_ϕ: reverse model



Unsupervised: Minimum Imputed Risk

- Minimum Empirical Risk Training



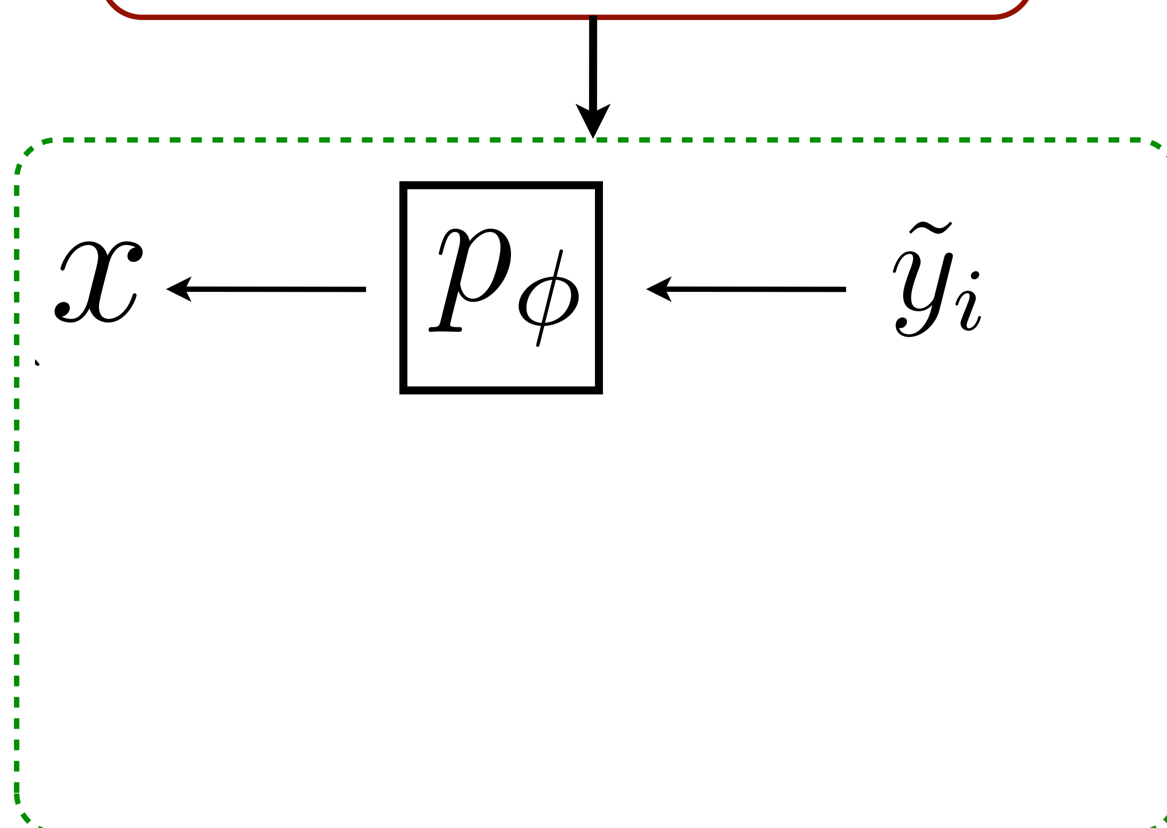
$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- Minimum Imputed Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_{x} p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

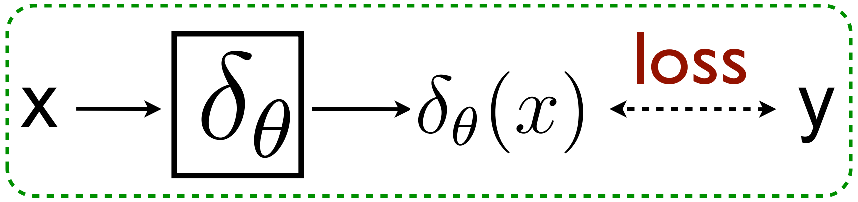
p_ϕ: reverse model

x: imputed input



Unsupervised: Minimum Imputed Risk

- Minimum Empirical Risk Training



$$\theta^* = \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

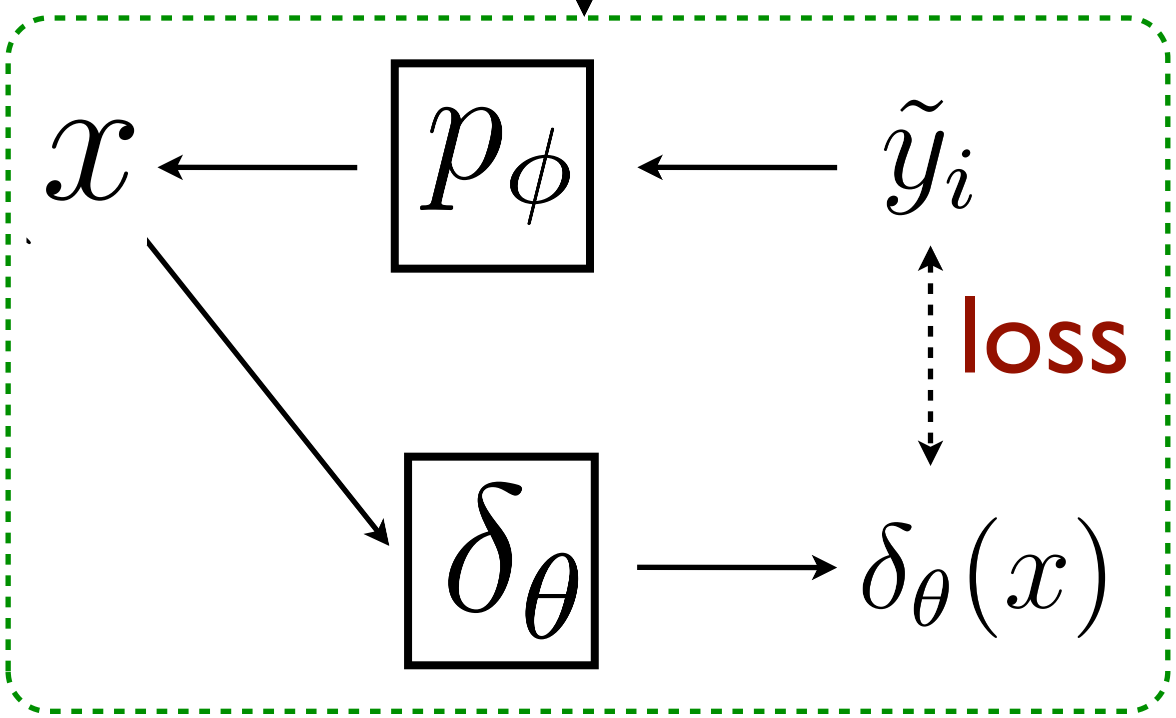
- Minimum Imputed Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i)\mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

p_ϕ: reverse model

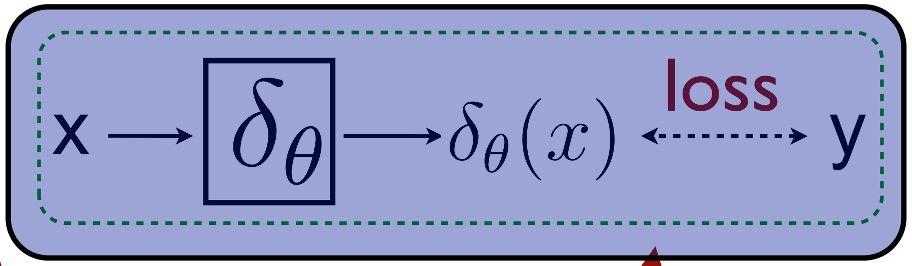
x: imputed input

δ_θ: forward system



Unsupervised: Minimum Imputed Risk

- Minimum Empirical Risk Training



$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

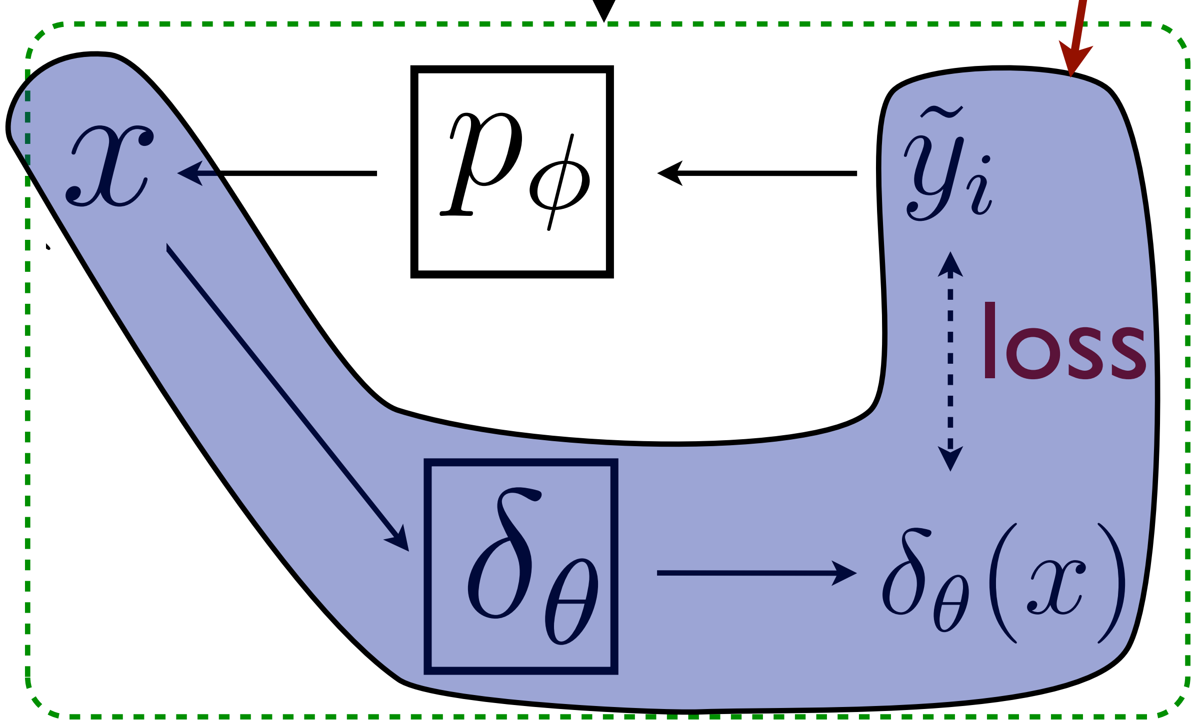
- Minimum Imputed Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

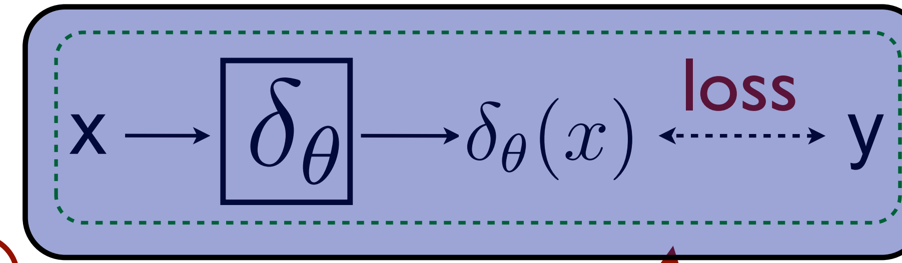
p_ϕ: reverse model

x: imputed input

δ_θ: forward system



Unsupervised: Minimum Imputed Risk



- Minimum Empirical Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- Minimum Imputed Risk Training

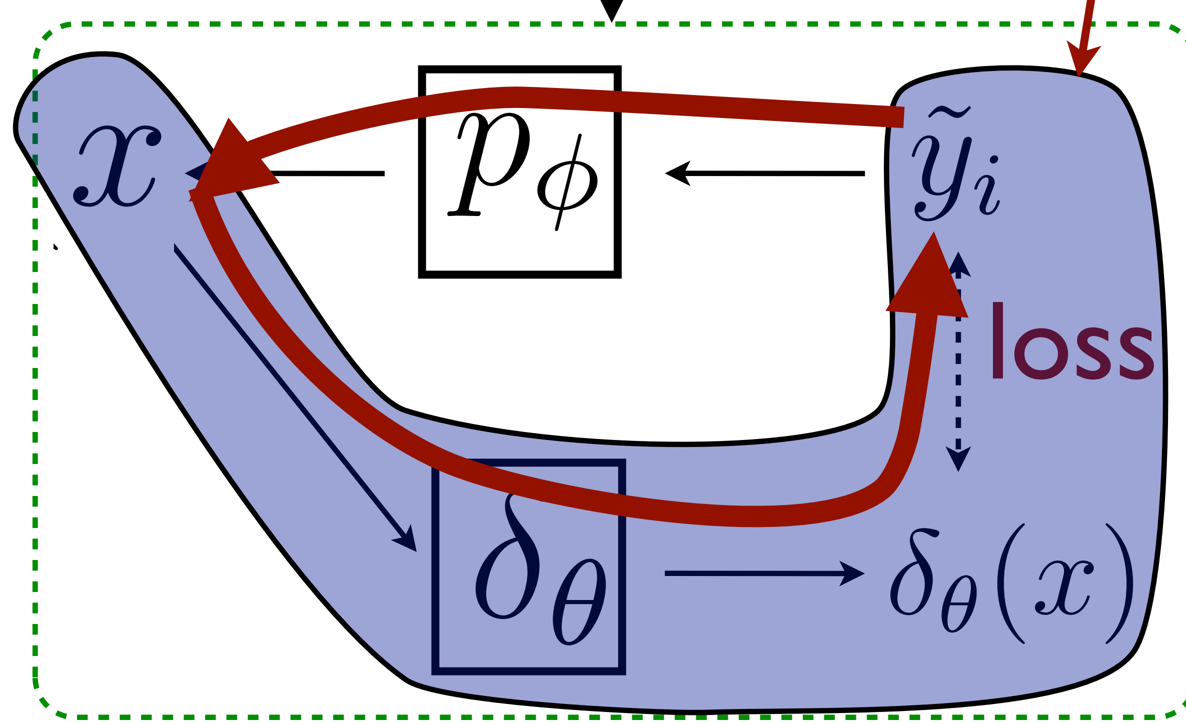
$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_{x} p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

p_ϕ: reverse model

x: imputed input

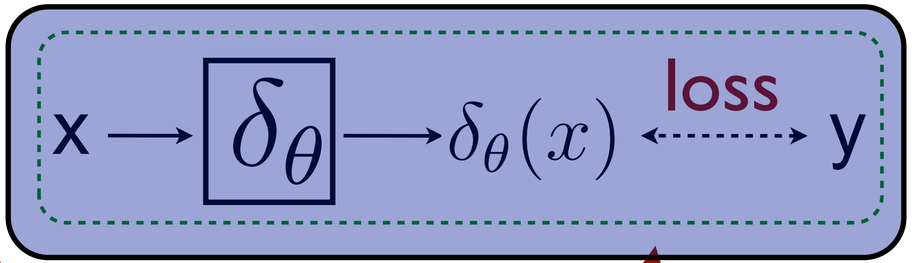
δ_θ: forward system

Round trip translation



Unsupervised: Minimum Imputed Risk

- Minimum Empirical Risk Training



$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathrm{L}(\delta_\theta(x_i), \tilde{y}_i)$$

- Minimum Imputed Risk Training

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_{x} p_\phi(x \mid \tilde{y}_i) \mathrm{L}(\delta_\theta(x), \tilde{y}_i)$$

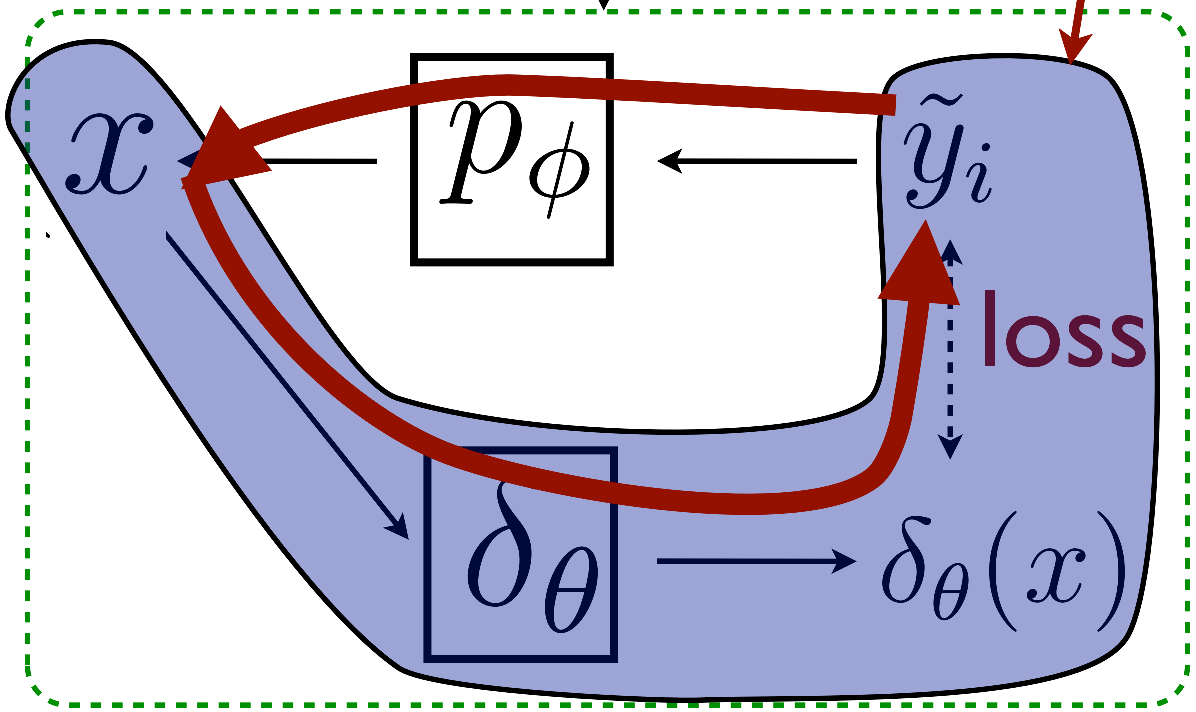
p_ϕ: reverse model

x: imputed input

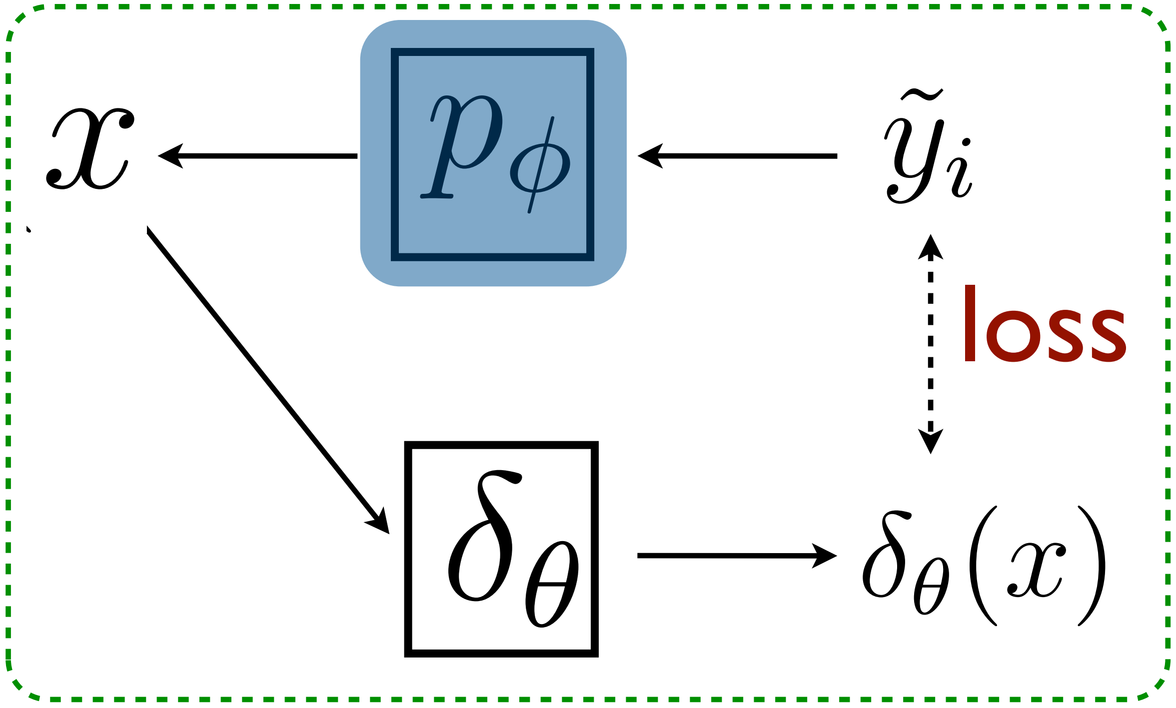
δ_θ: forward system

Round trip translation

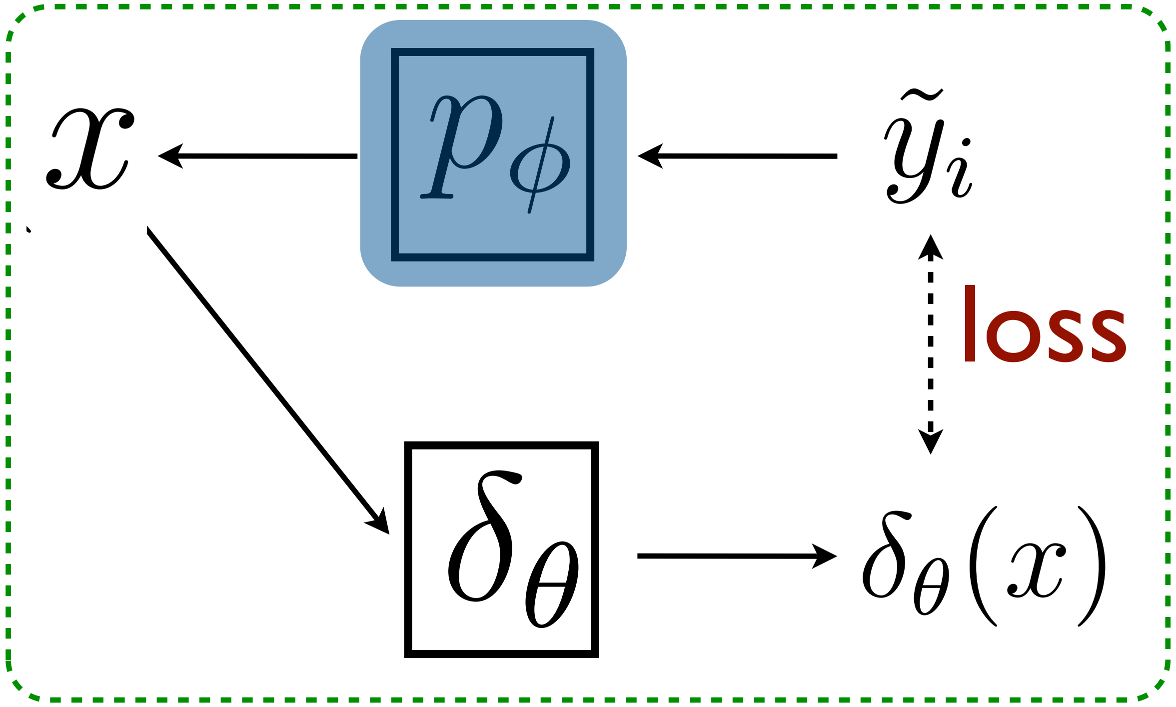
Speech recognition?



Training Reverse Model p_ϕ

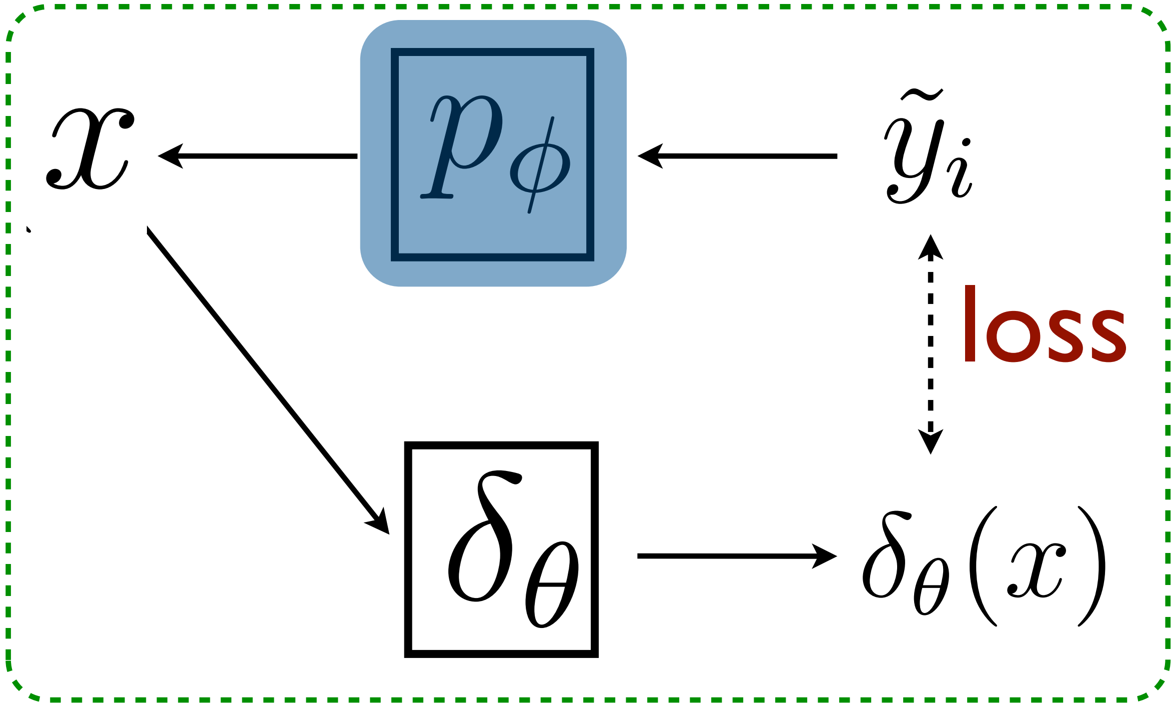


Training Reverse Model p_ϕ



Our goal is to train a good forward system δ_θ

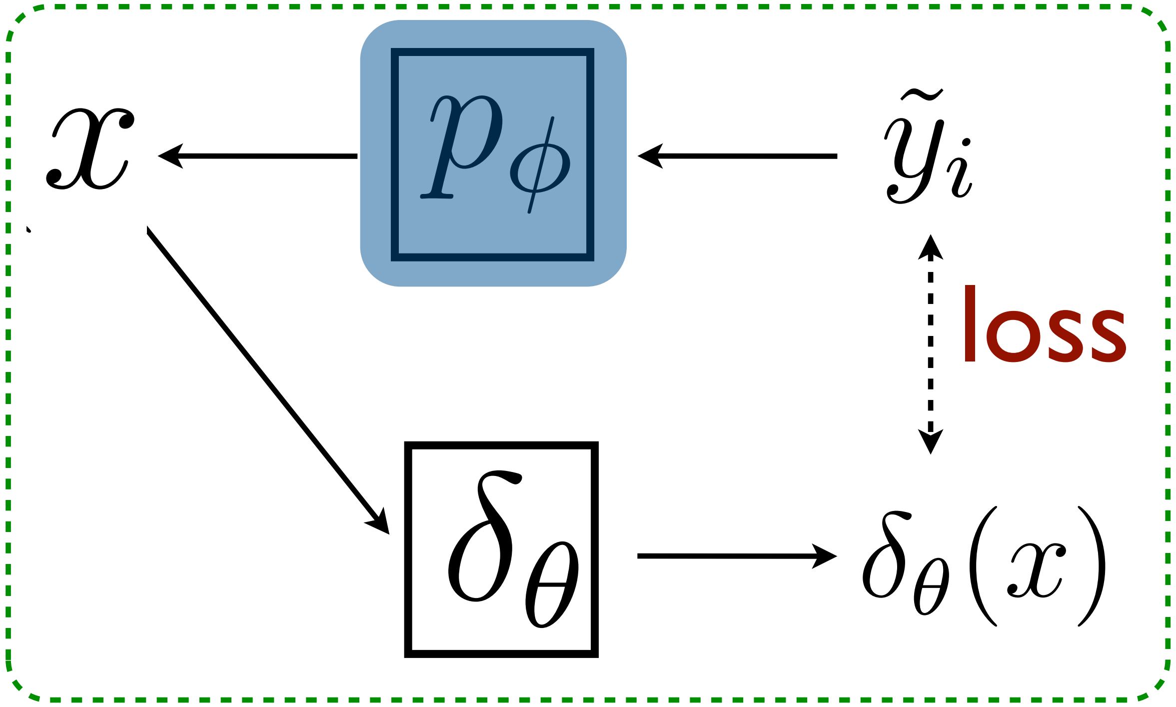
Training Reverse Model p_ϕ



Our goal is to train a good forward system δ_θ

p_ϕ and δ_θ are parameterized and trained separately

Training Reverse Model p_ϕ

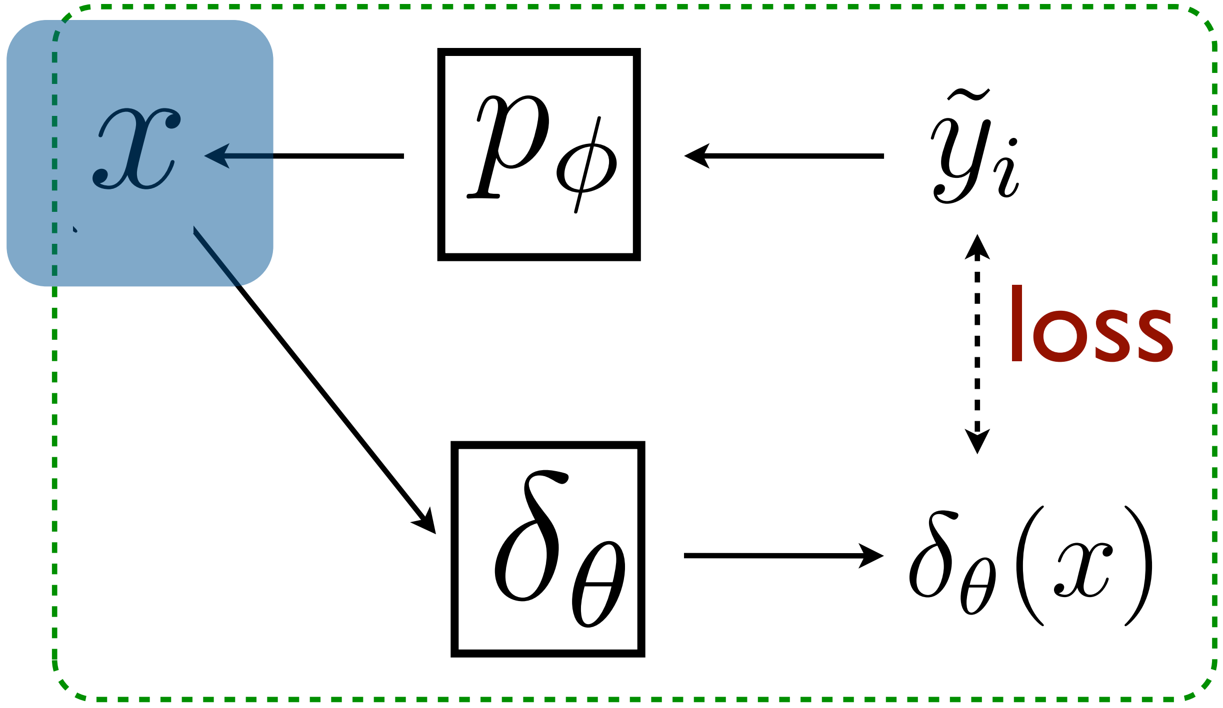


Our goal is to train a good forward system δ_θ

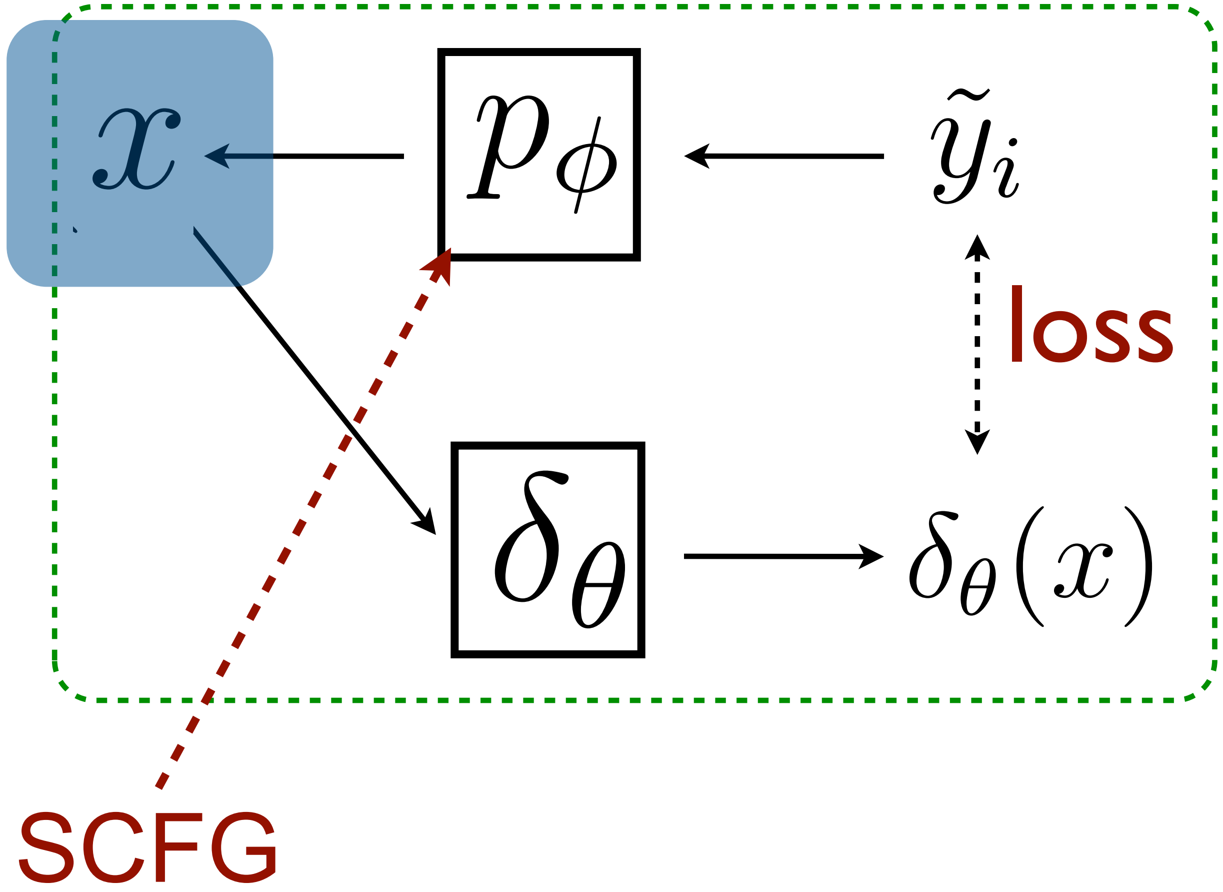
p_ϕ and δ_θ are parameterized and trained separately

p_ϕ is fixed when training δ_θ

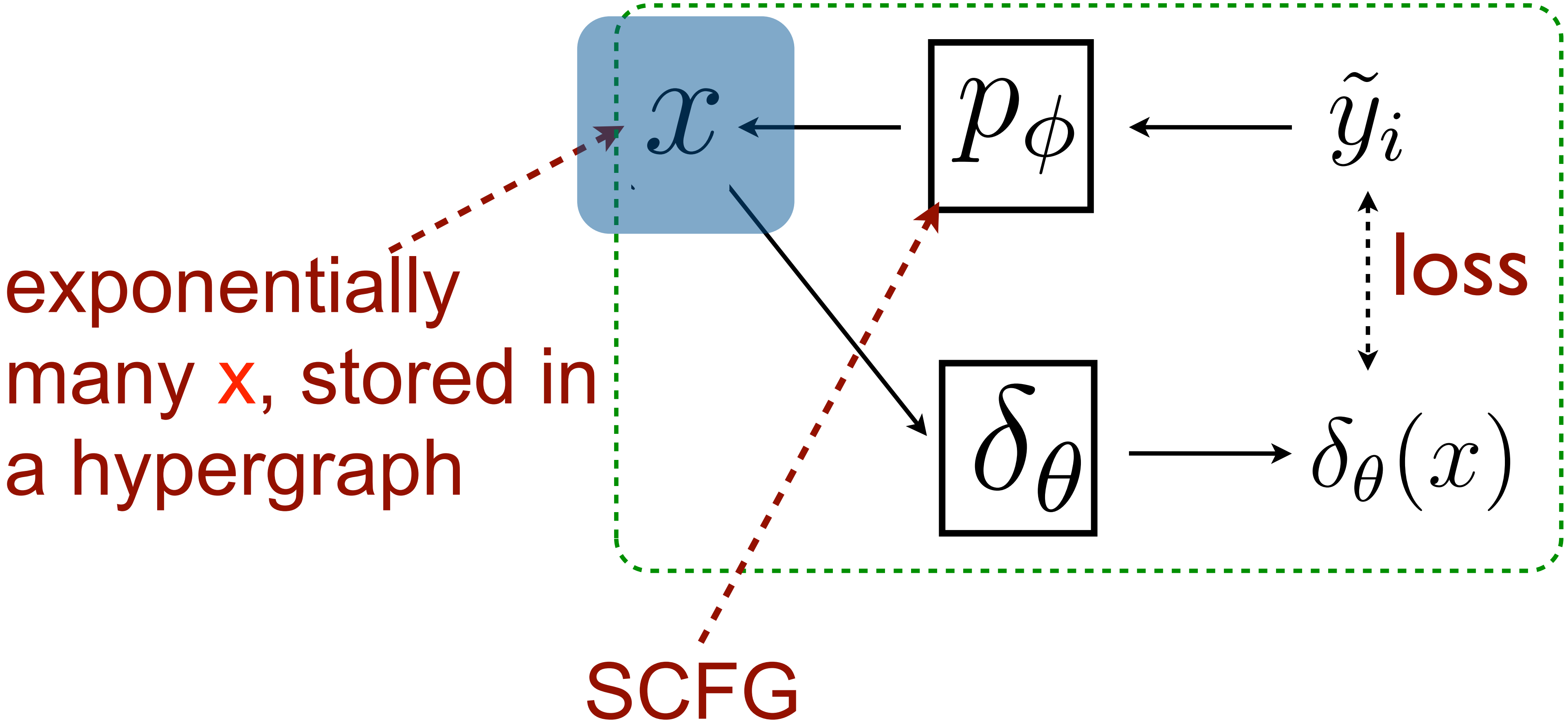
Approximating $p_\phi(x \mid \tilde{y}_i)$



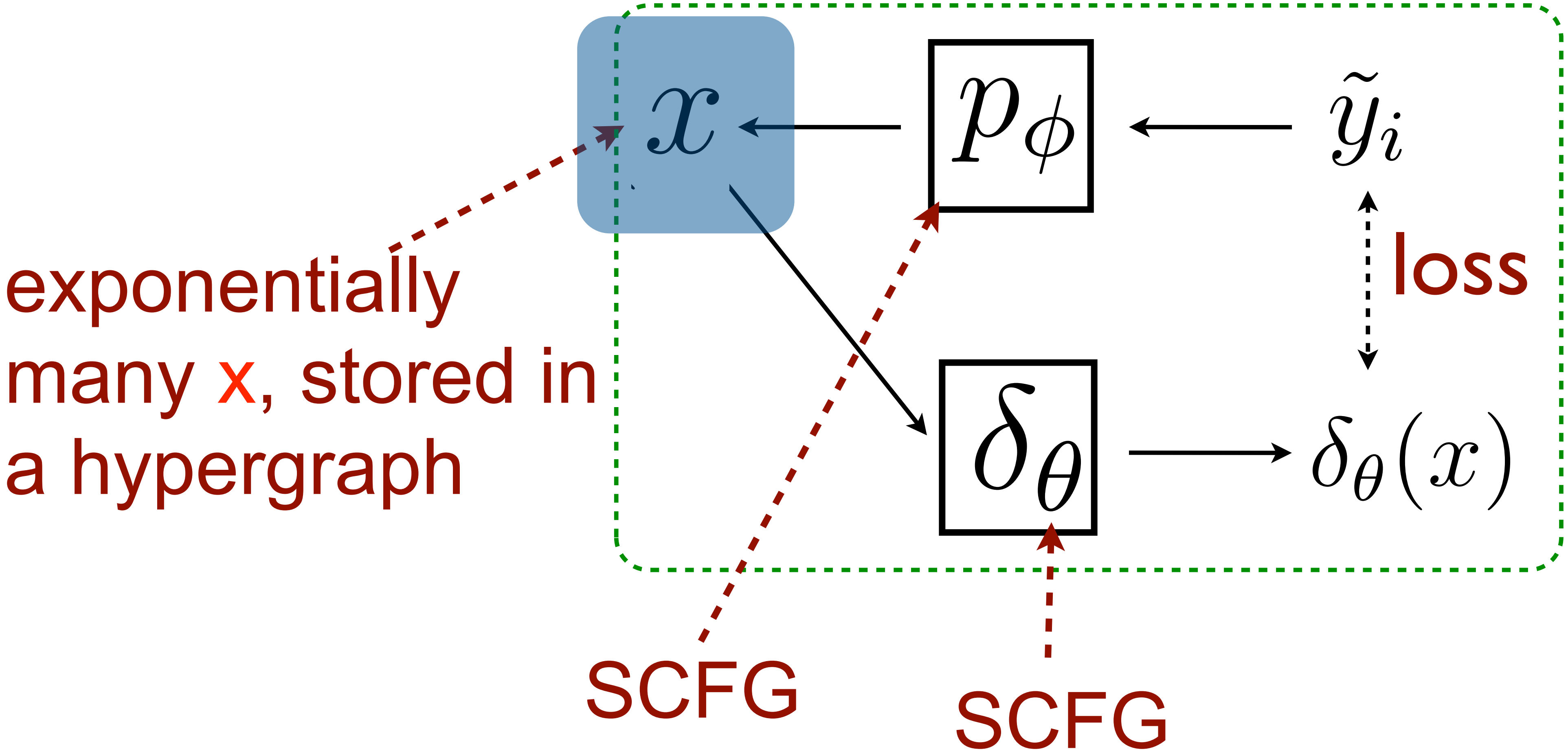
Approximating $p_\phi(x \mid \tilde{y}_i)$



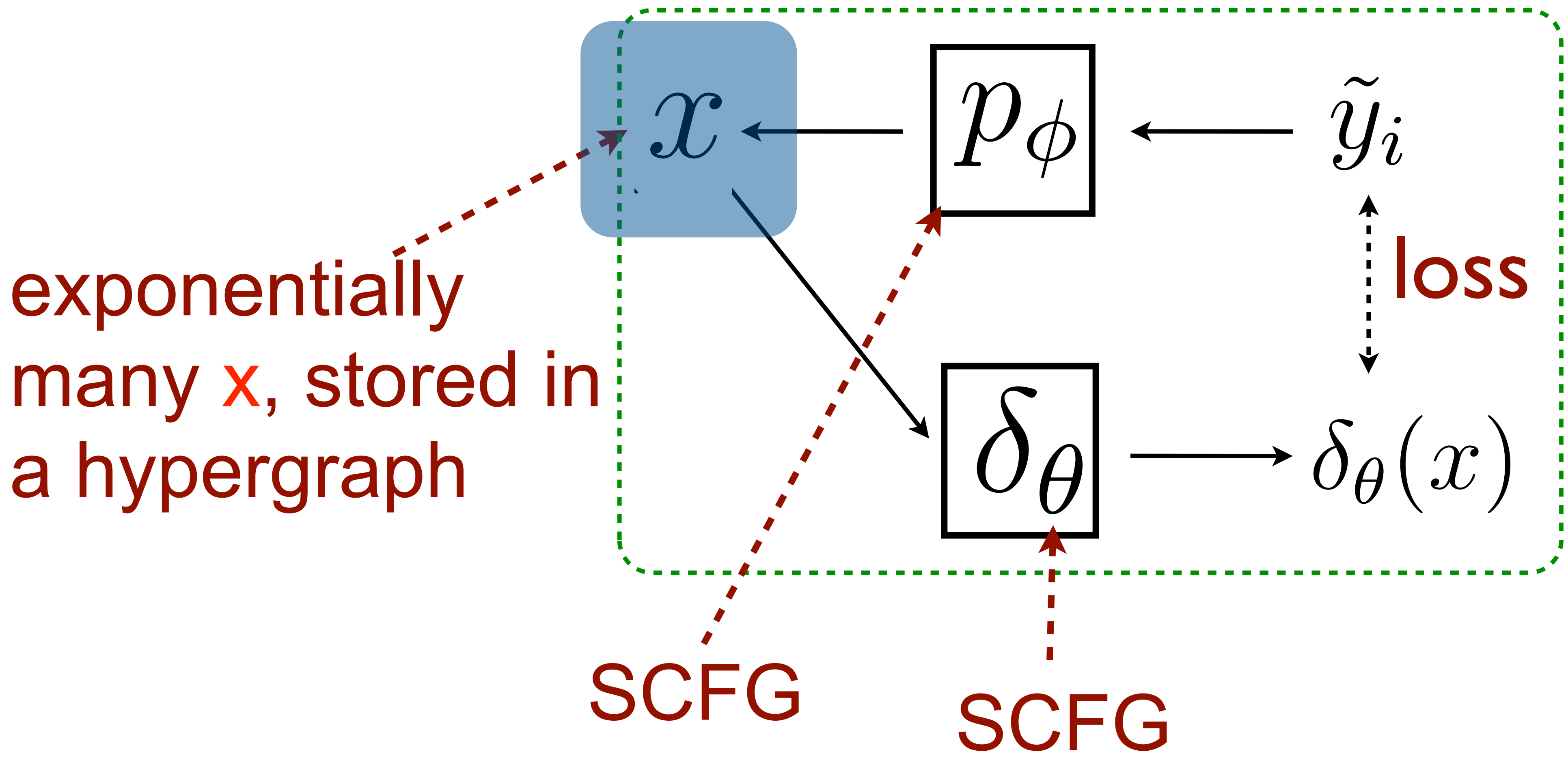
Approximating $p_\phi(x \mid \tilde{y}_i)$



Approximating $p_\phi(x \mid \tilde{y}_i)$

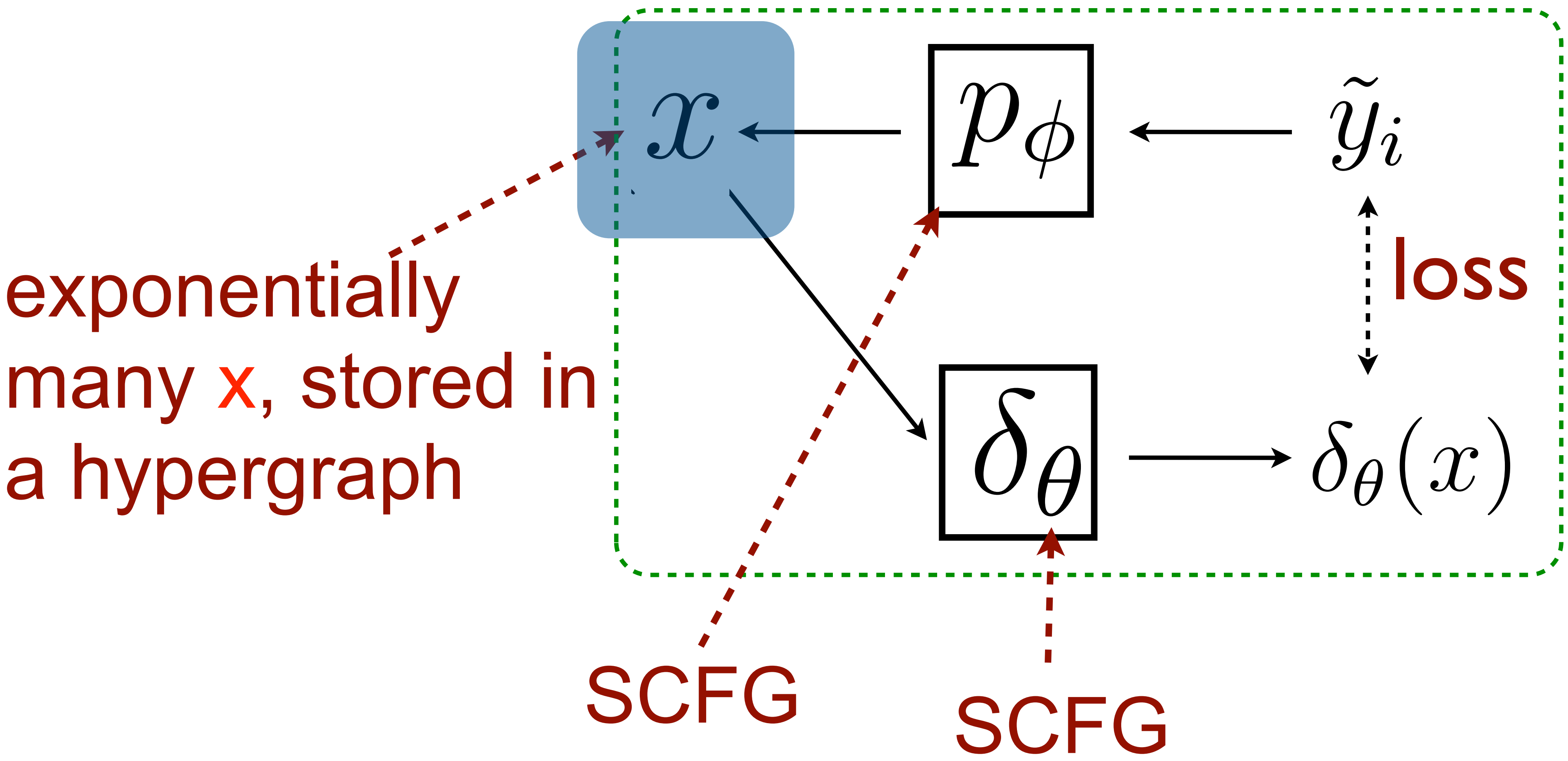


Approximating $p_\phi(x \mid \tilde{y}_i)$



CFG is not closed under composition!

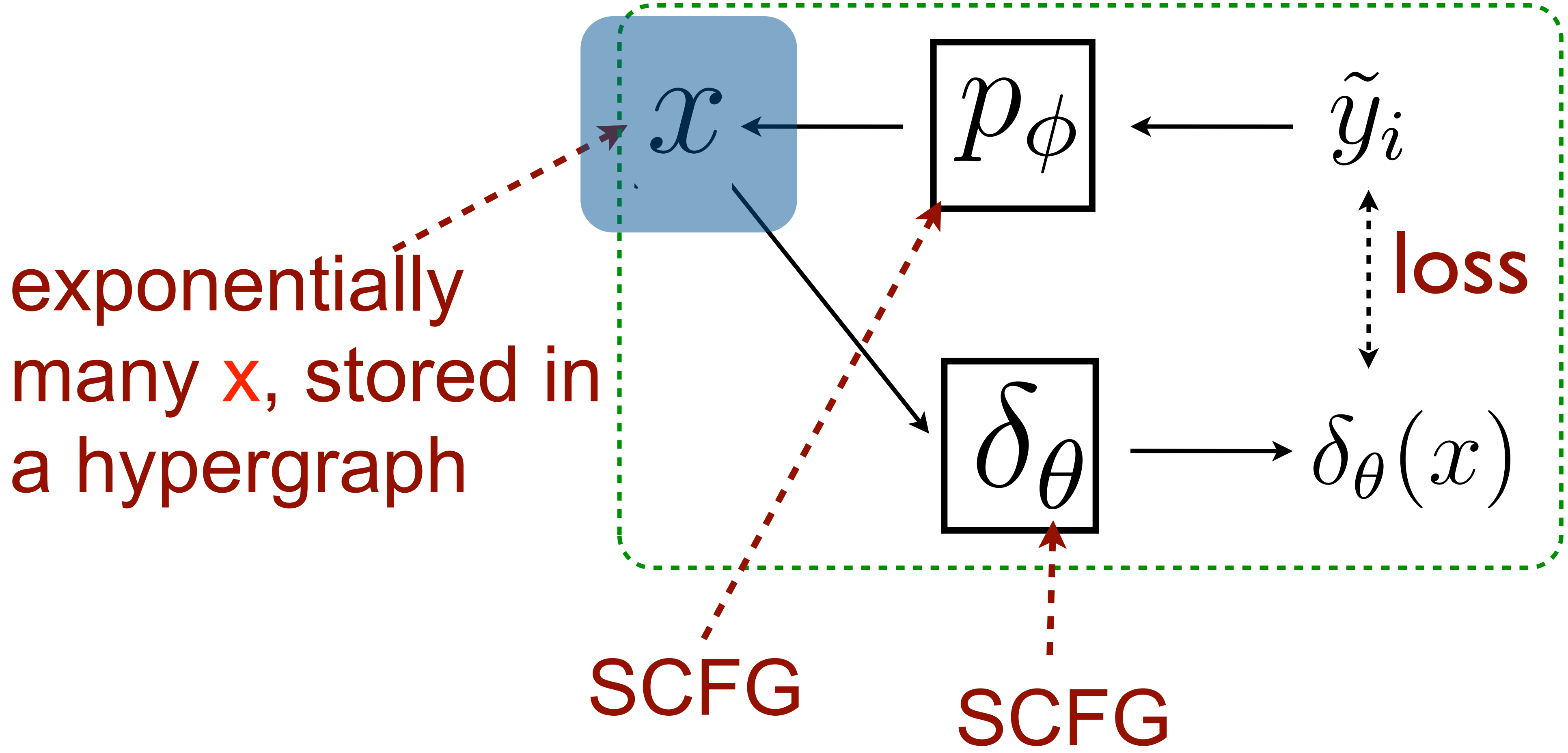
Approximating $p_\phi(x \mid \tilde{y}_i)$



CFG is not closed under composition!

- Approximations
  - k-best
  - sampling
  - lattice

Approximating $p_\phi(x \mid \tilde{y}_i)$

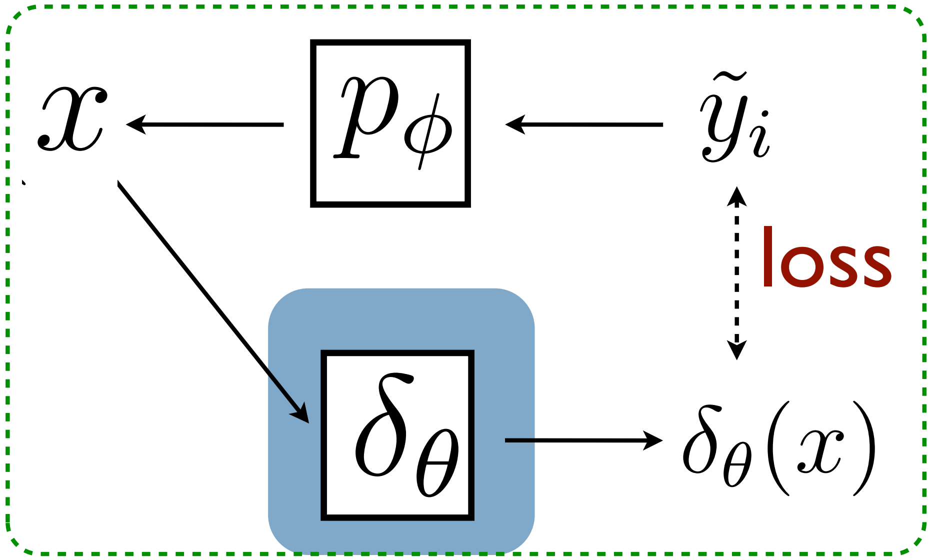


CFG is not closed under composition!

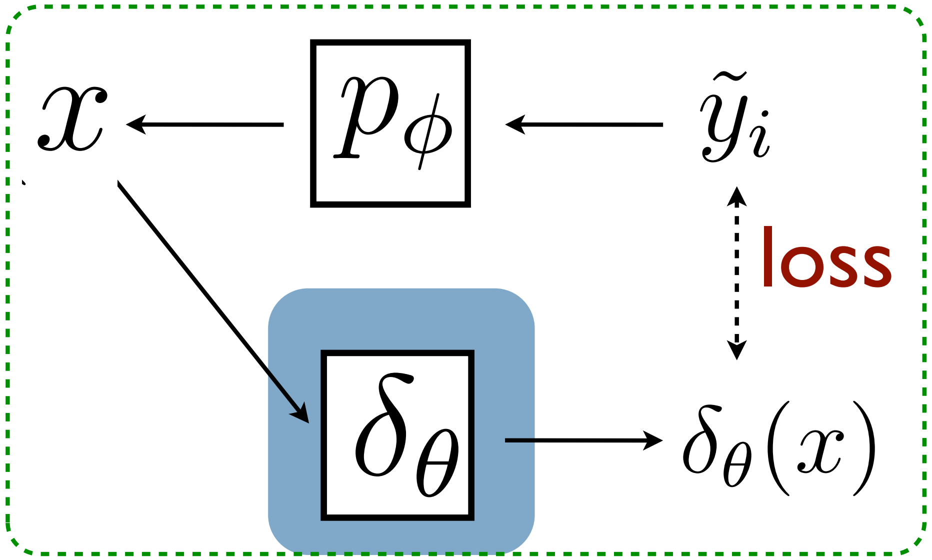
- Approximations
  - k-best
  - sampling
  - lattice

variational approximation
+
lattice decoding (Dyer et al., 2008)

The Forward System $\delta_\theta(x)$



The Forward System $\delta_\theta(x)$

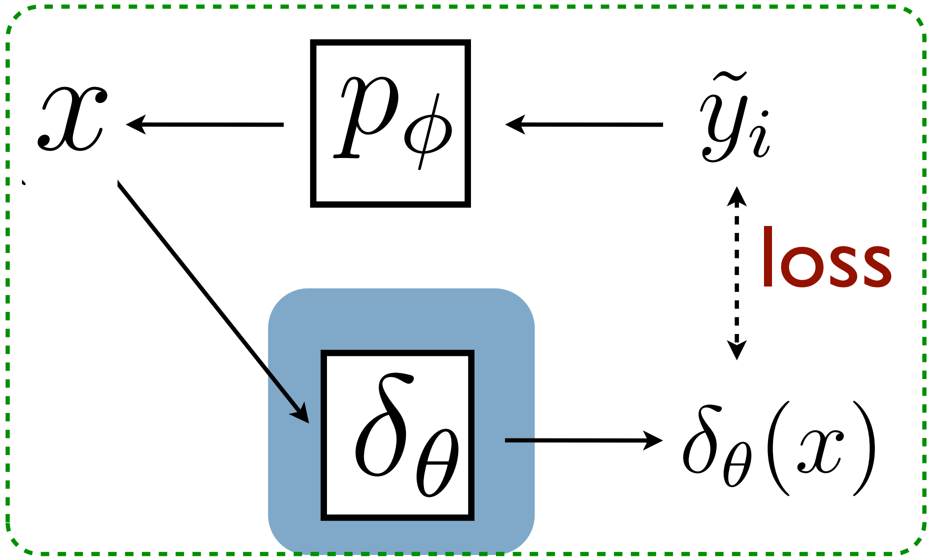


$$\delta_\theta(x) = \operatorname*{argmax}_y p_\theta(y \mid x)$$

- Deterministic Decoding
  - use **one-best** translation

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathbf{L}(\delta_\theta(x), \tilde{y}_i)$$

The Forward System $\delta_\theta(x)$



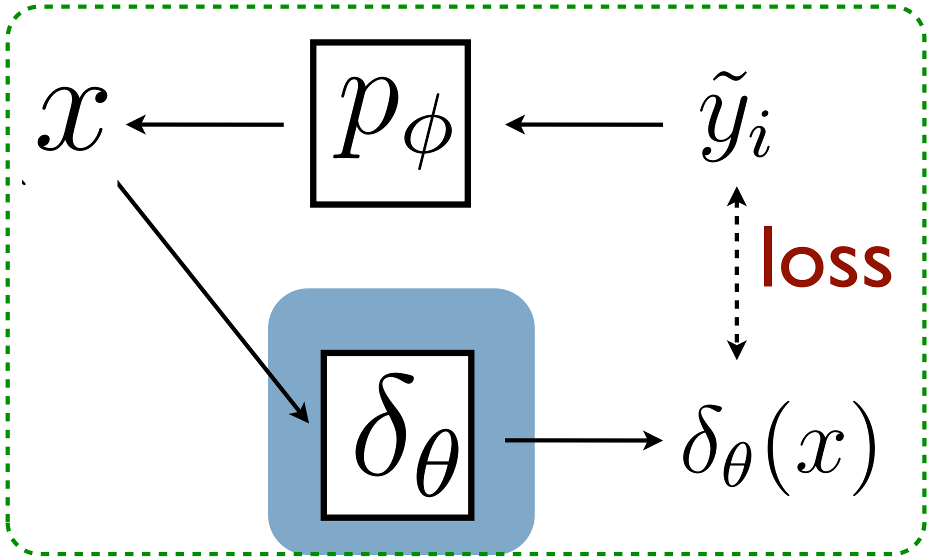
$$\delta_\theta(x) = \operatorname*{argmax}_y p_\theta(y \mid x)$$

- Deterministic Decoding
  - use **one-best** translation

the objective is not differentiable ☹

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathbf{L}(\delta_\theta(x), \tilde{y}_i)$$

The Forward System $\delta_\theta(x)$



$$\delta_\theta(x) = \underset{y}{\operatorname{argmax}}\, p_\theta(y \mid x)$$

- Deterministic Decoding
  - use **one-best** translation

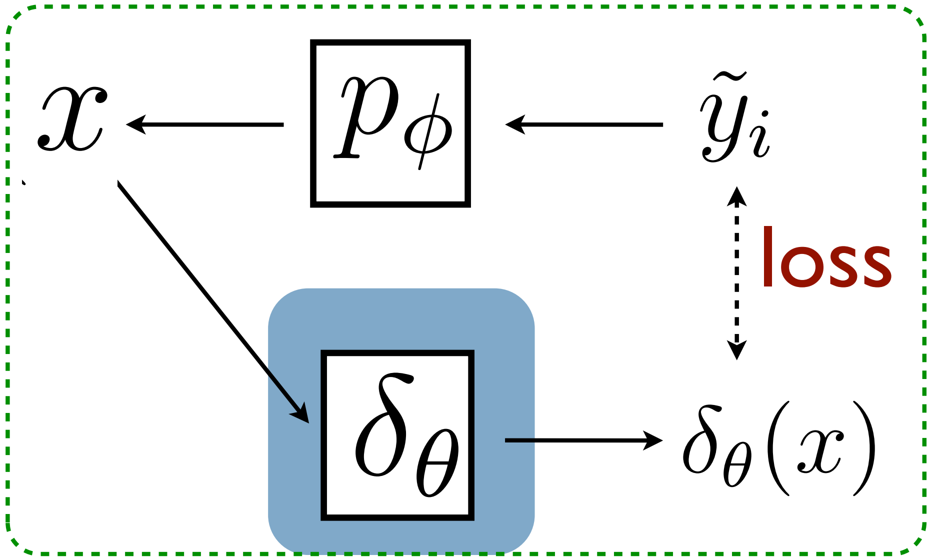
the objective is not differentiable ☹

$$\theta^* = \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N}\sum_{x} p_\phi(x \mid \tilde{y}_i)\mathbf{L}(\delta_\theta(x), \tilde{y}_i)$$

- Randomized Decoding
  - use a **distribution** of translations

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}\sum_{x} p_\phi(x \mid \tilde{y}_i)\sum_{y} p_\theta(y \mid x)\mathbf{L}(y, \tilde{y}_i)$$

The Forward System $\delta_\theta(x)$



$$\delta_\theta(x) = \underset{y}{\operatorname{argmax}}\, p_\theta(y \mid x)$$

the objective is not differentiable ☹

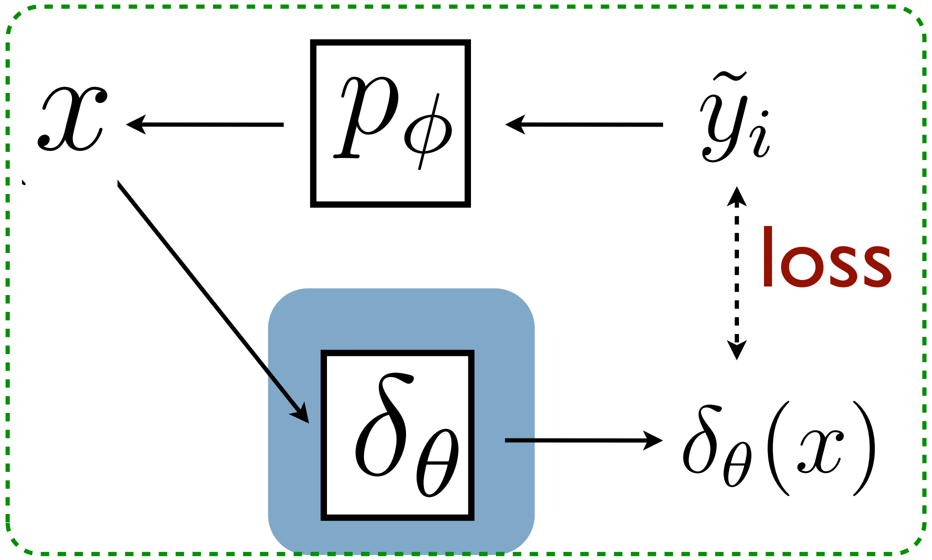
- Deterministic Decoding
  - use **one-best** translation

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathbf{L}(\delta_\theta(x), \tilde{y}_i)$$

- Randomized Decoding
  - use a **distribution** of translations

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \sum_y p_\theta(y \mid x) \mathbf{L}(y, \tilde{y}_i)$$

The Forward System $\delta_\theta(x)$



$$\delta_\theta(x) = \operatorname*{argmax}_y p_\theta(y \mid x)$$

- Deterministic Decoding
  - use **one-best** translation

the objective is not differentiable ☹

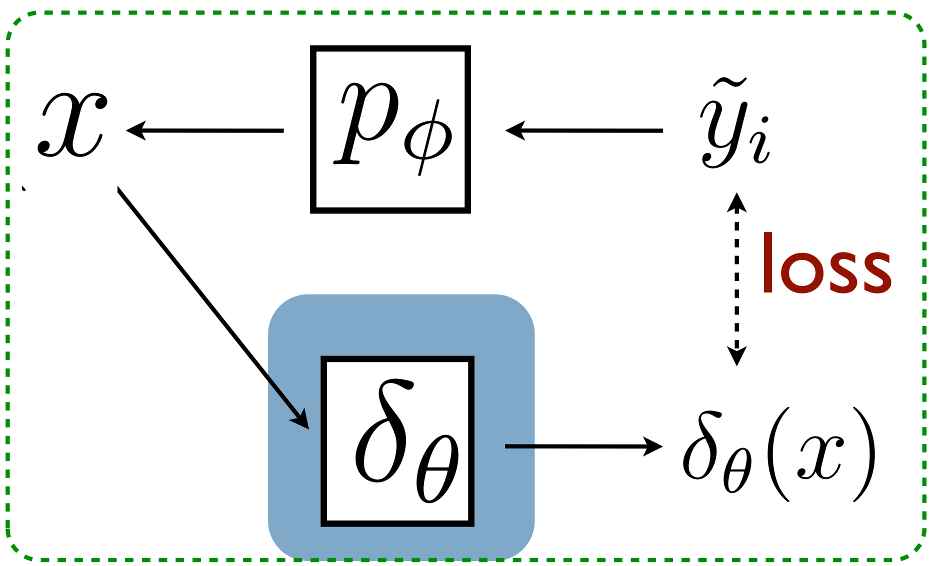
$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathbf{L}(\delta_\theta(x), \tilde{y}_i)$$

- Randomized Decoding
  - use a **distribution** of translations

expected loss

$$\theta^* = \operatorname*{argmin}_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \sum_y p_\theta(y \mid x) \mathbf{L}(y, \tilde{y}_i)$$

The Forward System $\delta_\theta(x)$



$$\delta_\theta(x) = \operatorname*{argmax}_y p_\theta(y \mid x)$$

- Deterministic Decoding
  - use **one-best** translation

the objective is not differentiable

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \mathbf{L}(\delta_\theta(x), \tilde{y}_i)$$

- Randomized Decoding
  - use a **distribution** of translations

differentiable

expected loss

$$\theta^* = \operatorname*{argmin}_\theta \frac{1}{N} \sum_{i=1}^{N} \sum_x p_\phi(x \mid \tilde{y}_i) \sum_y p_\theta(y \mid x) \mathbf{L}(y, \tilde{y}_i)$$

Experiments

- Supervised Training
  - require bitext
- Unsupervised Training
  - require monolingual English
- Semi-supervised Training
  - interpolation of supervised and unsupervised

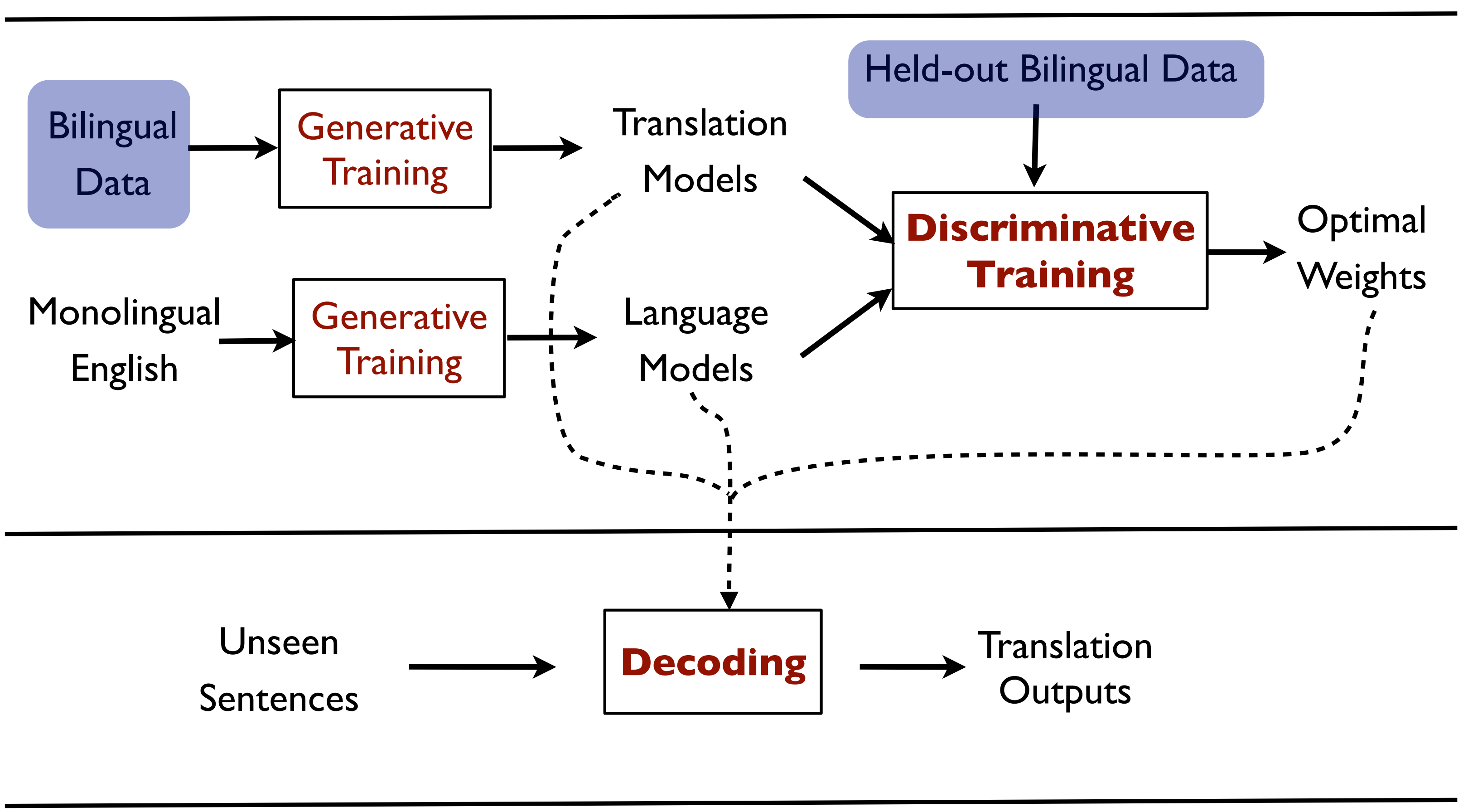
Semi-supervised Training

Semi-supervised Training

Training scenario	Test BLEU
Sup, (200, 200*16)	47.6

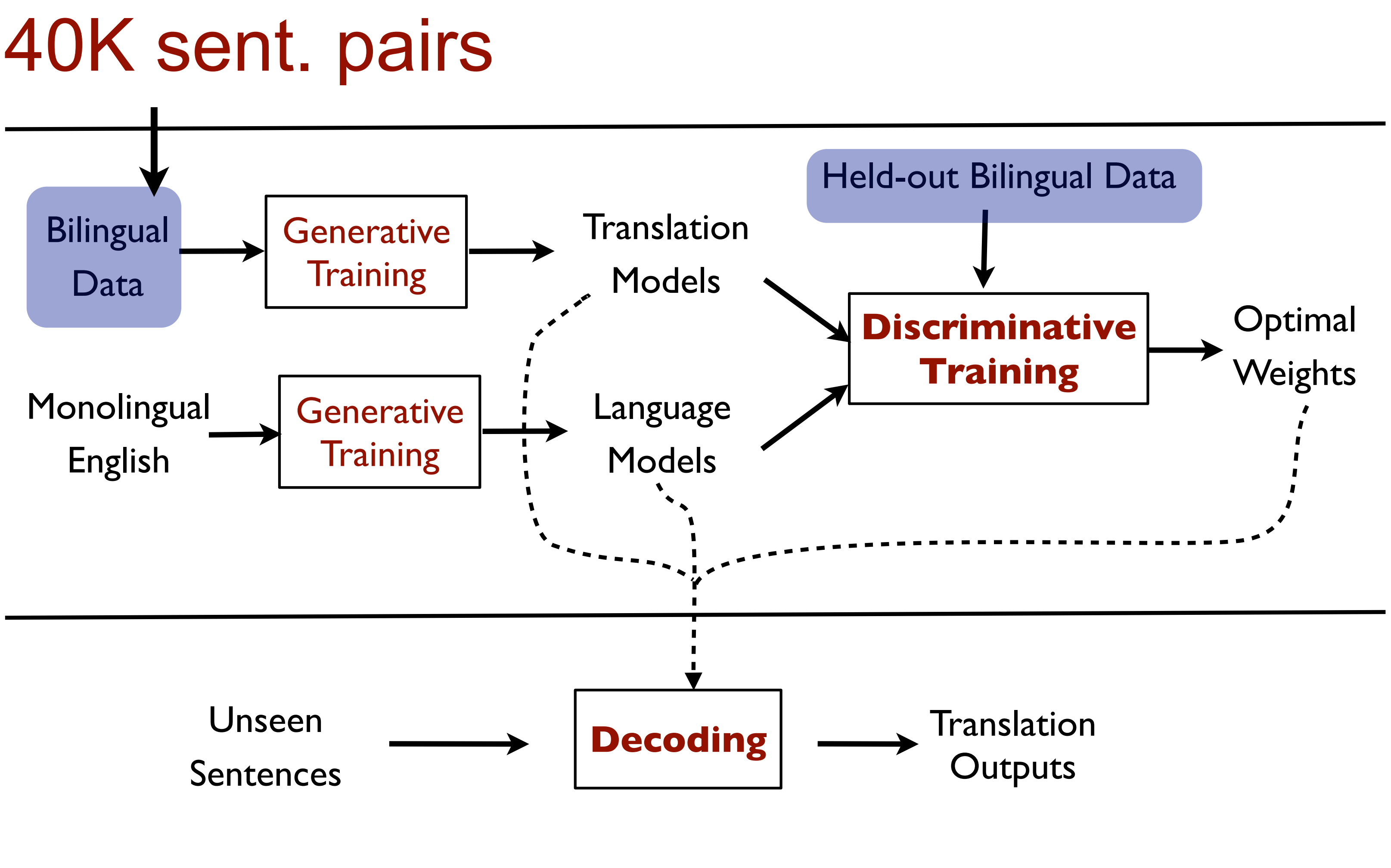
Semi-supervised Training

Training scenario	Test BLEU
Sup, (200, 200*16)	47.6



Semi-supervised Training

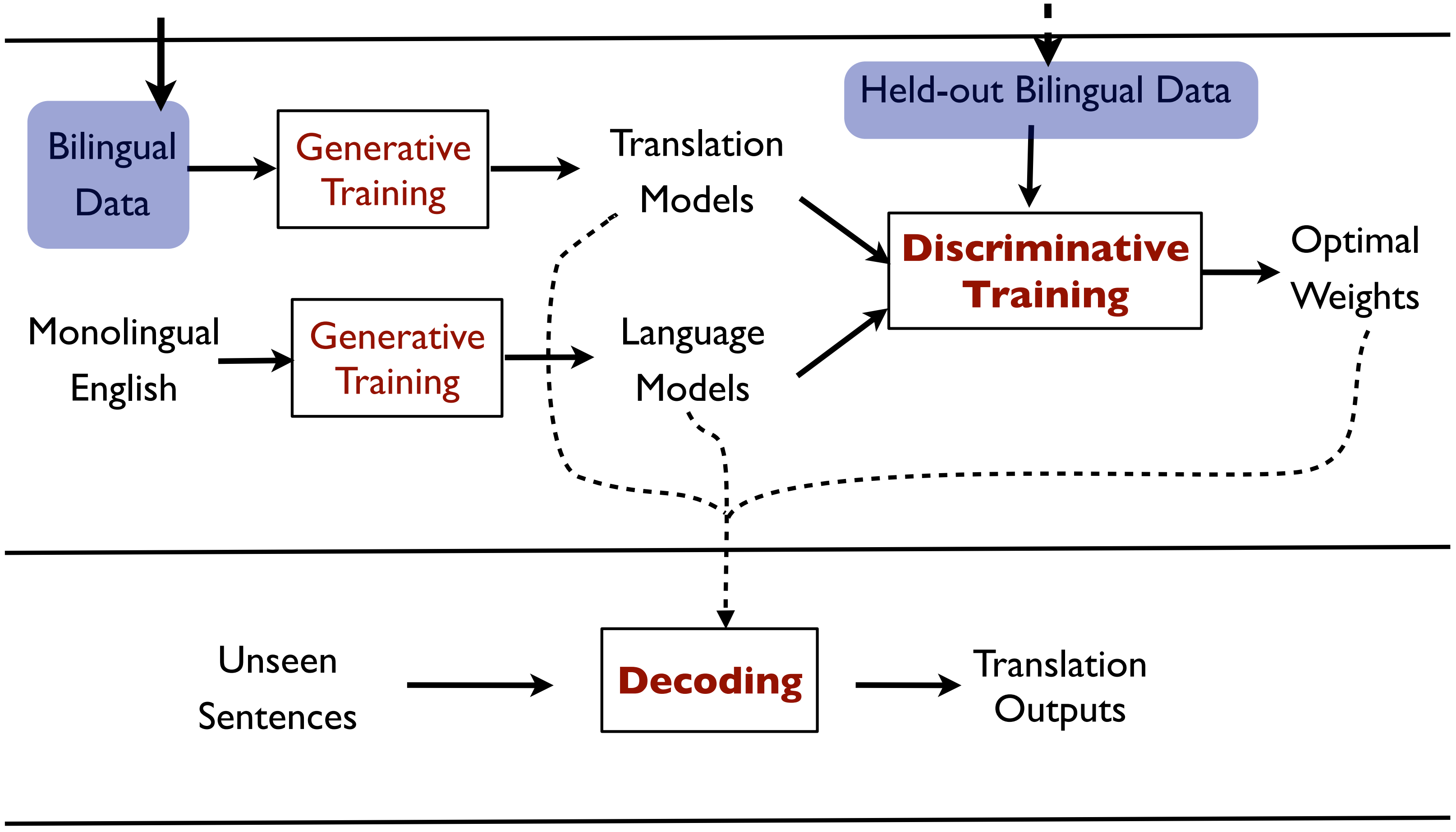
Training scenario	Test BLEU
Sup, (200, 200*16)	47.6



Semi-supervised Training

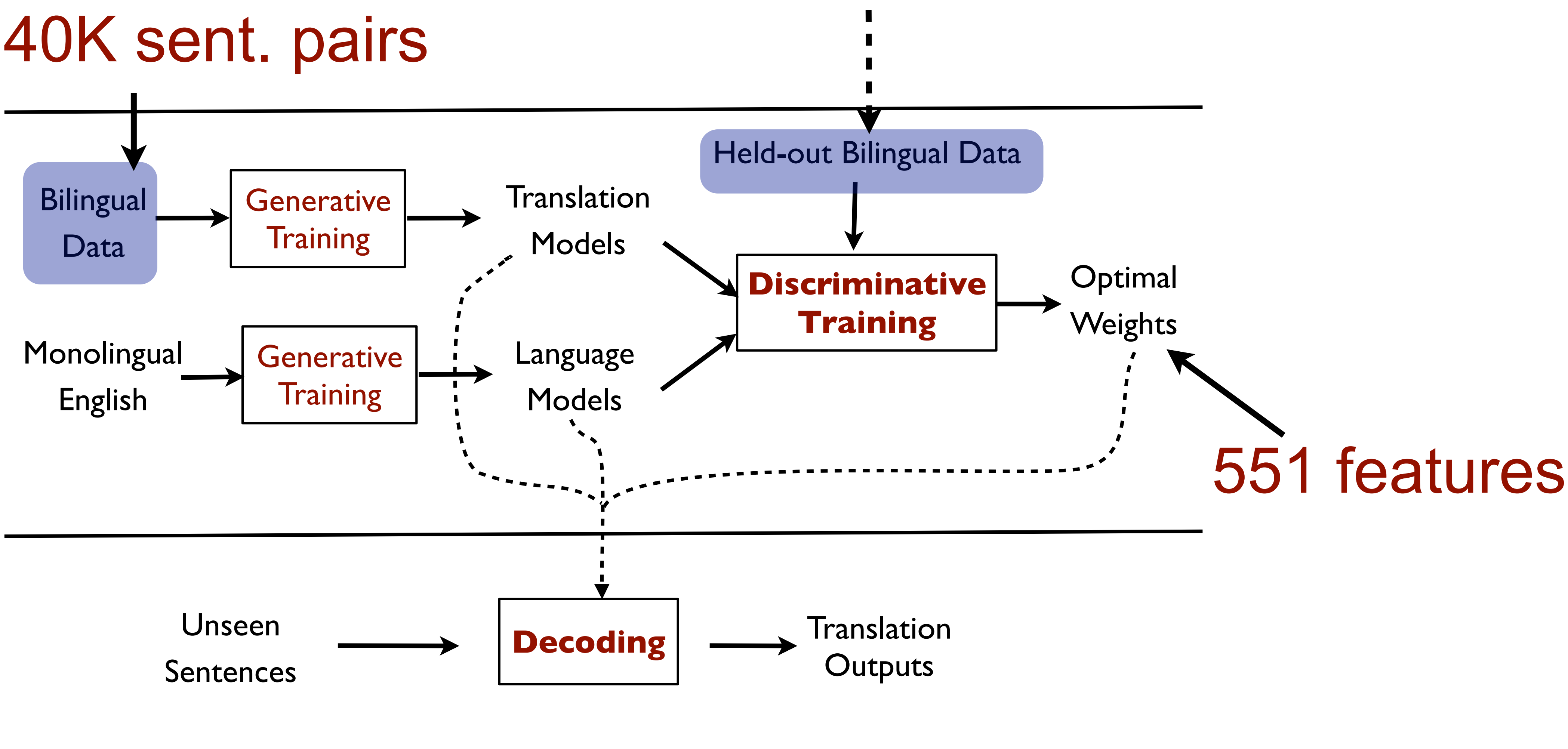
Training scenario	Test BLEU
Sup, (200, 200*16)	47.6

40K sent. pairs



Semi-supervised Training

Training scenario	Test BLEU
Sup, (200, 200*16)	47.6

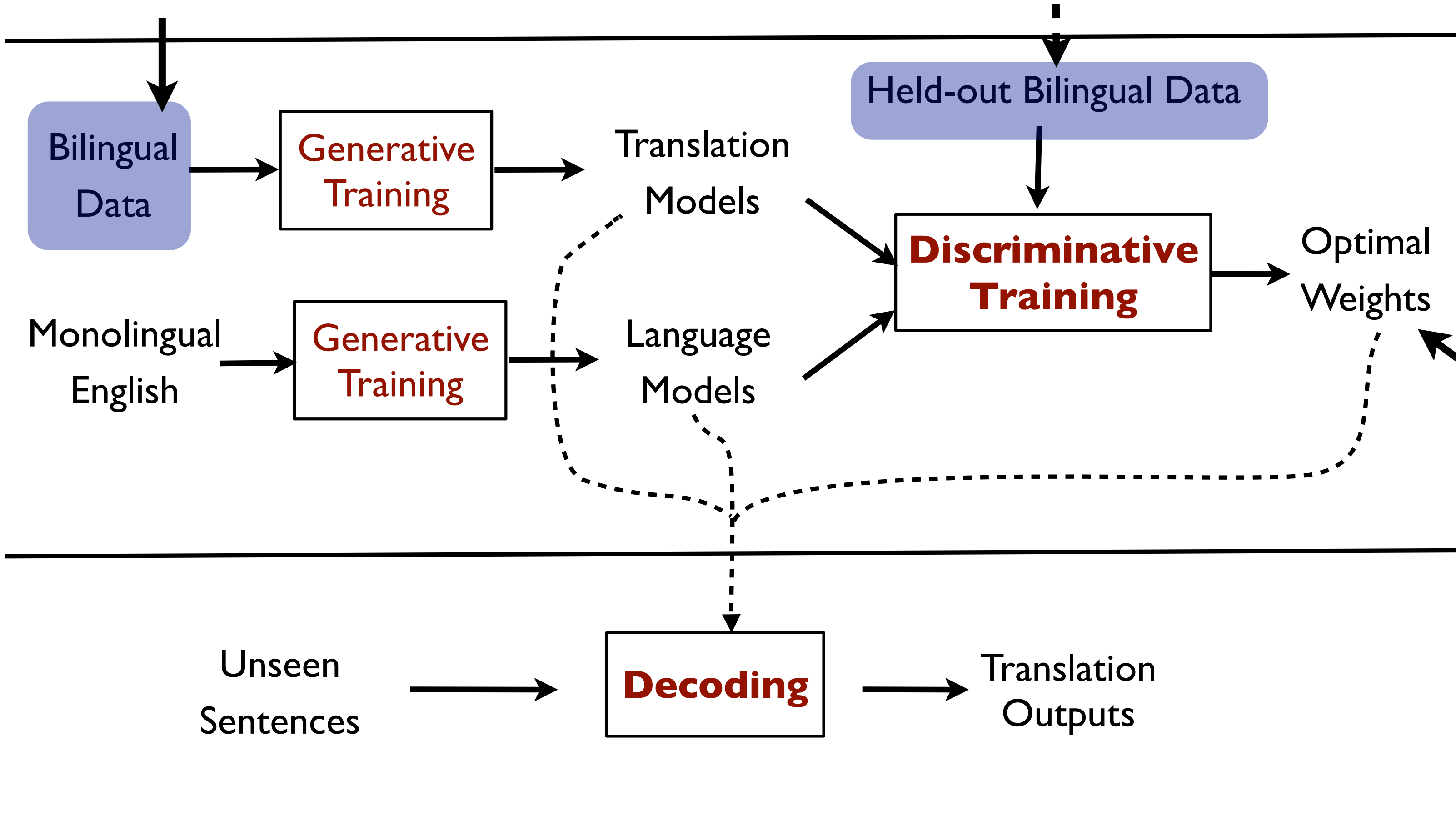


Semi-supervised Training

Training scenario	Test BLEU
Sup, (200, 200*16)	47.6
+Unsup, 100*16 Eng sentences	49.0
+Unsup, 200*16 Eng sentences	48.9

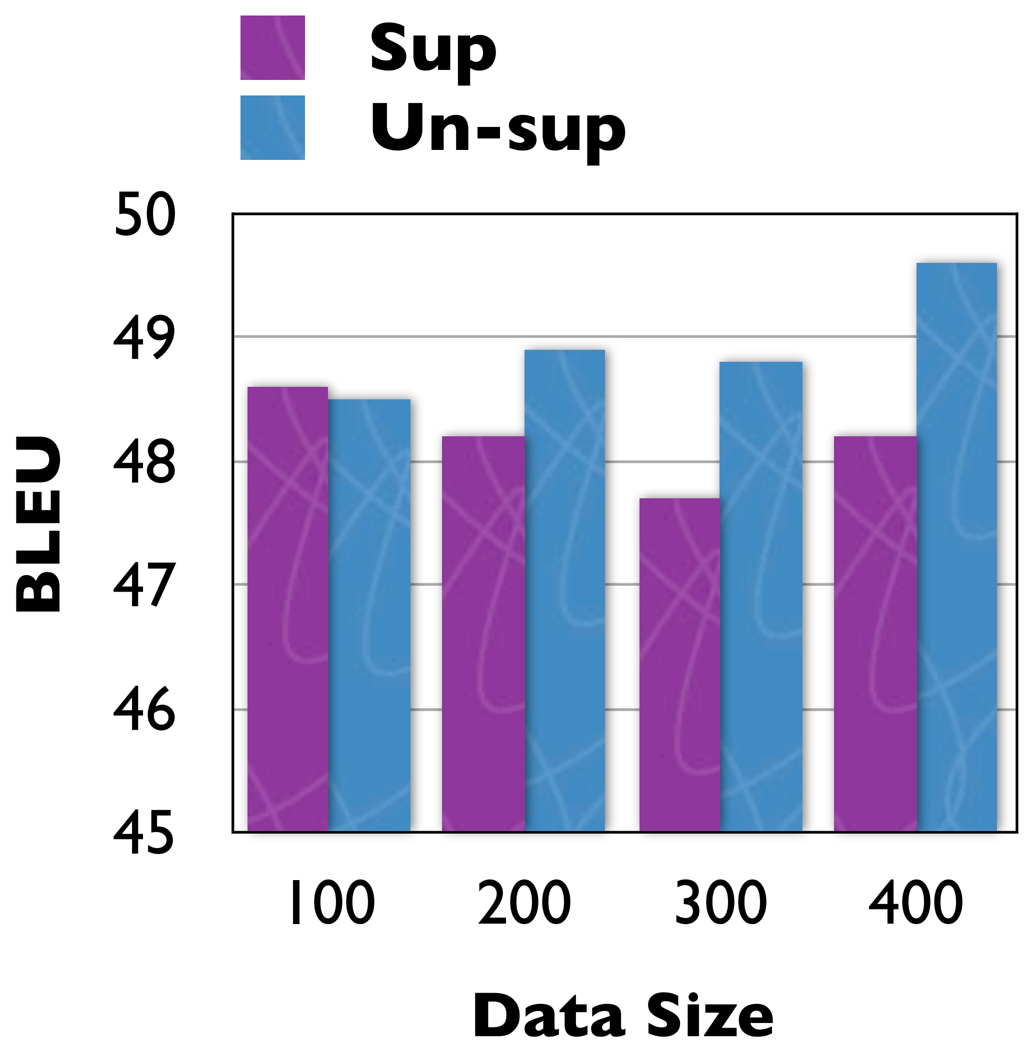
40K sent. pairs

Adding unsupervised data helps!



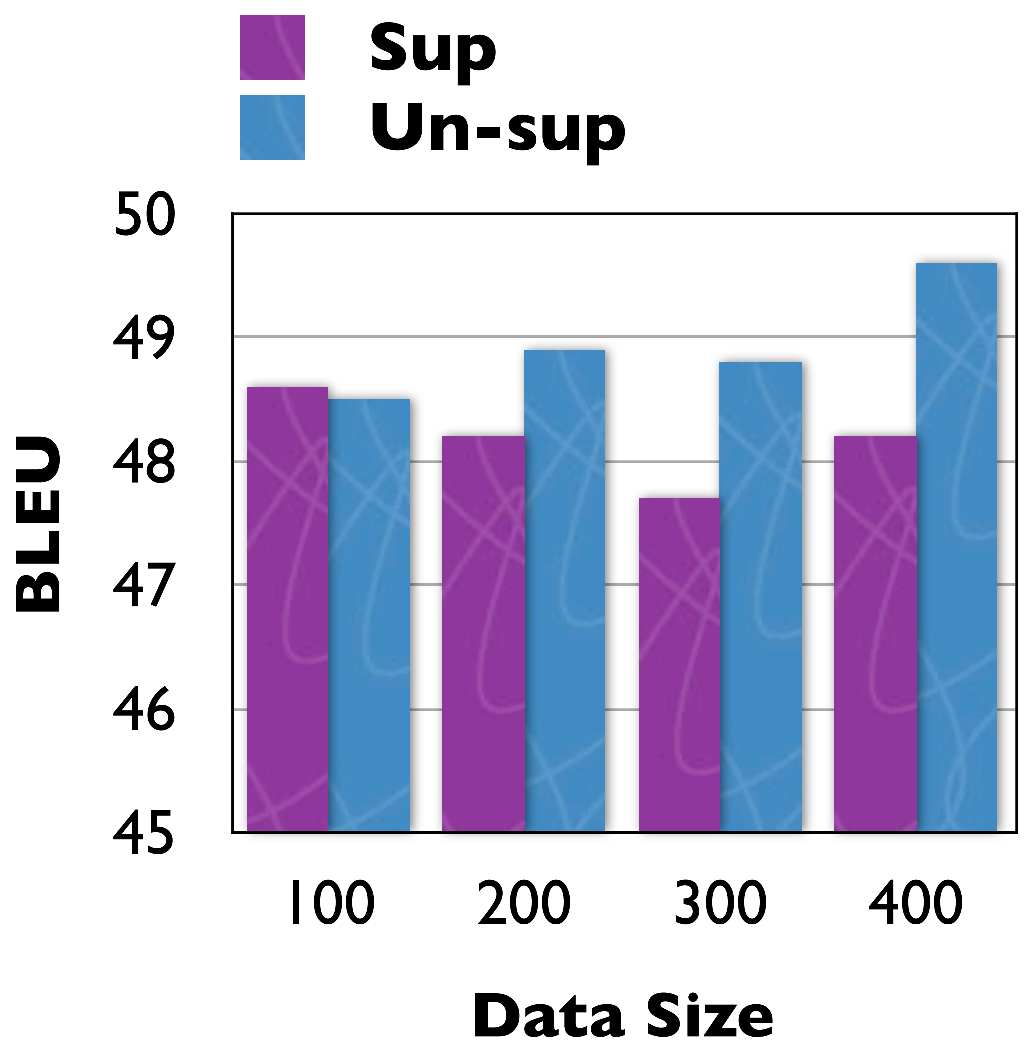
551 features

Supervised vs. Unsupervised



Unsupervised training performs as well as (and often better than) the supervised one!

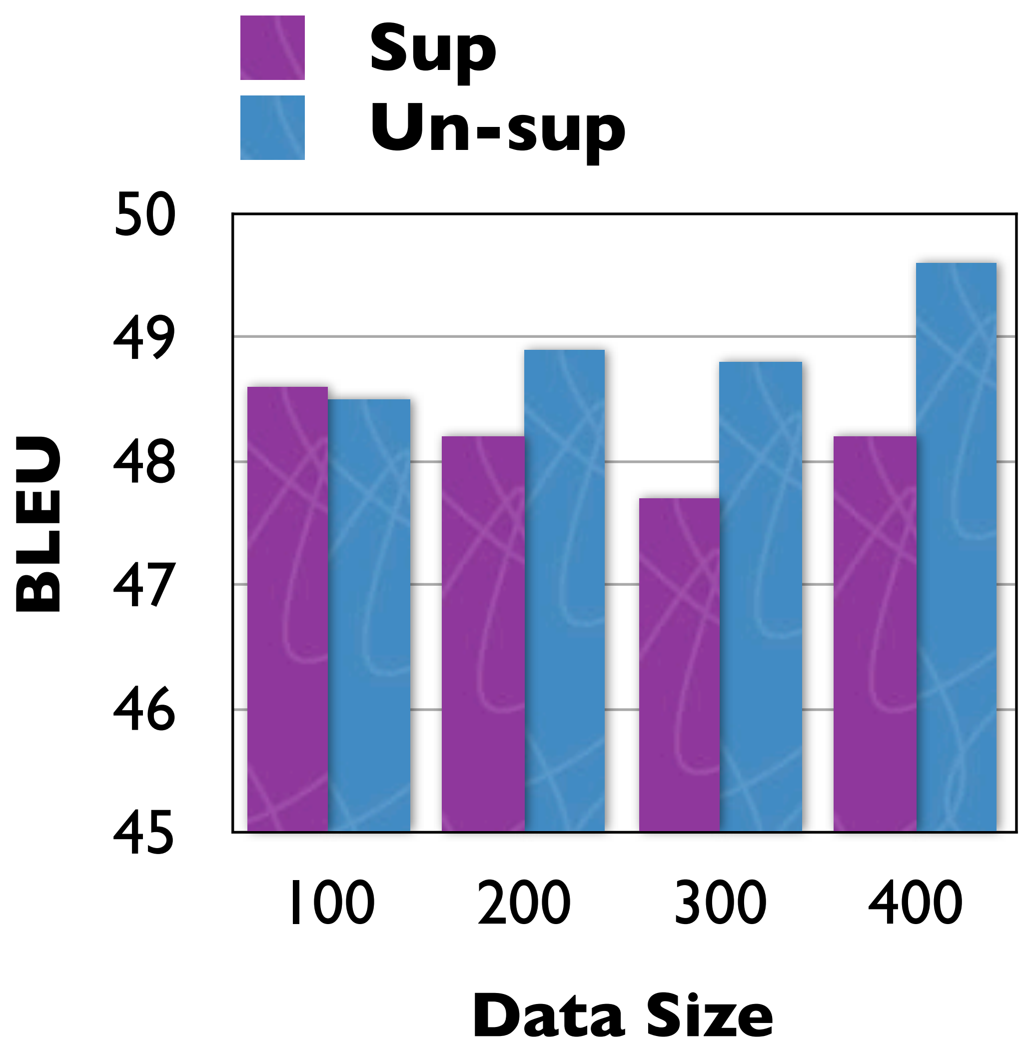
Supervised vs. Unsupervised



Unsupervised training performs as well as (and often better than) the supervised one!

Unsupervised uses **16** times of data as supervised. For example,

Supervised vs. Unsupervised

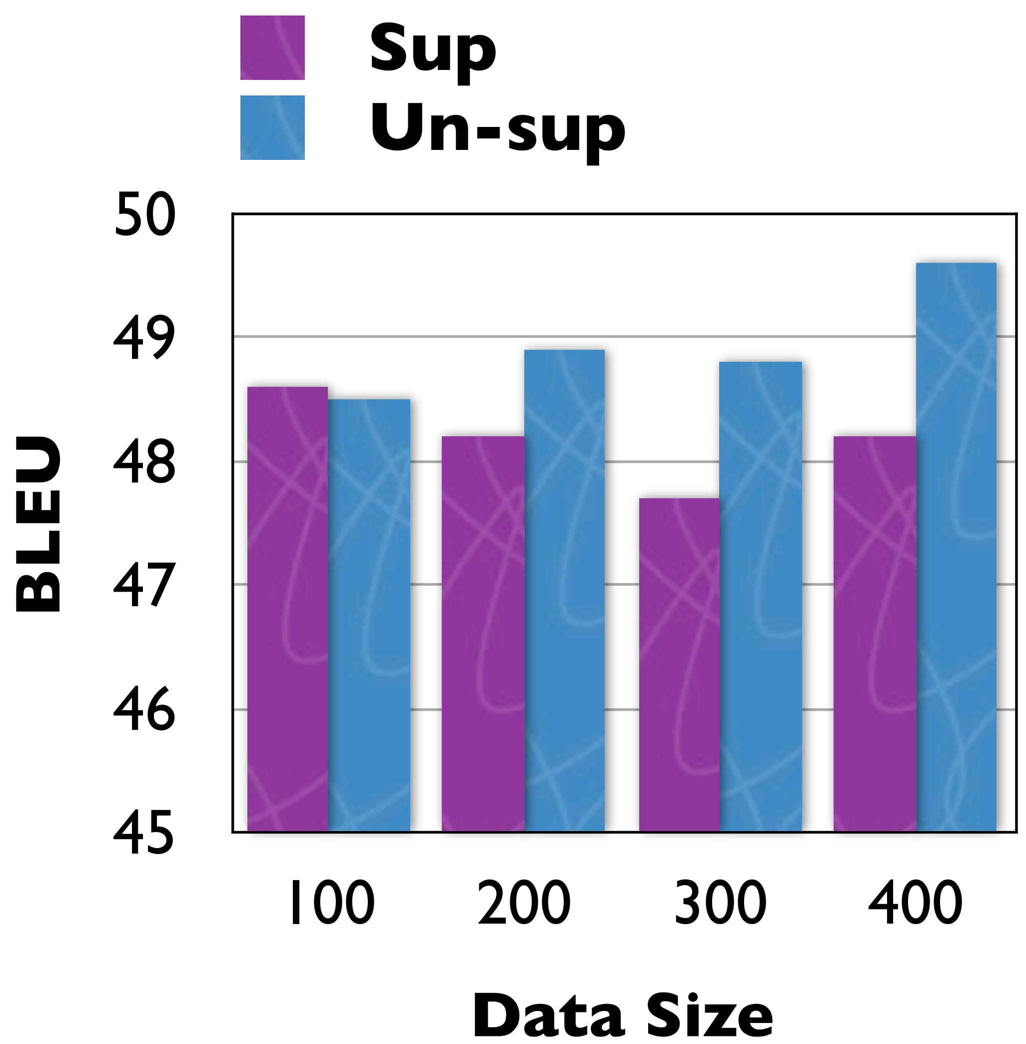


Unsupervised training performs as well as (and often better than) the supervised one!

Unsupervised uses **16** times of data as supervised. For example,

	Chinese	English
Sup	100	16*100

Supervised vs. Unsupervised

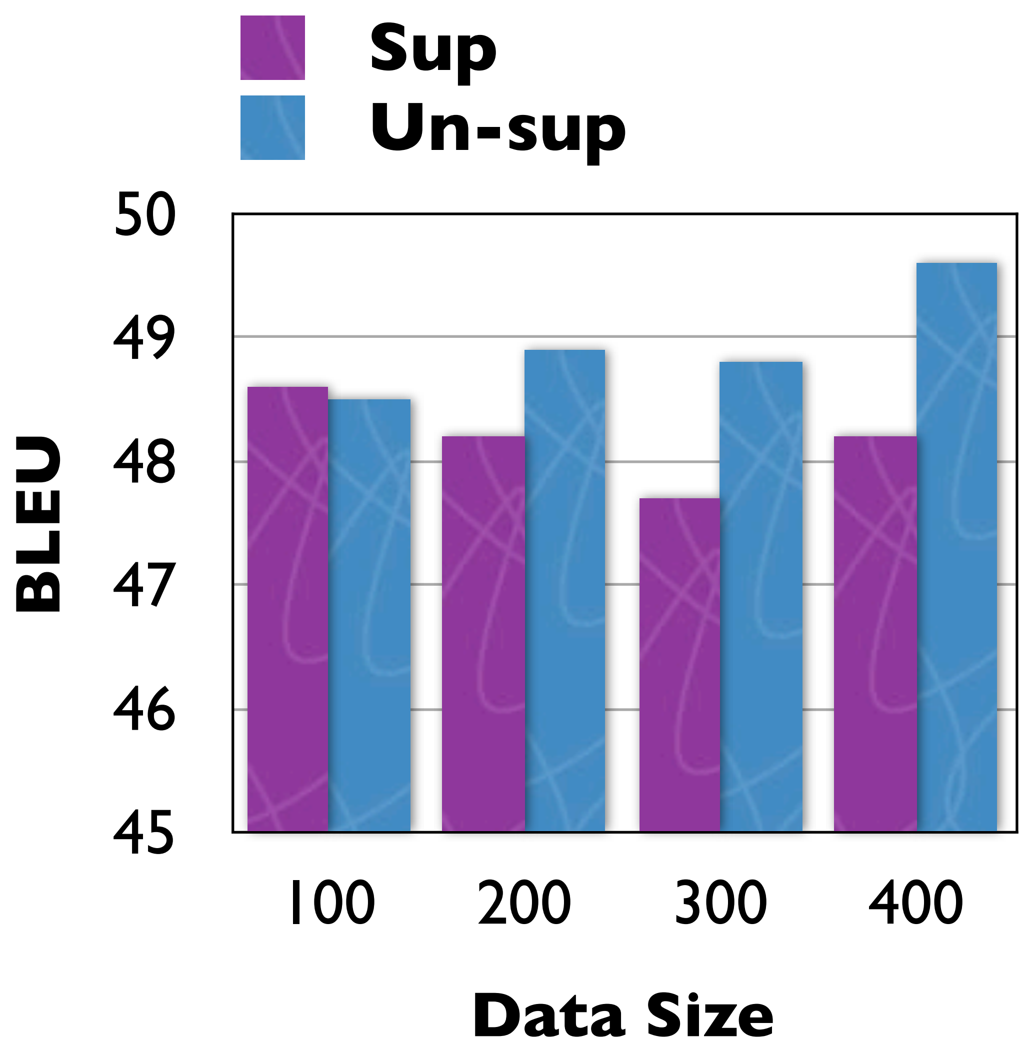


Unsupervised training performs as well as (and often better than) the supervised one!

Unsupervised uses **16** times of data as supervised. For example,

	Chinese	English
Sup	100	16*100
Unsup	16*100	16*16*100

Supervised vs. Unsupervised



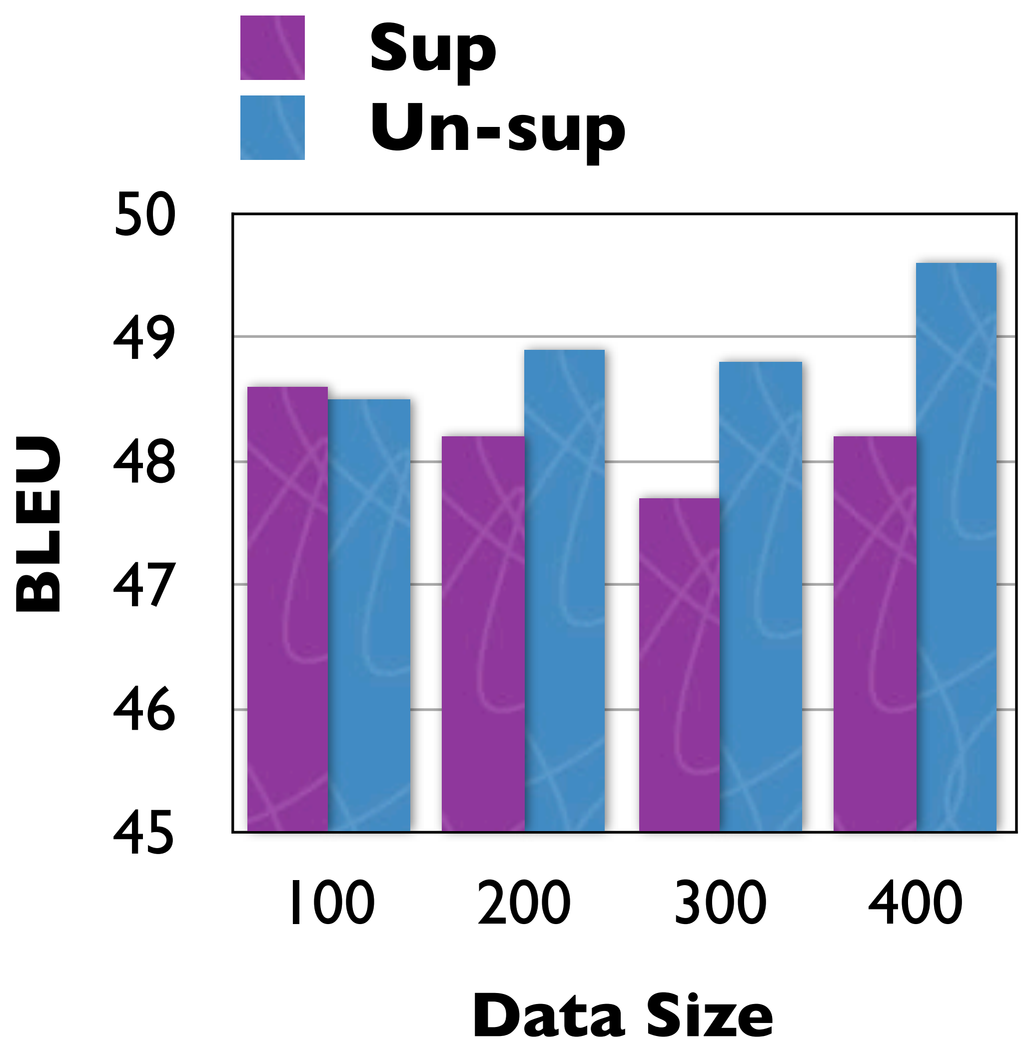
Unsupervised training performs as well as (and often better than) the supervised one!

Unsupervised uses **16** times of data as supervised. For example,

	Chinese	English
Sup	100	16*100
Unsup	16*100	16*16*100

But, fair comparison!

Supervised vs. Unsupervised



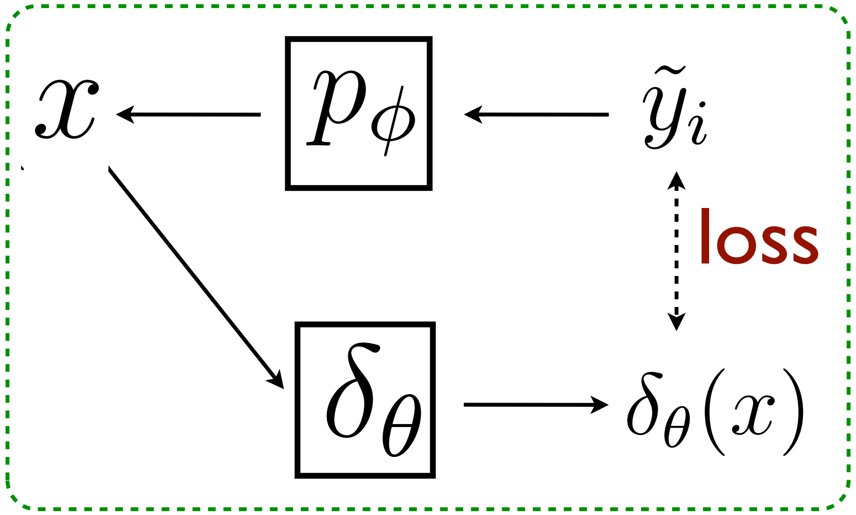
Unsupervised training performs as well as (and often better than) the supervised one!

Unsupervised uses **16** times of data as supervised. For example,

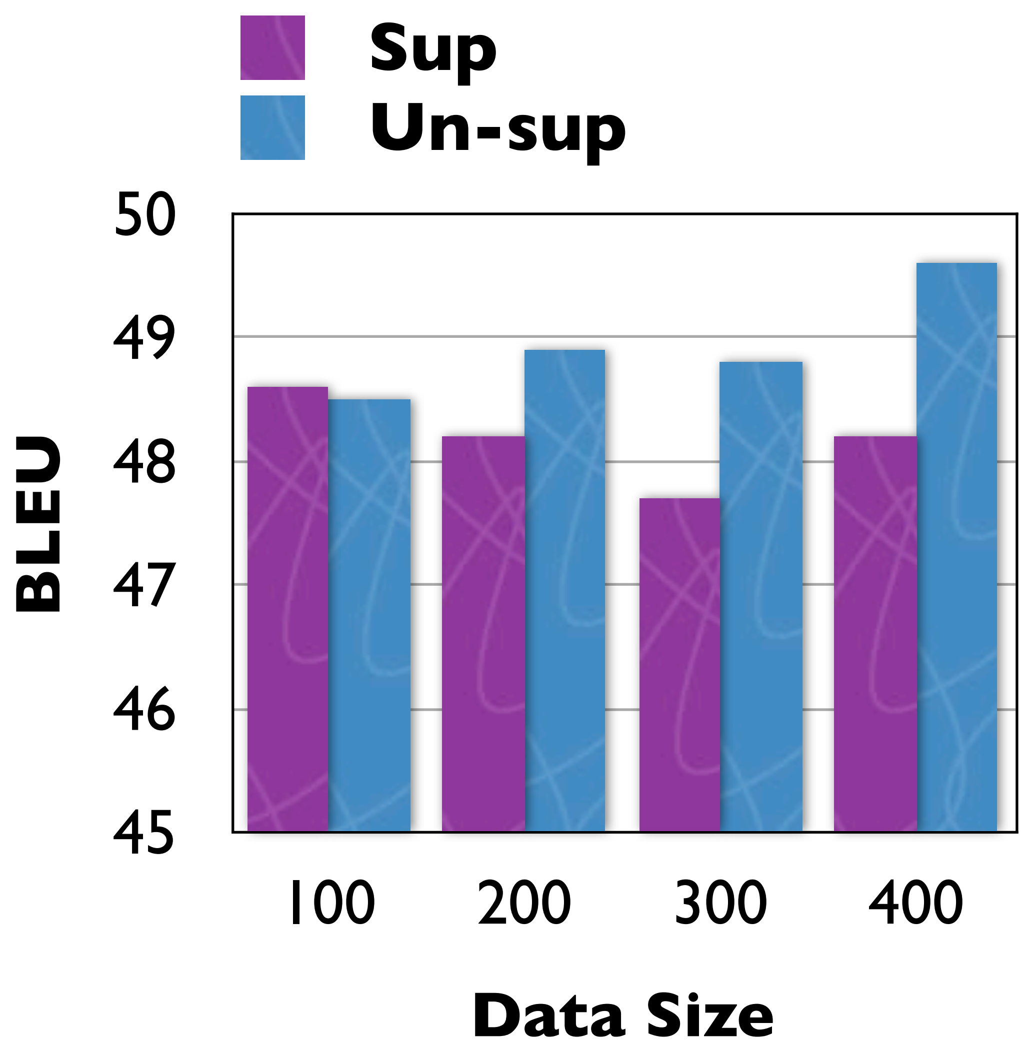
	Chinese	English
Sup	100	16*100
Unsup	16*100	16*16*100

But, fair comparison!

- More experiments



Supervised vs. Unsupervised



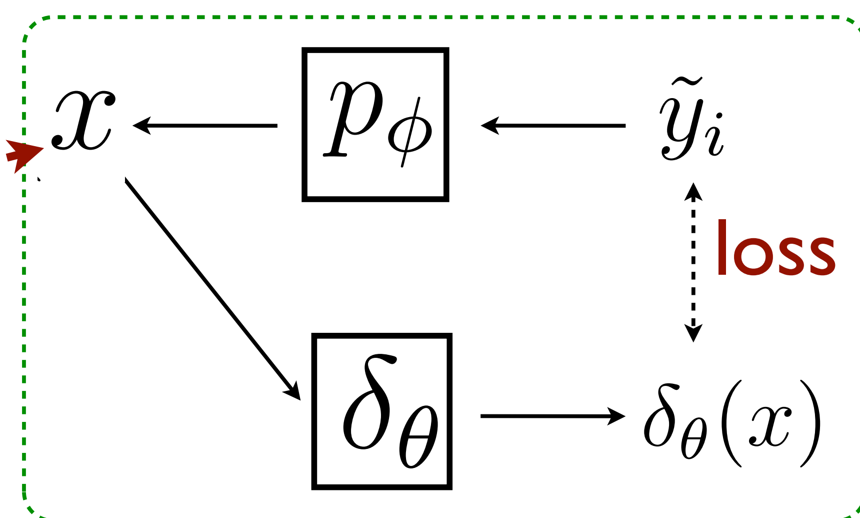
Unsupervised training performs as well as (and often better than) the supervised one!

Unsupervised uses **16** times of data as supervised. For example,

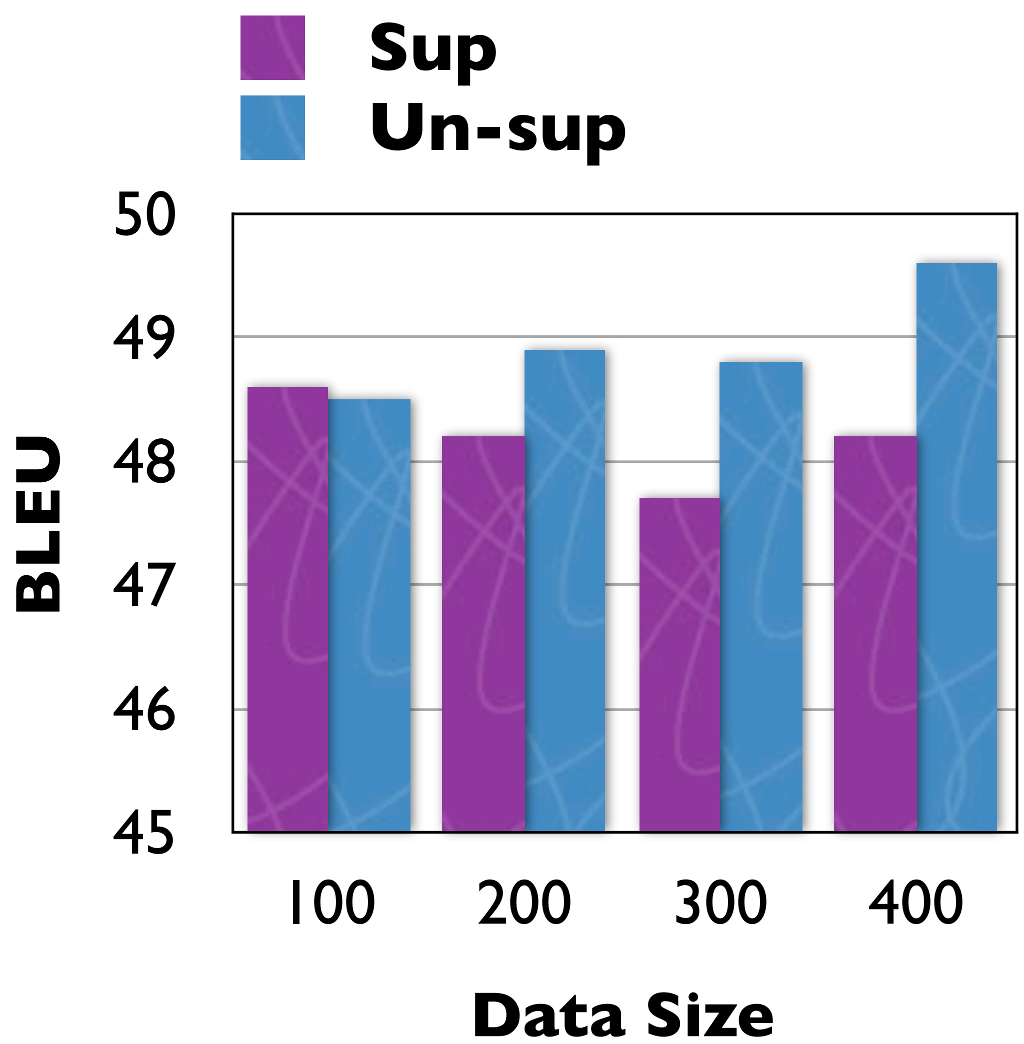
	Chinese	English
Sup	100	16*100
Unsup	16*100	16*16*100

But, fair comparison!

- More experiments
  - different k-best size



Supervised vs. Unsupervised



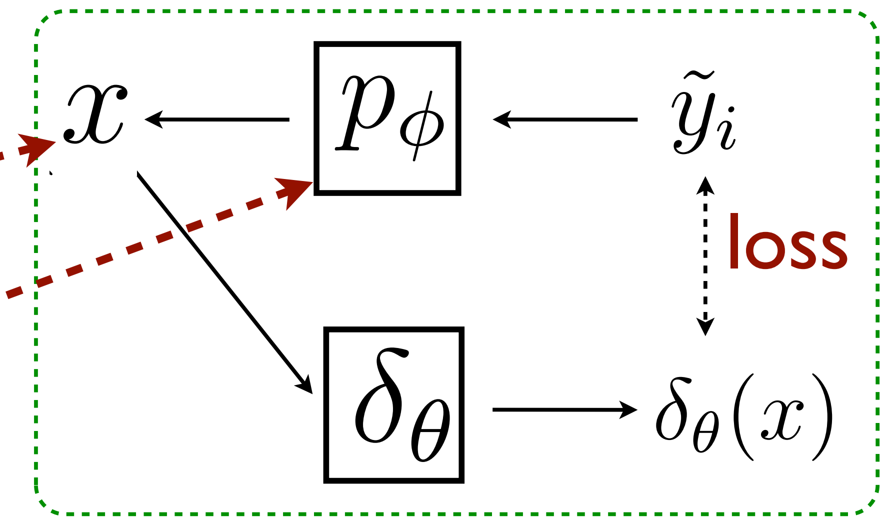
Unsupervised training performs as well as (and often better than) the supervised one!

Unsupervised uses **16** times of data as supervised. For example,

	Chinese	English
Sup	100	16*100
Unsup	16*100	16*16*100

But, fair comparison!

- More experiments
  - different k-best size
  - different reverse model



Outline

- Hypergraph as Hypothesis Space
- Unsupervised Discriminative Training
  - minimum imputed risk
  - contrastive language model estimation
- Variational Decoding
- First- and Second-order Expectation Semirings

Language Modeling

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) = \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) \quad = \quad \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

$w \in W_n$ → a set of n-grams occurred in y

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) \;=\; \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

W_n → a set of n-grams occurred in y

Locally normalized

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) = \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

$w \in W_n$ → a set of n-grams occurred in y

Locally normalized

- Global Log-linear Model (whole-sentence maximum-entropy LM)

(Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)}$$

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) \quad = \quad \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

W_n → a set of n-grams occurred in y

Locally normalized

- Global Log-linear Model (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) \quad = \quad \frac{e^{f(y)\cdot\theta}}{Z(*)}$$

Globally normalized

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) = \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

W_n → a set of n-grams occurred in y

Locally normalized

- Global Log-linear Model (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)}$$

$$Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

Globally normalized

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) = \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

$w \in W_n$ → a set of n-grams occurred in y

Locally normalized

- Global Log-linear Model (whole-sentence maximum-entropy LM)

(Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)}$$

$$Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

Globally normalized

Σ^* ← All English sentences with any length!

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) = \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

$w \in W_n$ → a set of n-grams occurred in y

Locally normalized

- Global Log-linear Model (whole-sentence maximum-entropy LM)

(Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)}$$

$$Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

Σ^* → All English sentences with any length!

Globally normalized

Sampling

Language Modeling

- Language Model $p_\theta(y)$
  - assign a probability to an English sentence y
  - typically use an n-gram model

$$p(y) = \prod_{w \in W_n} p(r(w) \mid h(w))^{c_w(y)}$$

W_n → a set of n-grams occurred in y

Locally normalized

- Global Log-linear Model (whole-sentence maximum-entropy LM)

(Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)}$$

$$Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

Σ^* ← All English sentences with any length!

Globally normalized

Sampling slow ☹

Contrastive Estimation

- Global Log-linear Model
  (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{\mathrm{Z}(*)} \qquad \mathrm{Z}(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

Contrastive Estimation

- Global Log-linear Model (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{\mathrm{Z}(*)} \qquad \mathrm{Z}(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{\mathrm{Z}(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$

Contrastive Estimation

- Global Log-linear Model
(whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{\mathrm{Z}(*)} \qquad \mathrm{Z}(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{\mathrm{Z}(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$

Contrastive Estimation

- Global Log-linear Model
  (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)} \qquad Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{Z(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$

$\tilde{y}$

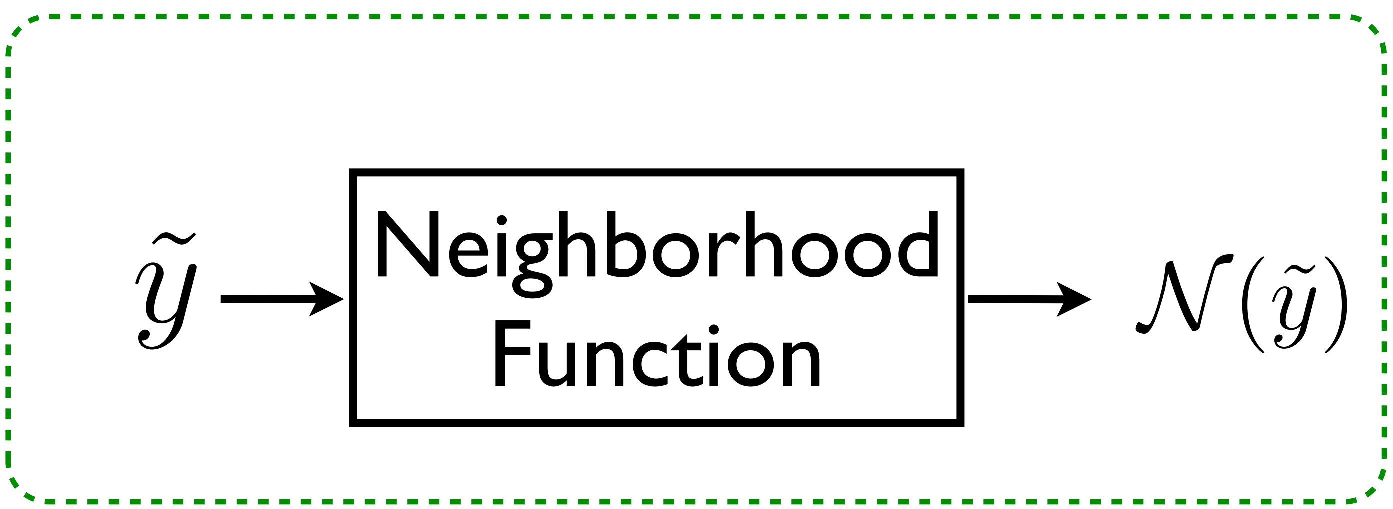
Contrastive Estimation

- Global Log-linear Model (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)} \qquad Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{Z(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$



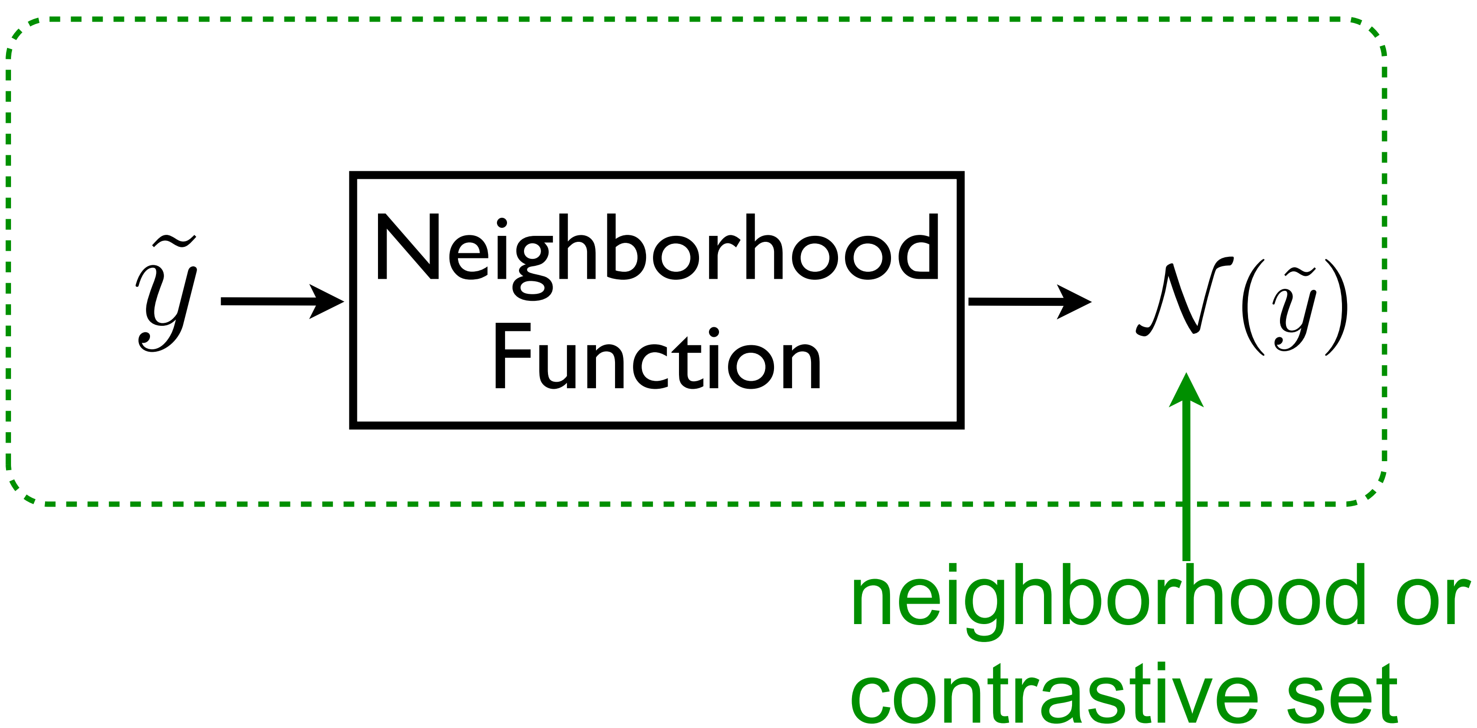
Contrastive Estimation

- Global Log-linear Model
  (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{\mathrm{Z}(*)} \qquad \mathrm{Z}(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{\mathrm{Z}(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$



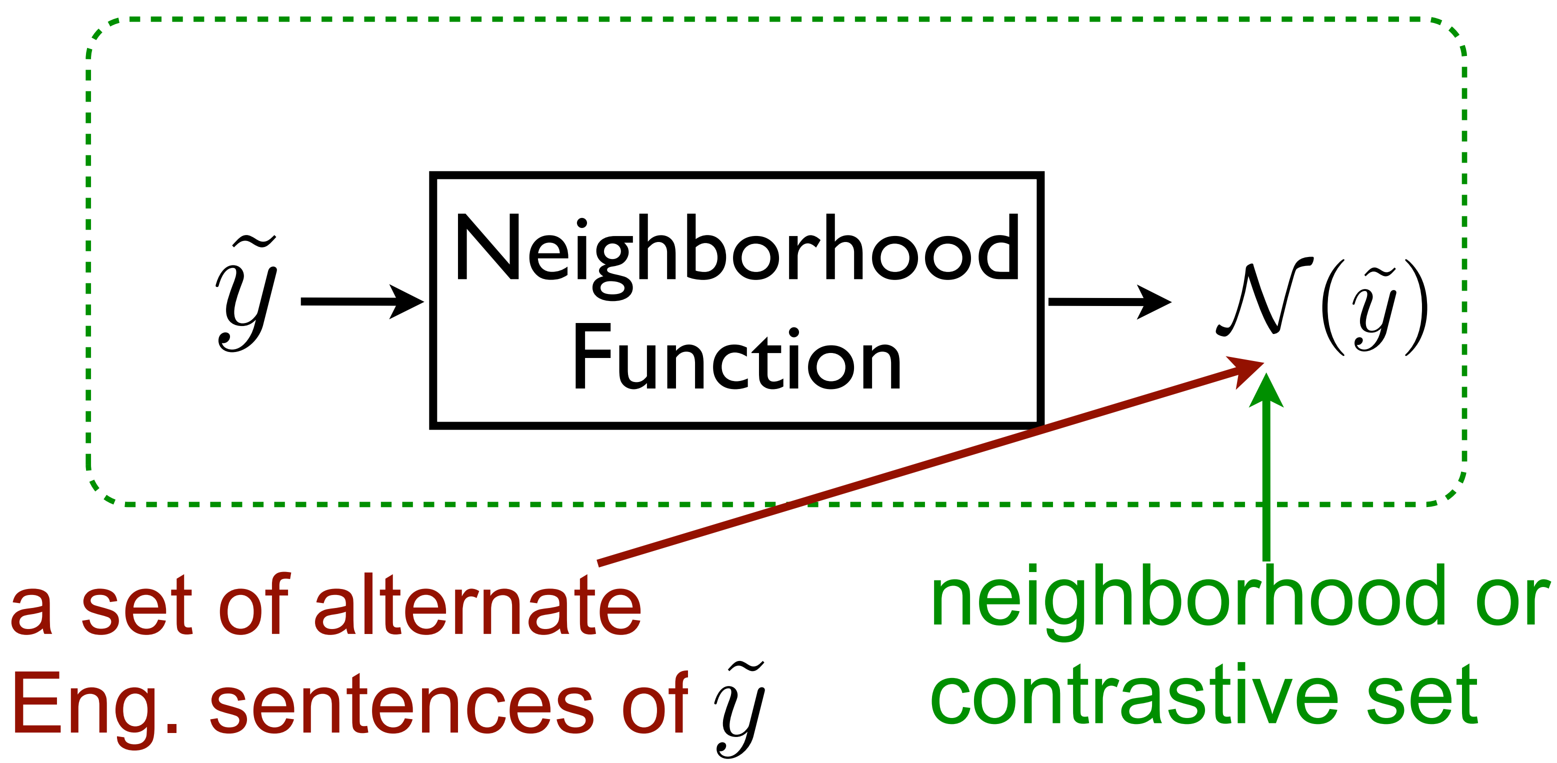
Contrastive Estimation

- Global Log-linear Model (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)} \qquad Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{Z(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$



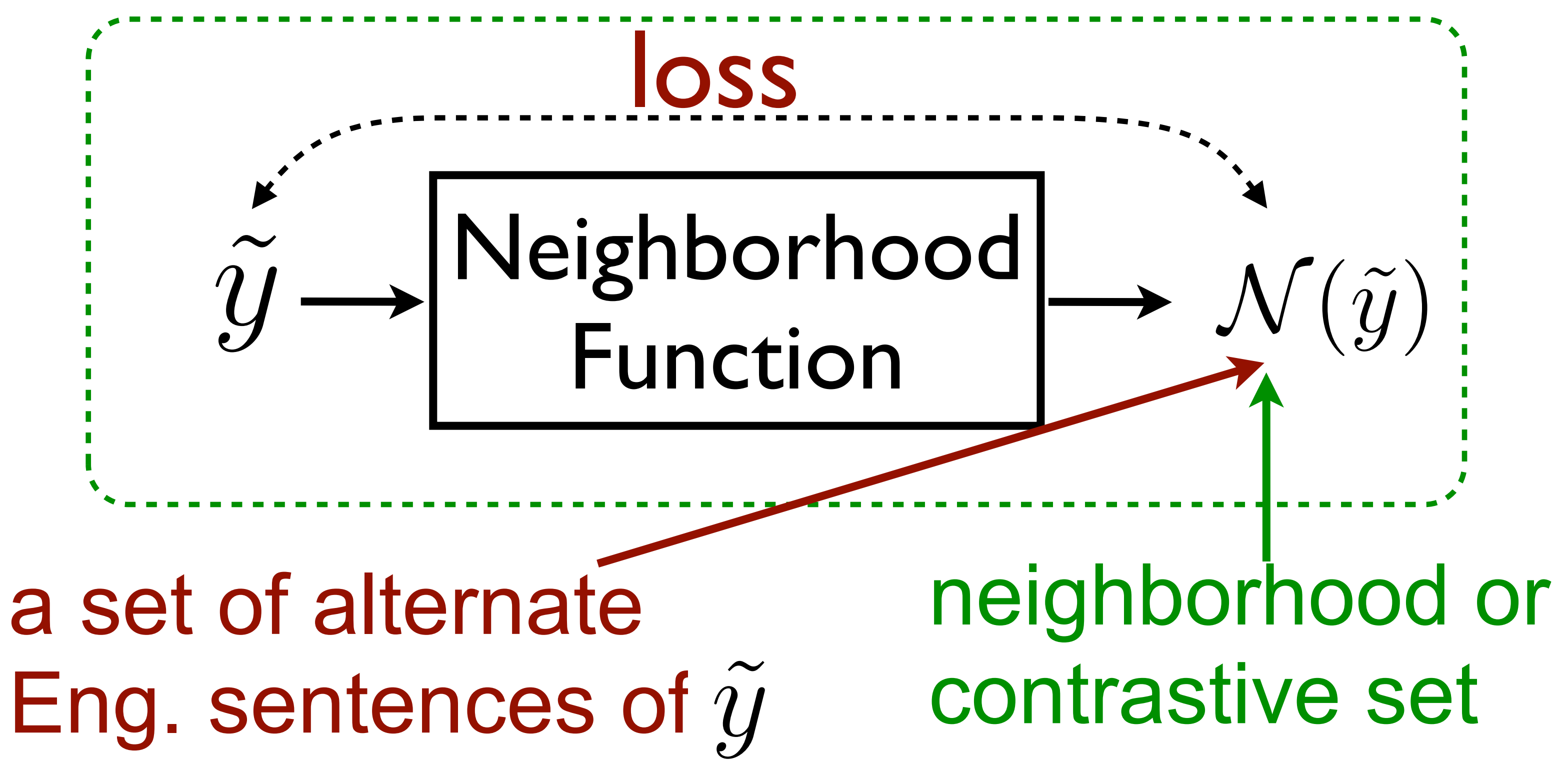
Contrastive Estimation

- Global Log-linear Model (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{\mathrm{Z}(*)} \qquad \mathrm{Z}(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{\mathrm{Z}(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$



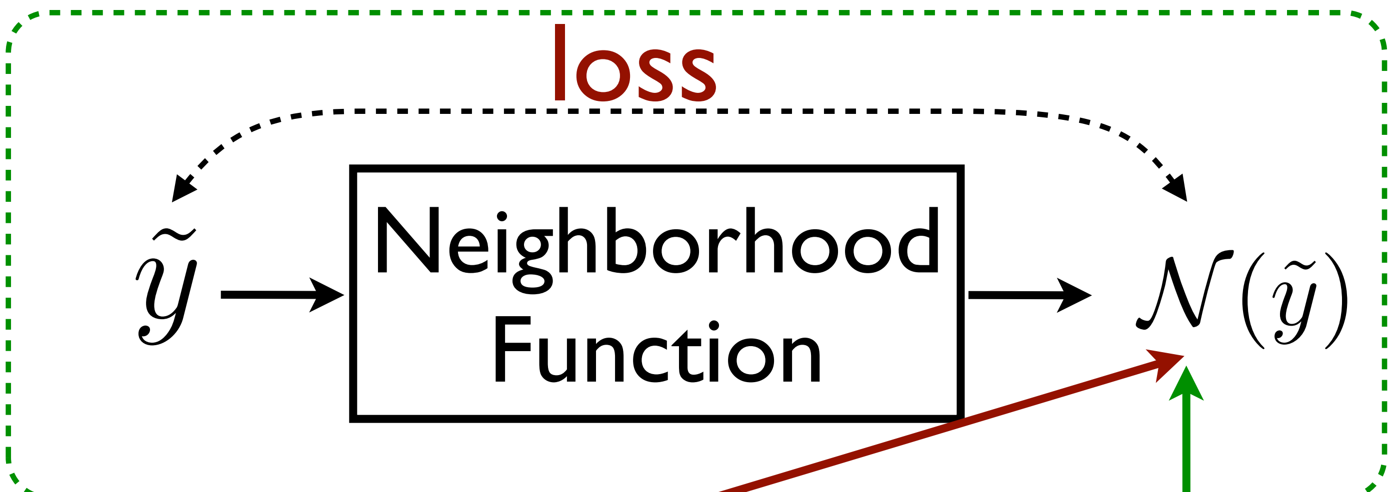
Contrastive Estimation

- Global Log-linear Model
  (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)} \qquad Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{Z(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$



a set of alternate Eng. sentences of $\tilde{y}$

neighborhood or contrastive set

Contrastive Estimation

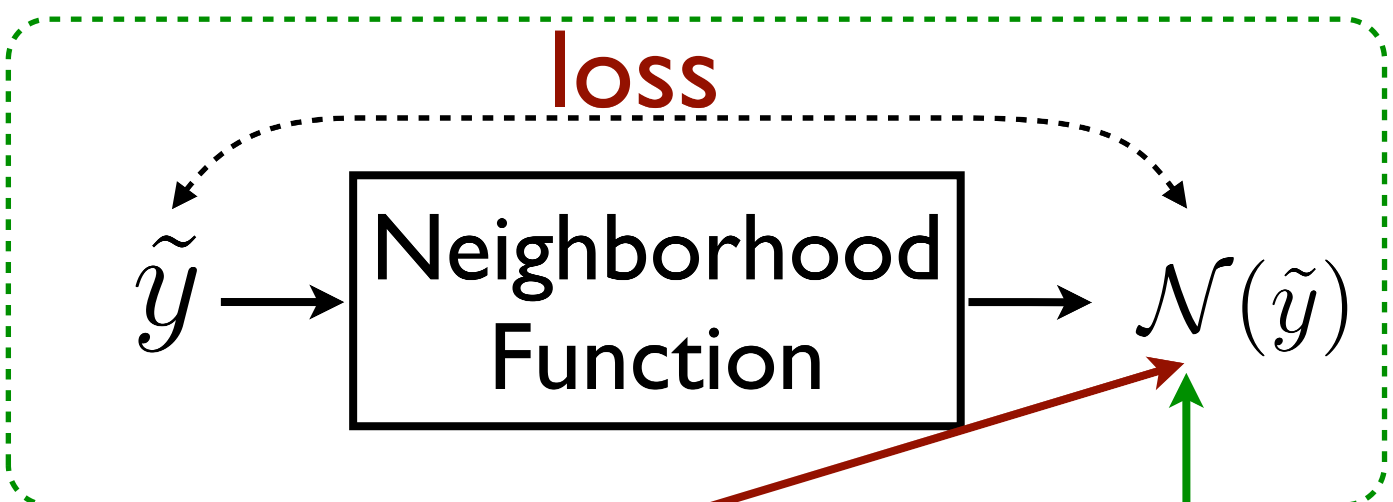
- Global Log-linear Model (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)} \qquad Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{Z(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$

improve both speed and accuracy



a set of alternate Eng. sentences of $\tilde{y}$

neighborhood or contrastive set

Contrastive Estimation

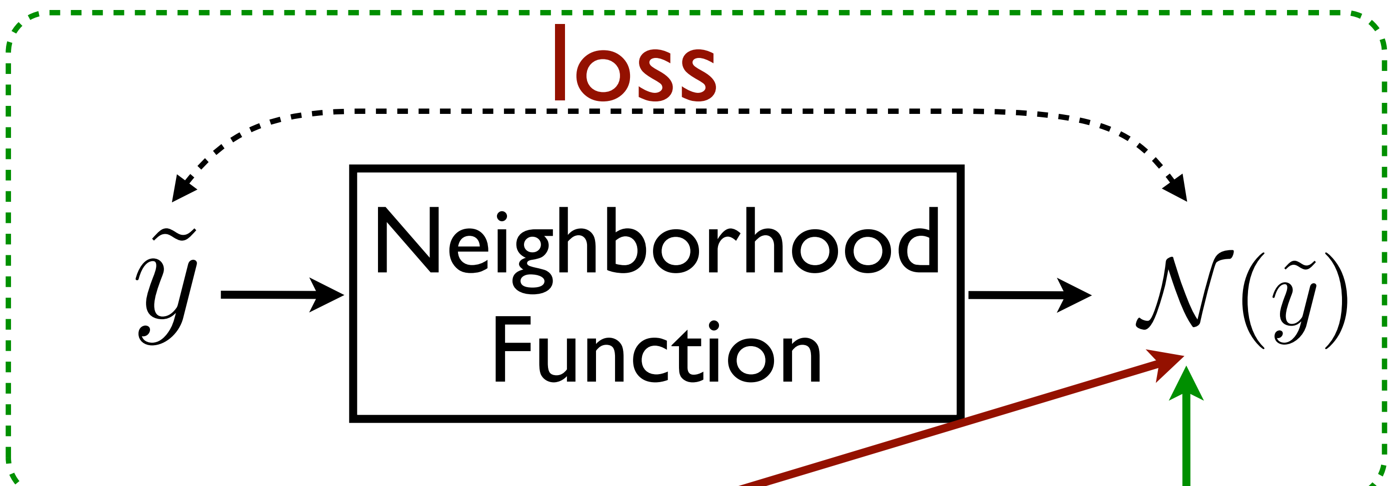
- Global Log-linear Model (whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)} \qquad Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{Z(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$

improve both speed and accuracy



not proposed for language modeling

a set of alternate Eng. sentences of $\tilde{y}$

neighborhood or contrastive set

Contrastive Estimation

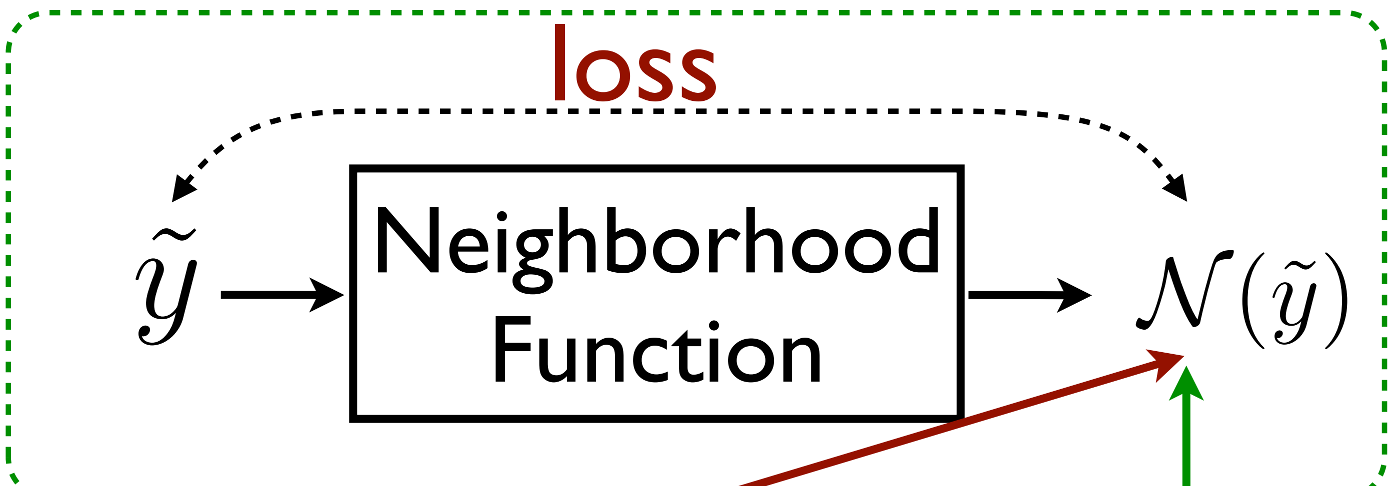
- Global Log-linear Model
(whole-sentence maximum-entropy LM) (Rosenfeld et al., 2001)

$$p_\theta(y) = \frac{e^{f(y)\cdot\theta}}{Z(*)} \qquad Z(*) \stackrel{\text{def}}{=} \sum_{y' \in \Sigma^*} e^{f(y')\cdot\theta}$$

- Contrastive Estimation (CE) (Smith and Eisner, 2005)

$$p_\theta(\tilde{y}) = \frac{e^{f(\tilde{y})\cdot\theta}}{Z(\tilde{y})} = \frac{e^{f(\tilde{y})\cdot\theta}}{\sum_{y' \in \mathcal{N}(\tilde{y})} e^{f(y')\cdot\theta}}$$

improve both speed and accuracy



a set of alternate Eng. sentences of $\tilde{y}$

neighborhood or contrastive set

not proposed for language modeling

train to recover the original English as much as possible

Contrastive Language Model Estimation

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
- Step-3: discriminative training

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
- Step-3: discriminative training

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG) ← neighborhood function
  - an English-to-English SCFG

$$X \rightarrow \langle \text{lead to}, \text{ result in} \rangle$$

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
- Step-3: discriminative training

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

$$X \rightarrow \langle \text{lead to}, \ \text{result in} \rangle$$

paraphrase

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG

- Step-3: discriminative training

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

paraphrase

$$X \rightarrow \langle \text{lead to}, \ \text{result in} \rangle$$

$$X \rightarrow \langle X_0 \text{ at beijing}, \text{beijing 's } X_0 \rangle$$

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
- Step-3: discriminative training

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

paraphrase

$$X \rightarrow \langle \text{lead to} , \ \text{result in} \rangle$$

$$X \rightarrow \langle X_0 \text{ at beijing} , \text{beijing 's } X_0 \rangle$$

$$X \rightarrow \langle X_0 \text{ of } X_1 , \ X_0 \text{ of the } X_1 \rangle$$

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
- Step-3: discriminative training

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

$$X \rightarrow \langle \text{lead to}, \text{ result in} \rangle$$ paraphrase

$$X \rightarrow \langle X_0 \text{ at beijing}, \text{beijing 's } X_0 \rangle$$

$$X \rightarrow \langle X_0 \text{ of } X_1, X_0 \text{ of the } X_1 \rangle$$ insertion

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
- Step-3: discriminative training

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

$$X \rightarrow \langle \text{lead to} \,,\ \text{result in} \rangle$$

paraphrase

$$X \rightarrow \langle X_0 \text{ at beijing} \,,\ \text{beijing 's } X_0 \rangle$$

$$X \rightarrow \langle X_0 \text{ of } X_1 \,,\ X_0 \text{ of the } X_1 \rangle$$

insertion

$$X \rightarrow \langle X_0 \text{ 's } X_1 \,,\ X_1 \text{ of } X_0 \rangle$$

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
- Step-3: discriminative training

Contrastive Language Model Estimation

- Step-1: extract a confusion grammar (CG)
  - an English-to-English SCFG

neighborhood function

$$X \rightarrow \langle \text{lead to}, \ \text{result in} \rangle$$ paraphrase

$$X \rightarrow \langle X_0 \text{ at beijing}, \text{beijing 's } X_0 \rangle$$

$$X \rightarrow \langle X_0 \text{ of } X_1, \ X_0 \text{ of the } X_1 \rangle$$ insertion

$$X \rightarrow \langle X_0 \text{ 's } X_1, \ X_1 \text{ of } X_0 \rangle$$ re-ordering

- Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
- Step-3: discriminative training

Step-1: Extracting a Confusion Grammar (CG)

Step-1: Extracting a Confusion Grammar (CG)

- Deriving a CG from a bilingual grammar
  - use Chinese side as pivots

Step-1: Extracting a Confusion Grammar (CG)

- Deriving a CG from a bilingual grammar
  - use Chinese side as pivots

Bilingual Rule	Confusion Rule

Step-1: Extracting a Confusion Grammar (CG)

- Deriving a CG from a bilingual grammar
  - use Chinese side as pivots

Bilingual Rule	Confusion Rule
$X \rightarrow \langle \text{ mao}, \text{ a cat} \rangle$	
$X \rightarrow \langle \text{ mao}, \text{ the cat} \rangle$	

Step-1: Extracting a Confusion Grammar (CG)

- Deriving a CG from a bilingual grammar
  - use Chinese side as pivots

Bilingual Rule	Confusion Rule
$X \rightarrow \langle \text{ mao}, \text{ a cat} \rangle$	$X \rightarrow \langle \text{ a cat}, \text{ the cat} \rangle$
$X \rightarrow \langle \text{ mao}, \text{ the cat} \rangle$	$X \rightarrow \langle \text{ the cat}, \text{ a cat} \rangle$

Step-1: Extracting a Confusion Grammar (CG)

- Deriving a CG from a bilingual grammar
  - use Chinese side as pivots

Bilingual Rule	Confusion Rule
$X \rightarrow \langle \text{ mao}, \text{ a cat} \rangle$	$X \rightarrow \langle \text{ a cat}, \text{ the cat} \rangle$
$X \rightarrow \langle \text{ mao}, \text{ the cat} \rangle$	$X \rightarrow \langle \text{ the cat}, \text{ a cat} \rangle$
$X \rightarrow \langle X_0 \text{ de } X_1, X_0 \text{ on } X_1 \rangle$	
$X \rightarrow \langle X_0 \text{ de } X_1, X_1 \text{ of } X_0 \rangle$	

Step-1: Extracting a Confusion Grammar (CG)

- Deriving a CG from a bilingual grammar
  - use Chinese side as pivots

Bilingual Rule	Confusion Rule
$X \rightarrow \langle\ \text{mao},\ \text{a cat}\rangle$	$X \rightarrow \langle\ \text{a cat},\ \text{the cat}\rangle$
$X \rightarrow \langle\ \text{mao},\ \text{the cat}\rangle$	$X \rightarrow \langle\ \text{the cat},\ \text{a cat}\rangle$
$X \rightarrow \langle\ X_0 \text{ de } X_1, X_0 \text{ on } X_1\rangle$	$X \rightarrow \langle\ X_0 \text{ on } X_1, X_1 \text{ of } X_0\rangle$
$X \rightarrow \langle\ X_0 \text{ de } X_1, X_1 \text{ of } X_0\rangle$	$X \rightarrow \langle\ X_0 \text{ of } X_1, X_1 \text{ on } X_0\rangle$

Step-1: Extracting a Confusion Grammar (CG)

- Deriving a CG from a bilingual grammar
  - use Chinese side as pivots

Bilingual Rule	Confusion Rule
$X \rightarrow \langle\, \text{mao},\ \text{a cat} \rangle$	$X \rightarrow \langle\, \text{a cat},\ \text{the cat} \rangle$
$X \rightarrow \langle\, \text{mao},\ \text{the cat} \rangle$	$X \rightarrow \langle\, \text{the cat},\ \text{a cat} \rangle$
$X \rightarrow \langle\, X_0 \text{ de } X_1, X_0 \text{ on } X_1 \rangle$	$X \rightarrow \langle\, X_0 \text{ on } X_1, X_1 \text{ of } X_0 \rangle$
$X \rightarrow \langle\, X_0 \text{ de } X_1, X_1 \text{ of } X_0 \rangle$	$X \rightarrow \langle\, X_0 \text{ of } X_1, X_1 \text{ on } X_0 \rangle$

CG captures the confusion an MT system will have when translating an input.

Step-1: Extracting a Confusion Grammar (CG)

- Deriving a CG from a bilingual grammar
  - use Chinese side as pivots

Bilingual Rule	Confusion Rule
$X \rightarrow \langle \text{mao}, \text{ a cat} \rangle$	$X \rightarrow \langle \text{a cat}, \text{ the cat} \rangle$
$X \rightarrow \langle \text{mao}, \text{ the cat} \rangle$	$X \rightarrow \langle \text{the cat}, \text{ a cat} \rangle$
$X \rightarrow \langle X_0 \text{ de } X_1, X_0 \text{ on } X_1 \rangle$	$X \rightarrow \langle X_0 \text{ on } X_1, X_1 \text{ of } X_0 \rangle$
$X \rightarrow \langle X_0 \text{ de } X_1, X_1 \text{ of } X_0 \rangle$	$X \rightarrow \langle X_0 \text{ of } X_1, X_1 \text{ on } X_0 \rangle$

CG captures the confusion an MT system will have when translating an input.

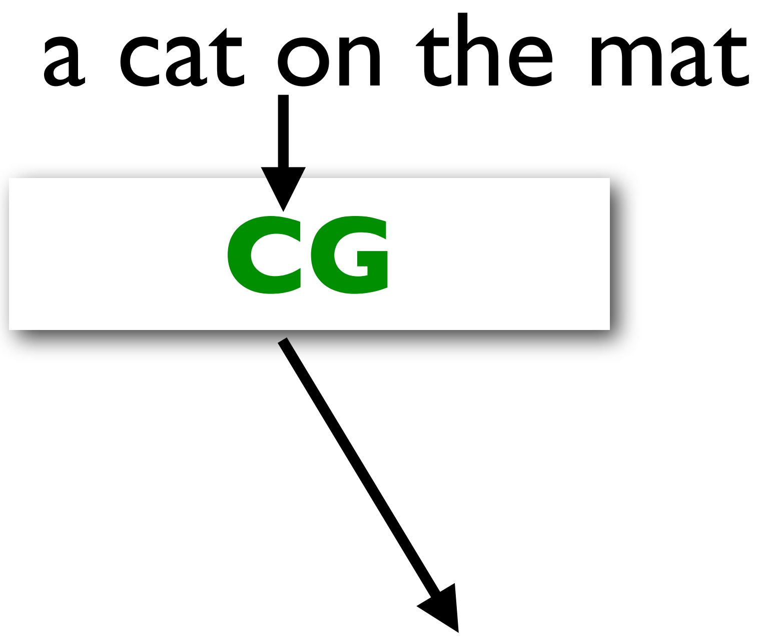
Our neighborhood function is **learned** and **MT-specific**.

Step-2: Generating Contrastive Sets

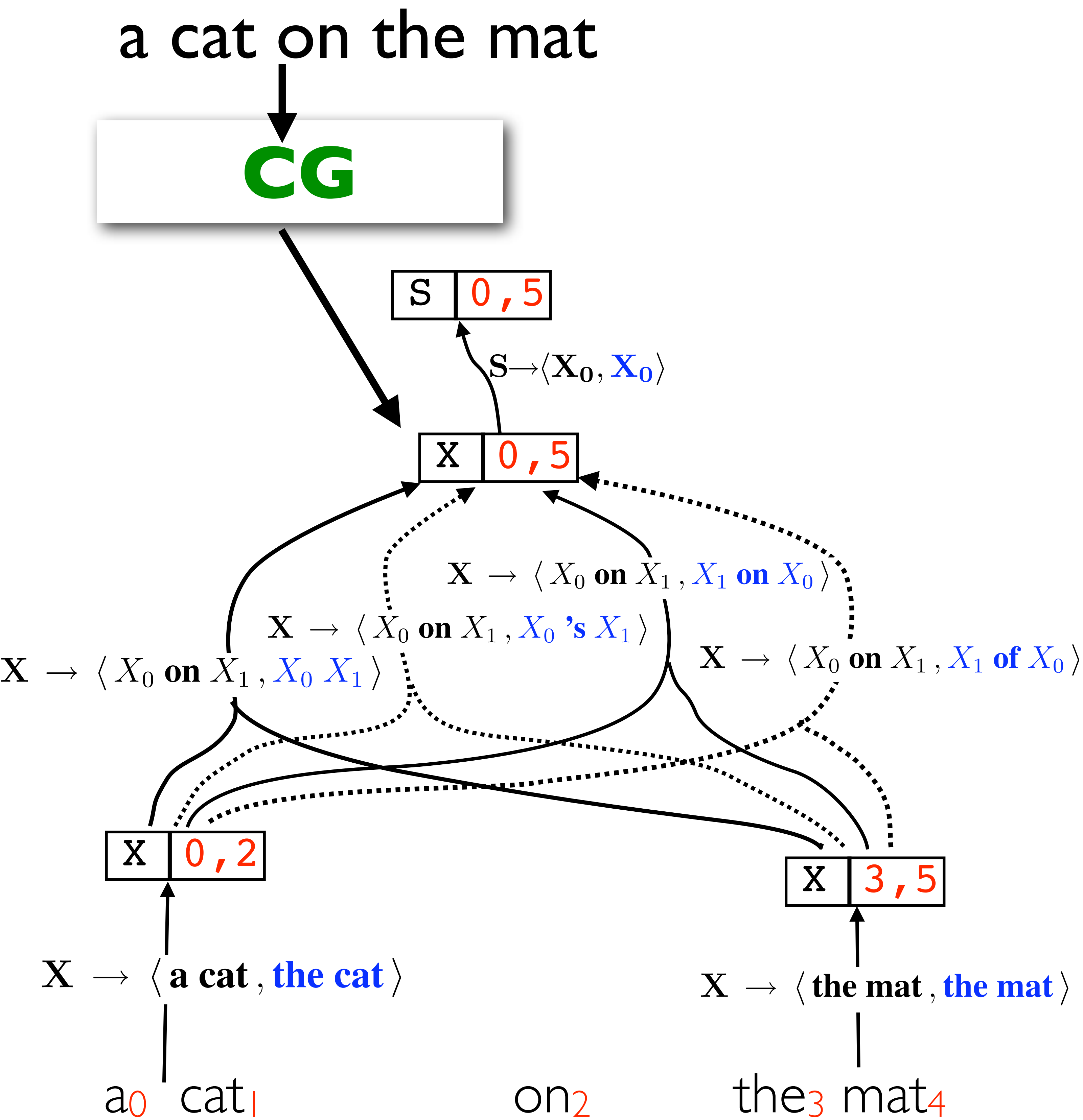
Step-2: Generating Contrastive Sets

a cat on the mat

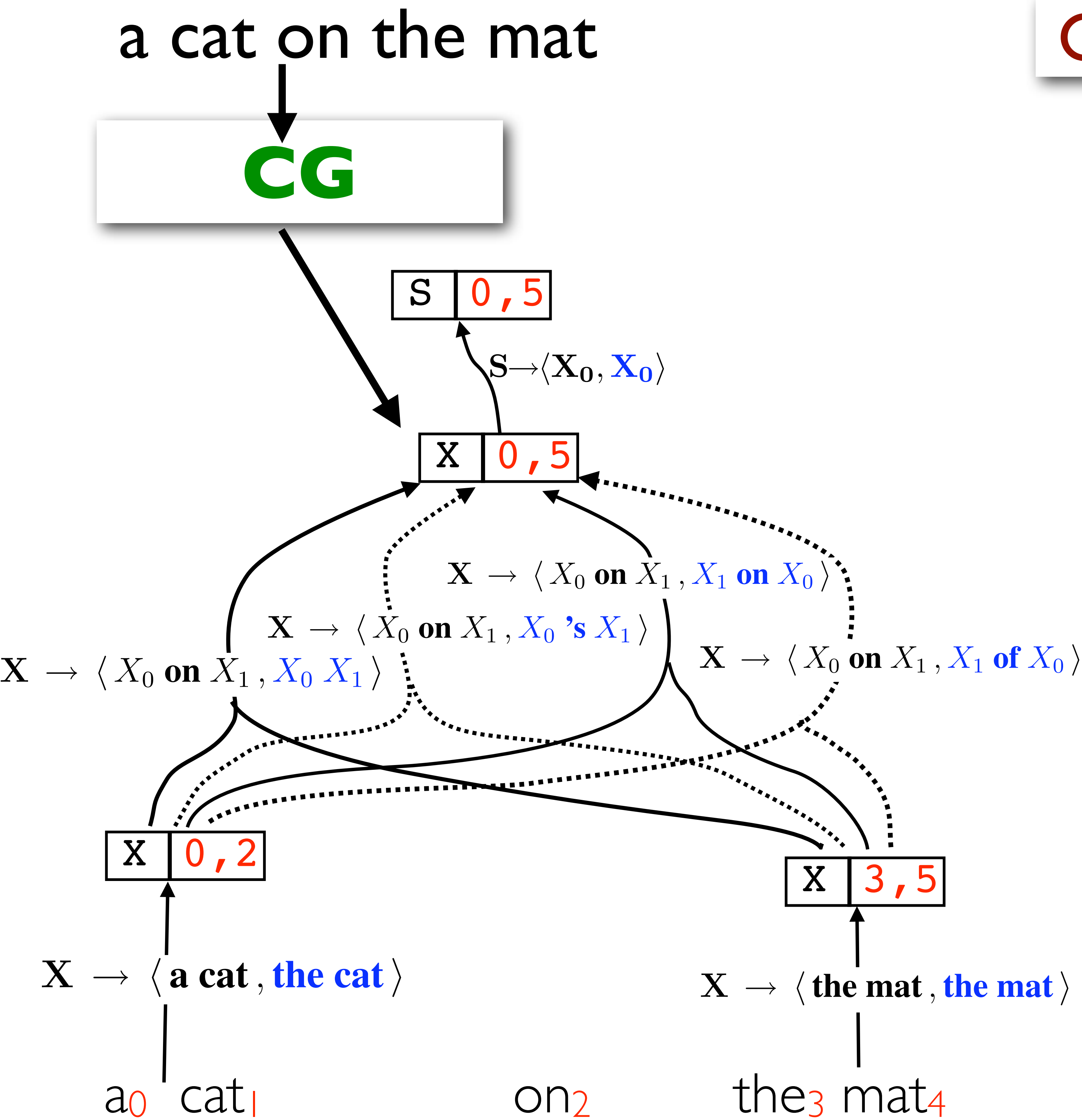
Step-2: Generating Contrastive Sets



Step-2: Generating Contrastive Sets



Step-2: Generating Contrastive Sets



Contrastive set:

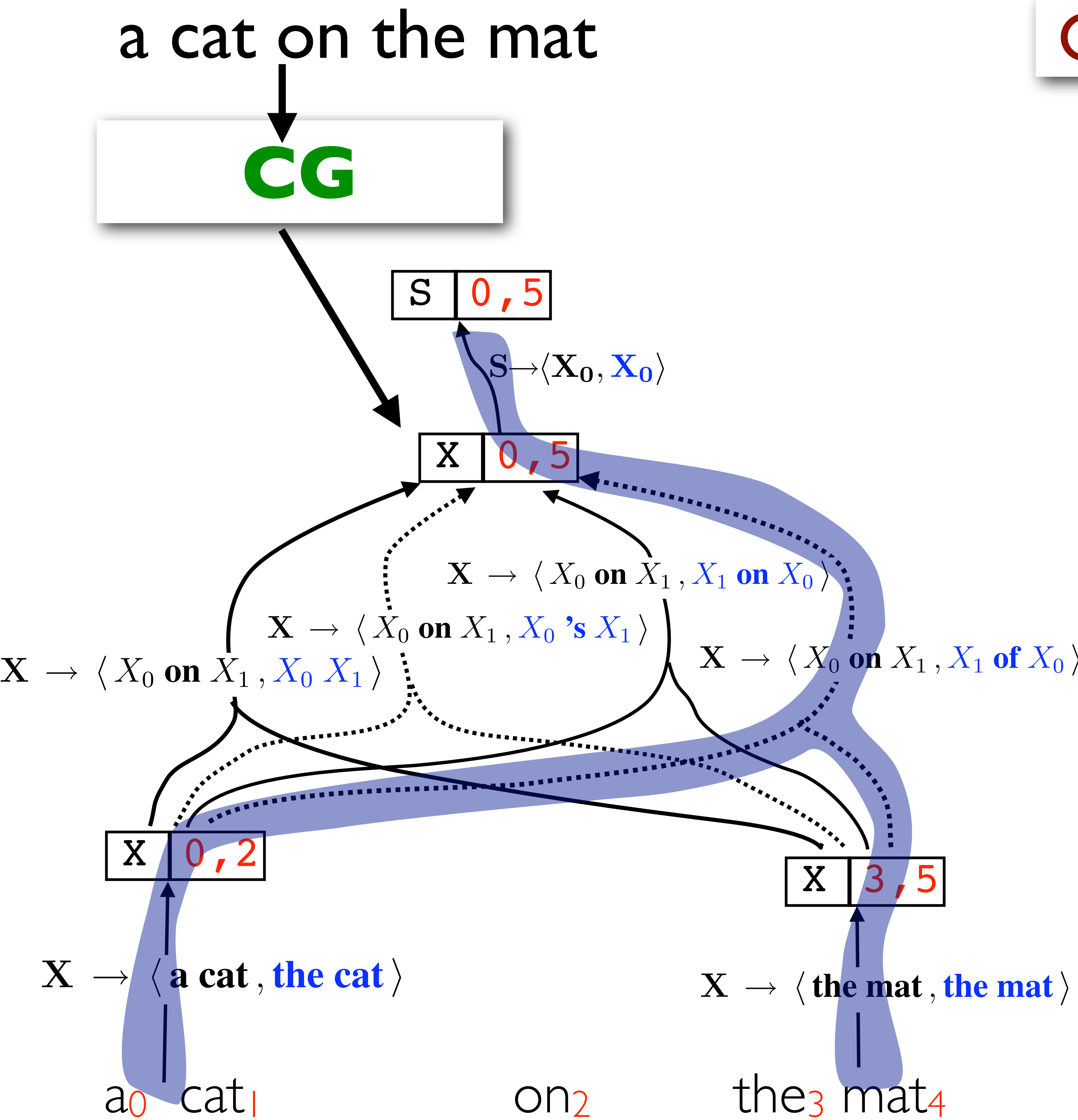
the cat the mat

the cat 's the mat

the mat on the cat

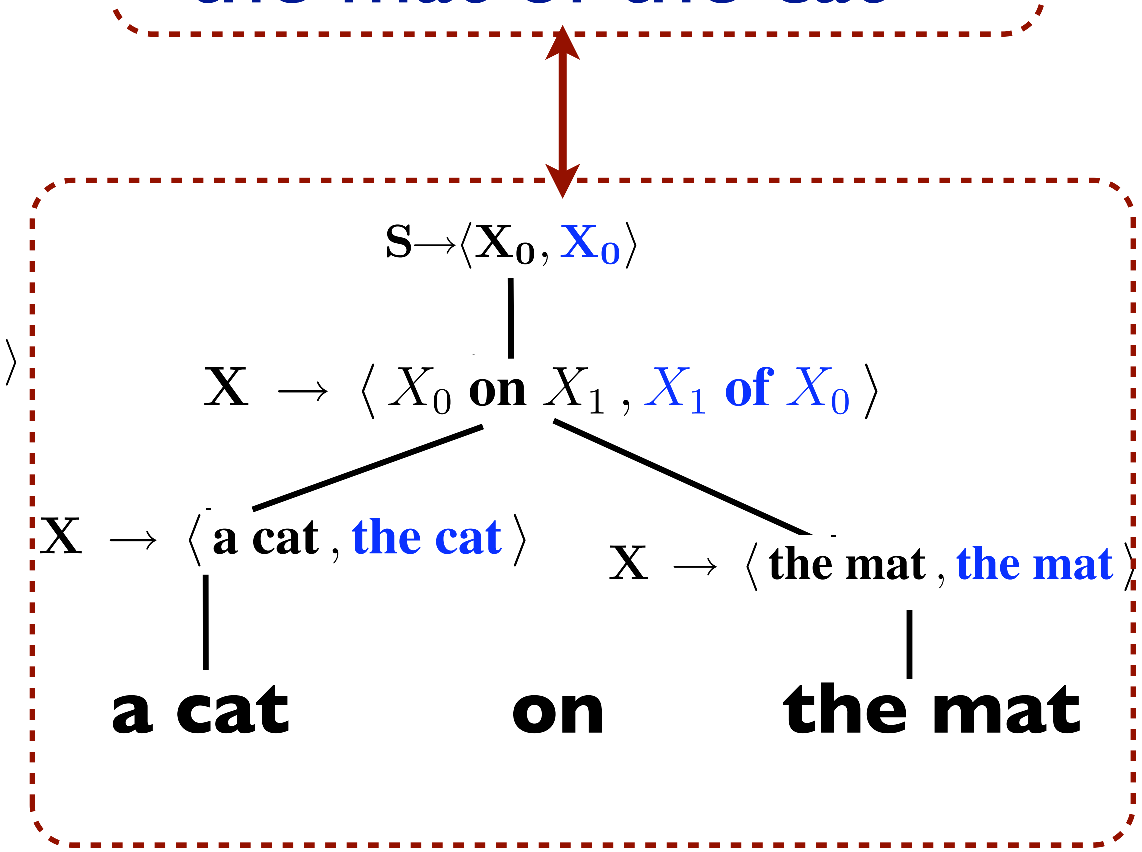
the mat of the cat

Step-2: Generating Contrastive Sets

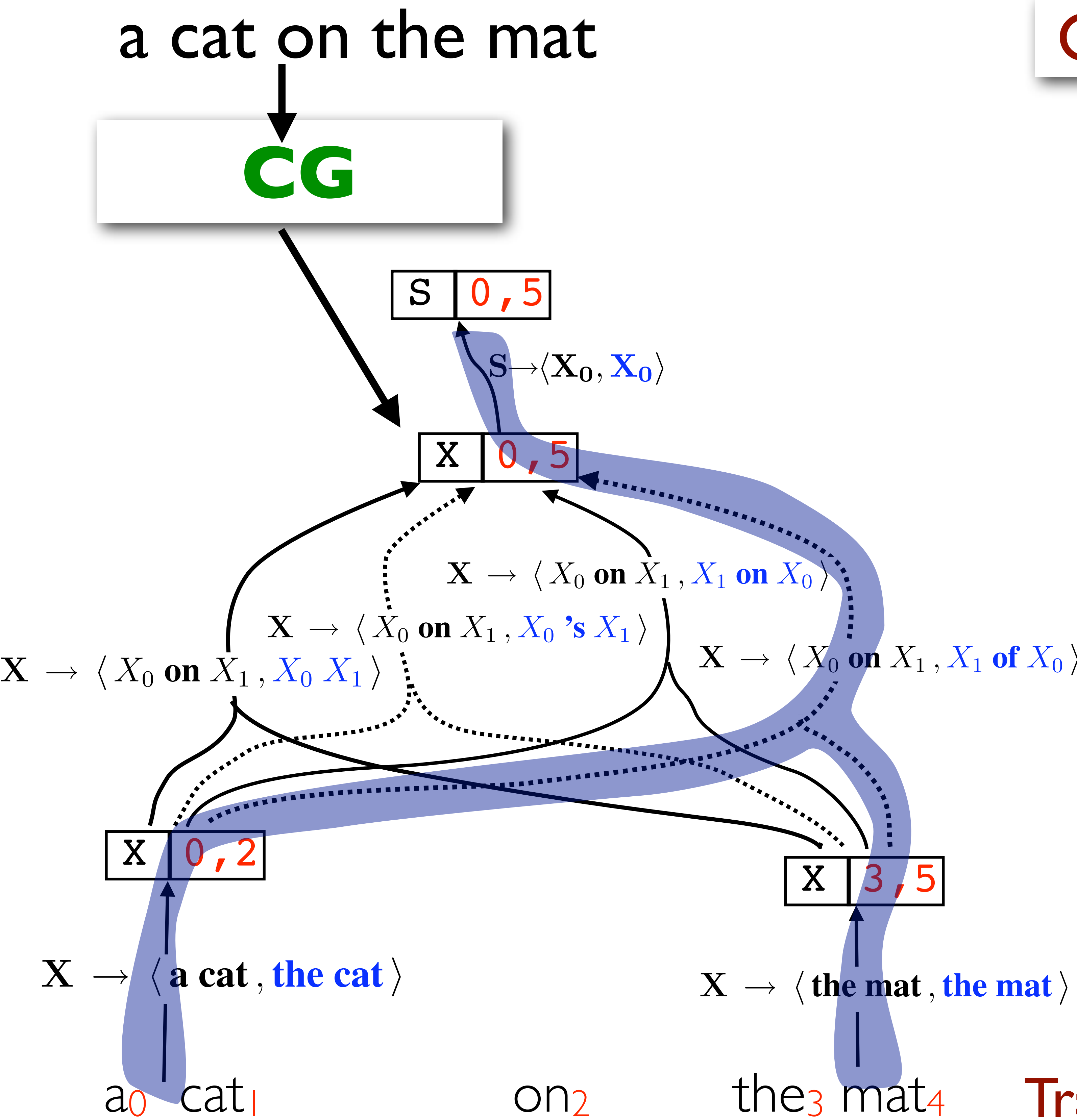


Contrastive set:

- the cat the mat
- the cat 's the mat
- the mat on the cat
- the mat of the cat



Step-2: Generating Contrastive Sets



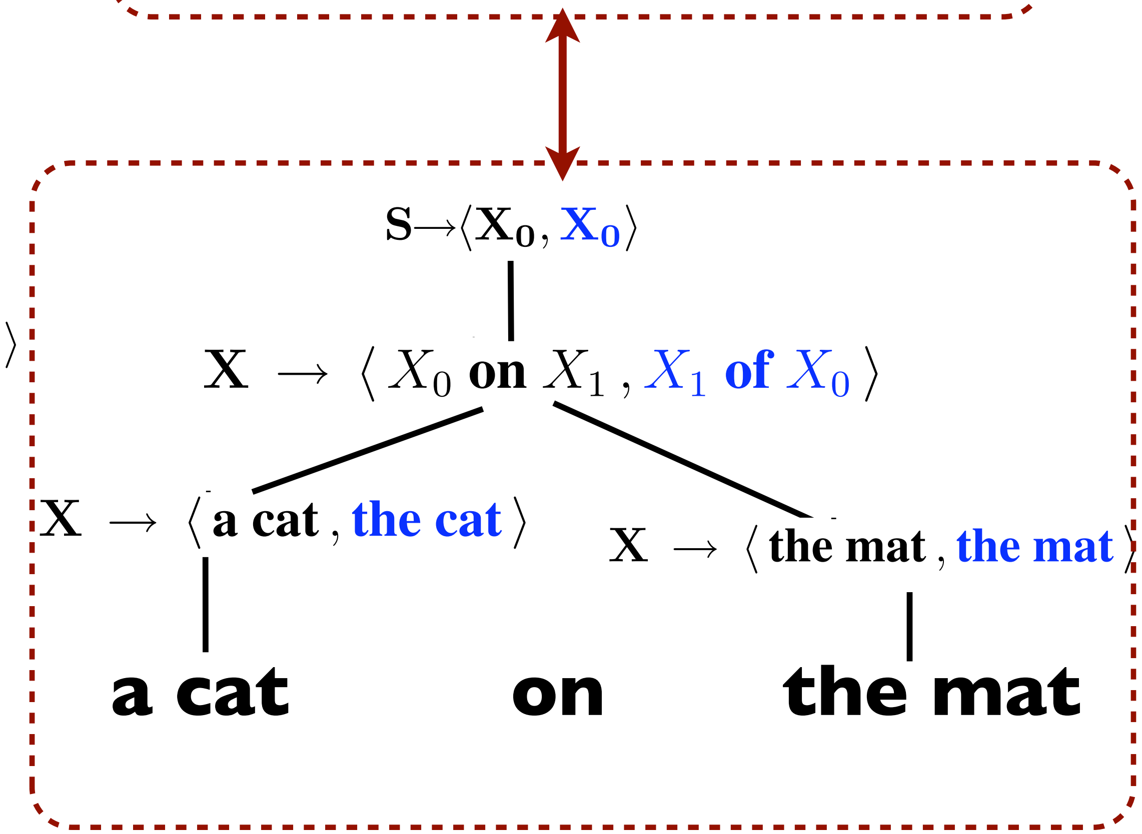
Contrastive set:

the cat the mat

the cat 's the mat

the mat on the cat

the mat of the cat



Translating "dianzi shang de mao"?

Step-3: Discriminative Training

Step-3: Discriminative Training

- Training Objective

$$\theta^* \ = \ \arg\min_{\theta} \sum_i \sum_{y \in \mathcal{N}(\tilde{y}_i)} \mathrm{L}(y, \tilde{y}_i) p_\theta(y \mid \tilde{y}_i)$$

Step-3: Discriminative Training

- Training Objective

$$\theta^* = \arg\min_\theta \sum_i \sum_{y \in \mathcal{N}(\tilde{y}_i)} \mathrm{L}(y, \tilde{y}_i) p_\theta(y \mid \tilde{y}_i)$$

contrastive set

Step-3: Discriminative Training

- Training Objective

expected loss

$$\theta^* = \arg\min_{\theta} \sum_i \sum_{y \in \mathcal{N}(\tilde{y}_i)} \mathrm{L}(y, \tilde{y}_i) p_\theta(y \mid \tilde{y}_i)$$

contrastive set

Step-3: Discriminative Training

- Training Objective

expected loss

$$\theta^* = \arg\min_\theta \sum_i \sum_{y \in \mathcal{N}(\tilde{y}_i)} \mathrm{L}(y, \tilde{y}_i) p_\theta(y \mid \tilde{y}_i)$$

contrastive set

CE maximizes the conditional likelihood

Step-3: Discriminative Training

- Training Objective

expected loss

$$\theta^* = \arg\min_{\theta} \sum_i \sum_{y \in \mathcal{N}(\tilde{y}_i)} \mathrm{L}(y, \tilde{y}_i) p_\theta(y \mid \tilde{y}_i)$$

contrastive set

CE maximizes the conditional likelihood

- Iterative Training
  - Step-2: for each English sentence, generate a contrastive set (or neighborhood) using the CG
  - Step-3: discriminative training

Applying the Contrastive Model

Applying the Contrastive Model

- We can use the contrastive model as a regular language model

Applying the Contrastive Model

- We can use the contrastive model as a regular language model

- We can incorporate the contrastive model into an end-to-end MT system as a feature

Applying the Contrastive Model

- We can use the contrastive model as a regular language model
- We can incorporate the contrastive model into an end-to-end MT system as a feature
- We may also use the contrastive model to generate paraphrase sentences
(if the loss function measures semantic similarity)
  - the rules in CG are symmetric

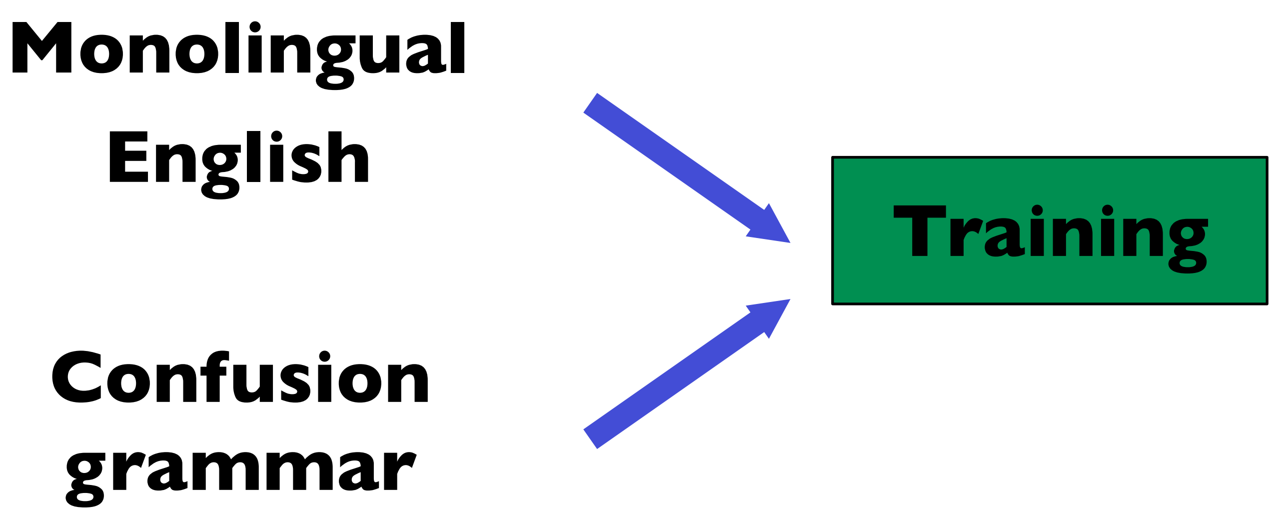
Test on Synthesized Hypergraphs of English Data

Test on Synthesized Hypergraphs of English Data

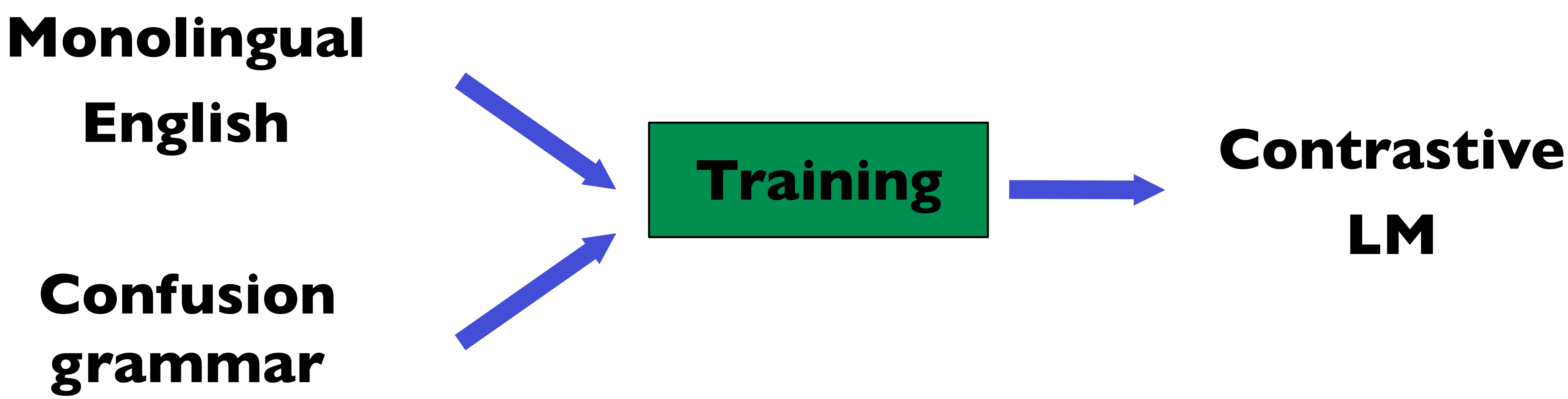
Monolingual English

Confusion grammar

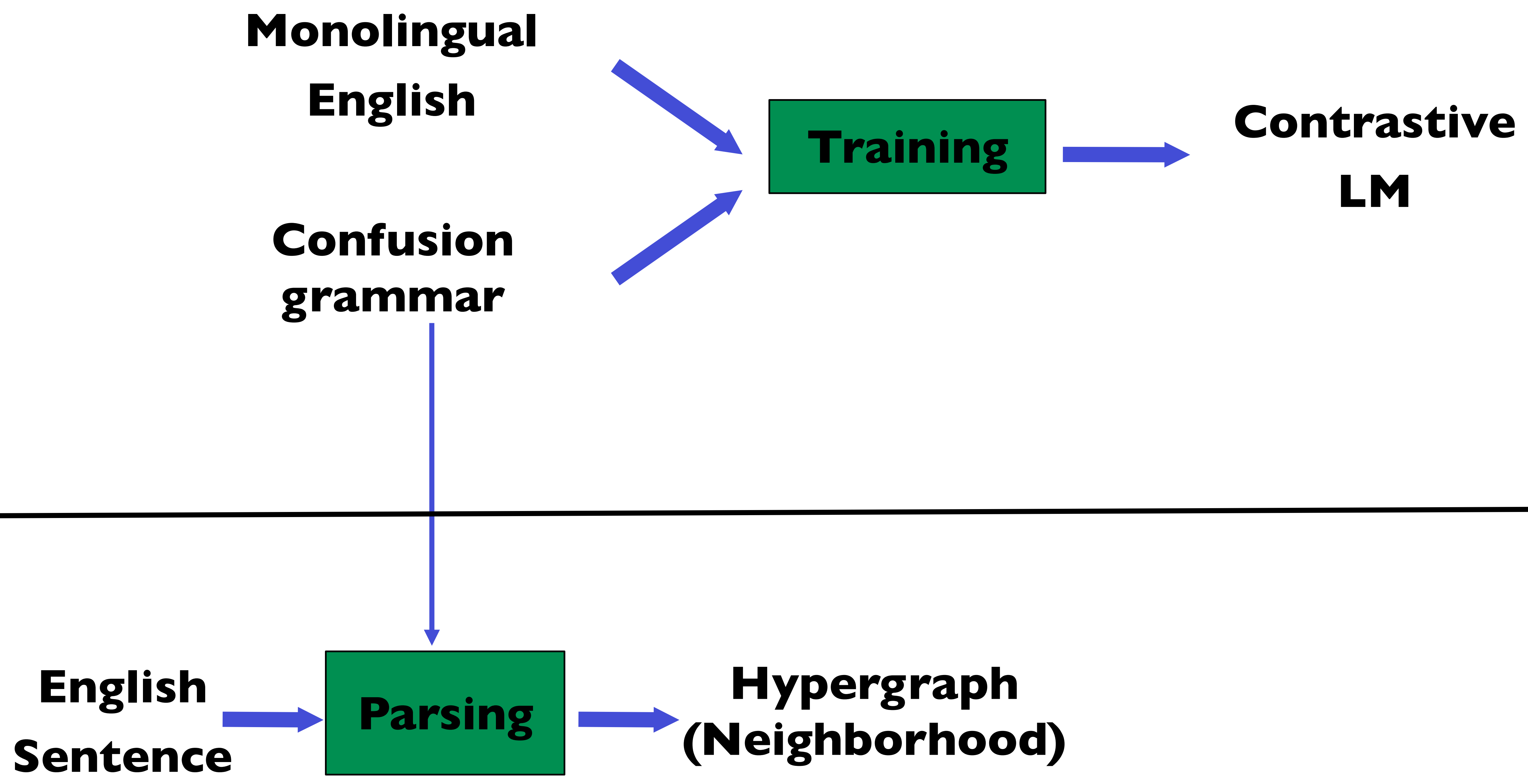
Test on Synthesized Hypergraphs of English Data



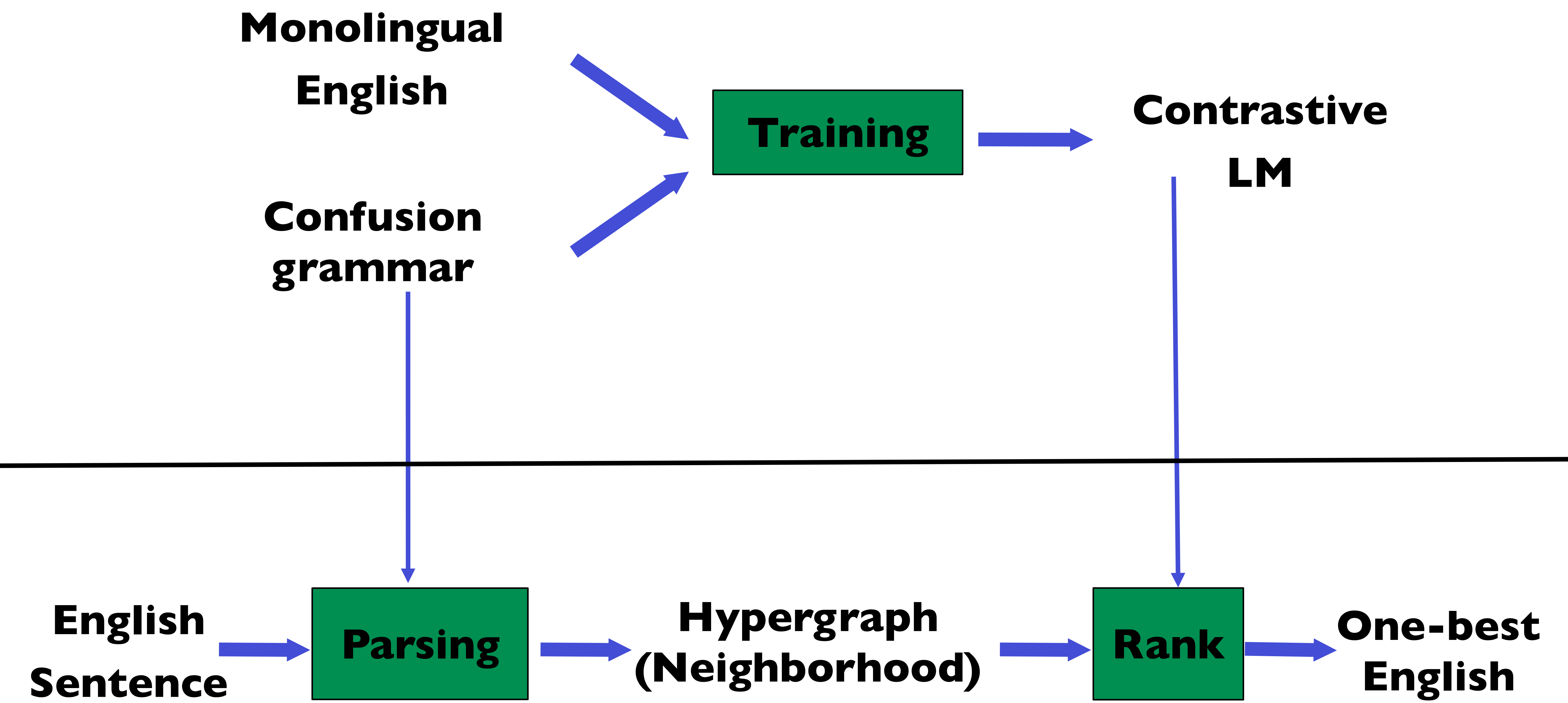
Test on Synthesized Hypergraphs of English Data



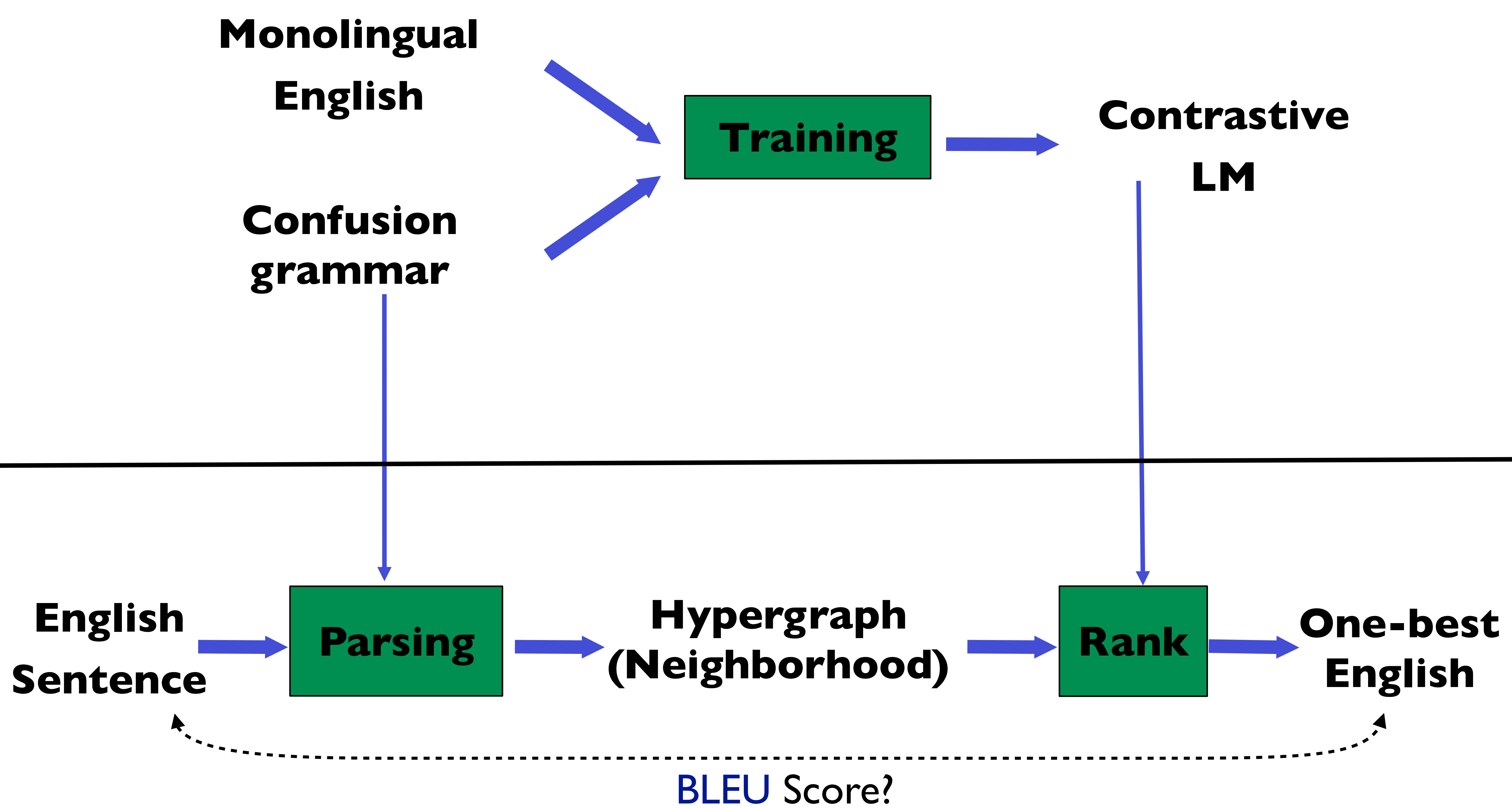
Test on Synthesized Hypergraphs of English Data



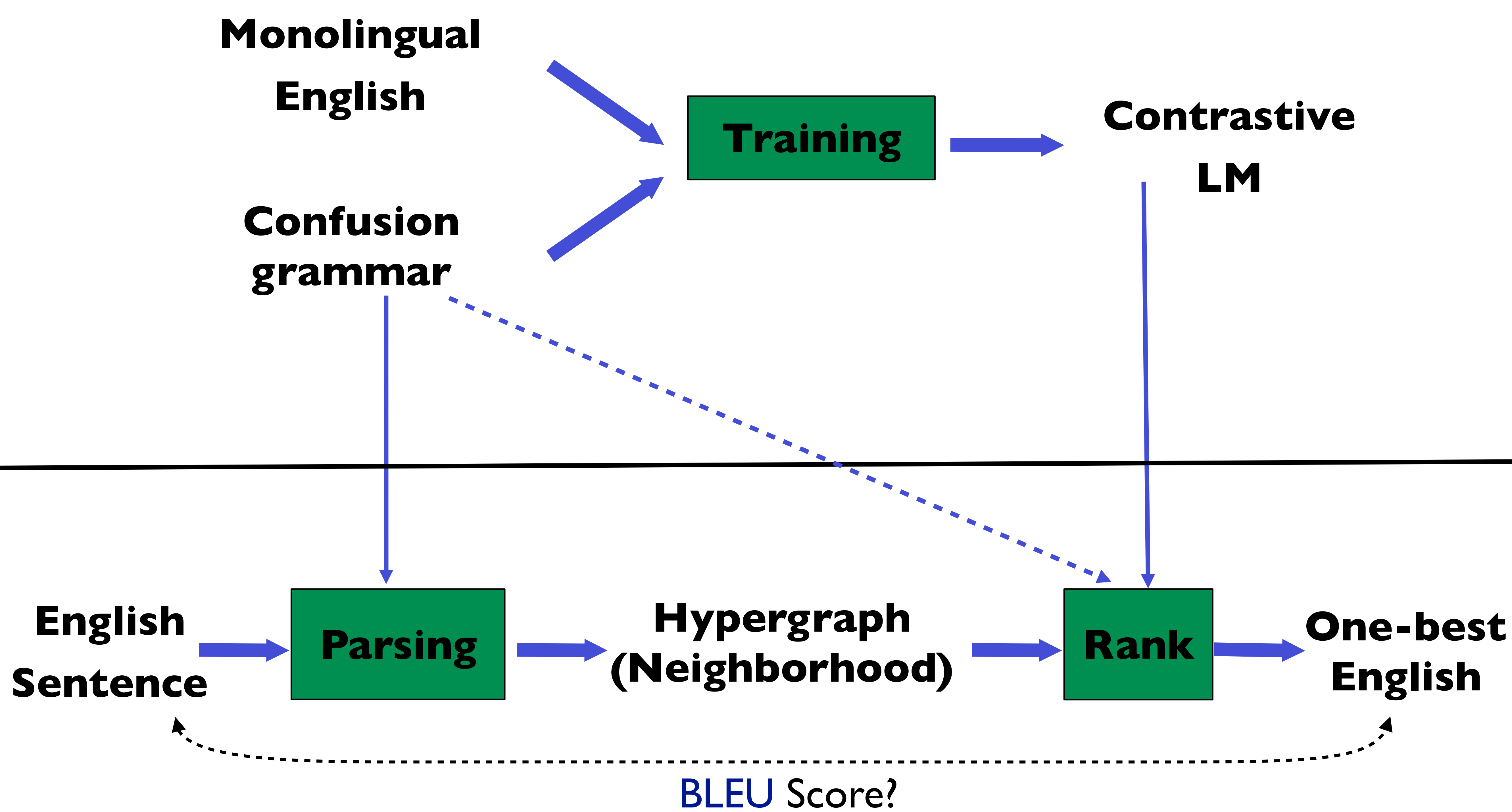
Test on Synthesized Hypergraphs of English Data



Test on Synthesized Hypergraphs of English Data

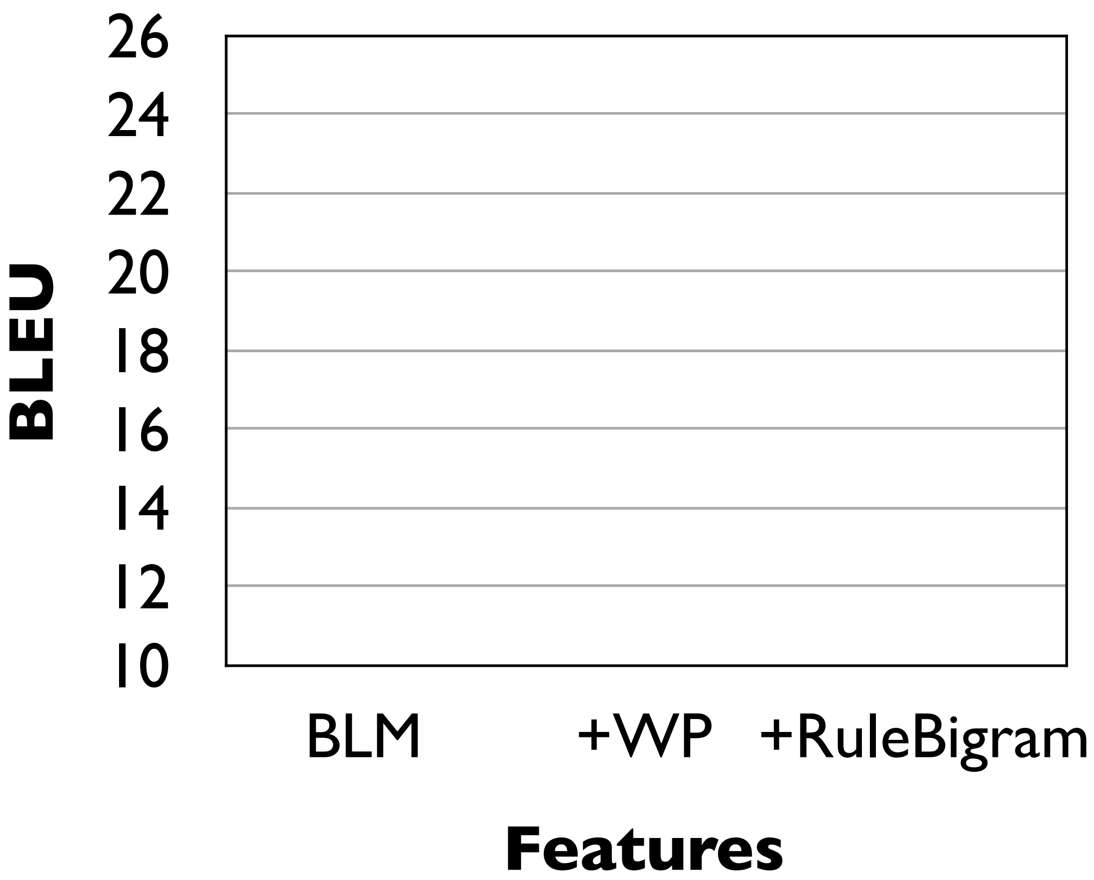


Test on Synthesized Hypergraphs of English Data

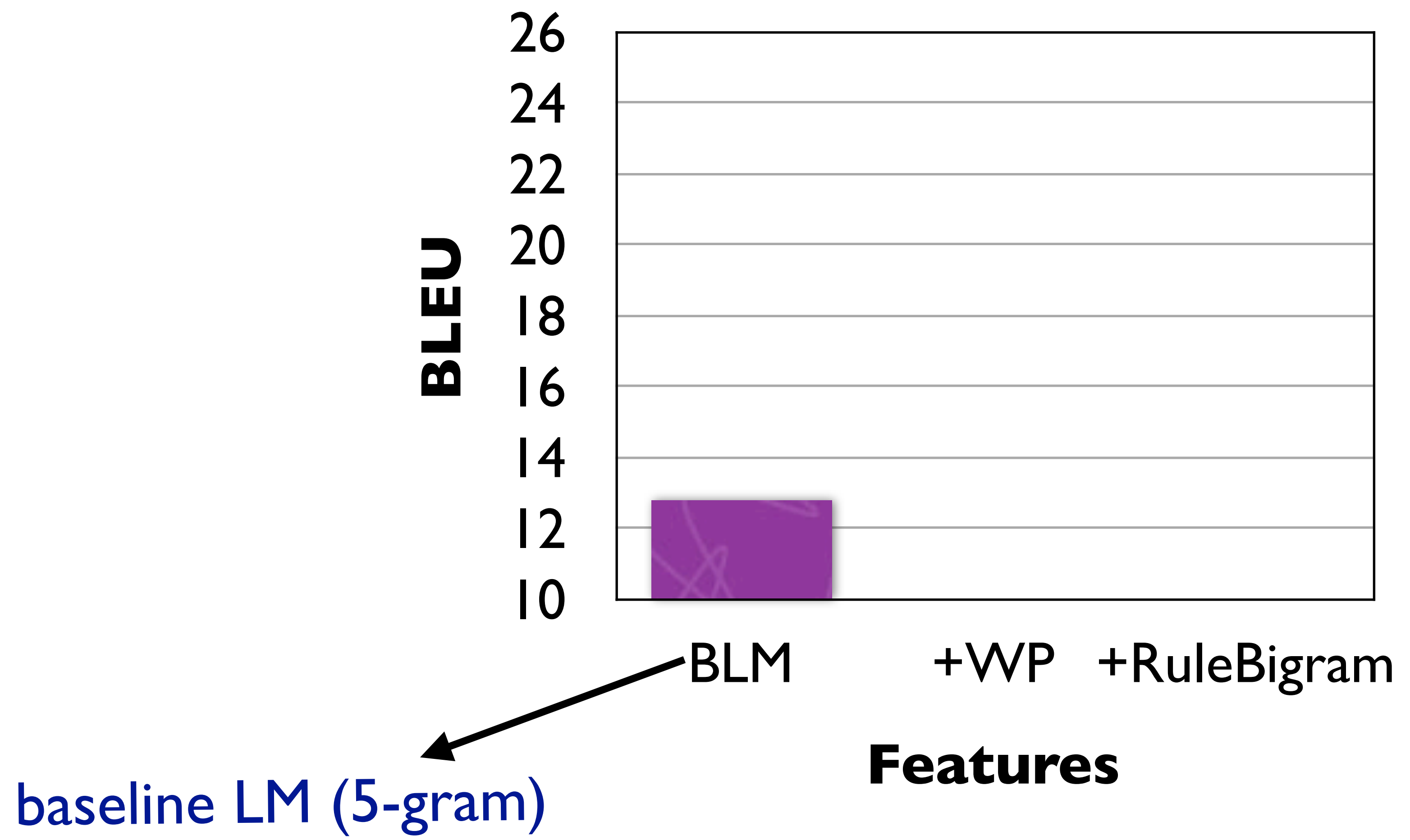


Results on Synthesized Hypergraphs

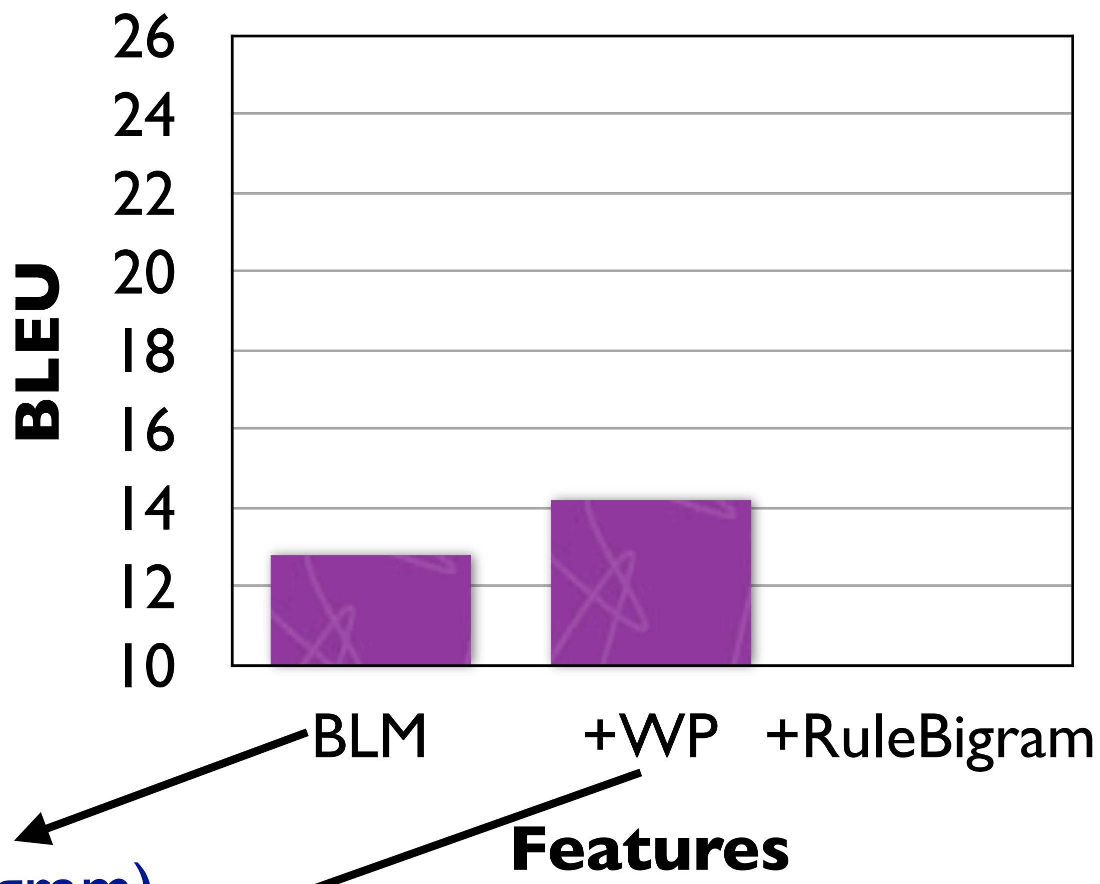
Results on Synthesized Hypergraphs



Results on Synthesized Hypergraphs



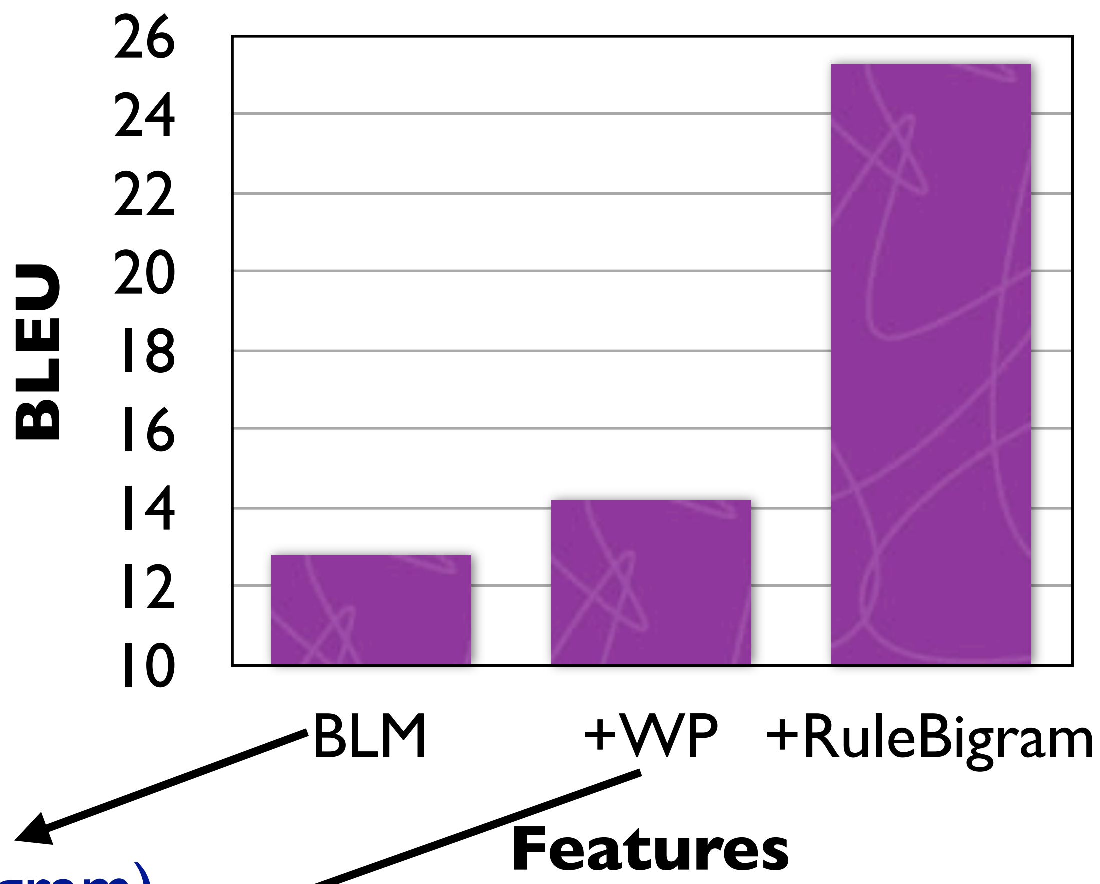
Results on Synthesized Hypergraphs



baseline LM (5-gram)

word penalty

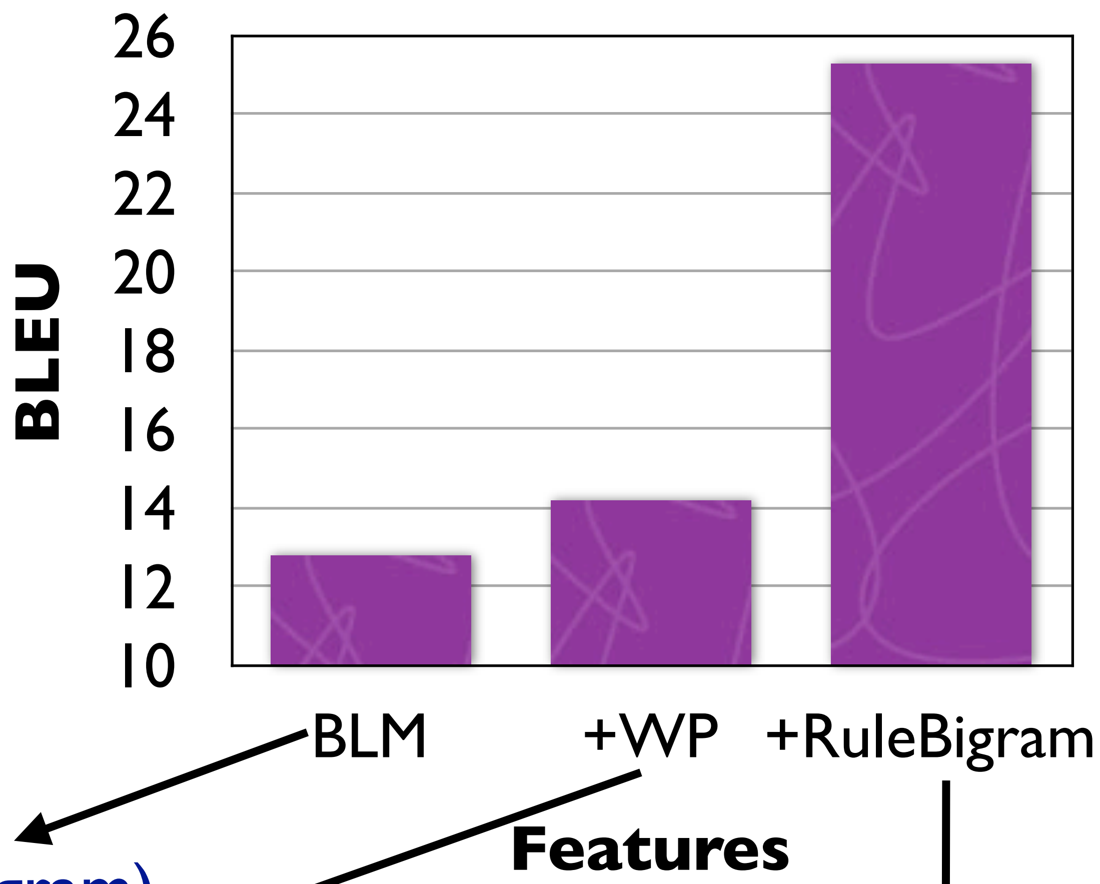
Results on Synthesized Hypergraphs



baseline LM (5-gram)

word penalty

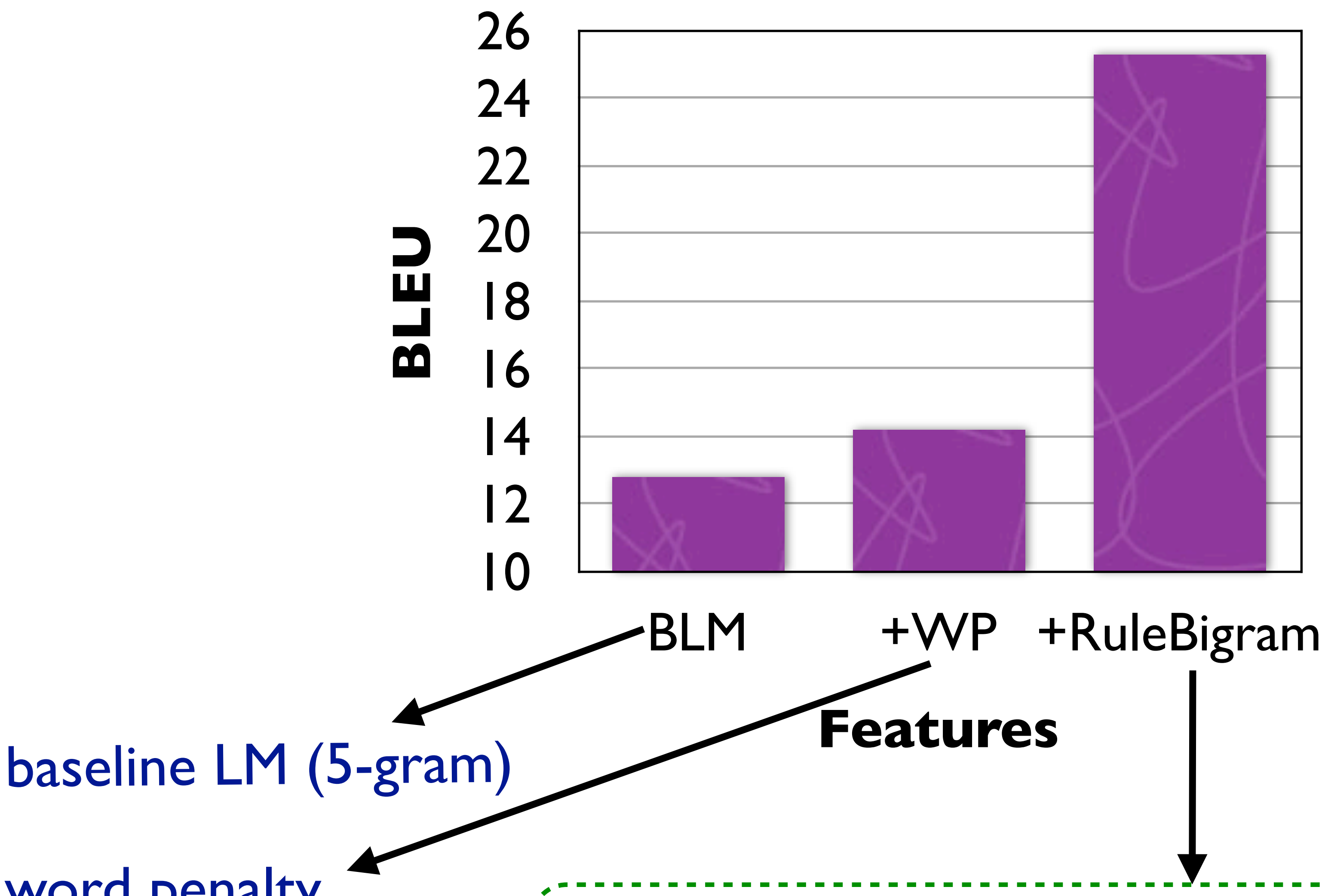
Results on Synthesized Hypergraphs



baseline LM (5-gram)

word penalty

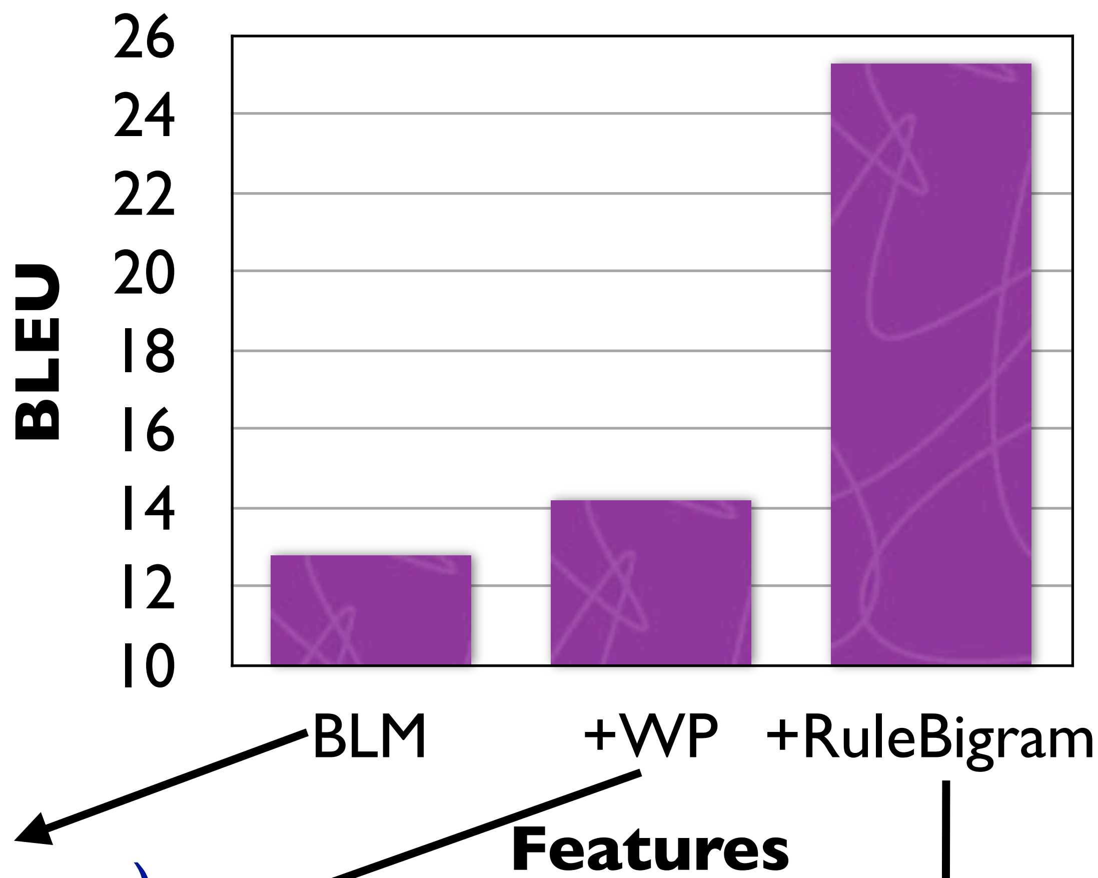
Results on Synthesized Hypergraphs



- Target side of a confusion rule

"*on the* X_1 *issue of* X_2"

Results on Synthesized Hypergraphs



baseline LM (5-gram)

word penalty

- Target side of a confusion rule

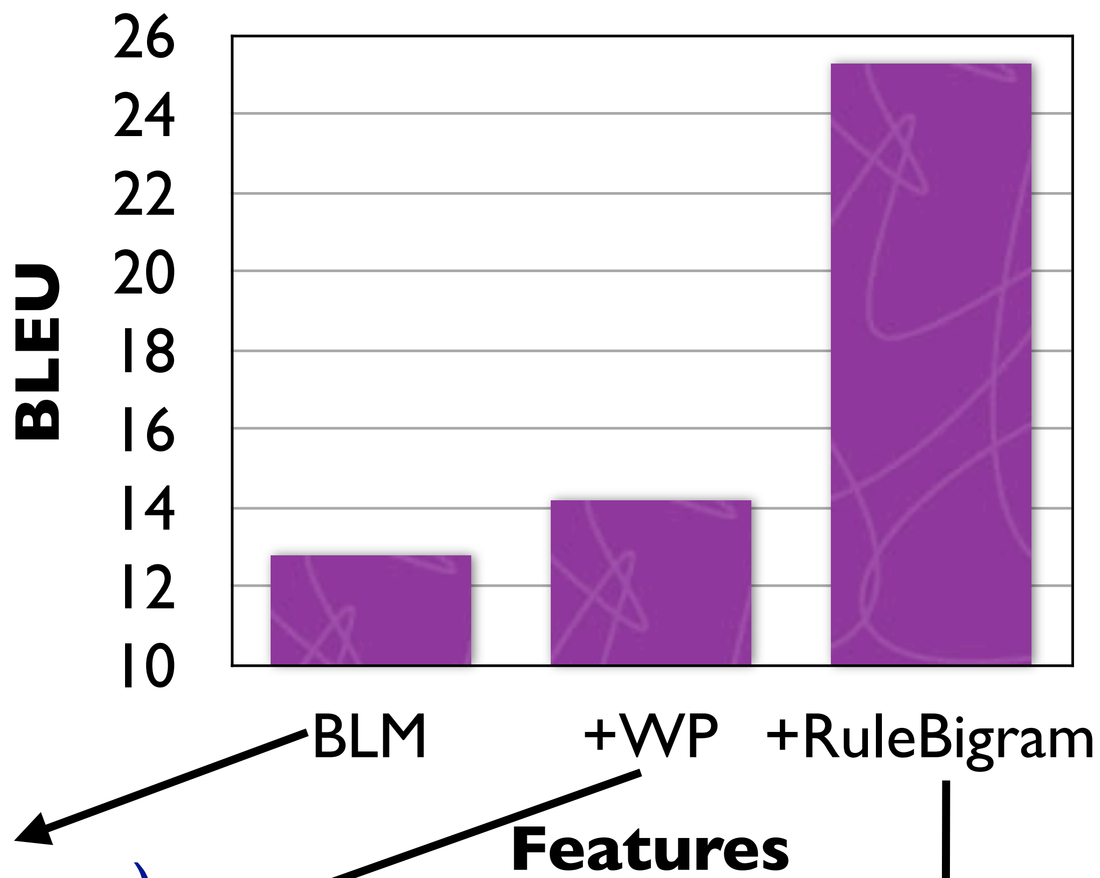
  "on the X_1 issue of X_2"

- Rule bigram features

  "on the" "the X" "X issue"

  "issue of" "of X"

Results on Synthesized Hypergraphs



The contrastive LM better **recovers** the original English than a regular n-gram LM.

baseline LM (5-gram)

word penalty

- Target side of a confusion rule

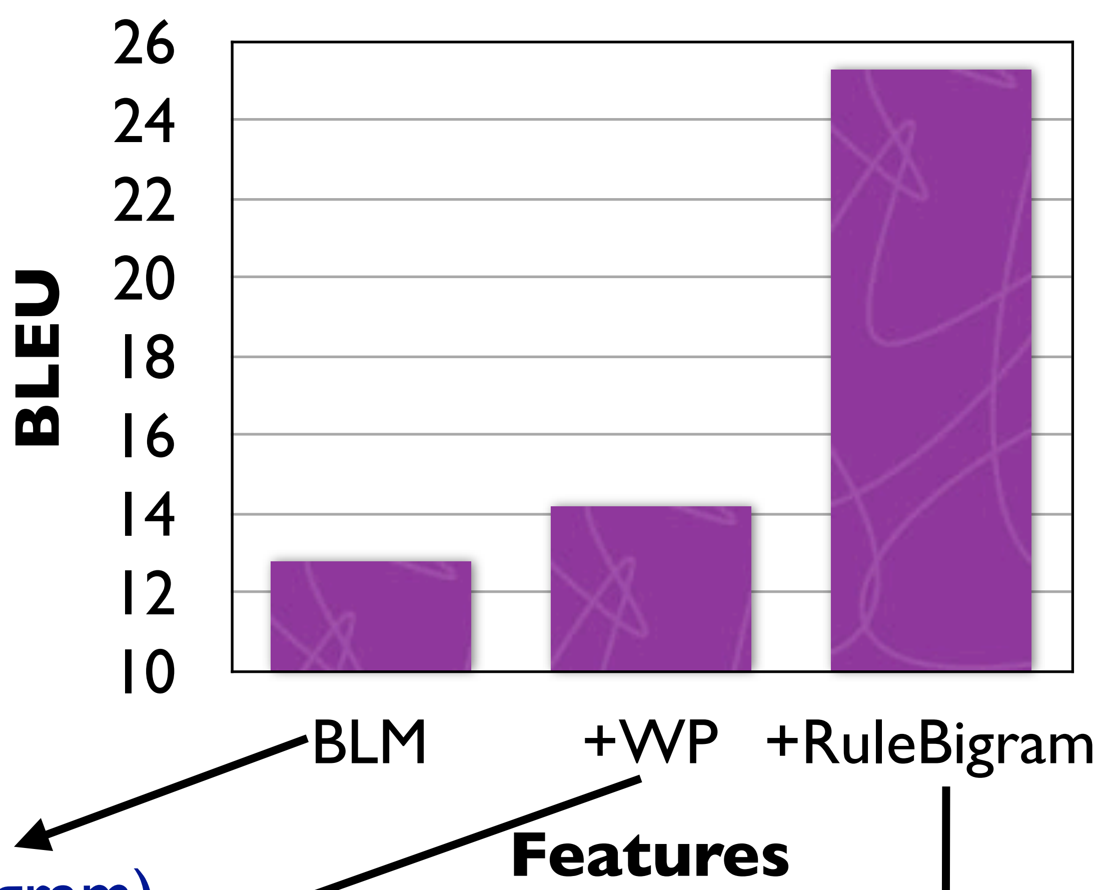
  “*on the* X_1 *issue of* X_2”

- Rule bigram features

  “*on the*” “*the X*” “*X issue*”

  “*issue of*” “*of X*”

Results on Synthesized Hypergraphs



baseline LM (5-gram)

word penalty

The contrastive LM better **recovers** the original English than a regular n-gram LM.

All the features look at only the target sides of confusion rules

- Target side of a confusion rule

  "*on the* X_1 *issue of* X_2"

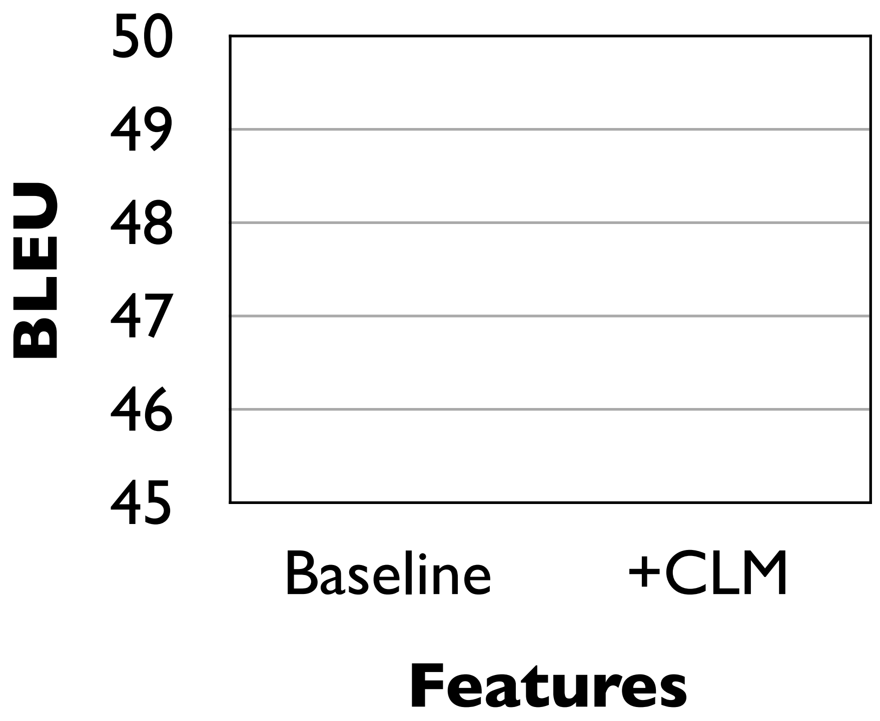
- Rule bigram features

  "*on the*" "*the X*" "*X issue*"

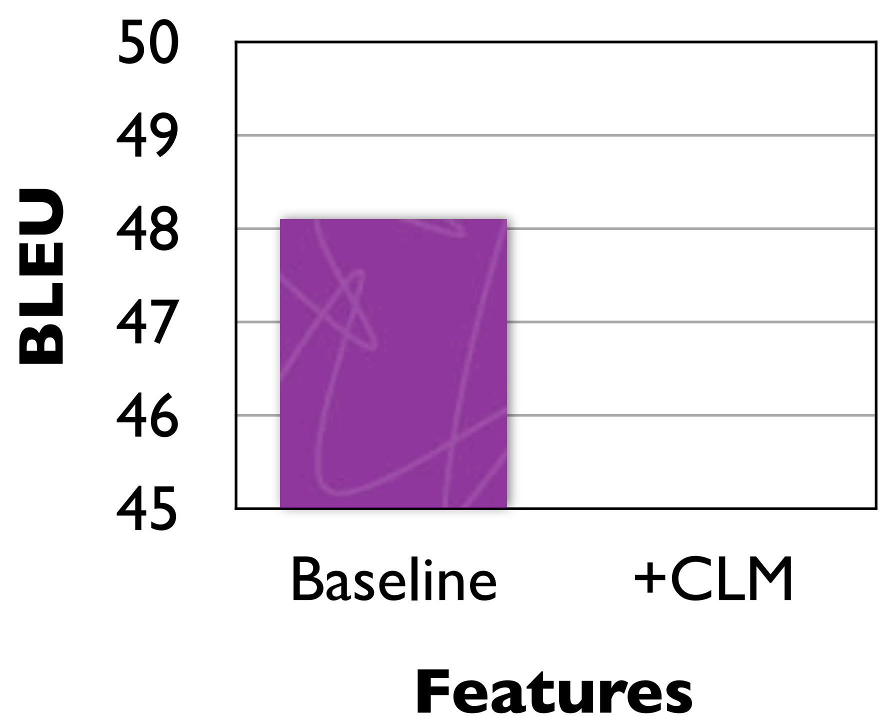
  "*issue of*" "*of X*"

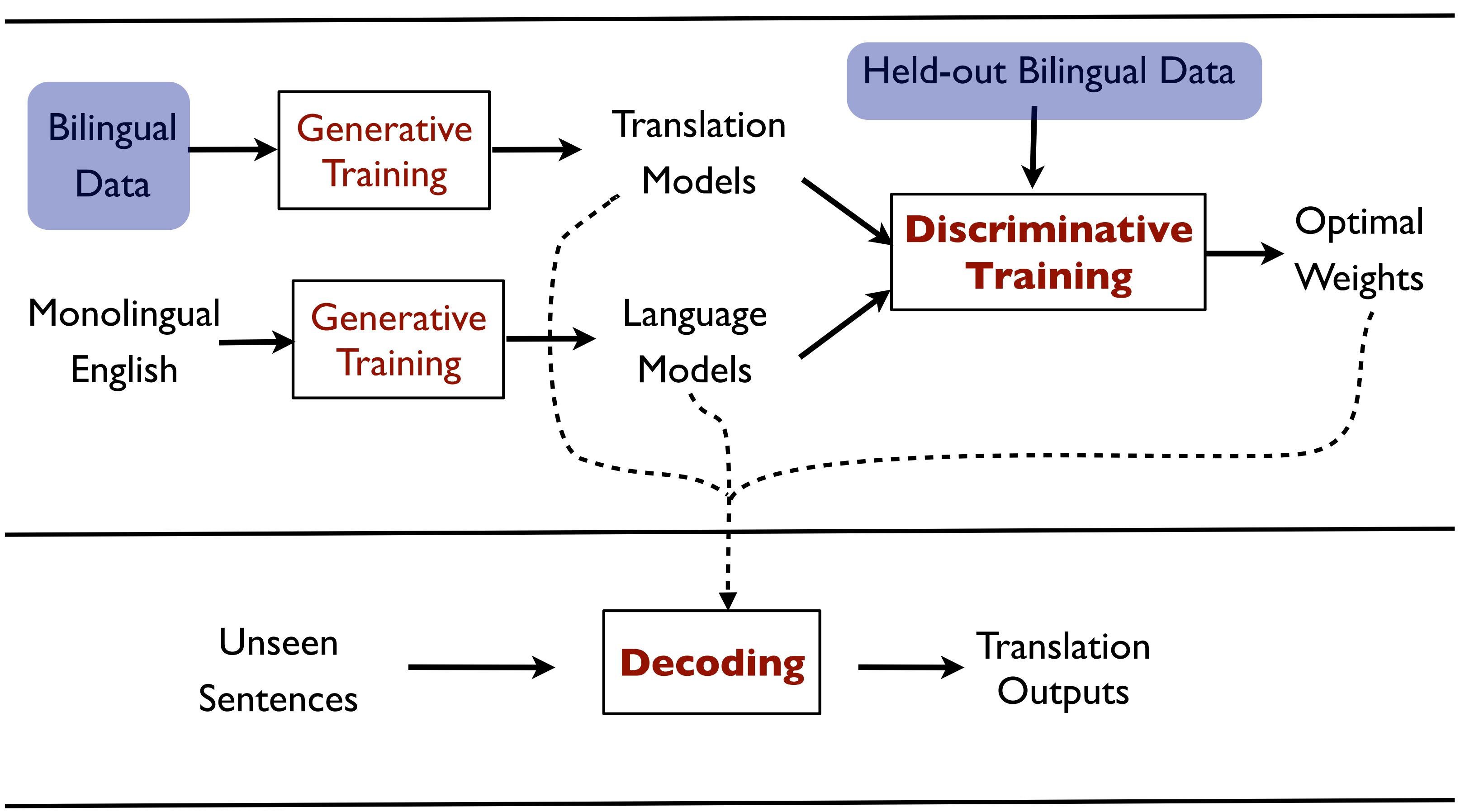
Results on MT Test Set

Results on MT Test Set

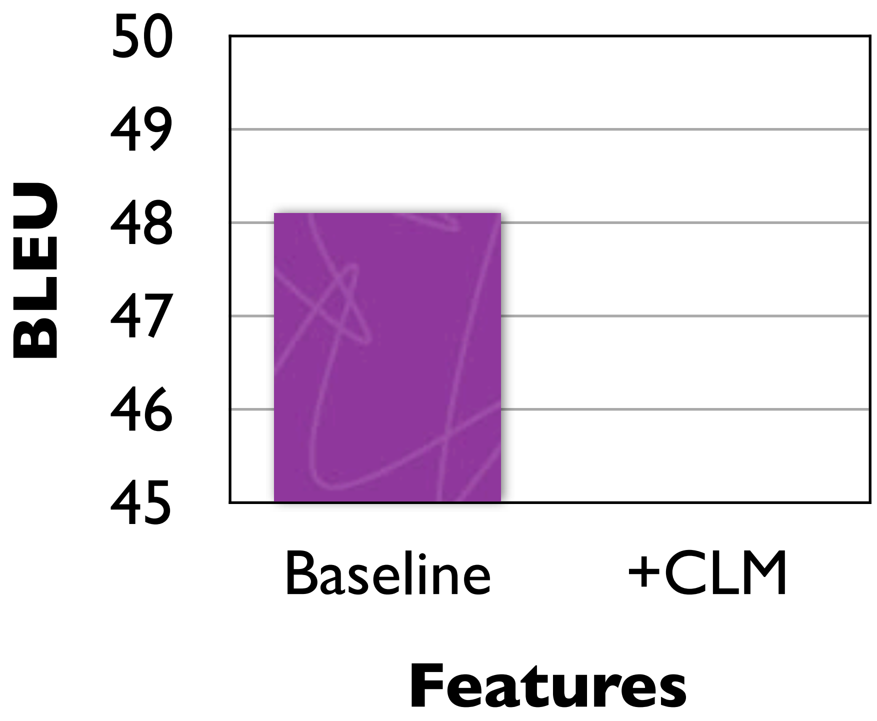


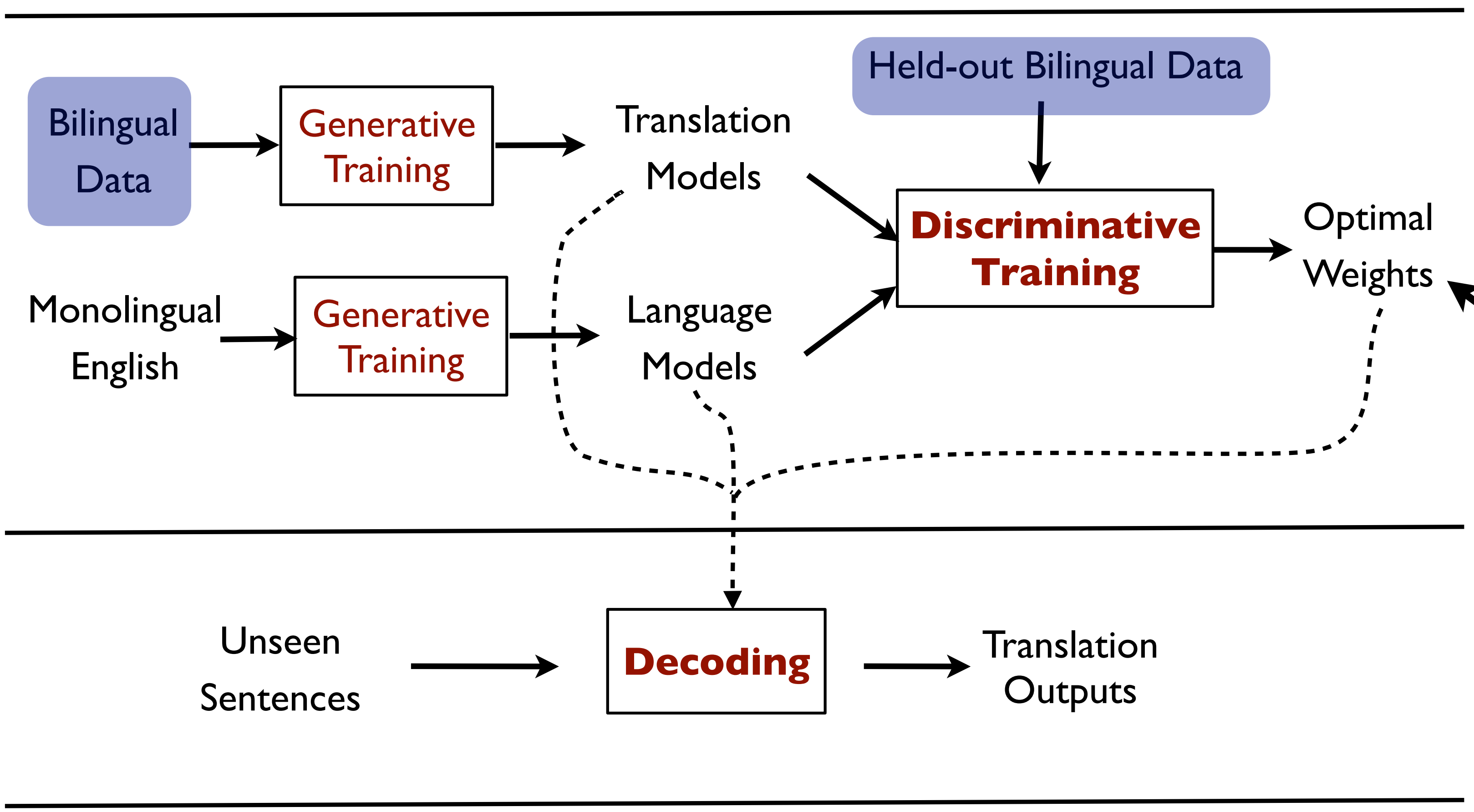
Results on MT Test Set





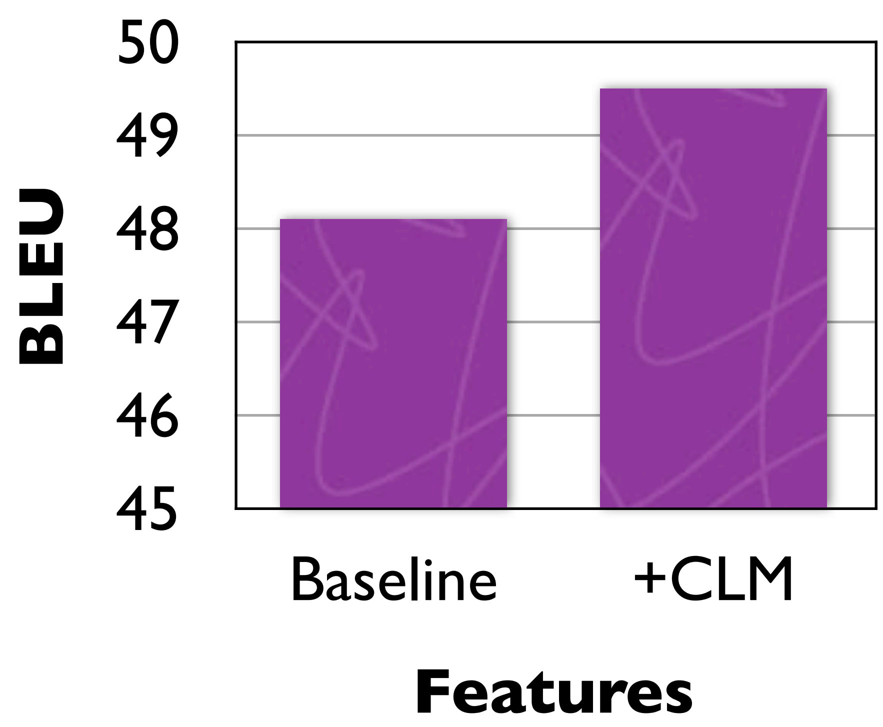
Results on MT Test Set



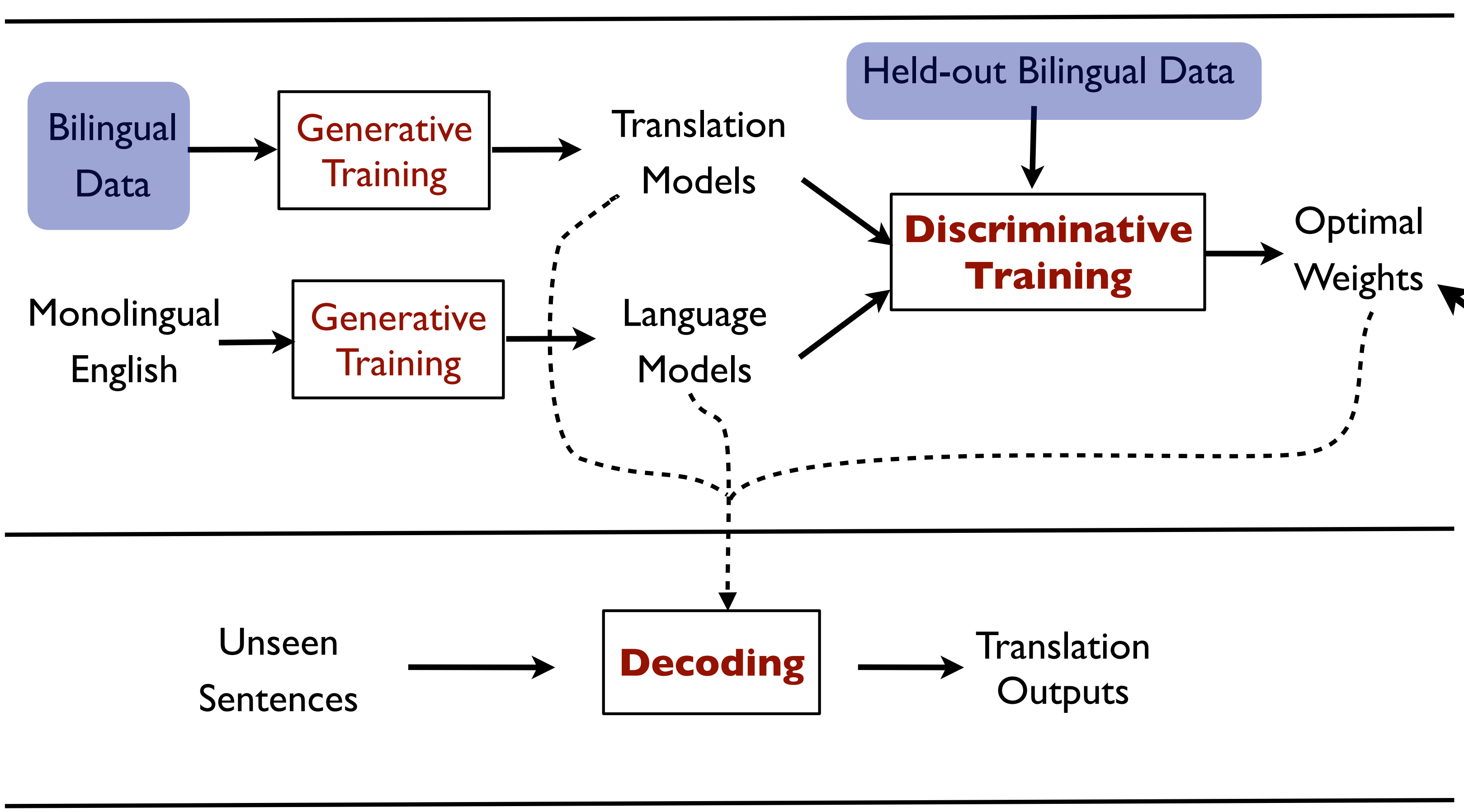


Add CLM as a feature

Results on MT Test Set



The contrastive LM helps to improve MT performance.



Add CLM as a feature

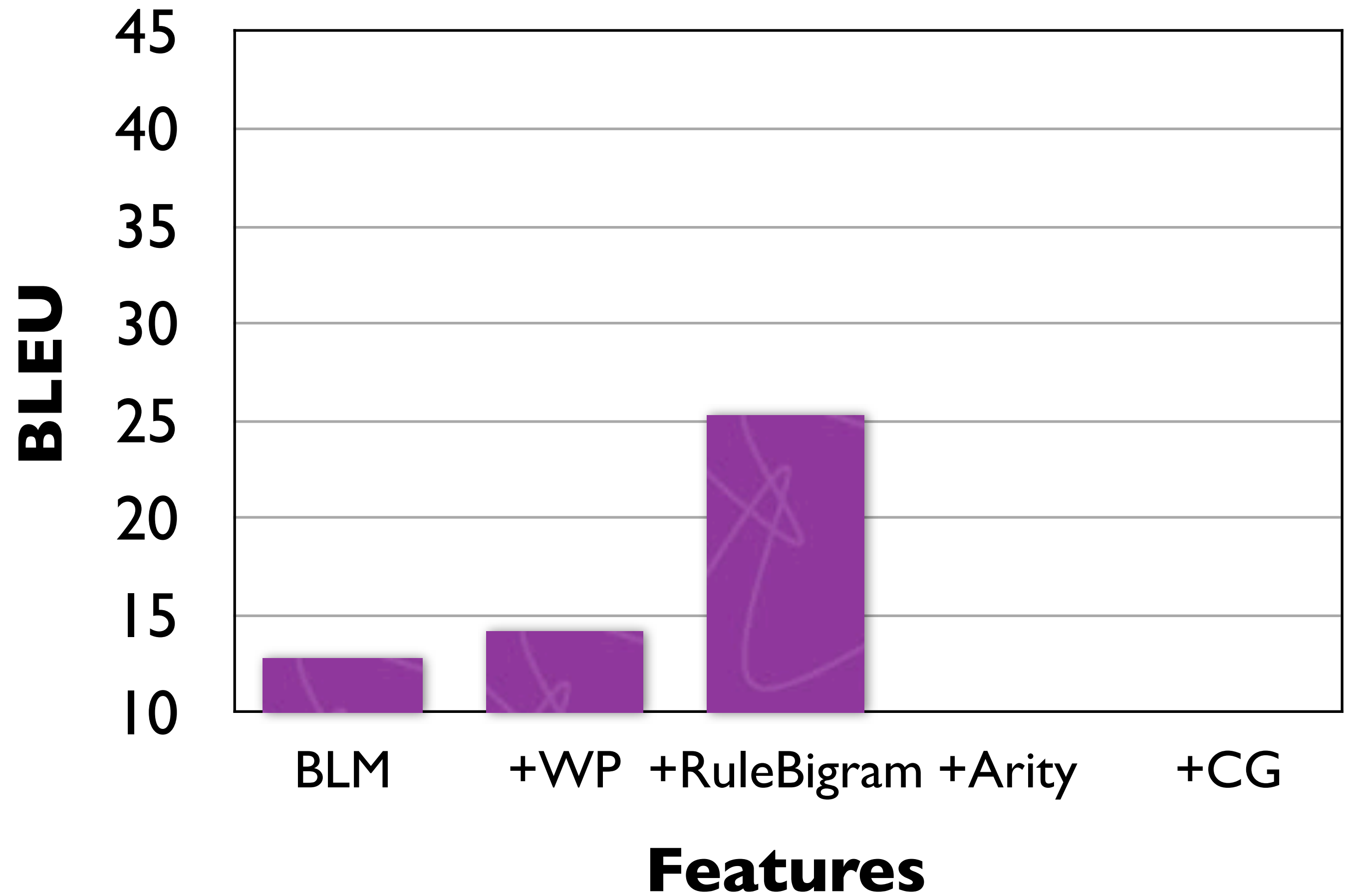
Adding Features on the CG itself

- On English Set

- On MT Set

Adding Features on the CG itself

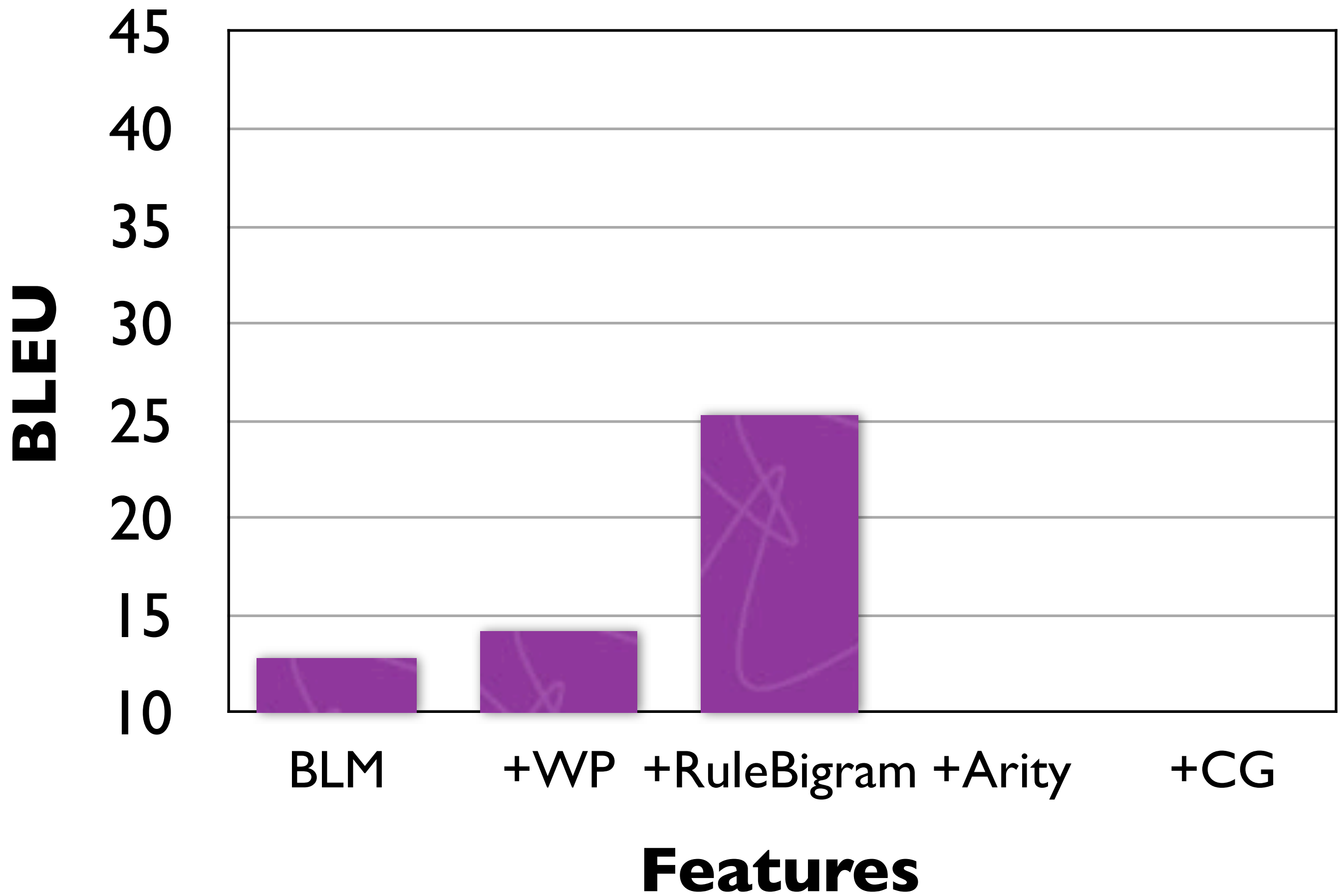
- On English Set



- On MT Set

Adding Features on the CG itself

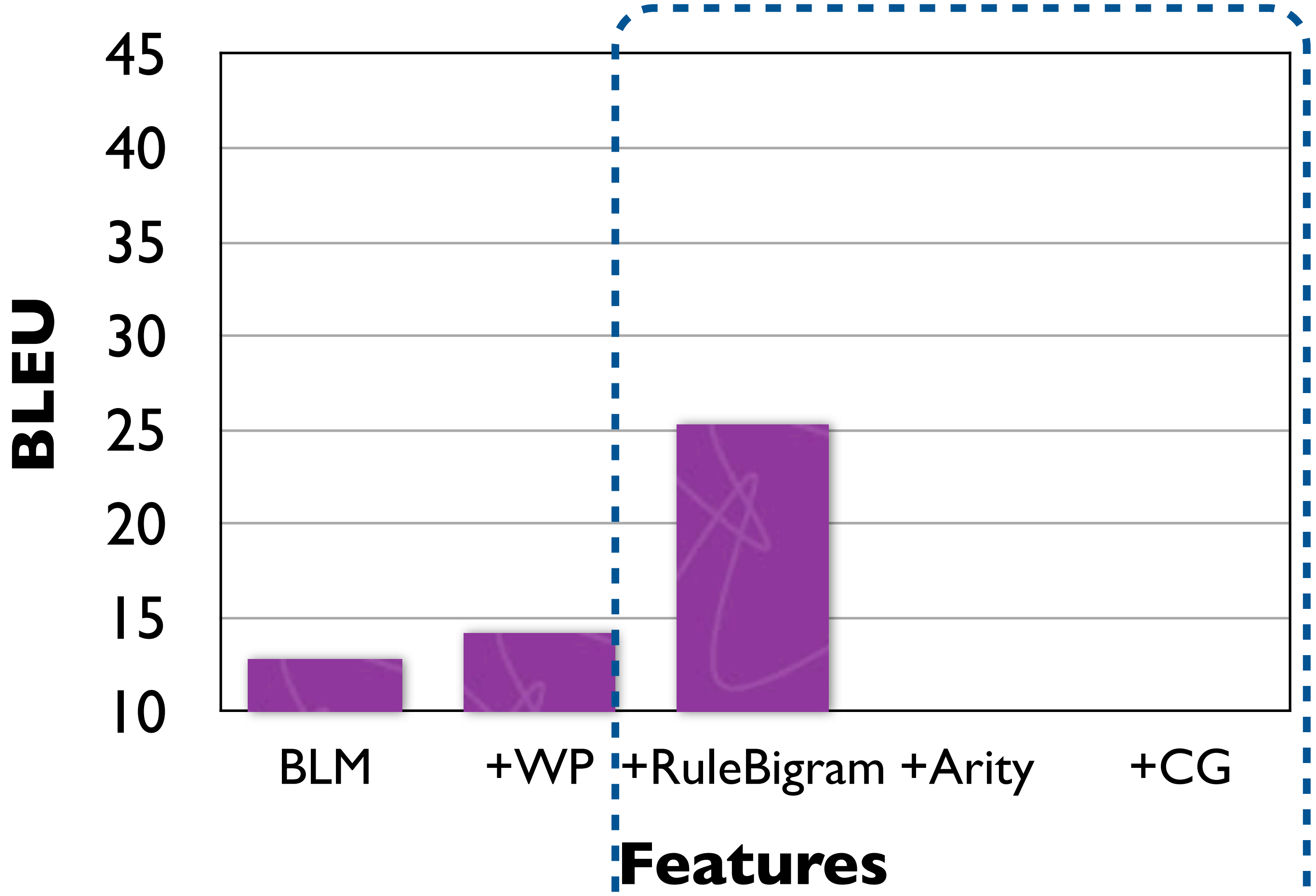
- On English Set



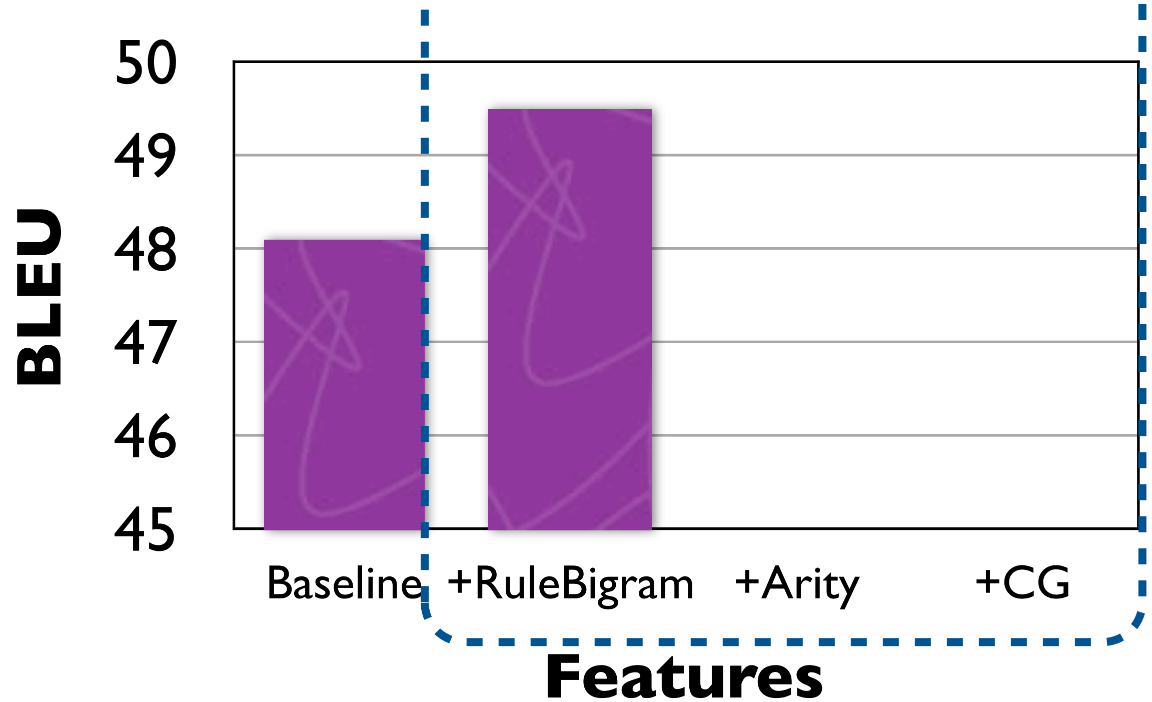
- On MT Set

Adding Features on the CG itself

- On English Set



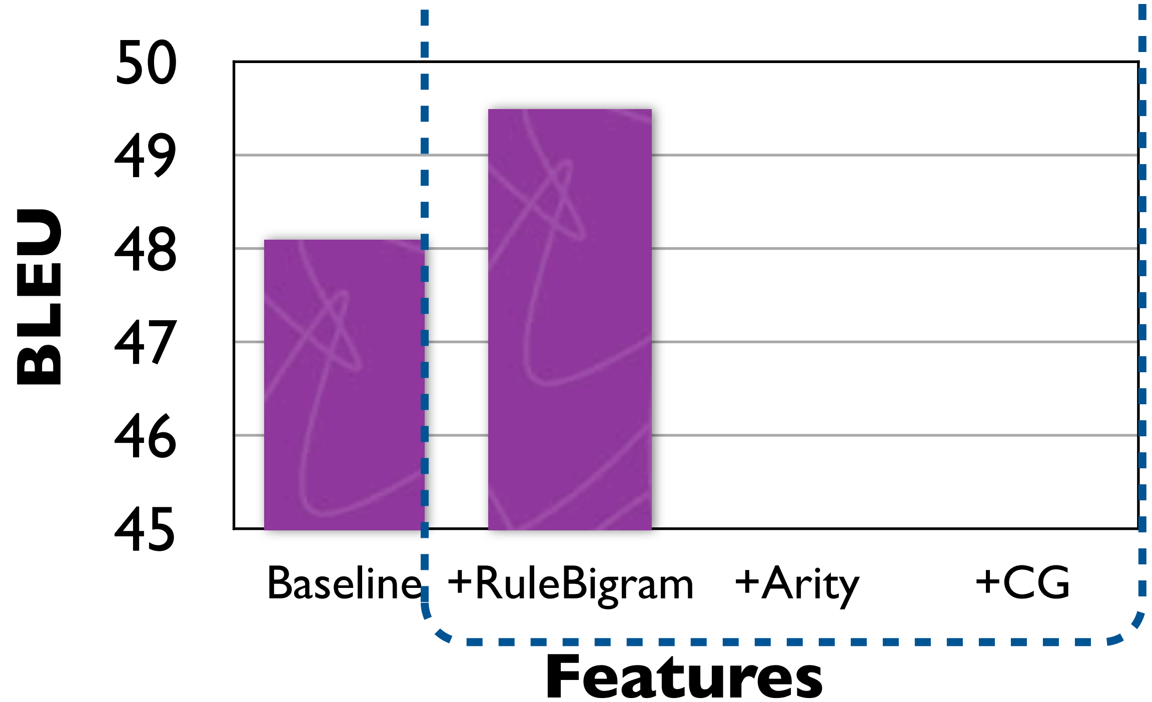
- On MT Set



Adding Features on the CG itself

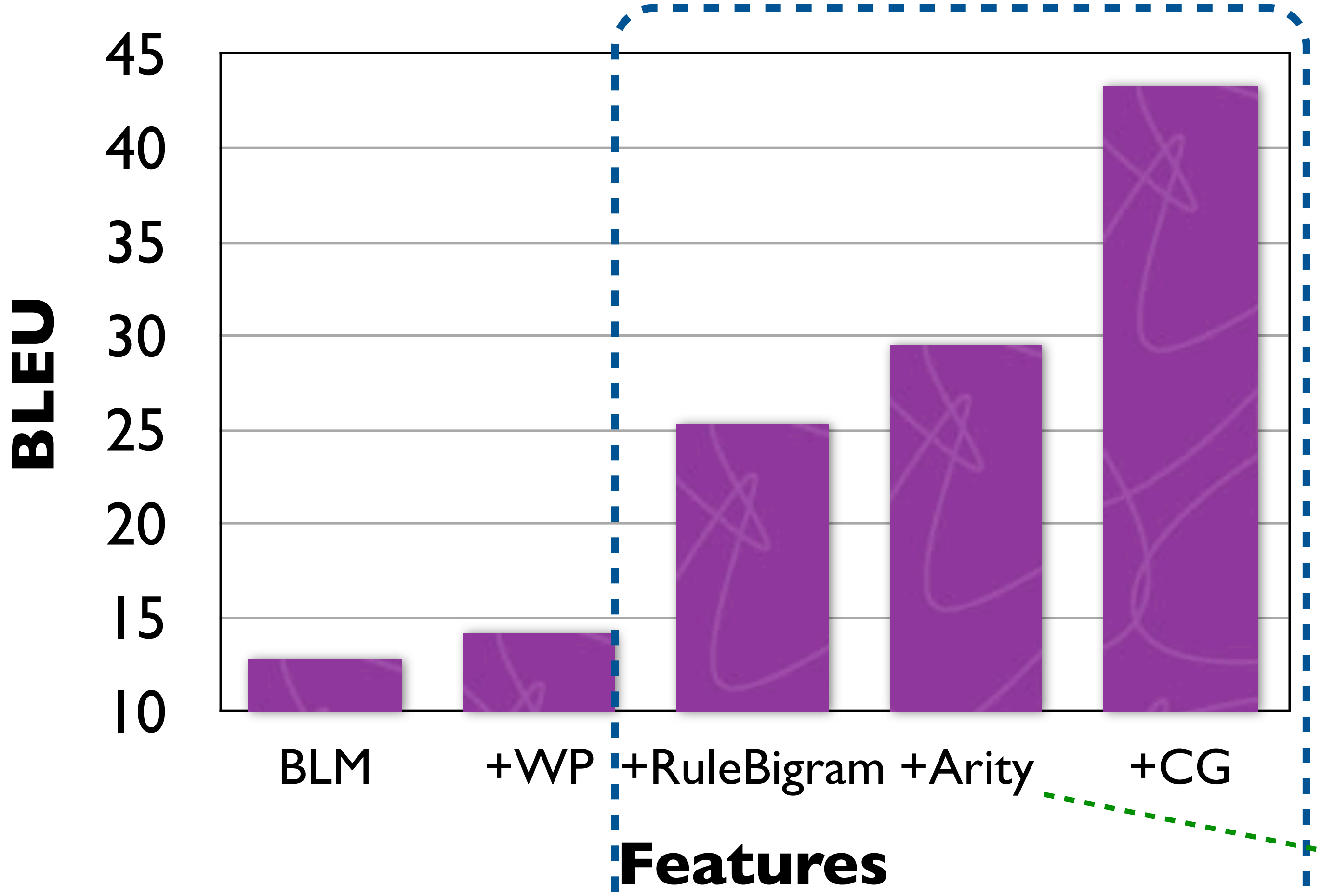
- On English Set

- On MT Set



Adding Features on the CG itself

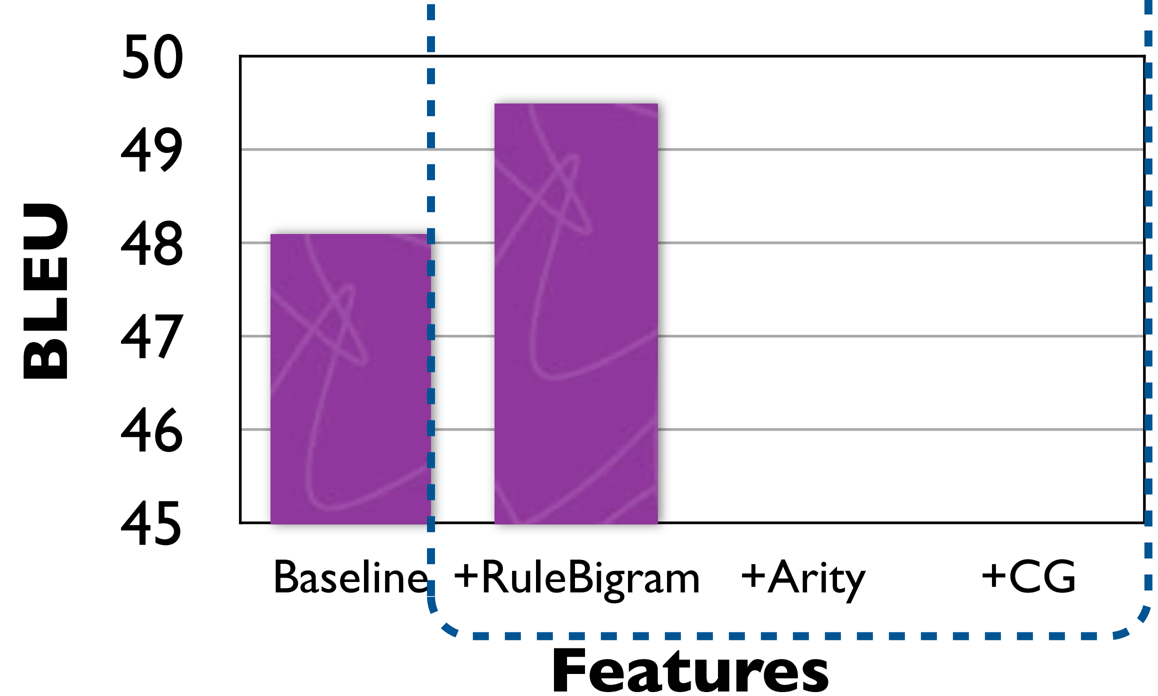
- On English Set



glue rules or regular confusion rules?

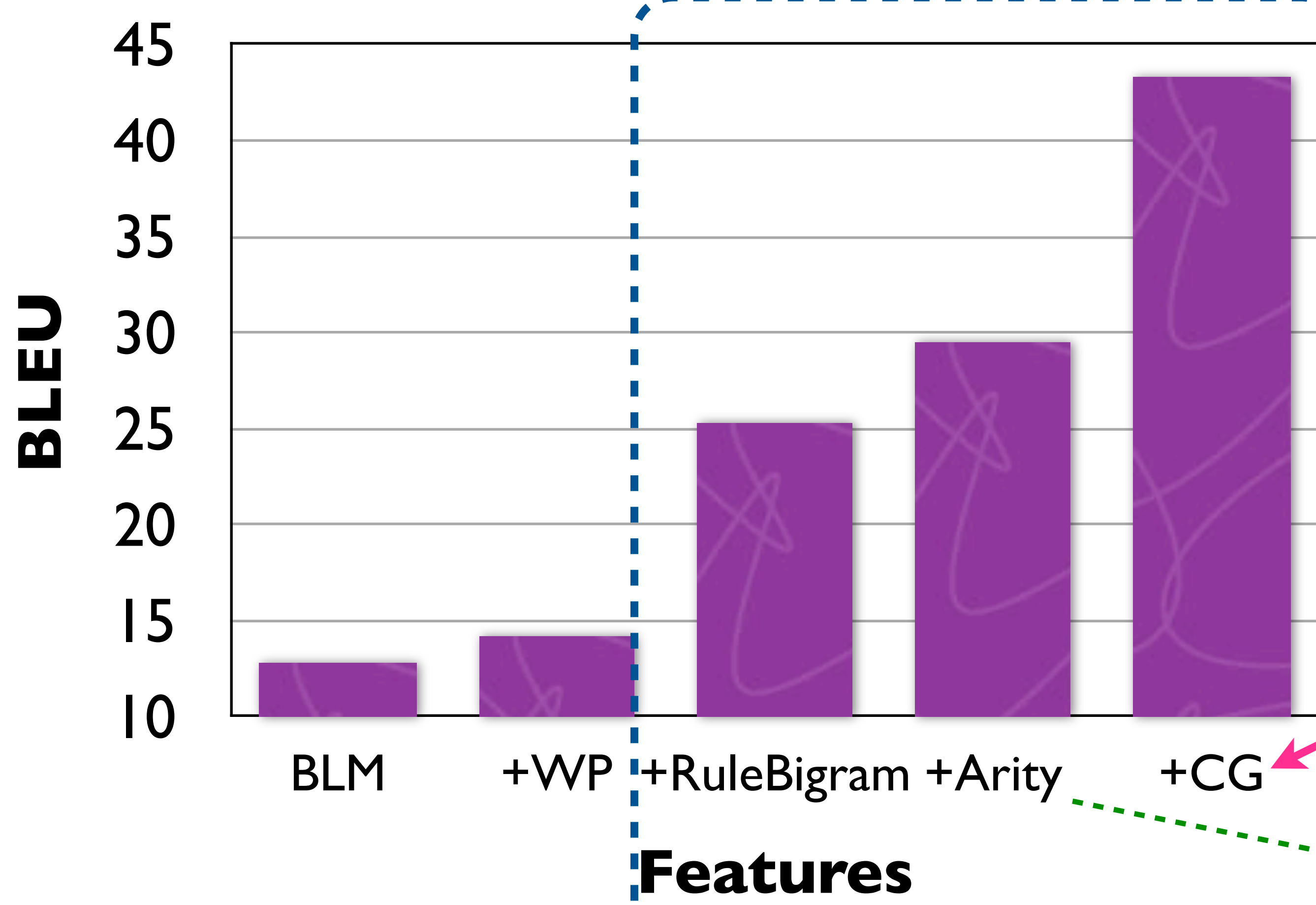
$$S \rightarrow \langle S_0\ X_1\ ,\ S_0\ X_1 \rangle$$
$$S \rightarrow \langle X_0\ ,\ X_0 \rangle$$

- On MT Set



Adding Features on the CG itself

- On English Set

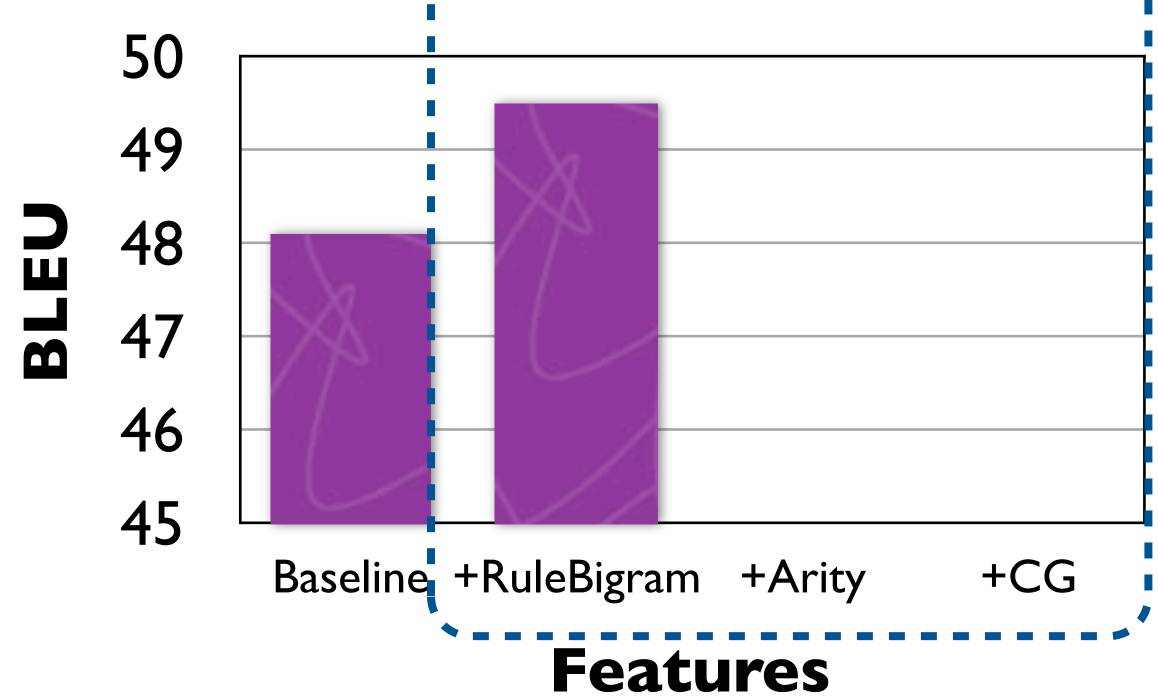


one big feature

glue rules or regular confusion rules?

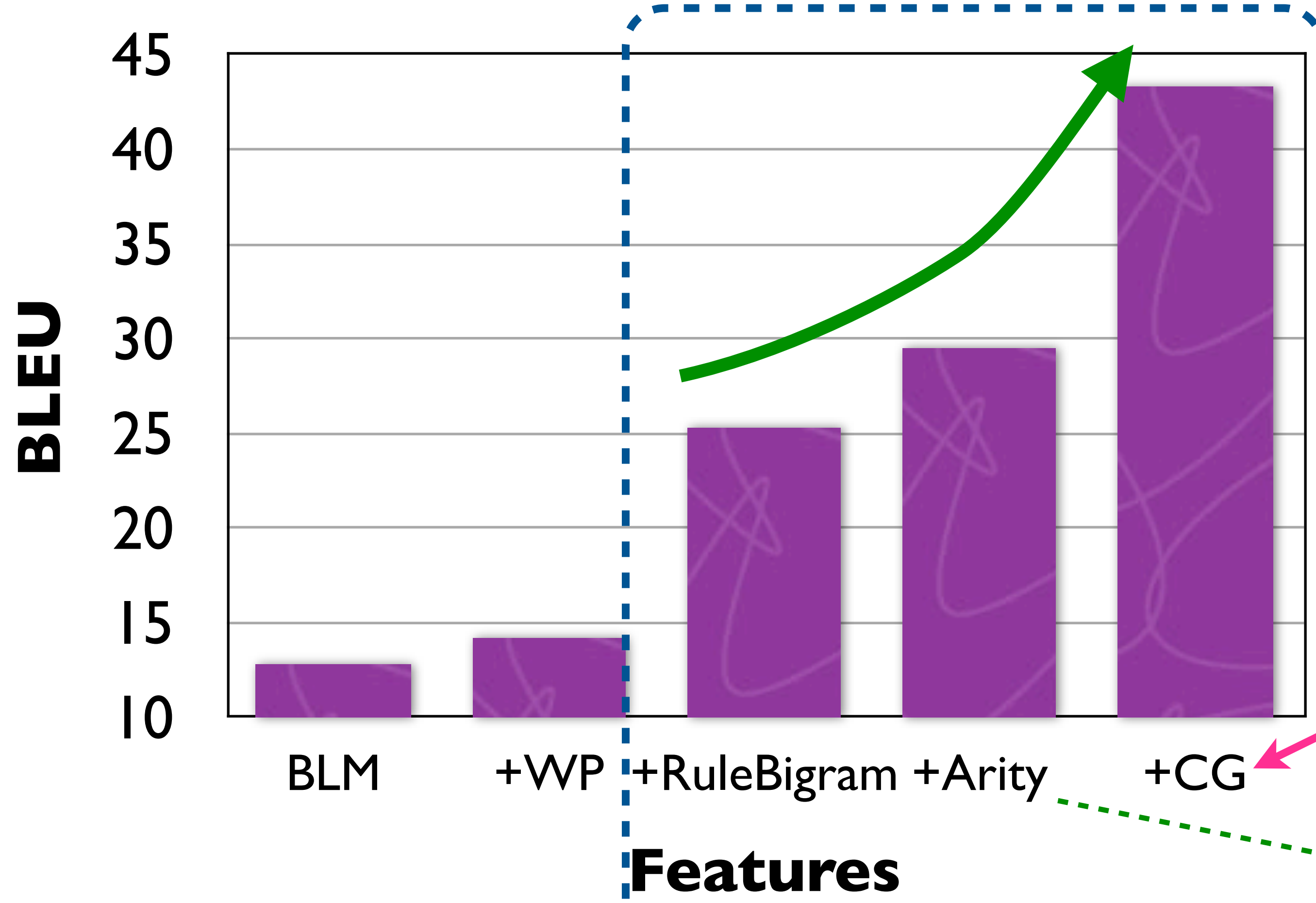
$$S \rightarrow \langle S_0 \, X_1 \, , S_0 \, X_1 \rangle$$
$$S \rightarrow \langle X_0 \, , X_0 \rangle$$

- On MT Set



Adding Features on the CG itself

- On English Set

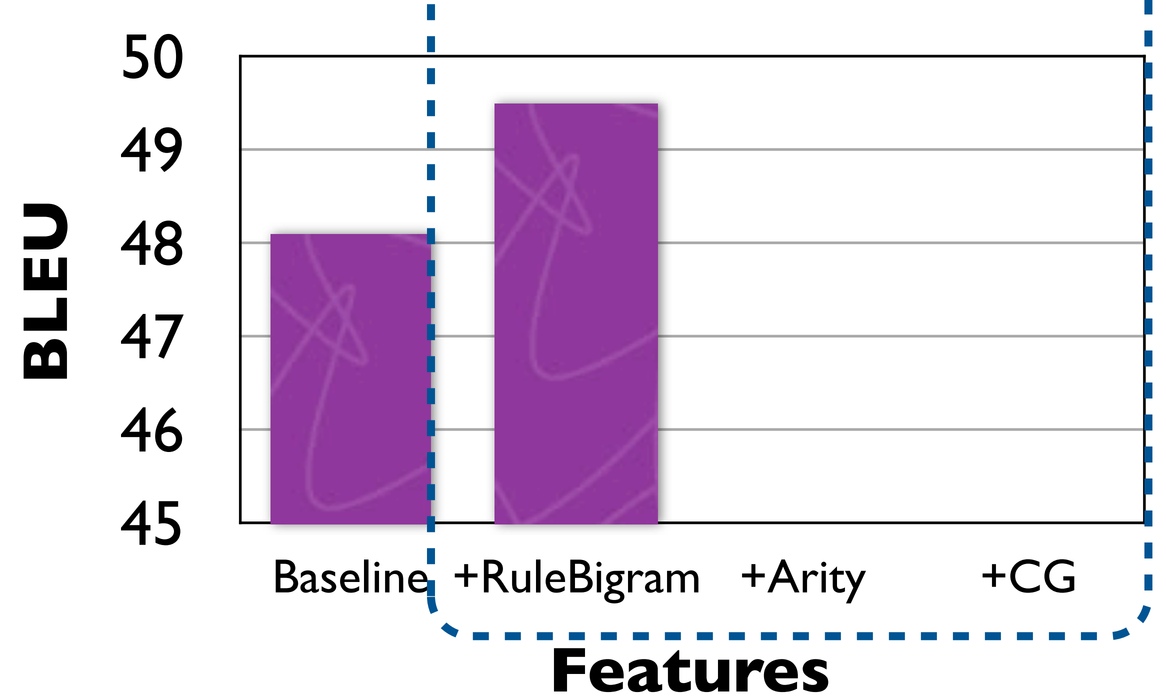


one big feature

glue rules or regular confusion rules?

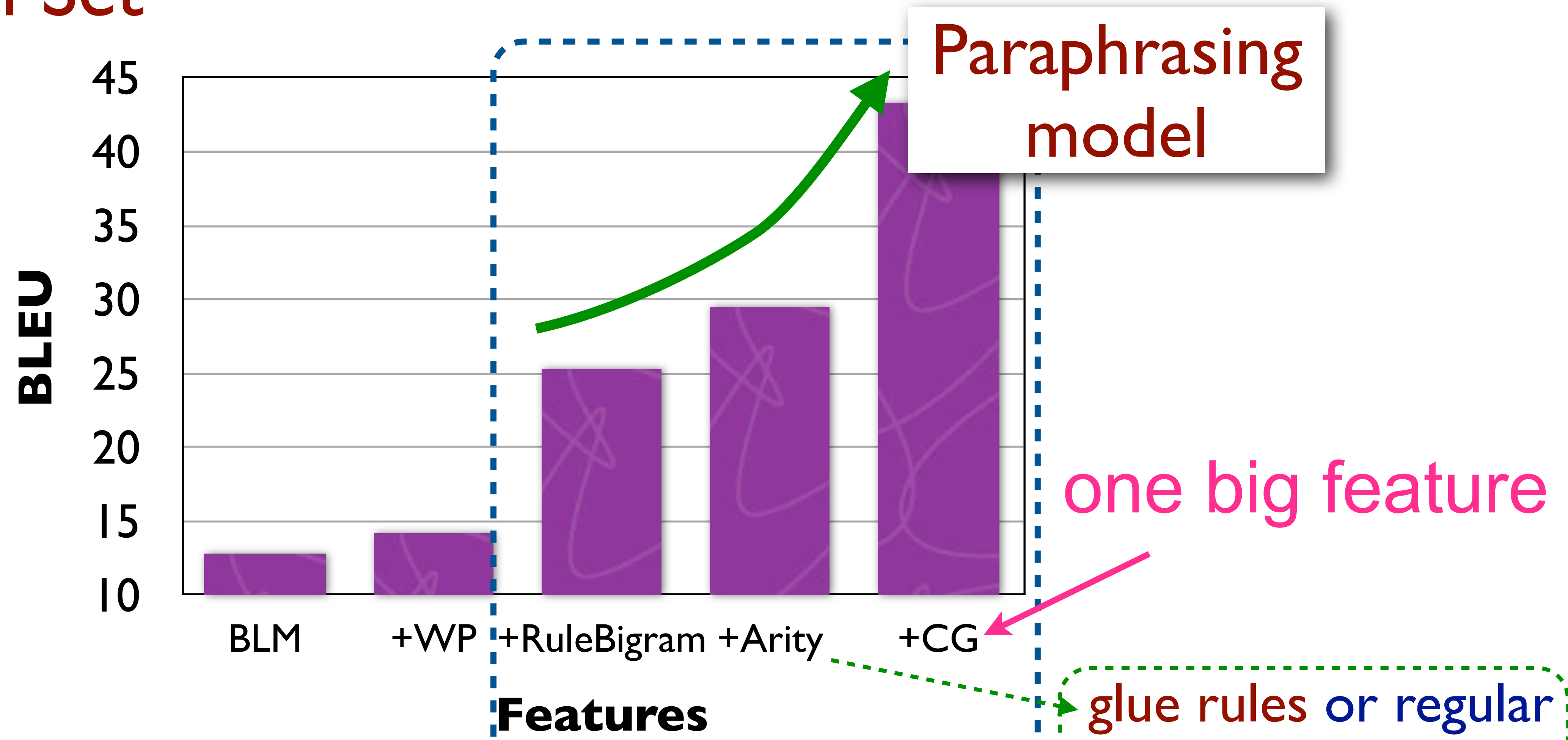
$$S \rightarrow \langle\, S_0\, X_1\,,\, S_0\, X_1 \,\rangle$$
$$S \rightarrow \langle\, X_0\,,\, X_0 \,\rangle$$

- On MT Set



Adding Features on the CG itself

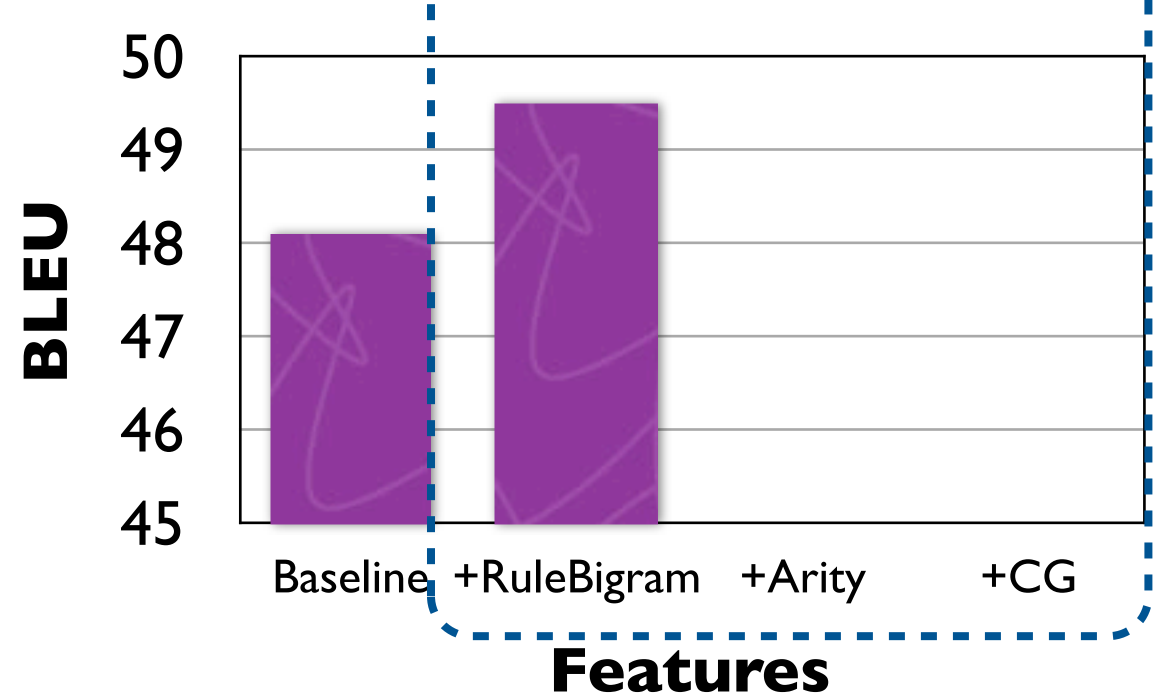
- On English Set



- On MT Set

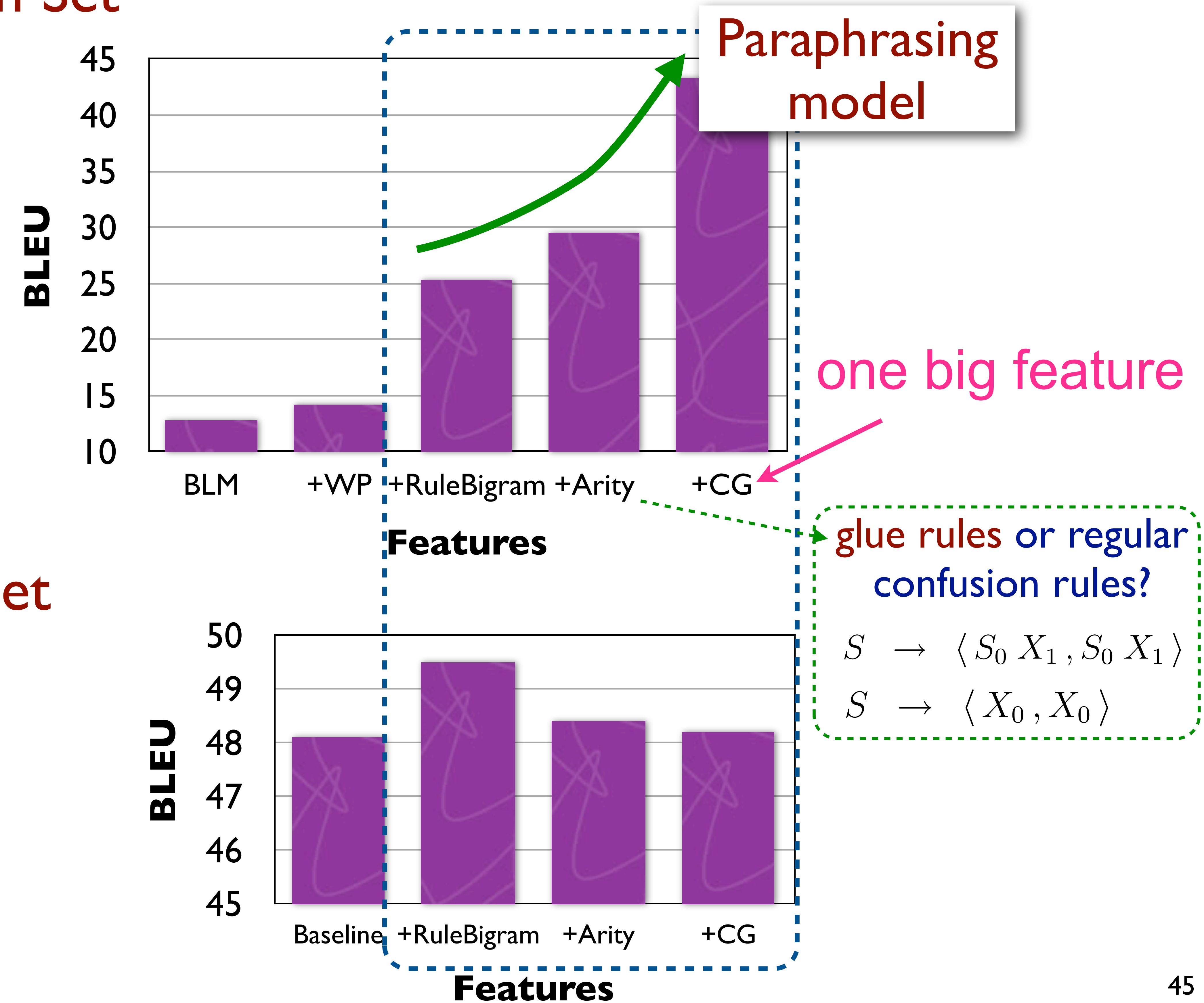
glue rules or regular confusion rules?

$$S \rightarrow \langle S_0 X_1, S_0 X_1 \rangle$$
$$S \rightarrow \langle X_0, X_0 \rangle$$



Adding Features on the CG itself

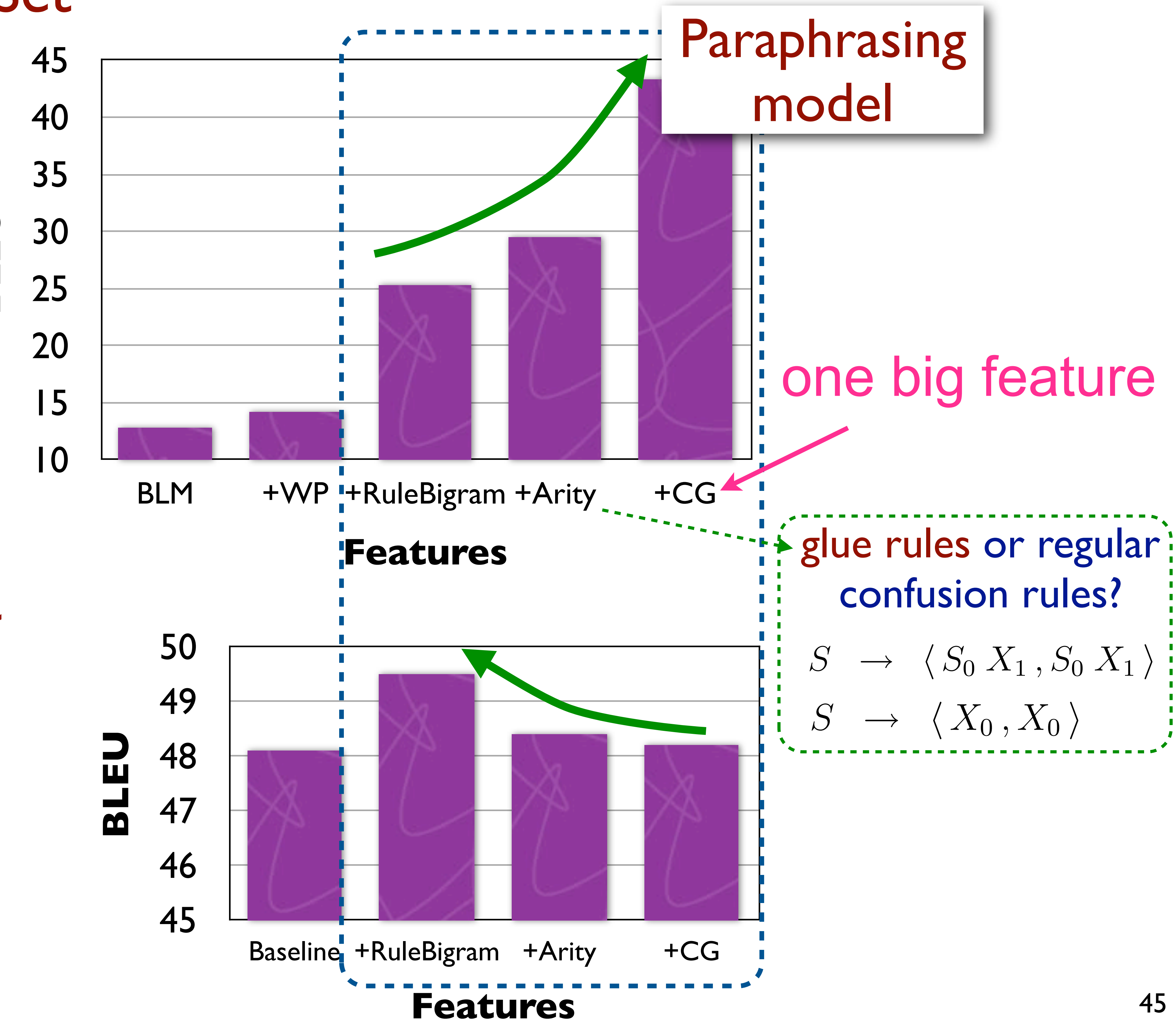
- On English Set
- On MT Set



Adding Features on the CG itself

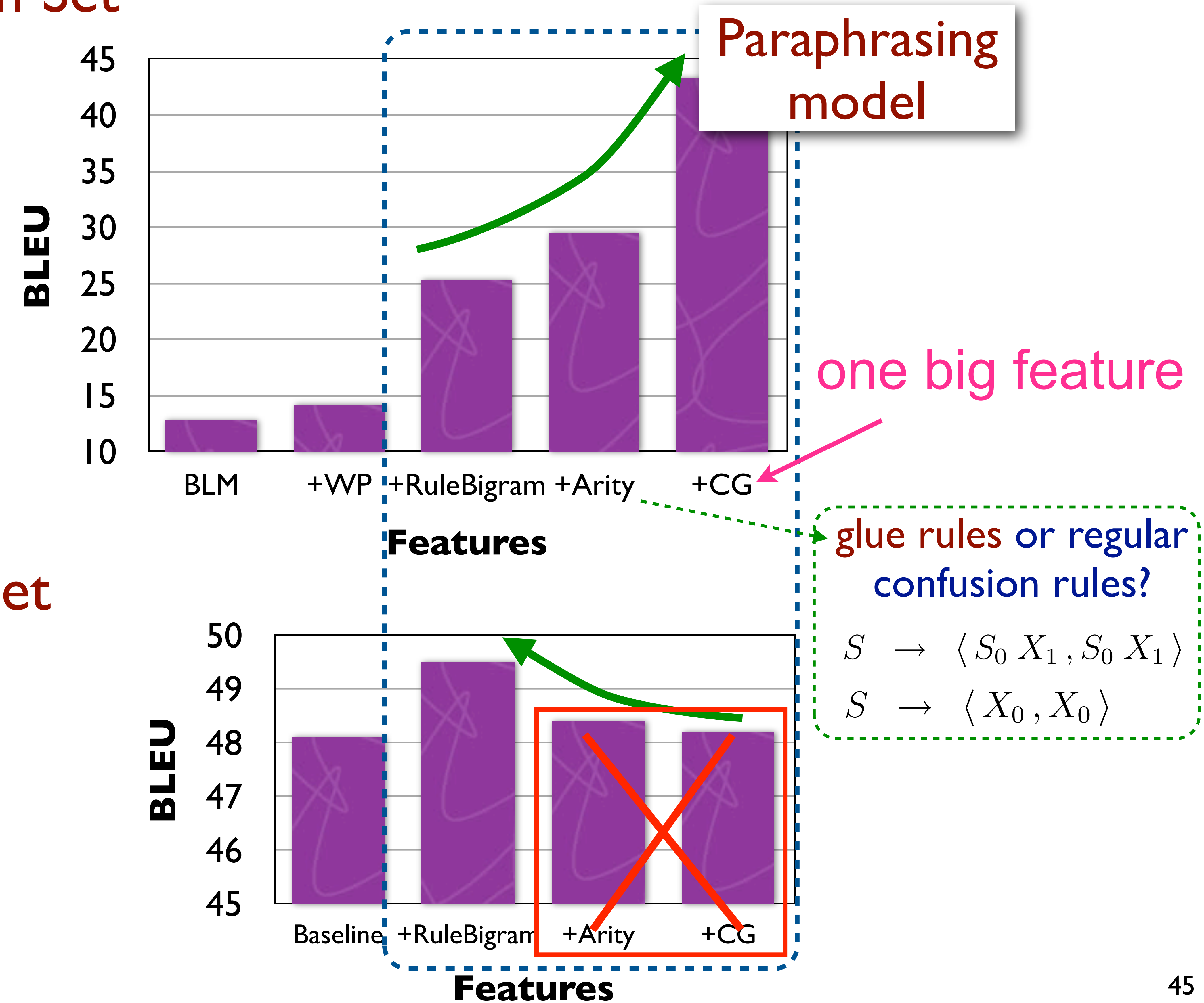
- On English Set

- On MT Set



Adding Features on the CG itself

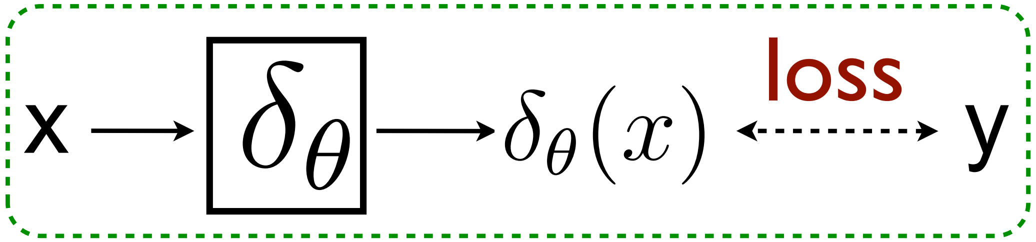
- On English Set
- On MT Set



Summary for Discriminative Training

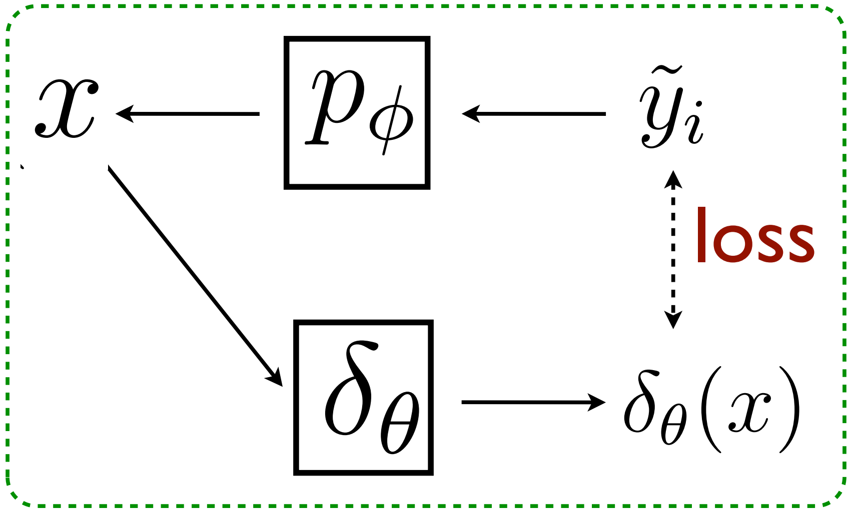
- Supervised: Minimum Empirical Risk

require
bitext



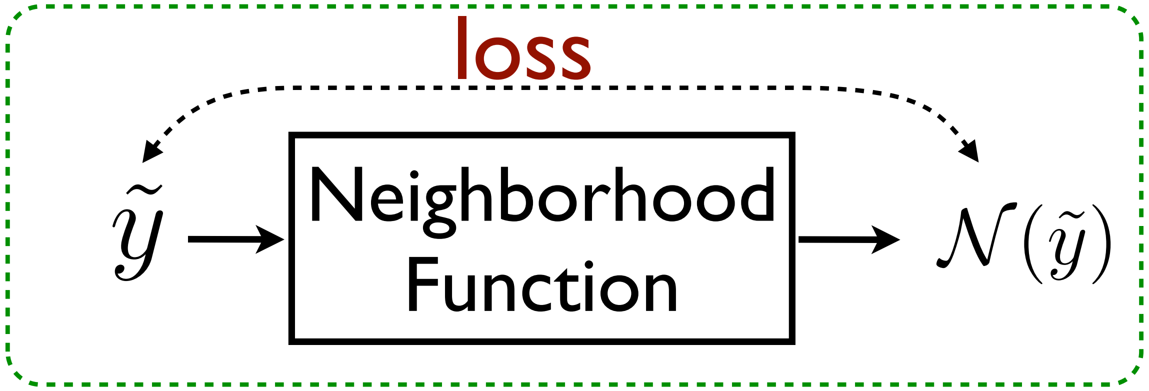
- Unsupervised: Minimum Imputed Risk

require
monolingual
English



- Unsupervised: Contrastive LM Estimation

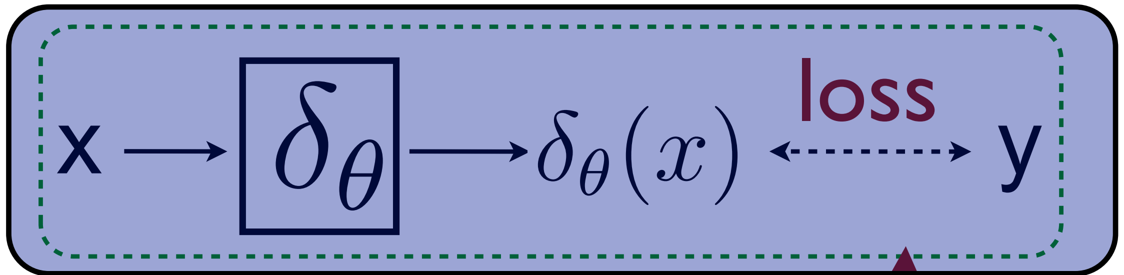
require
monolingual
English



Summary for Discriminative Training

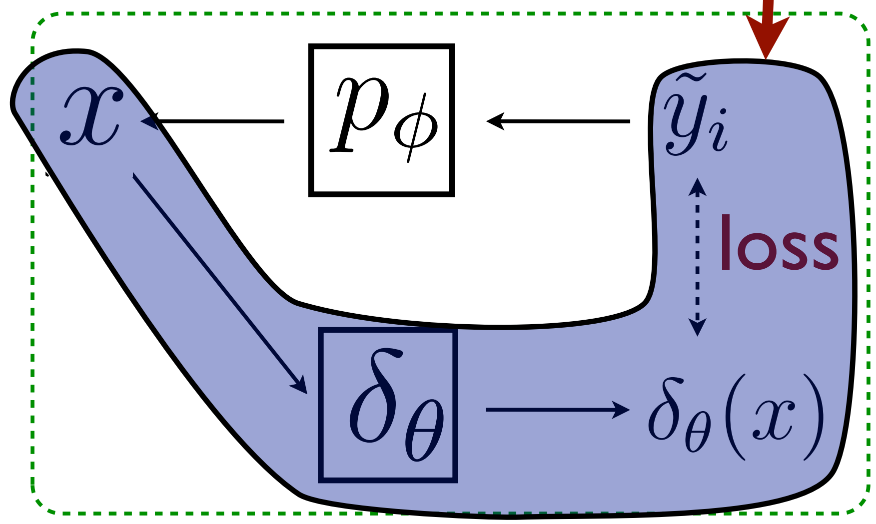
- Supervised: Minimum Empirical Risk

require bitext



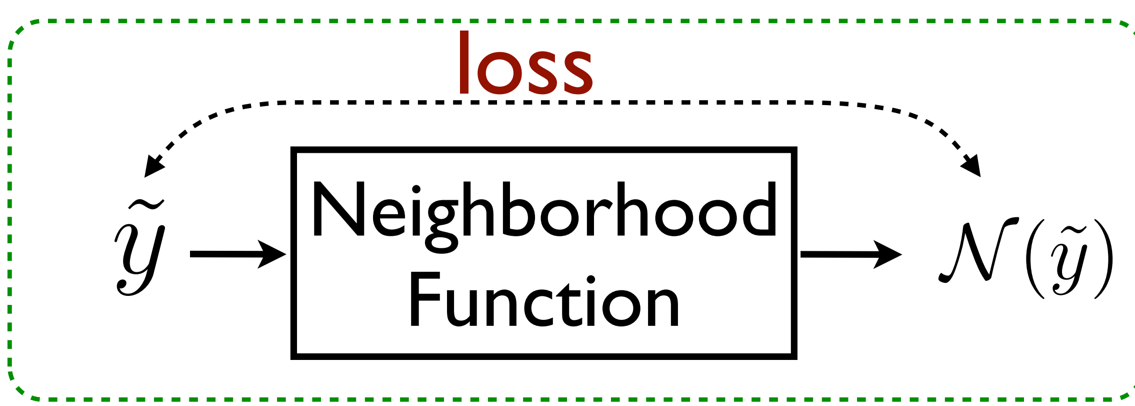
- Unsupervised: Minimum Imputed Risk

require monolingual English



- Unsupervised: Contrastive LM Estimation

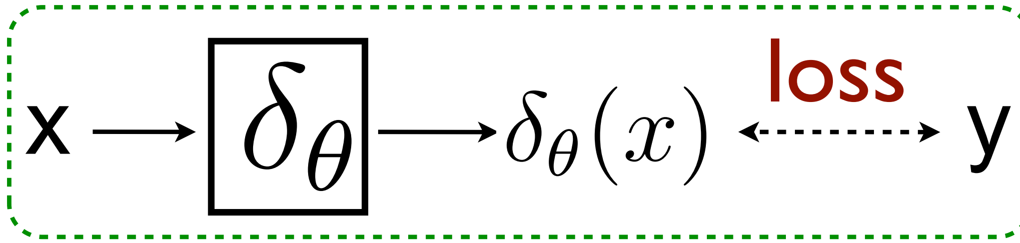
require monolingual English



Summary for Discriminative Training

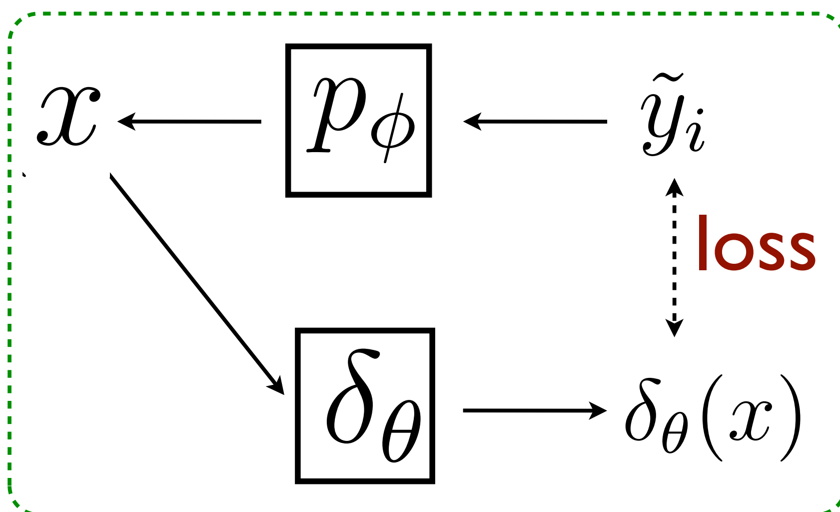
- Supervised Training

require bitext



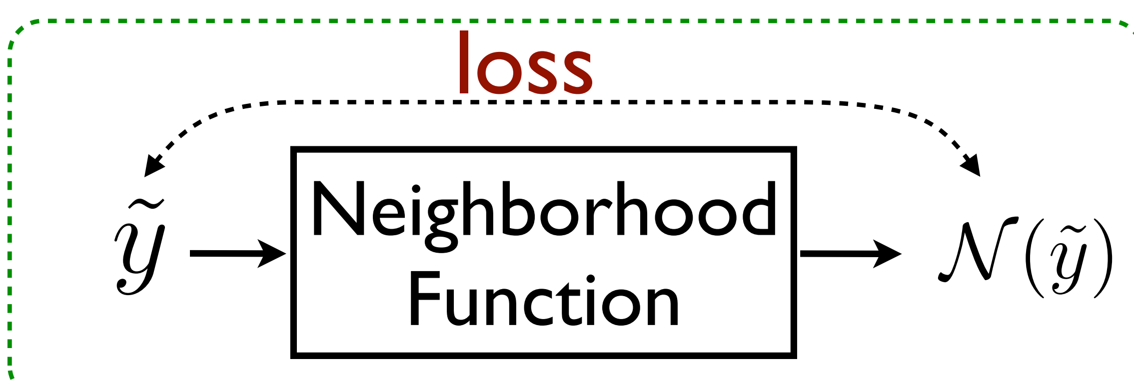
- Unsupervised: Minimum Imputed Risk

require monolingual English



- Unsupervised: Contrastive LM Estimation

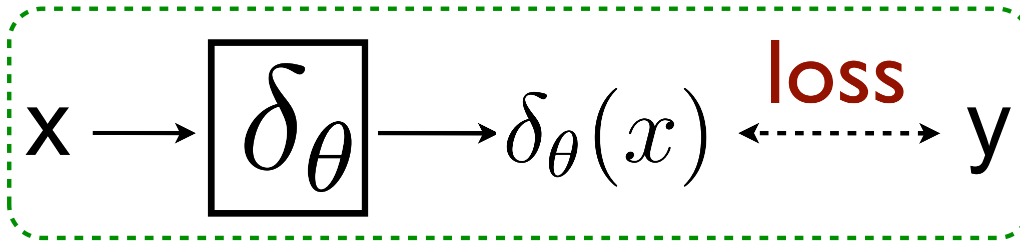
require monolingual English



Summary for Discriminative Training

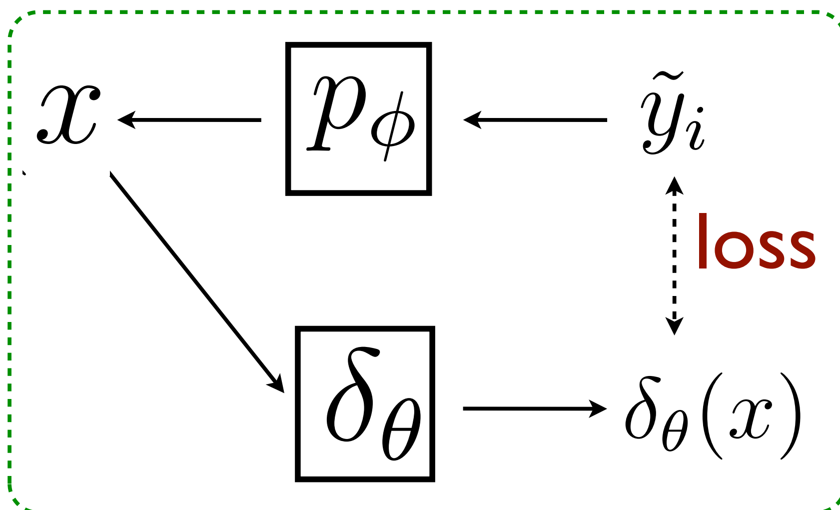
- Supervised Training

require bitext



- Unsupervised: Minimum Imputed Risk

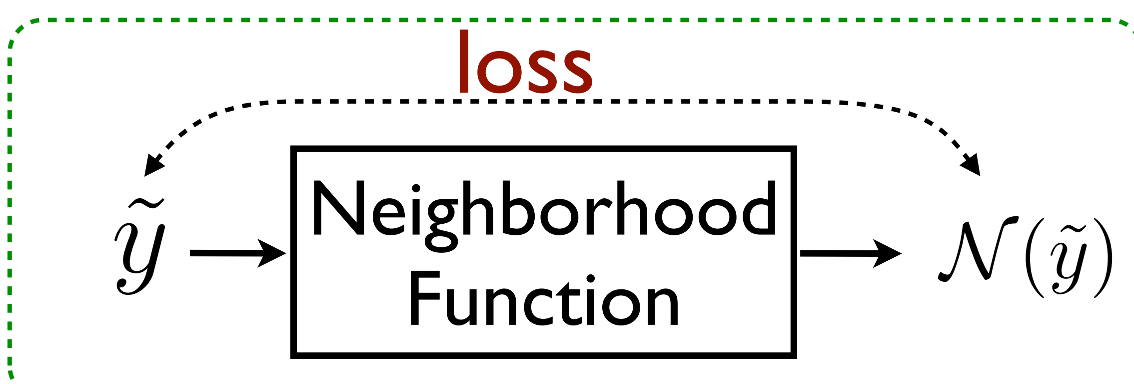
require monolingual English



require a reverse model

- Unsupervised: Contrastive LM Estimation

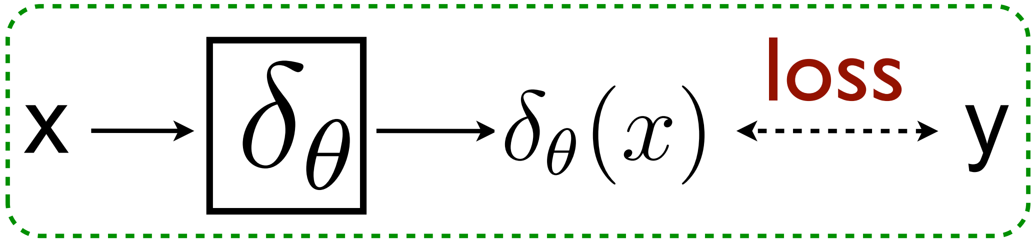
require monolingual English



Summary for Discriminative Training

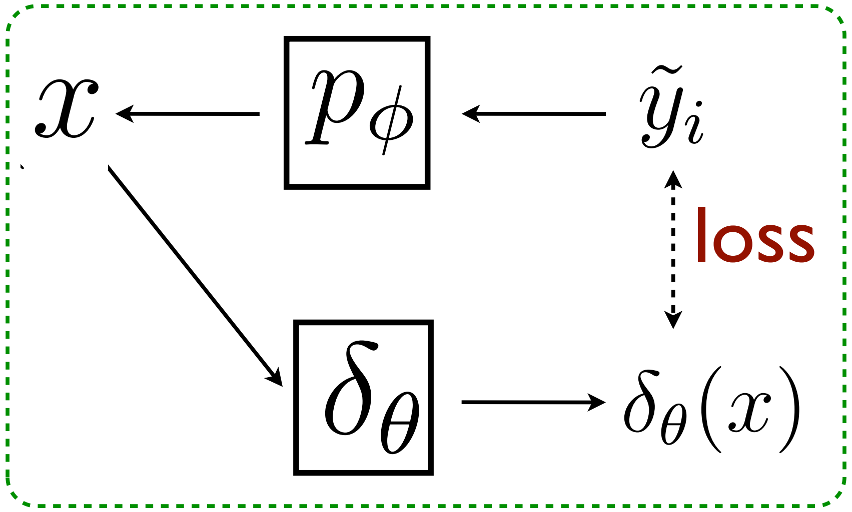
- Supervised Training

require bitext



- Unsupervised: Minimum Imputed Risk

require monolingual English

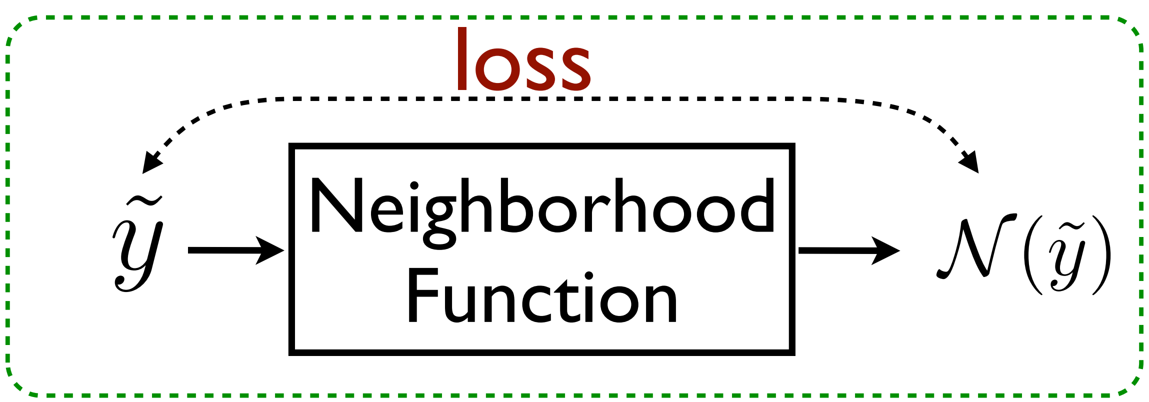


require a reverse model

can have both TM and LM features

- Unsupervised: Contrastive LM Estimation

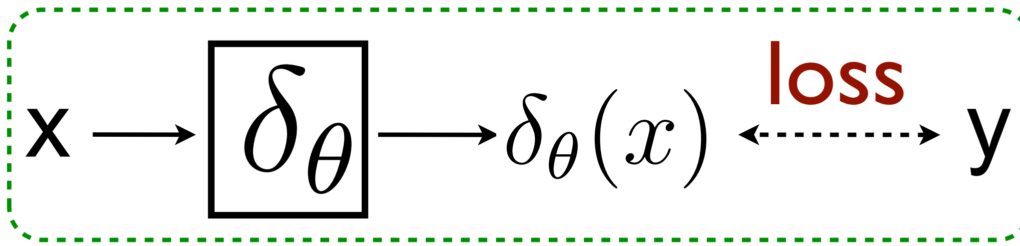
require monolingual English



Summary for Discriminative Training

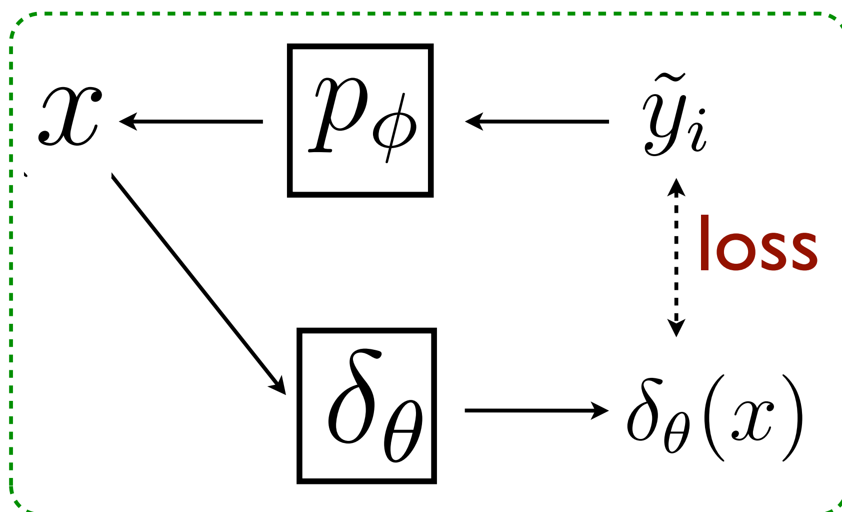
- Supervised Training

require bitext



- Unsupervised: Minimum Imputed Risk

require monolingual English

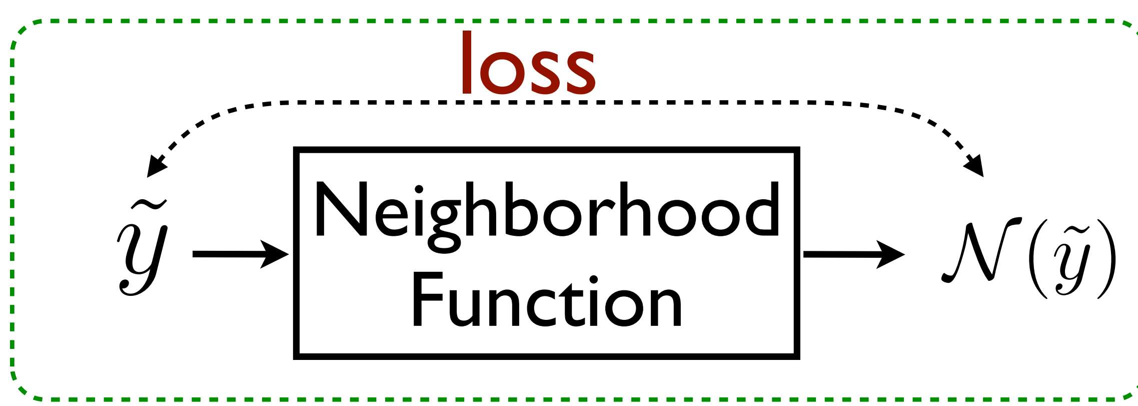


require a reverse model

can have both TM and LM features

- Unsupervised: Contrastive LM Estimation

require monolingual English

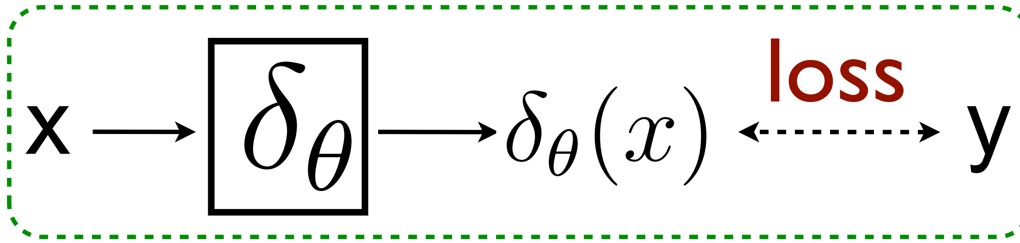


can have LM features only

Summary for Discriminative Training

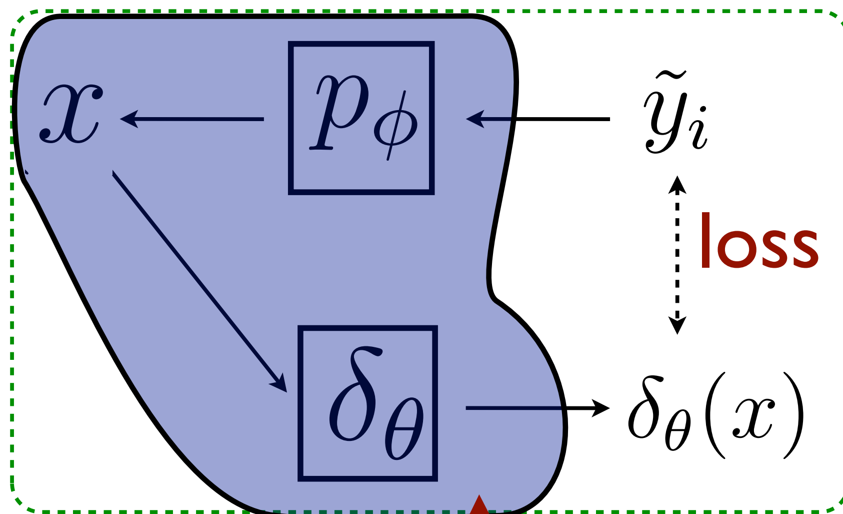
- Supervised Training

require bitext



- Unsupervised: Minimum Imputed Risk

require monolingual English

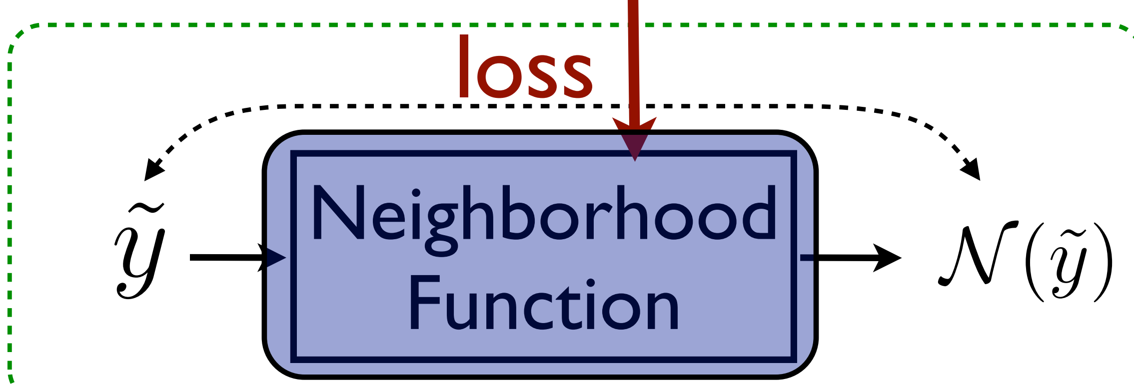


require a reverse model

can have both TM and LM features

- Unsupervised: Contrastive LM Estimation

require monolingual English



can have LM features only

Outline

- Hypergraph as Hypothesis Space
- Unsupervised Discriminative Training
  - minimum imputed risk
  - contrastive language model estimation
- Variational Decoding
- First- and Second-order Expectation Semirings

<table>
<tr><td>decoding
(e.g., mbr)</td><td>training
(e.g., mert)</td></tr>
<tr><td colspan="2">atomic inference operations
(e.g., finding one-best, k-best or expectation,
inference can be exact or approximate)</td></tr>
</table>

Variational Decoding

Variational Decoding

- We want to do inference under p, but it is intractable

Variational Decoding

- We want to do inference under p, but it is intractable

- Instead, we derive a simpler distribution q*

Variational Decoding

- We want to do inference under p, but it is intractable

- Instead, we derive a simpler distribution q*

- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

intractable MAP decoding

- Instead, we derive a simpler distribution q*

- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

intractable MAP decoding

$$y^* = \arg\max_y p(y \mid x)$$

- Instead, we derive a simpler distribution q*

- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

intractable MAP decoding

$$y^* \quad = \quad \arg\max_y \; p(y \mid x) = \quad \arg\max_y \sum_{d \in \mathrm{D}(x,y)} p(d \mid x)$$

- Instead, we derive a simpler distribution q*

- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

intractable MAP decoding (Sima'an 1996)

$$y^* = \arg\max_y p(y \mid x) = \arg\max_y \sum_{d \in \mathrm{D}(x,y)} p(d \mid x)$$

- Instead, we derive a simpler distribution q*

- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

intractable MAP decoding (Sima'an 1996)

$$y^* = \arg\max_y p(y \mid x) = \arg\max_y \sum_{d \in \mathrm{D}(x,y)} p(d \mid x)$$

- Instead, we derive a simpler distribution q*

tractable estimation

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p || q)$$

- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

intractable MAP decoding (Sima'an 1996)

$$y^* = \arg\max_y p(y \mid x) = \arg\max_y \sum_{d \in \mathrm{D}(x,y)} p(d \mid x)$$

- Instead, we derive a simpler distribution q*

tractable estimation

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p || q)$$

- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

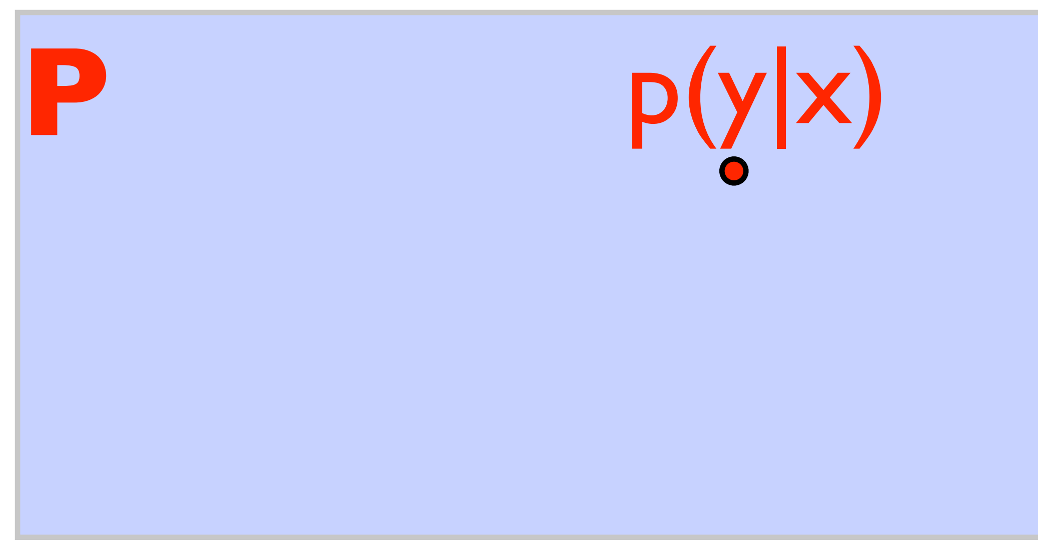
intractable MAP decoding (Sima'an 1996)

$$y^* = \arg\max_y p(y \mid x) = \arg\max_y \sum_{d \in \mathrm{D}(x,y)} p(d \mid x)$$

- Instead, we derive a simpler distribution q*

tractable estimation

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p || q)$$



- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

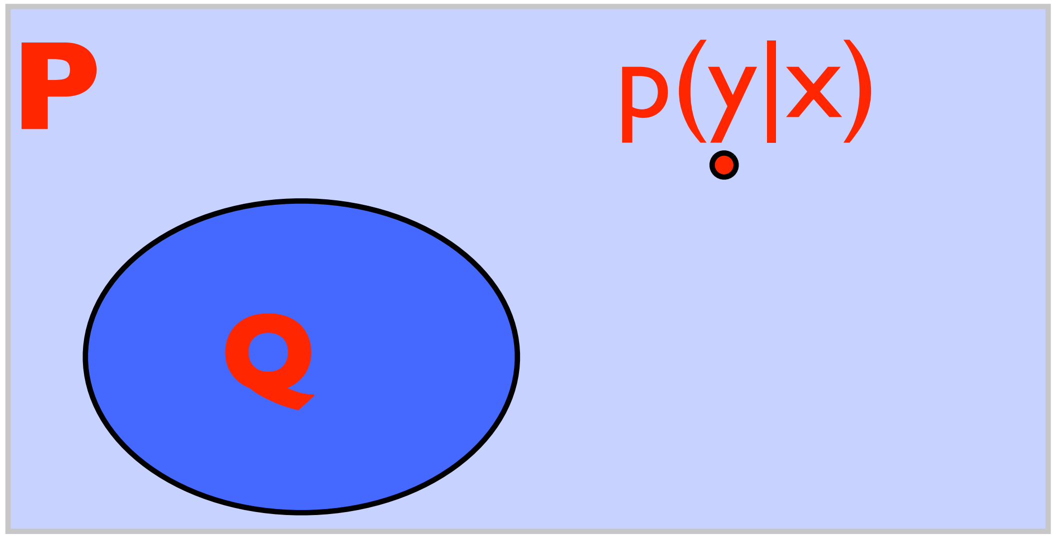
intractable MAP decoding (Sima'an 1996)

$$y^* = \arg\max_y p(y \mid x) = \arg\max_y \sum_{d \in \mathrm{D}(x,y)} p(d \mid x)$$

- Instead, we derive a simpler distribution q*

tractable estimation

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

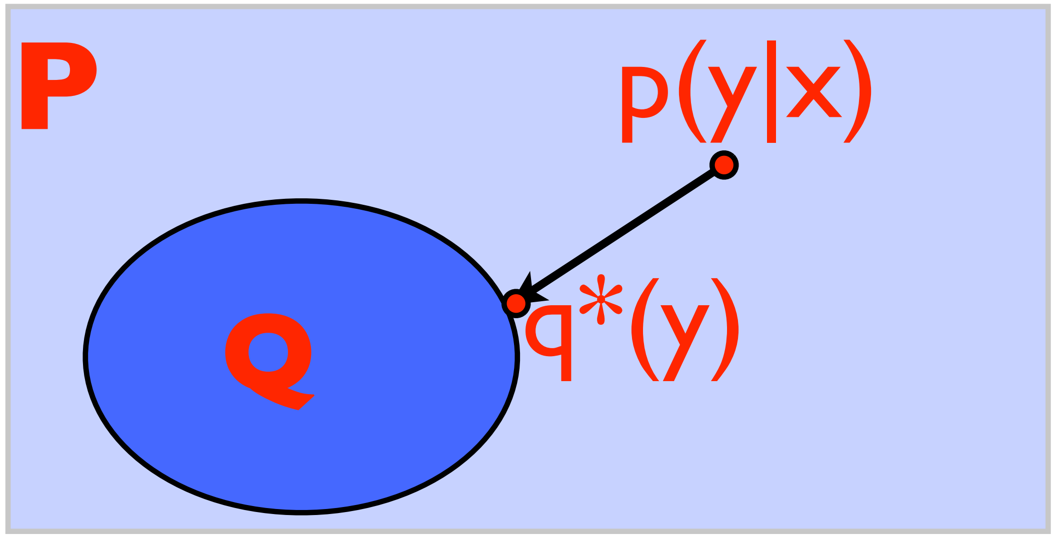
intractable MAP decoding (Sima'an 1996)

$$y^* = \arg\max_y p(y \mid x) = \arg\max_y \sum_{d \in \mathrm{D}(x,y)} p(d \mid x)$$

- Instead, we derive a simpler distribution q*

tractable estimation

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

Variational Decoding

- We want to do inference under p, but it is intractable

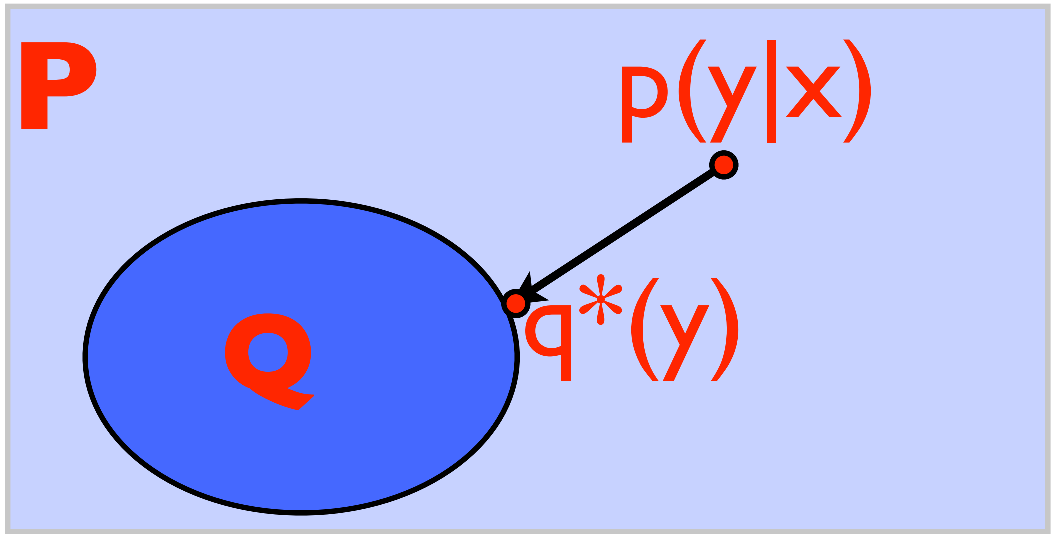
intractable MAP decoding (Sima'an 1996)

$$y^* = \arg\max_y p(y \mid x) = \arg\max_y \sum_{d \in D(x,y)} p(d \mid x)$$

- Instead, we derive a simpler distribution q*

tractable estimation

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p || q)$$



- Then, we will use q* as a surrogate for p in inference

tractable decoding

$$y^* = \arg\max_y q^*(y \mid x)$$

Variational Decoding for MT: an Overview

Variational Decoding for MT: an Overview

Sentence-specific decoding

Variational Decoding for MT: an Overview

Sentence-specific decoding

Three steps:

Variational Decoding for MT: an Overview

Sentence-specific decoding

Three steps:

1. Generate a hypergraph for the foreign sentence

Variational Decoding for MT: an Overview

Sentence-specific decoding

Three steps:

1 Generate a hypergraph for the foreign sentence

Foreign sentence x

Variational Decoding for MT: an Overview

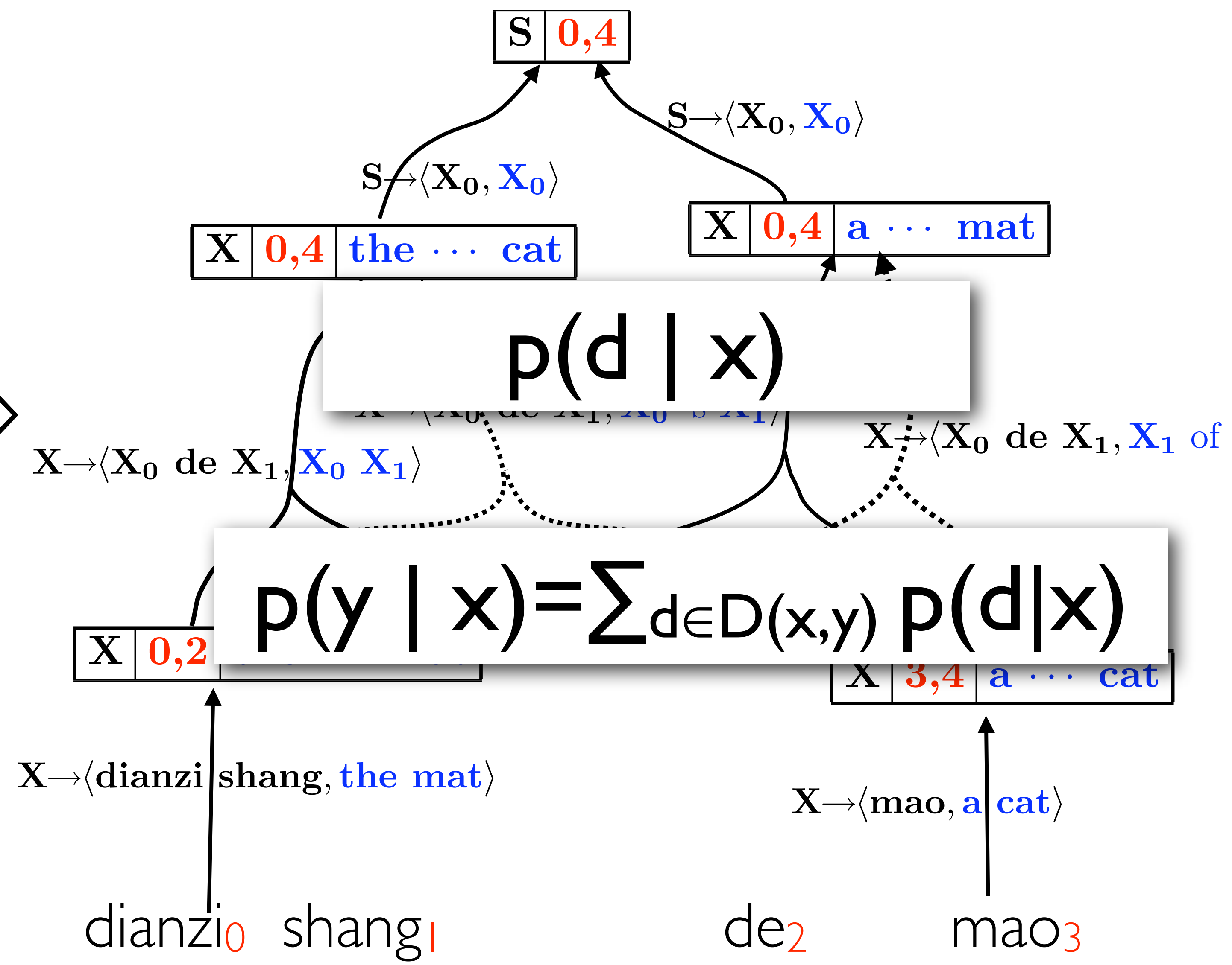
Sentence-specific decoding

Three steps:

MAP decoding under P is intractable

1. Generate a hypergraph for the foreign sentence

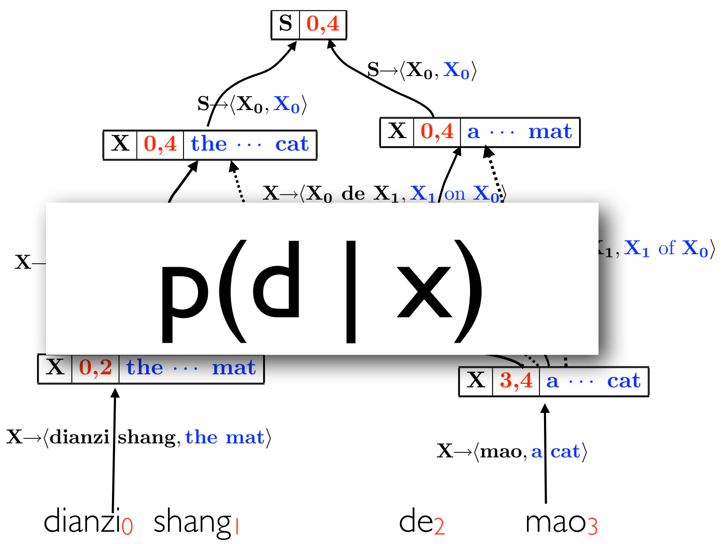
Foreign sentence x ⇒ SMT ⇒



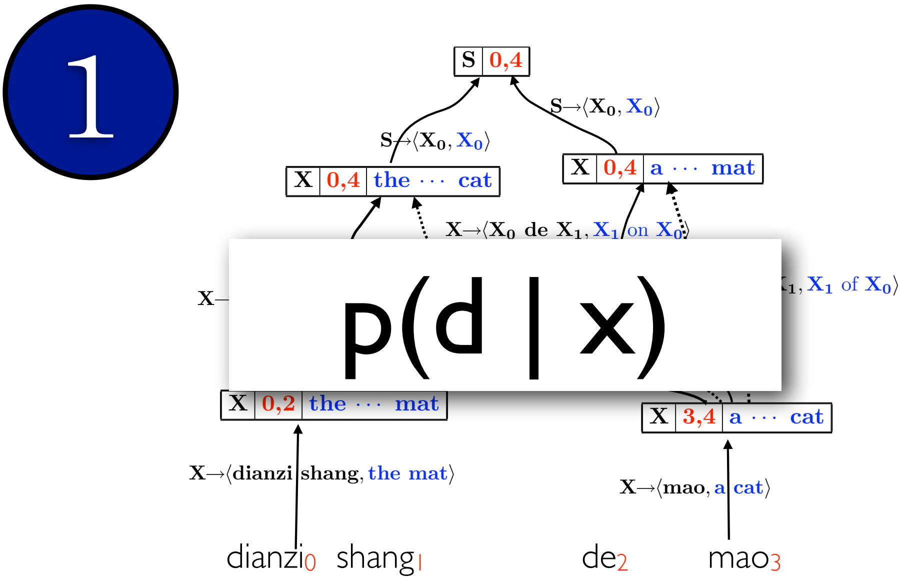
$p(d \mid x)$

$p(y \mid x) = \sum_{d \in D(x,y)} p(d|x)$

1

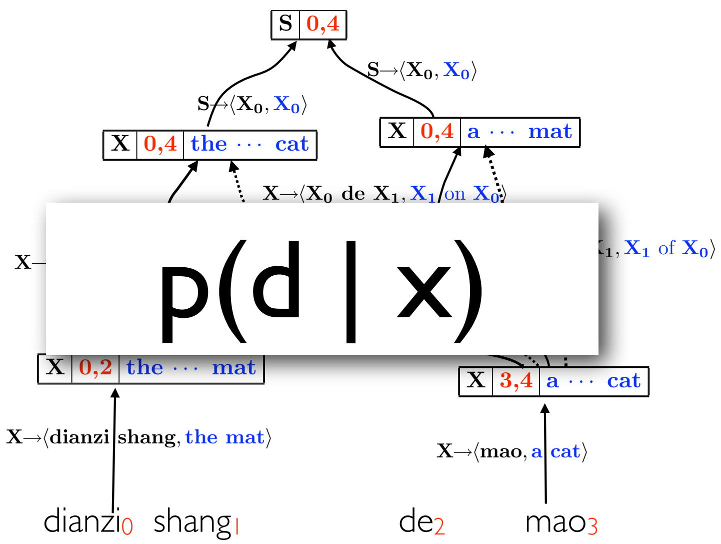


Generate a hypergraph



Generate a hypergraph

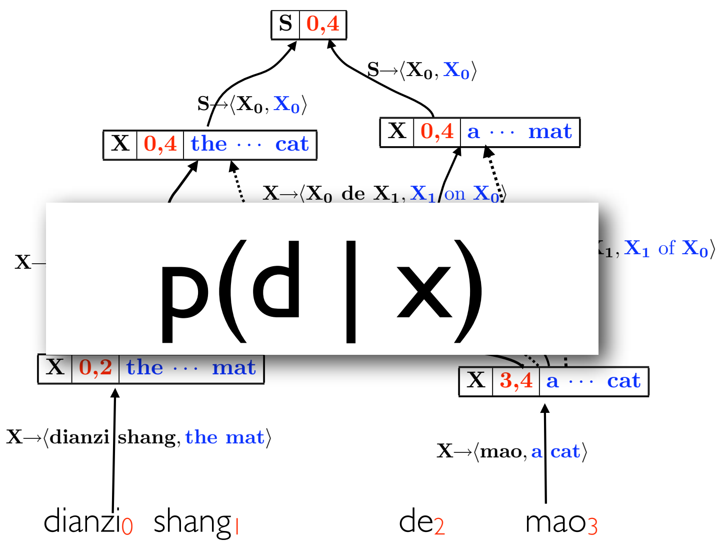
1



Generate a hypergraph

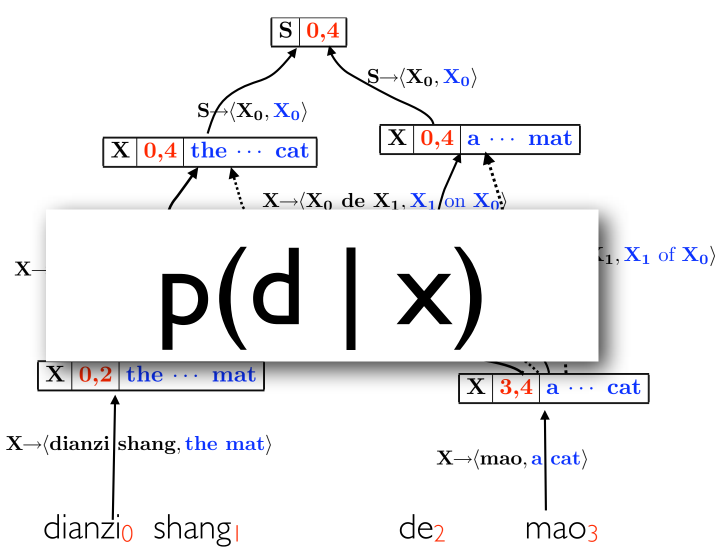
2

1

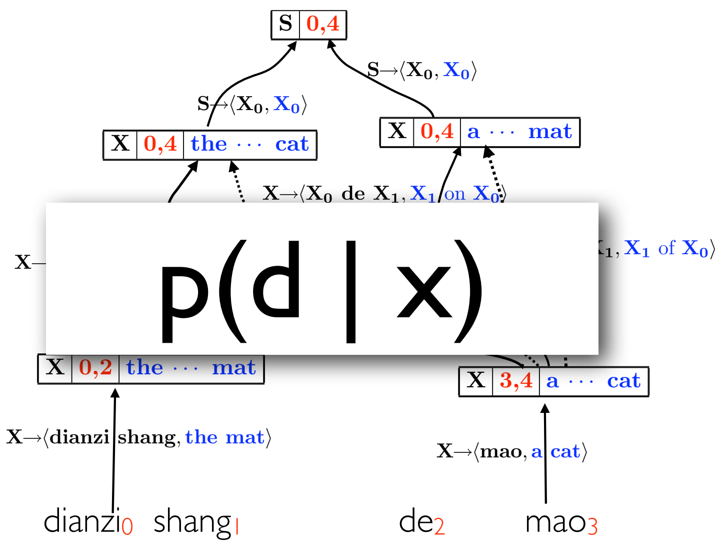


Generate a hypergraph

2

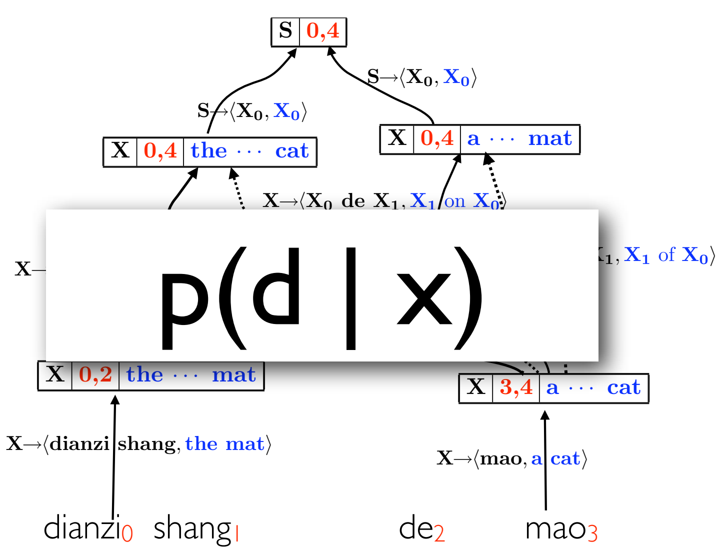


1



Generate a hypergraph

2

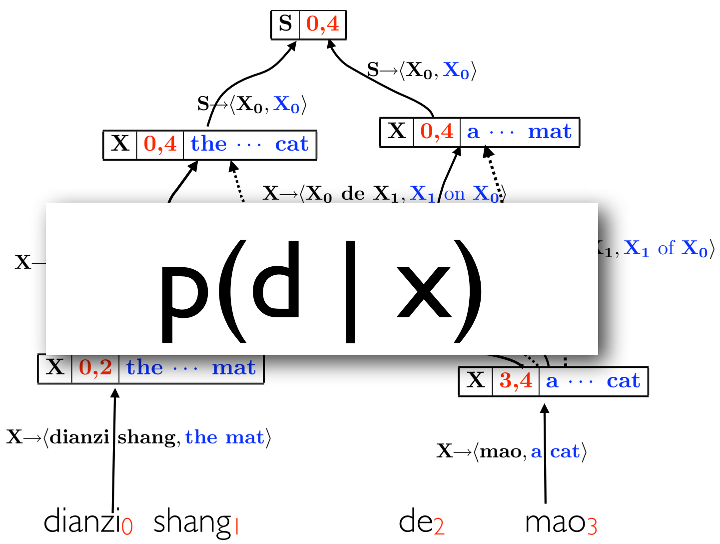


Estimate a model from the hypergraph by minimizing KL

q* is an n-gram model over output strings.

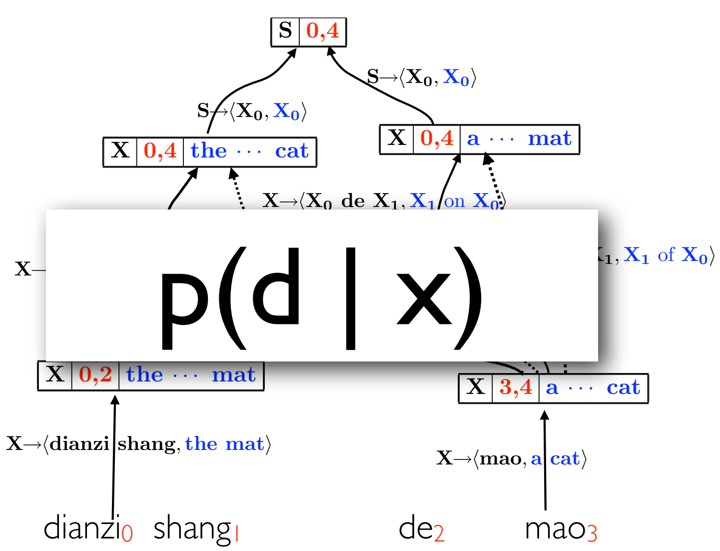
$$q^*(y \mid x)$$

1



Generate a hypergraph

2



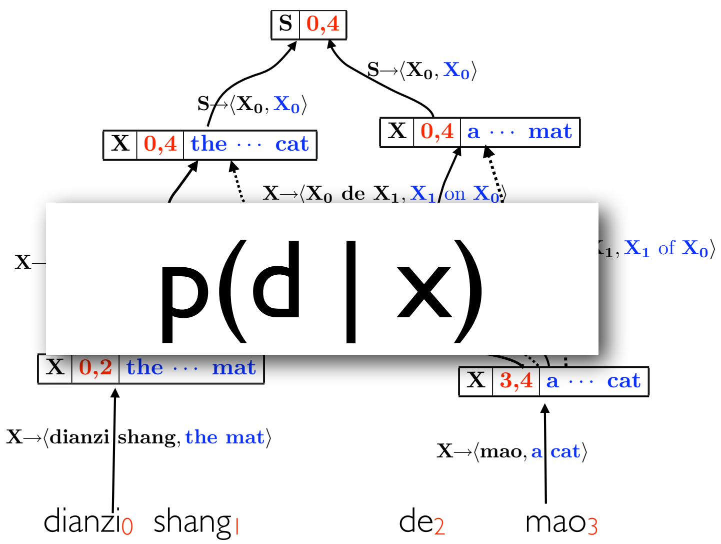
Estimate a model from the hypergraph by minimizing KL

q* is an n-gram model over output strings.

$$q^*(y \mid x)$$

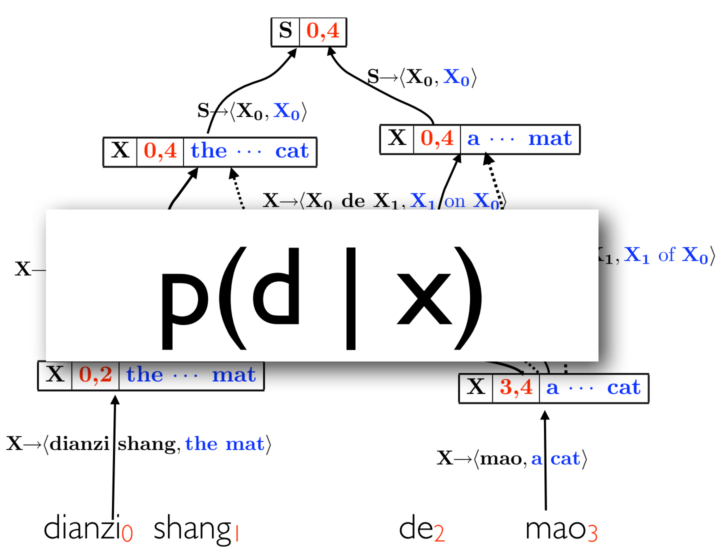
$$\approx \sum_{d \in D(x,y)} p(d|x)$$

1



Generate a hypergraph

2



Estimate a model from the hypergraph by minimizing KL

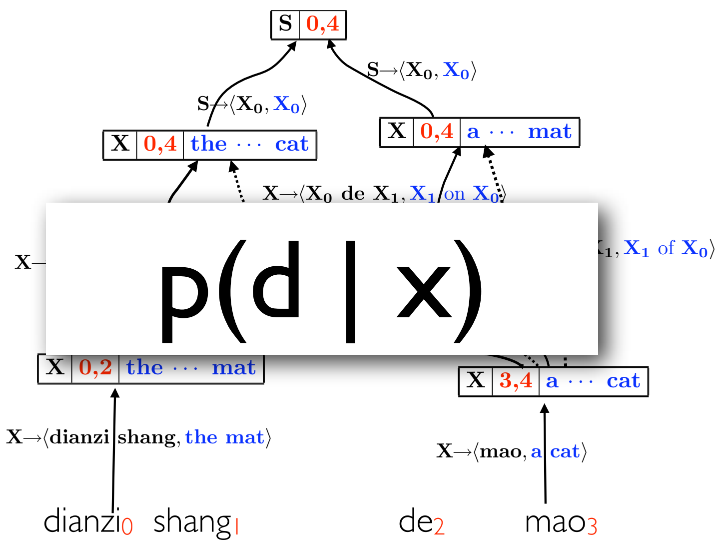
q* is an n-gram model over output strings.

$$q^*(y \mid x)$$

$$\approx \sum_{d \in D(x,y)} p(d|x)$$

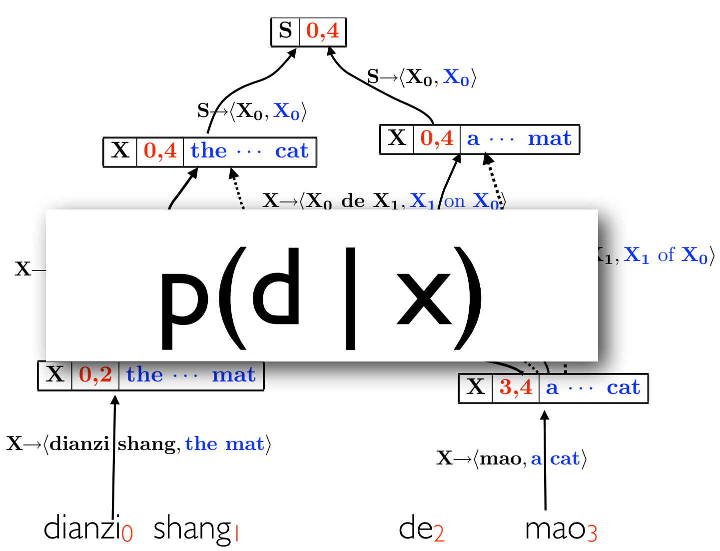
3

1



Generate a hypergraph

2



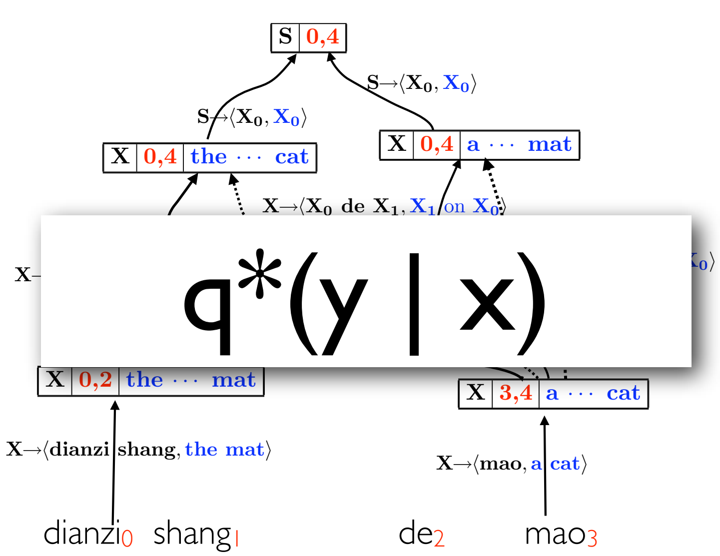
Estimate a model from the hypergraph by minimizing KL

q* is an n-gram model over output strings.

$$q^*(y \mid x)$$

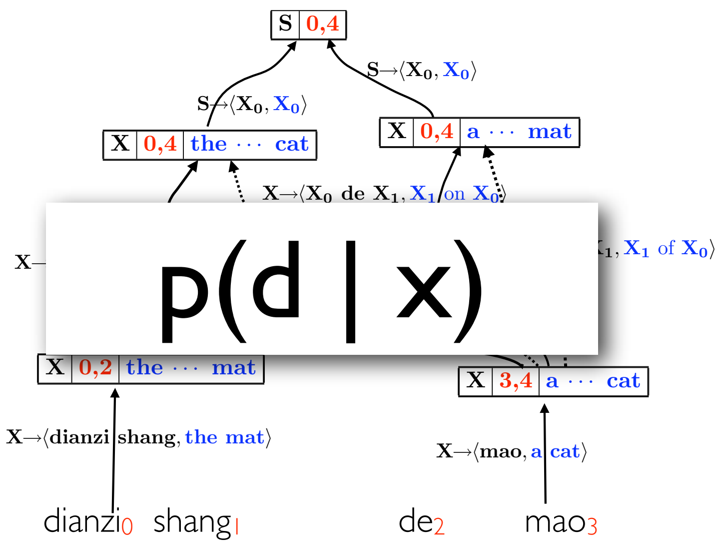
$$\approx \sum_{d \in D(x,y)} p(d|x)$$

3



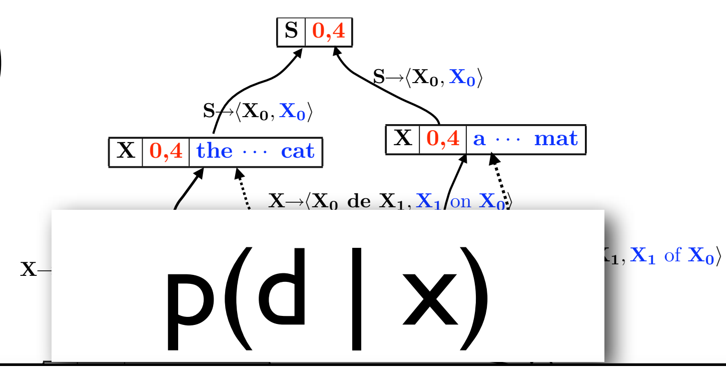
Decode using q* on the hypergraph

1



Generate a hypergraph

2



Estimate a model from the hypergraph by minimizing KL

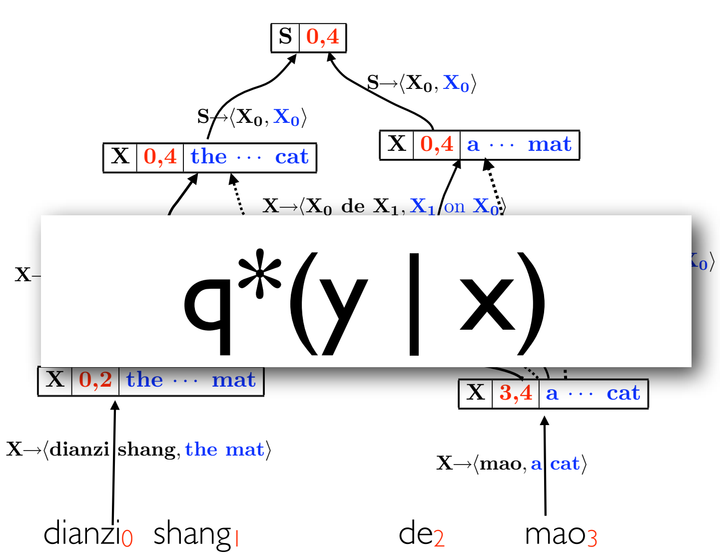
q* is an n-gram model over output strings.

$q^*(y \mid x)$

$\approx \sum_{d \in D(x,y)} p(d|x)$

Approximate a hypergraph with a lattice!

3



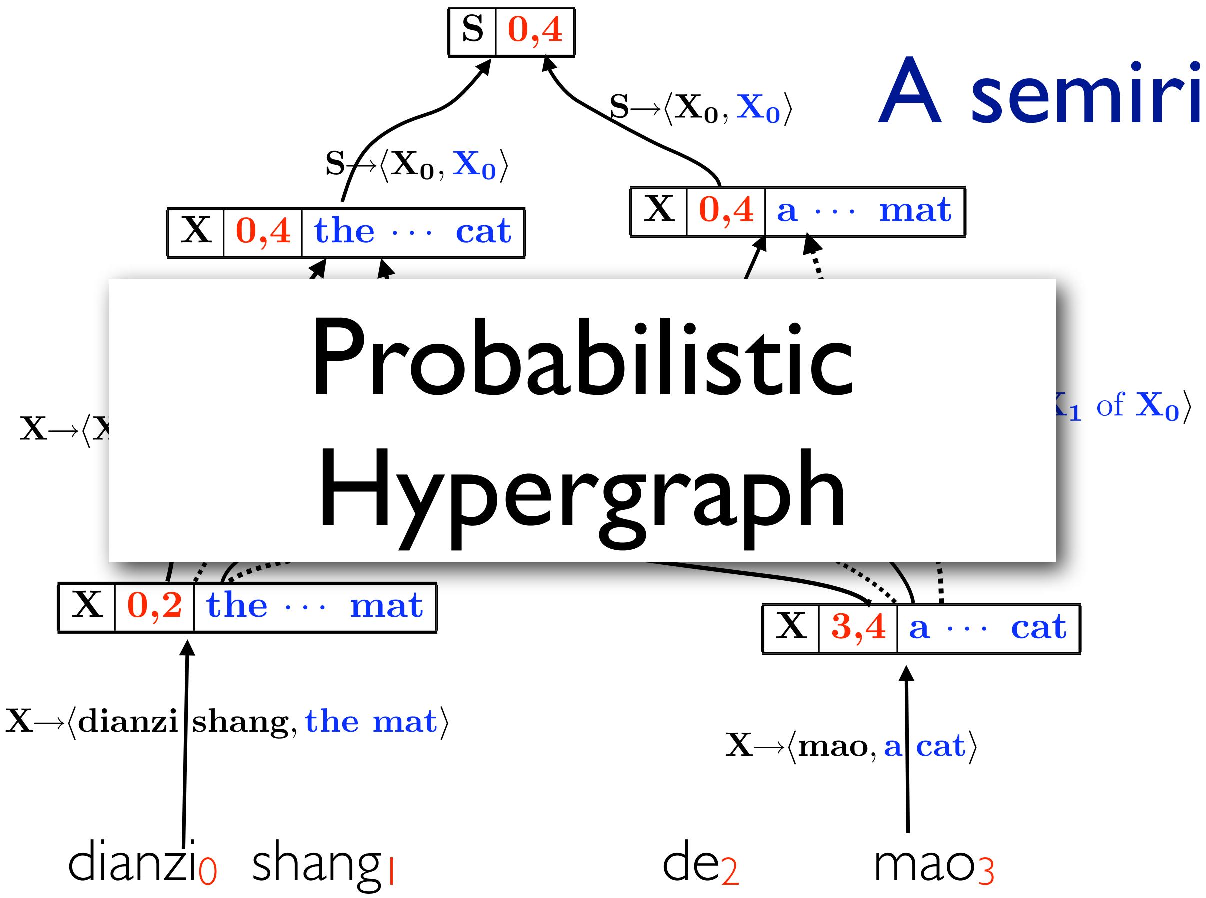
Decode using q* on the hypergraph

Outline

- Hypergraph as Hypothesis Space
- Unsupervised Discriminative Training
  - minimum imputed risk
  - contrastive language model estimation
- Variational Decoding
- First- and Second-order Expectation Semirings

decoding (e.g., mbr)	**training** (e.g., mert)
atomic inference operations (e.g., finding one-best, k-best or expectation, inference can be *exact* or *approximate*)	

A semiring framework to compute all of these

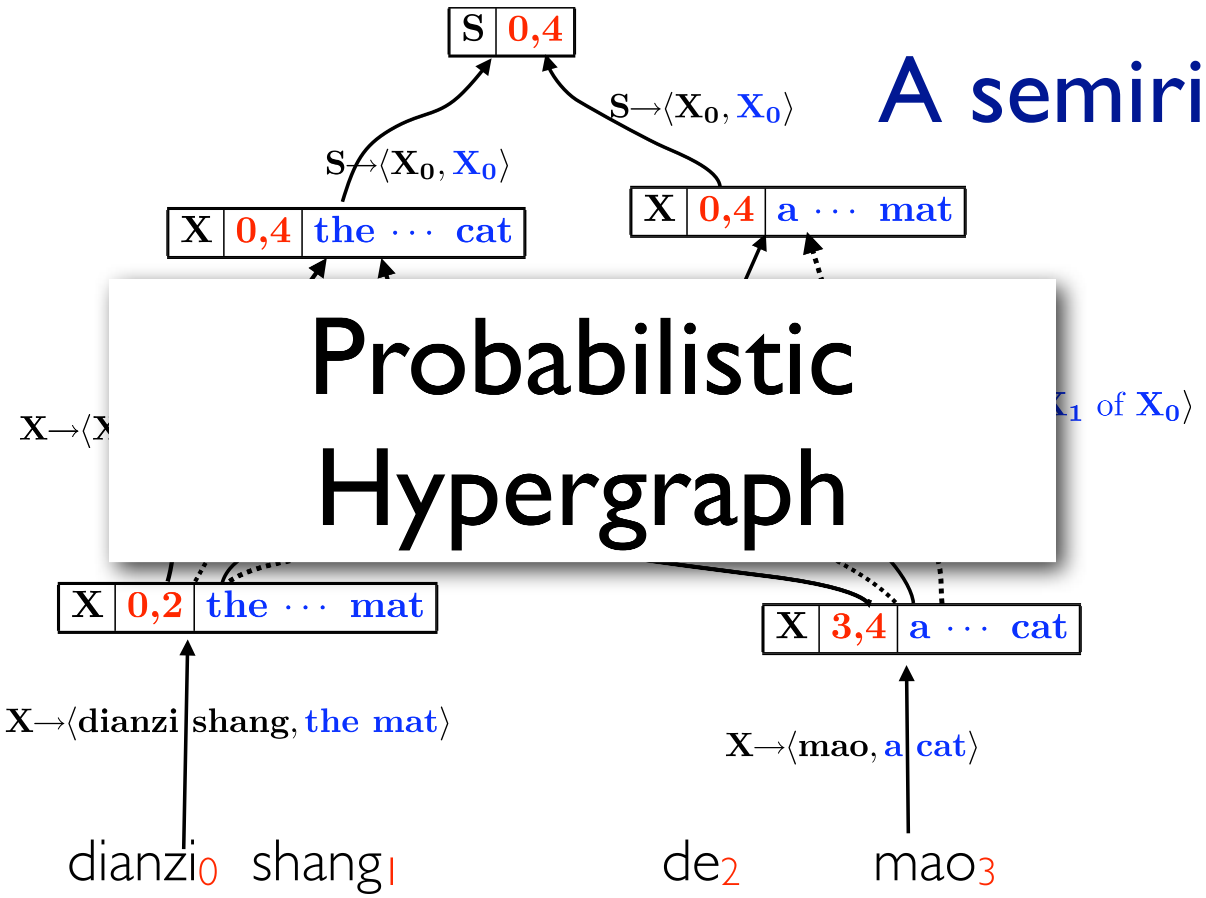


- "Decoding" quantities:
  - Viterbi
  - K-best
  - Counting
  - ......

- First-order expectations:
  - expectation
    - entropy
    - expected loss
    - cross-entropy
    - KL divergence
    - feature expectations
  - first-order gradient of Z

- Second-order expectations:
  - expectation over product
    - interaction between features
  - Hessian matrix of Z
    - second-order gradient descent
  - gradient of expectation
    - gradient of expected loss or entropy

A semiring framework to compute all of these

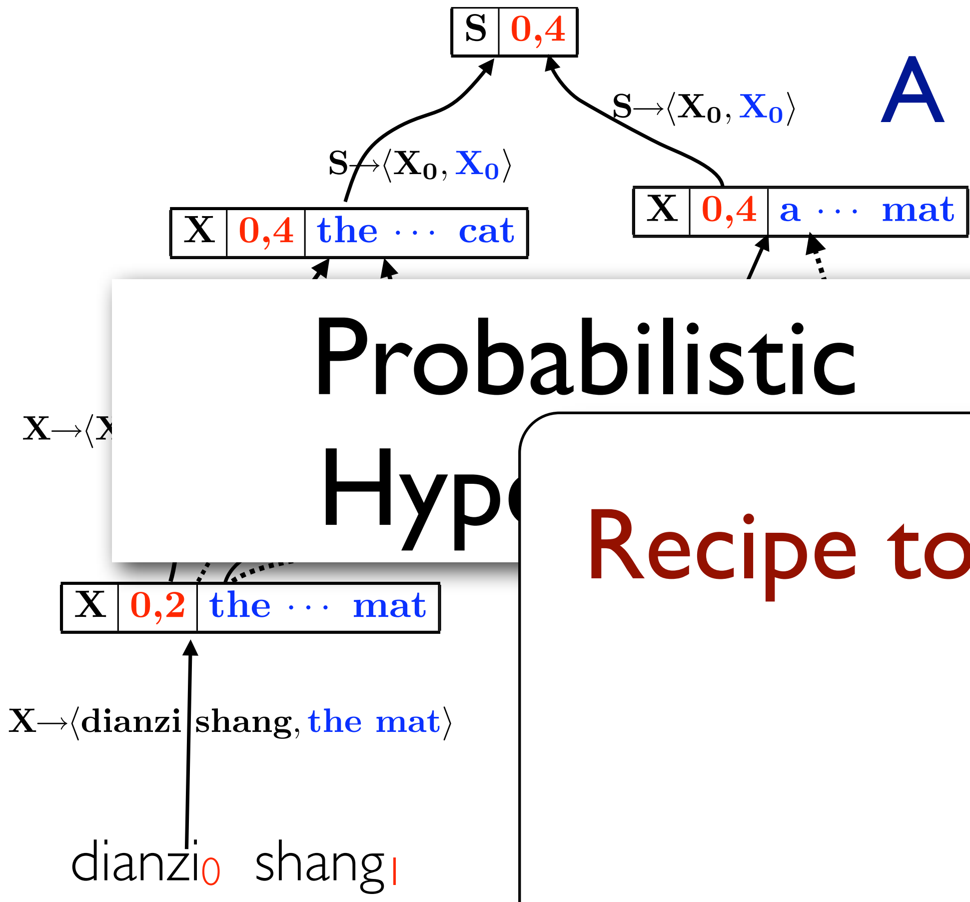


- "Decoding" quantities:
  - Viterbi
  - K-best
  - Counting
  - ......

- First-order expectations:
  - expectation
    - entropy
    - expected loss
    - cross-entropy
    - KL divergence
    - feature expectations
  - first-order gradient of Z

- Second-order expectations:
  - expectation over product
    - interaction between features
  - Hessian matrix of Z
    - second-order gradient descent
  - gradient of expectation
    - gradient of expected loss or entropy

A semiring framework to compute all of these



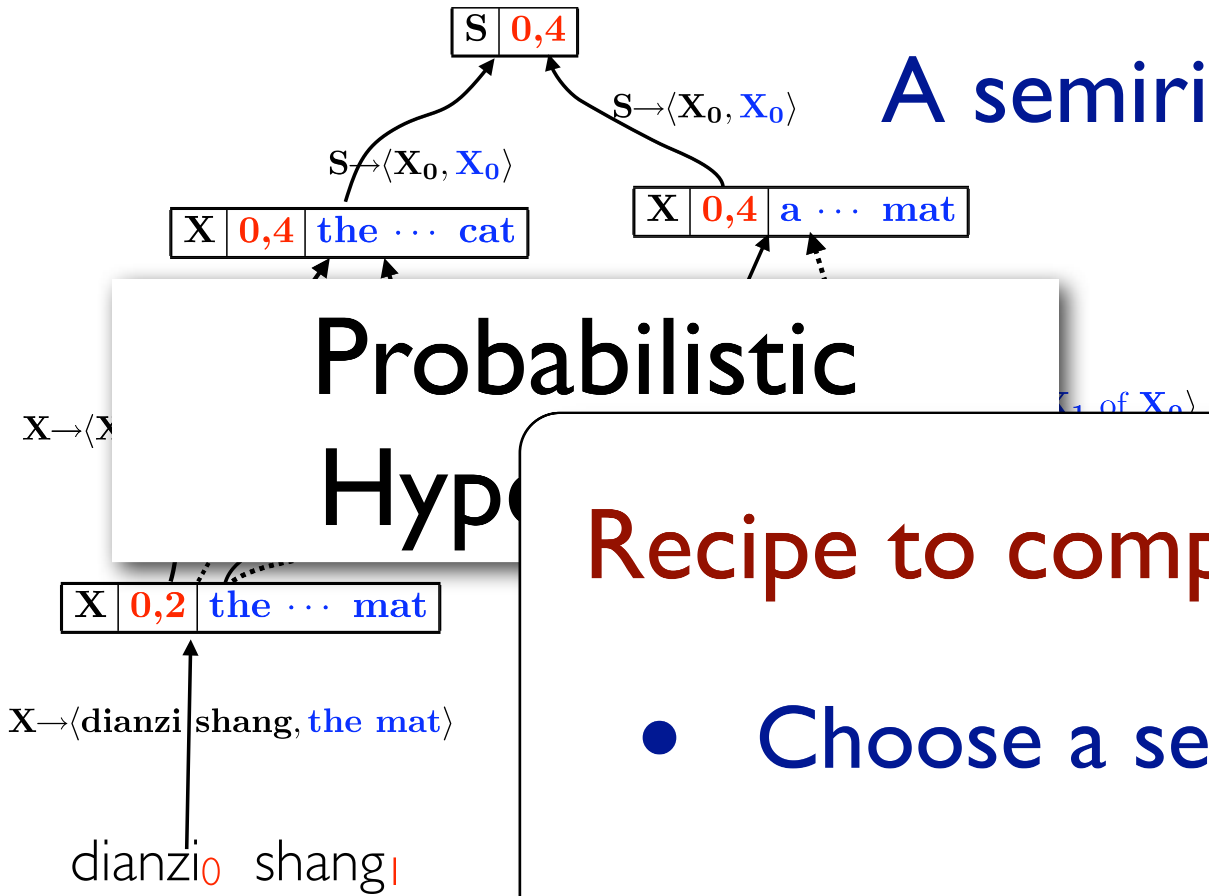
Probabilistic
Hyp

- “Decoding” quantities:

Recipe to compute a quantity:

- First-orde
  - expectat
    - entropy
    - expected loss
    - cross-entropy
    - KL divergence
    - feature expectations
  - first-order gradient of Z

- interaction between features
- Hessian matrix of Z
  - second-order gradient descent
- gradient of expectation
  - gradient of expected loss or entropy



A semiring framework to compute all of these

Probabilistic Hyp

- "Decoding" quantities:

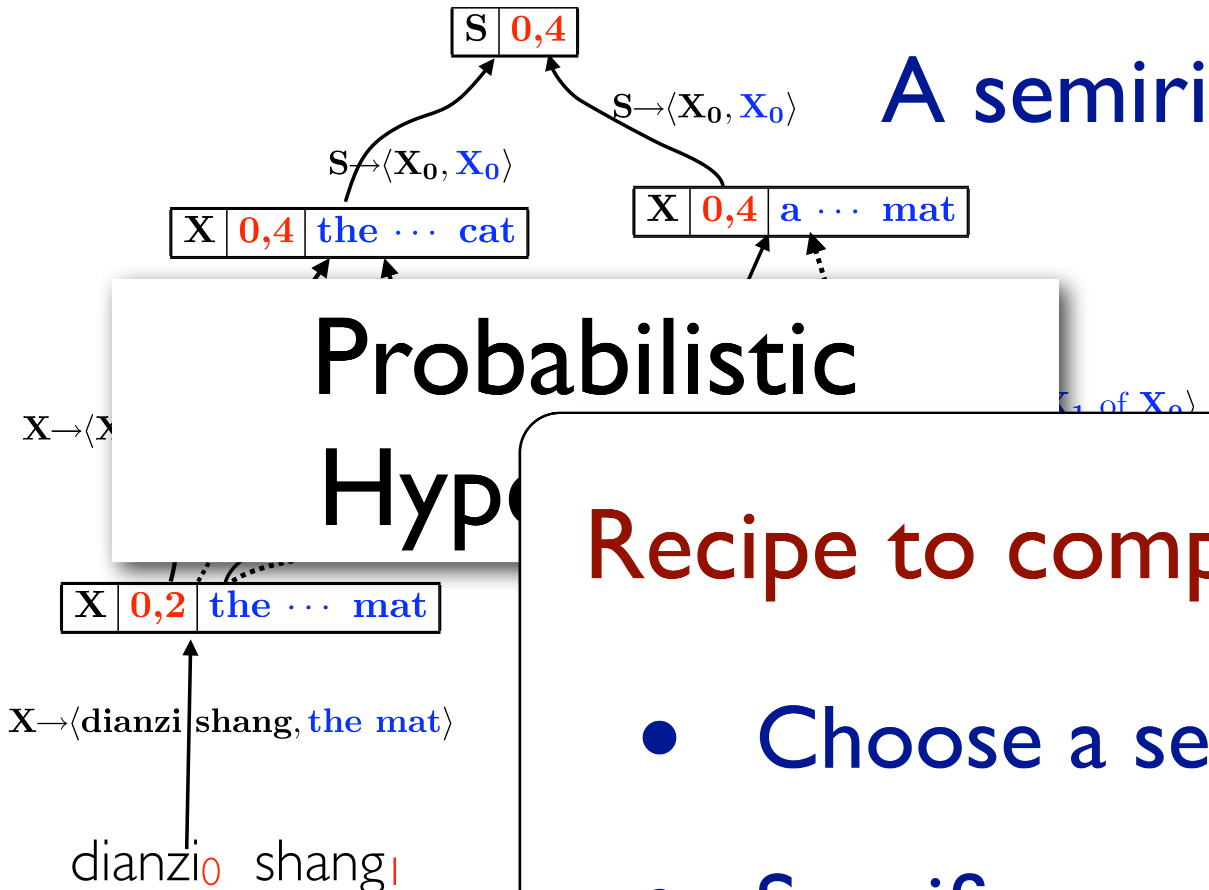
Recipe to compute a quantity:

- Choose a semiring

- First-orde
  - expectat
    - entropy
    - expected loss
    - cross-entropy
    - KL divergence
    - feature expectations
  - first-order gradient of Z

- interaction between features
- Hessian matrix of Z
  - second-order gradient descent
- gradient of expectation
  - gradient of expected loss or entropy

A semiring framework to compute all of these



Probabilistic Hyp

- "Decoding" quantities:

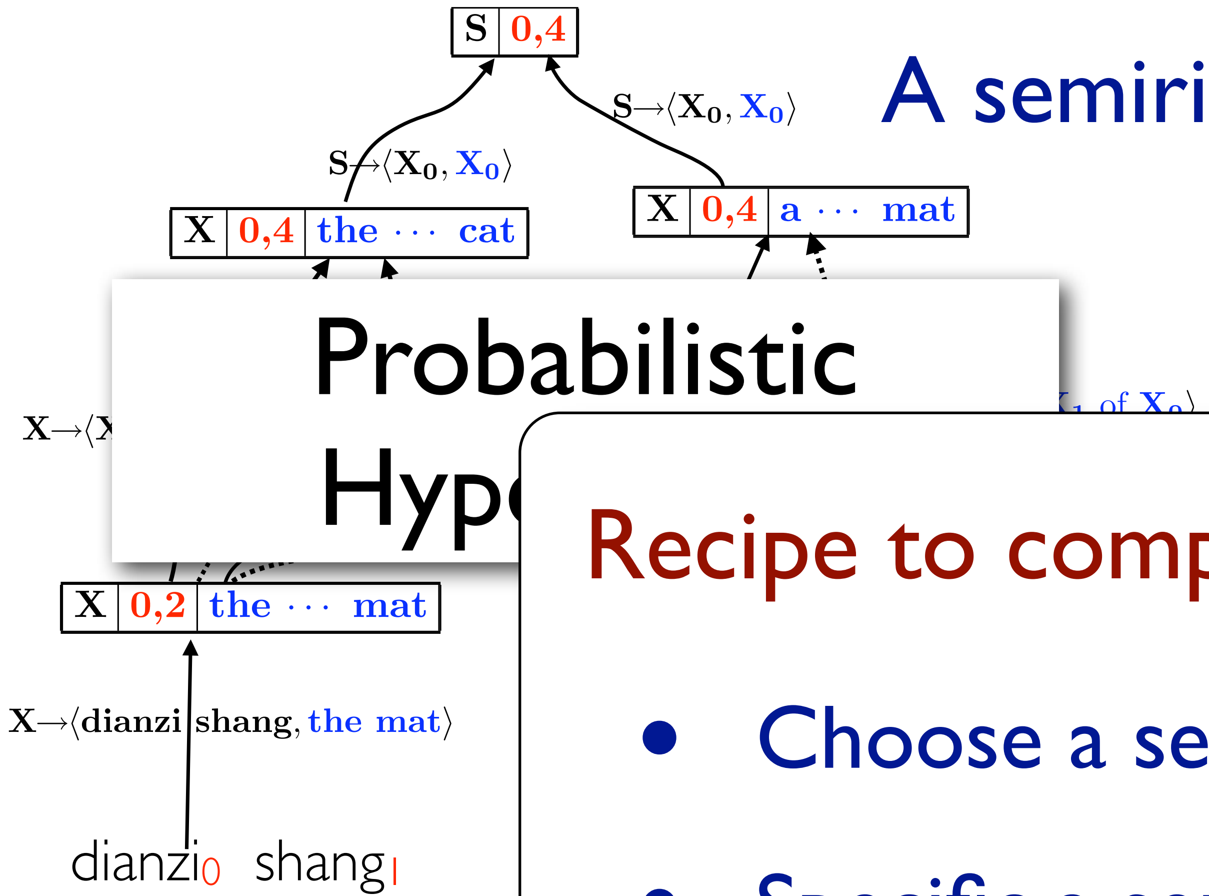
Recipe to compute a quantity:

- Choose a semiring
- Specific a semiring weight for each hyperedge

- First-orde
  - expectat
    - entropy
    - expected loss
    - cross-entropy
    - KL divergence
    - feature expectations
  - first-order gradient of Z

    - interaction between features
  - Hessian matrix of Z
    - second-order gradient descent
  - gradient of expectation
    - gradient of expected loss or entropy

A semiring framework to compute all of these



Probabilistic

Hyp

- "Decoding" quantities:

Recipe to compute a quantity:

- Choose a semiring
- Specific a semiring weight for each hyperedge
- Run the inside algorithm

- First-orde
  - expectat
    - entropy
    - expected loss
    - cross-entropy
    - KL divergence
    - feature expectations
  - first-order gradient of Z

- interaction between features
- Hessian matrix of Z
  - second-order gradient descent
- gradient of expectation
  - gradient of expected loss or entropy

Applications of Expectation Semirings: a Summary

Quantity	k_e	$k_{\textbf{root}}$	**Final**
Expectation	$\langle p_e, p_e r_e \rangle$	$\langle Z, \bar{r} \rangle$	$\bar{r}/Z$
Entropy	$r_e \stackrel{\text{def}}{=} \log p_e$, so $k_e = \langle p_e, p_e \log p_e \rangle$	$\langle Z, \bar{r} \rangle$	$\log Z - \bar{r}/Z$
Cross-entropy	$\langle q_e \rangle$ $r_e \stackrel{\text{def}}{=} \log q_e$, so $k_e = \langle p_e, p_e \log q_e \rangle$	$\langle Z_q \rangle$ $\langle Z_p, \bar{r} \rangle$	$\log Z_q - \bar{r}/Z_p$
Bayes risk	$r_e \stackrel{\text{def}}{=} \mathrm{L}_e$, so $k_e = \langle p_e, p_e \mathrm{L}_e \rangle$	$\langle Z, \bar{r} \rangle$	$\bar{r}/Z$
First-order gradient	$\langle p_e, \nabla p_e \rangle$	$\langle Z, \nabla Z \rangle$	∇Z
Covariance matrix	$\langle p_e, p_e r_e, p_e s_e, p_e r_e s_e \rangle$	$\langle Z, \bar{r}, \bar{s}, \bar{t} \rangle$	$\frac{\bar{t}}{Z} - \frac{\bar{r}\,\bar{s}^{\mathbf{T}}}{Z^2}$
Hessian matrix	$\langle p_e, \nabla p_e, \nabla p_e, \nabla^2 p_e \rangle$	$\langle Z, \nabla Z, \nabla Z, \nabla^2 Z \rangle$	$\nabla^2 Z$
Gradient of expectation	$\langle p_e, p_e r_e, \nabla p_e, (\nabla p_e) r_e + p_e(\nabla r_e) \rangle$	$\langle Z, \bar{r}, \nabla Z, \nabla \bar{r} \rangle$	$\frac{Z \nabla \bar{r} - \bar{r} \nabla Z}{Z^2}$
Gradient of entropy	$\langle p_e, p_e \log p_e, \nabla p_e, (1 + \log p_e) \nabla p_e \rangle$	$\langle Z, \bar{r}, \nabla Z, \nabla \bar{r} \rangle$	$\frac{\nabla Z}{Z} - \frac{Z \nabla \bar{r} - \bar{r} \nabla Z}{Z^2}$
Gradient of risk	$\langle p_e, p_e \mathrm{L}_e, \nabla p_e, \mathrm{L}_e \nabla p_e \rangle$	$\langle Z, \bar{r}, \nabla Z, \nabla \bar{r} \rangle$	$\frac{Z \nabla \bar{r} - \bar{r} \nabla Z}{Z^2}$

Inference, Training and Decoding on Hypergraphs

- Unsupervised Discriminative Training
  - minimum imputed risk (In Preparation)
  - contrastive language model estimation (In Preparation)
- Variational Decoding
  (Li et al., ACL 2009)
- First- and Second-order Expectation Semirings
  (Li and Eisner, EMNLP 2009)

My Other MT Research

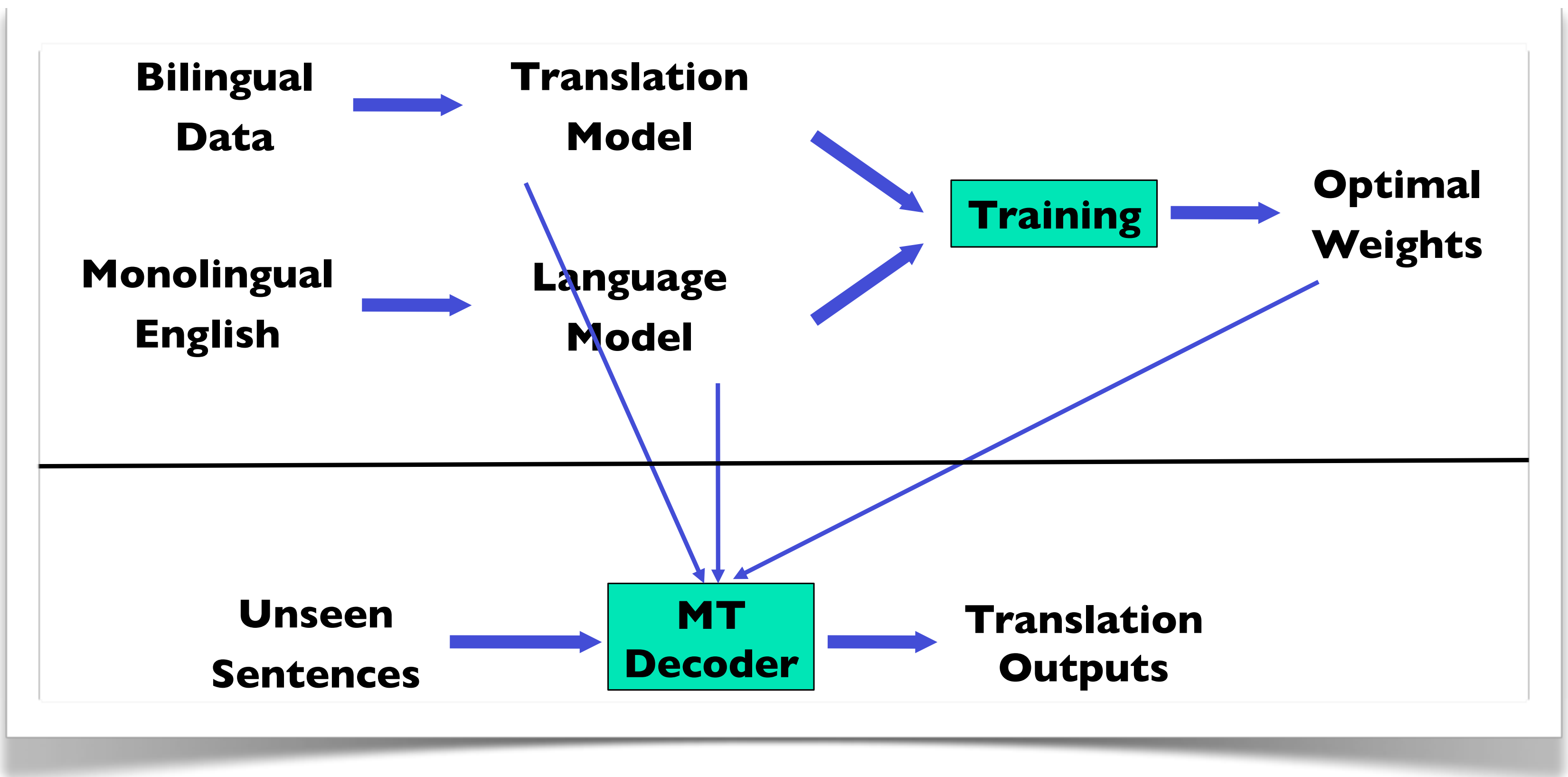
- **Training methods (supervised)**
  - Discriminative forest reranking with Perceptron (Li and Khudanpur, GALE book chapter 2009)
  - Discriminative n-gram language models (Li and Khudanpur, AMTA 2008)
- **Algorithms**
  - Oracle extraction from hypergraphs (Li and Khudanpur, NAACL 2009)
  - Efficient intersection between n-gram LM and CFG (Li and Khudanpur, ACL SSST 2008)
- **Others**
  - System combination (Smith et al., GALE book chapter 2009)
  - Unsupervised translation induction for Chinese abbreviations (Li and Yarowsky, ACL 2008)

Research other than MT

- **Information extraction**
  - Relation extraction between formal and informal phrases (Li and Yarowsky, EMNLP 2008)
- **Spoken dialog management**
  - Optimal dialog in consumer-rating systems using a POMDP (Li et al., SIGDial 2008)

Joshua project

- An open-source parsing-based MT toolkit (Li et al. 2009)
  - support Hiero (Chiang, 2007) and SAMT (Venugopal et al., 2007)
- Team members
  - **Zhifei Li**, Chris Callison-Burch, Chris Dyer, Sanjeev Khudanpur, Wren Thornton, Jonathan Weese, Juri Ganitkevitch, Lane Schwartz, and Omar Zaidan



Only rely on word-aligner and SRI LM!

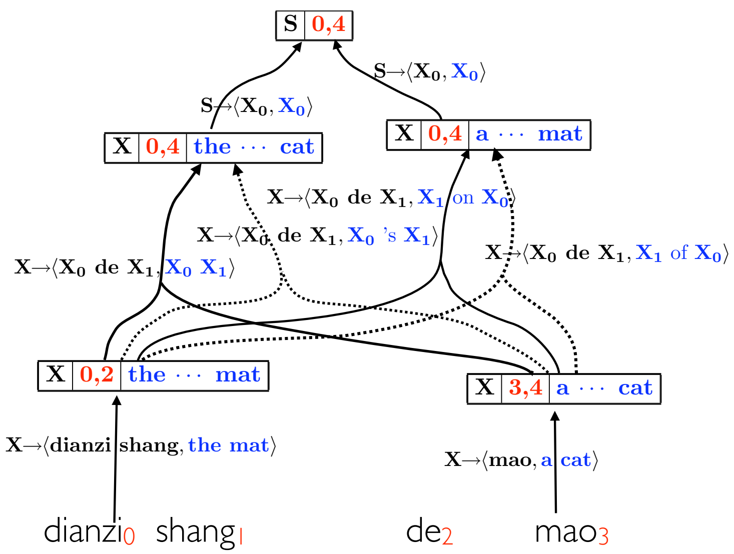
All the methods presented have been implemented in Joshua!

Thank you!

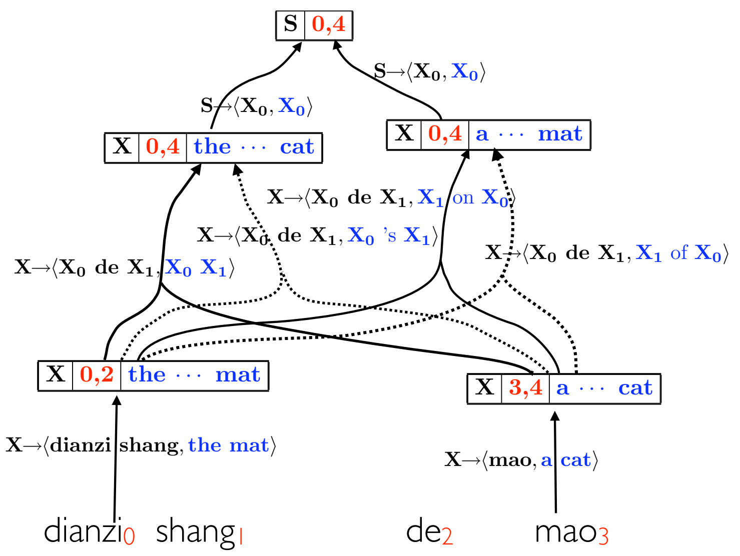
XieXie!

谢谢!

Decoding over a hypergraph



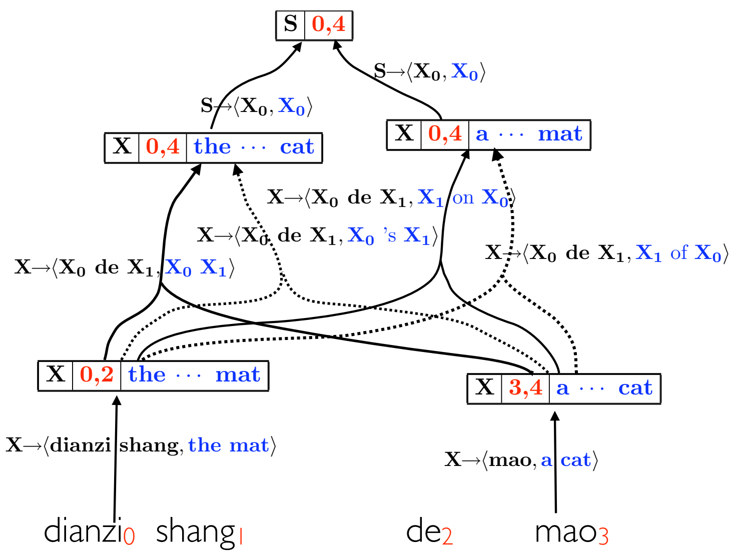
Decoding over a hypergraph



Given a hypergraph of possible translations

(generated for a given foreign sentence by already-trained model)

Decoding over a hypergraph



Given a hypergraph of possible translations

(generated for a given foreign sentence by already-trained model)

Pick a single translation to output

(why not just pick the tree with the highest weight?)

Spurious Ambiguity

- Statistical models in MT exhibit **spurious ambiguity**
  - Many **different derivations** (e.g., trees or segmentations) generate the **same translation string**
- Tree-based MT systems
  - derivation tree ambiguity
- Regular phrase-based MT systems
  - phrase segmentation ambiguity

Spurious Ambiguity in Derivation Trees

Spurious Ambiguity in Derivation Trees

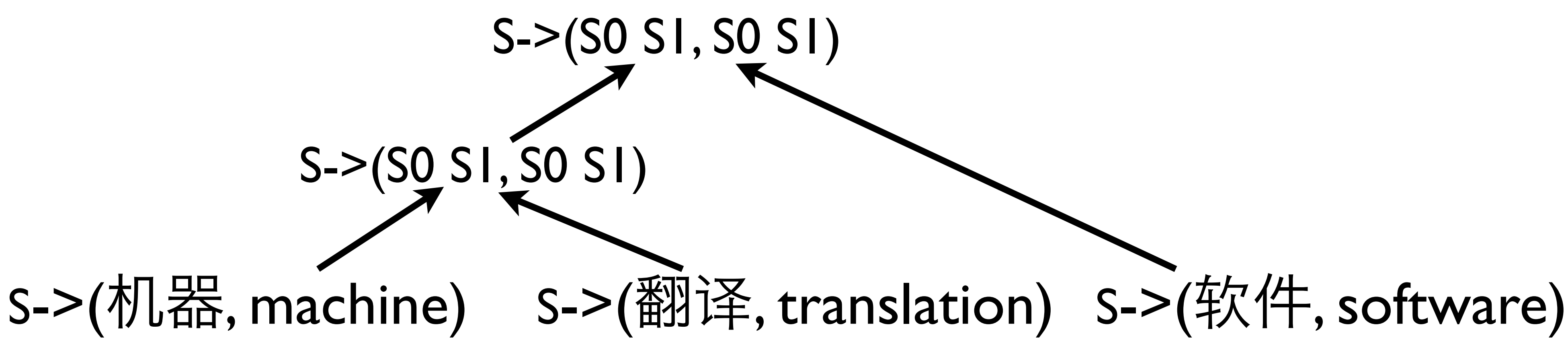
机器 翻译 软件 machine translation software

jiqi fanyi yuanjian

Spurious Ambiguity in Derivation Trees

机器 翻译 软件 machine translation software

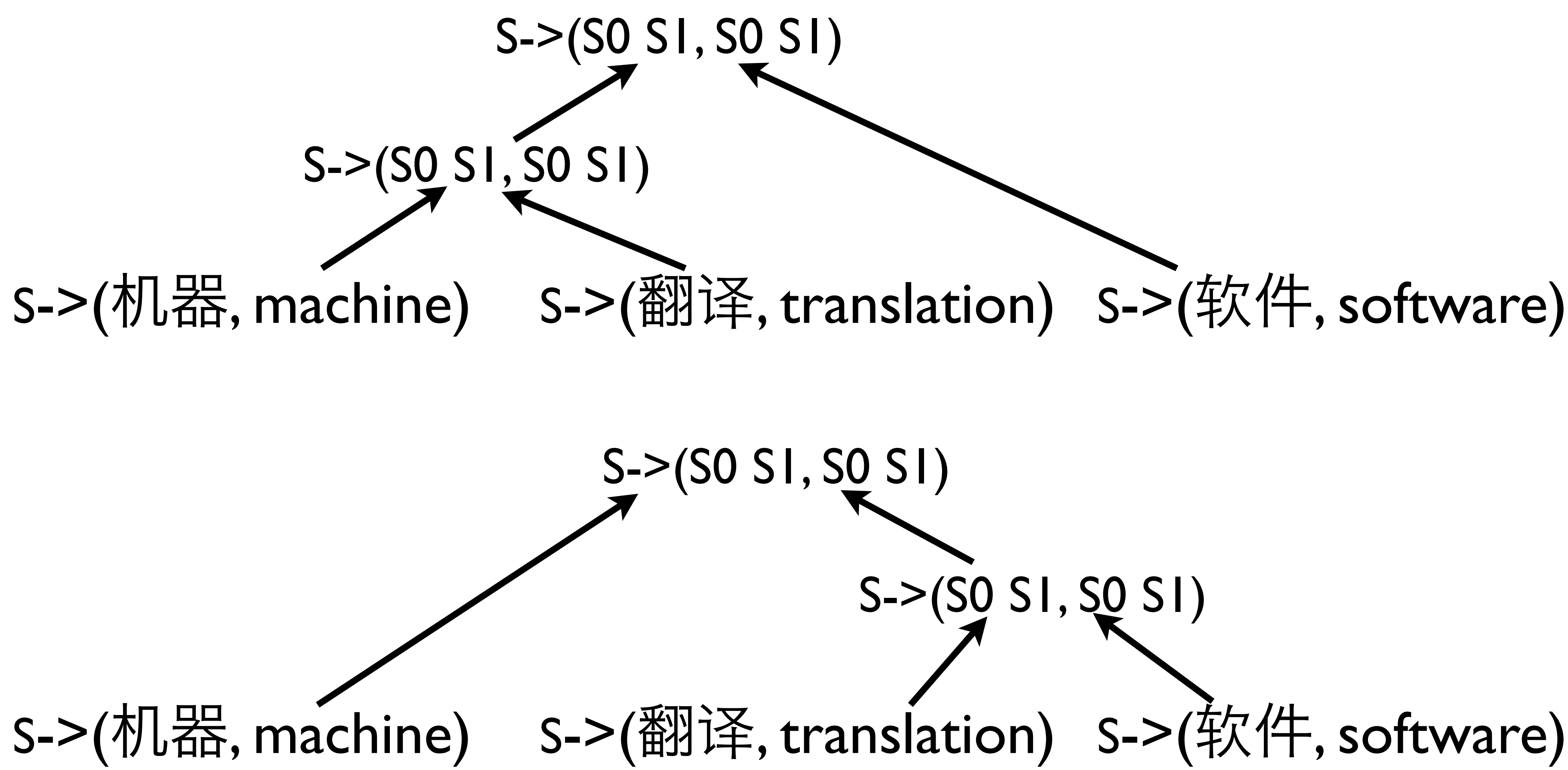
jiqi fanyi yuanjian



Spurious Ambiguity in Derivation Trees

机器 翻译 软件 machine translation software

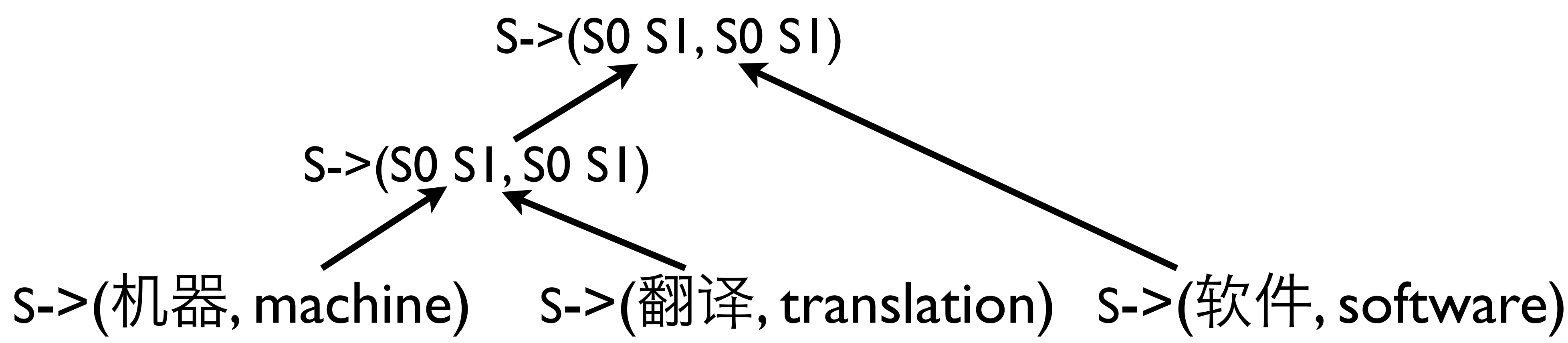
jiqi fanyi yuanjian

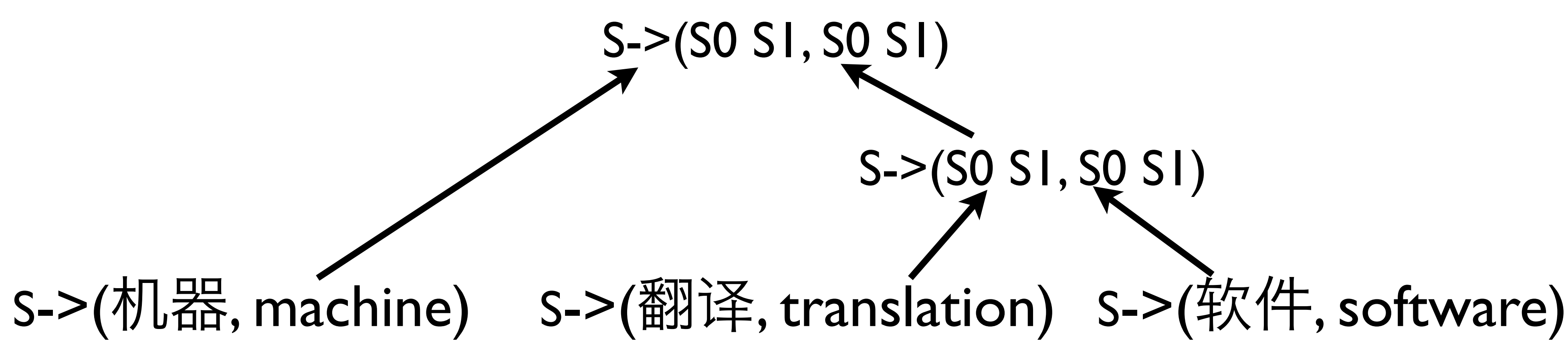


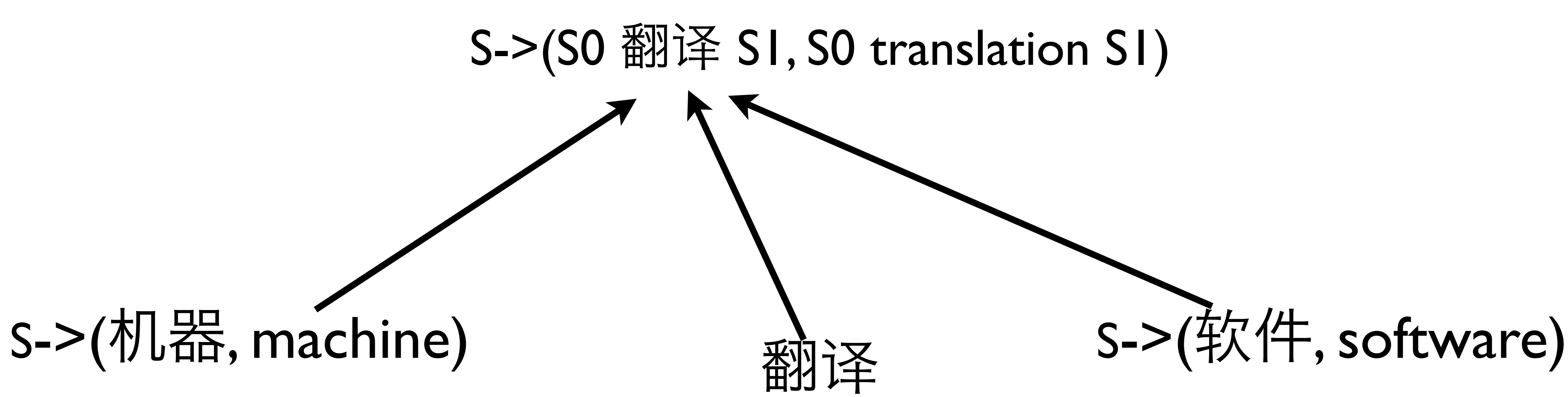
Spurious Ambiguity in Derivation Trees

机器 翻译 软件 machine translation software

jiqi fanyi yuanjian





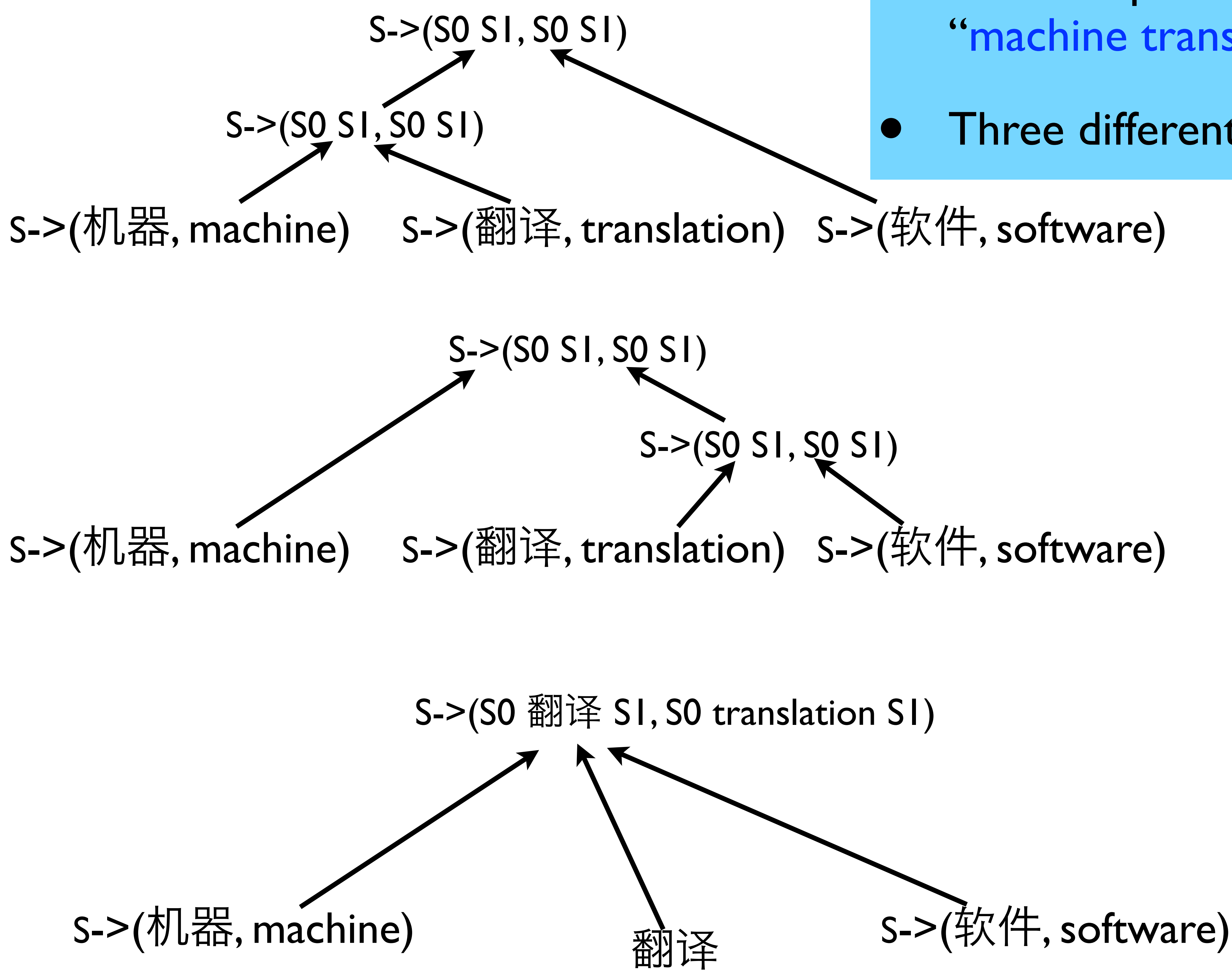


Spurious Ambiguity in Derivation Trees

机器 翻译 软件 machine translation software

jiqi fanyi yuanjian

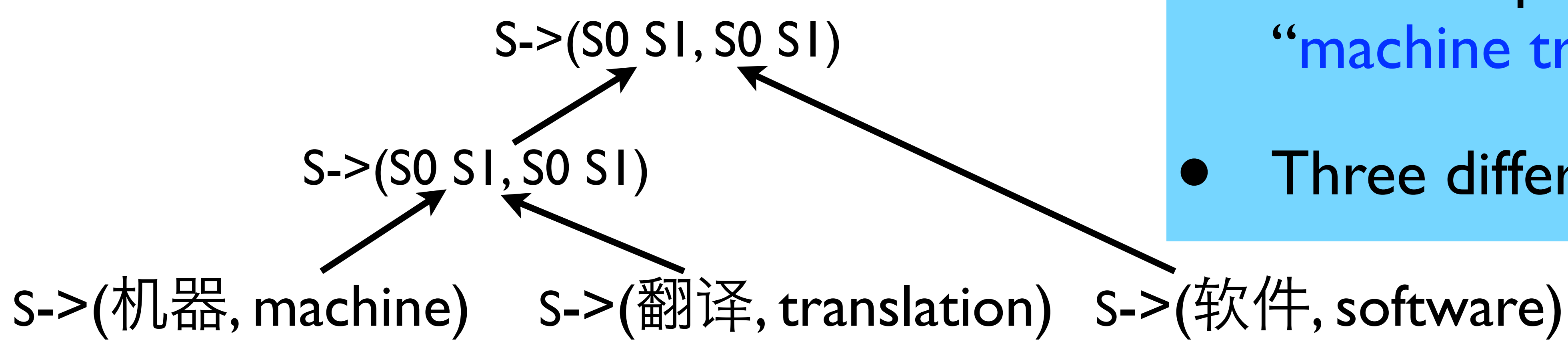
- Same output: "machine translation software"
- Three different derivation trees



Spurious Ambiguity in Derivation Trees

机器 翻译 软件 machine translation software

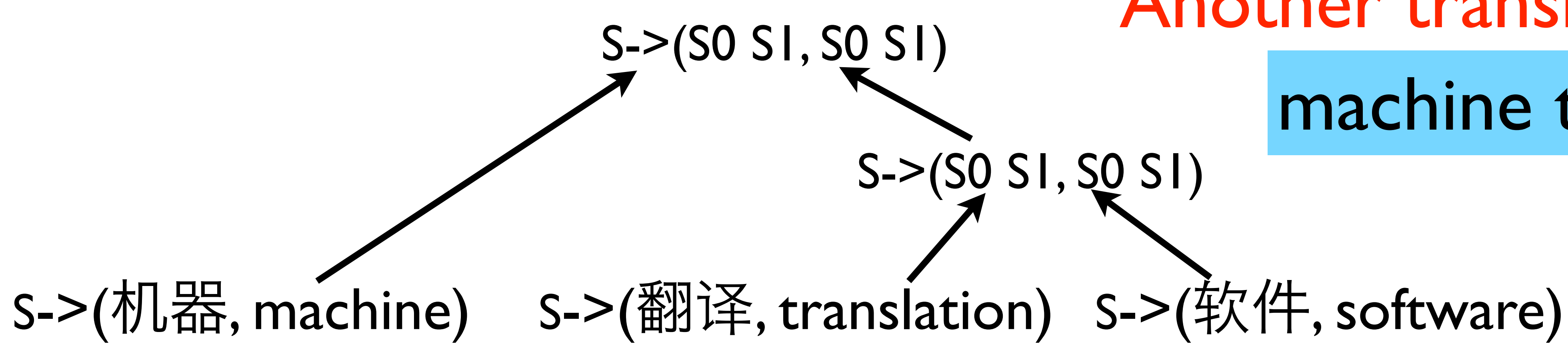
jiqi fanyi yuanjian

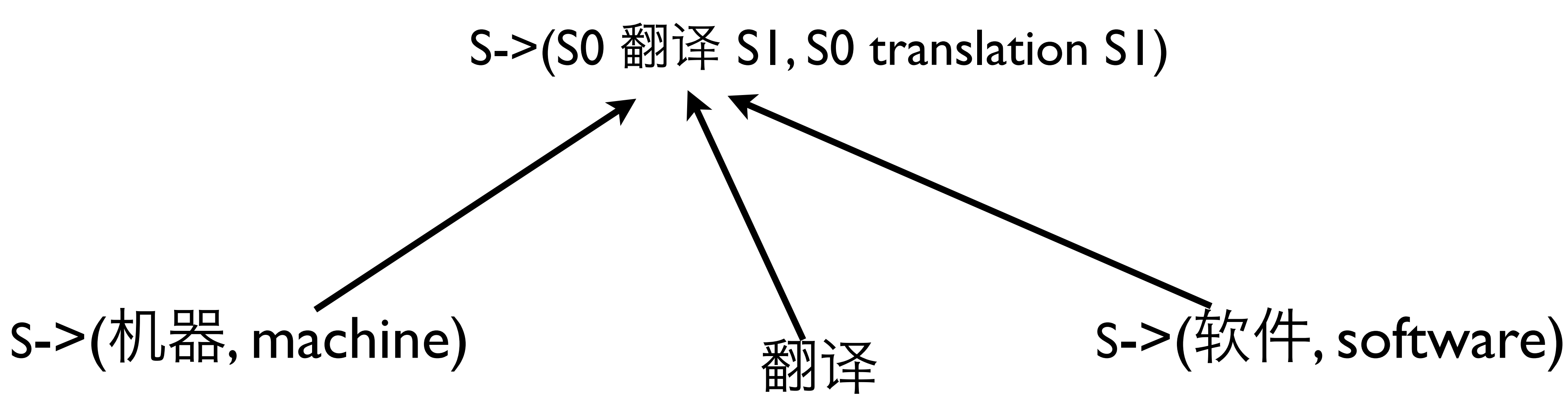


- Same output: "machine translation software"
- Three different derivation trees

Another translation:

machine transfer software





MAP, Viterbi and N-best Approximations

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$\begin{aligned} y^* &= \arg \max_{y \in \text{Trans}(x)} p(y|x) \\ &= \arg \max_{y \in \text{Trans}(x)} \sum_{d \in \text{D}(x,y)} p(y, d|x) \end{aligned}$$

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$
\begin{aligned}
y^* &= \arg\max_{y \in \mathrm{Trans}(x)} p(y|x) \\
&= \arg\max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y)} p(y,d|x)
\end{aligned}
$$

NP-hard (Sima'an 1996)

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$
\begin{aligned}
y^* &= \arg \max_{y \in \mathrm{Trans}(x)} p(y|x) \\
&= \arg \max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y)} p(y, d|x)
\end{aligned}
$$

NP-hard (Sima'an 1996)

- Viterbi approximation

$$
y^* = \arg \max_{y \in \mathrm{Trans}(x)} \max_{d \in \mathrm{D}(x,y)} p(y, d|x)
$$

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$\begin{aligned} y^* &= \arg \max_{y \in \mathrm{Trans}(x)} p(y|x) \\ &= \arg \max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y)} p(y,d|x) \end{aligned}$$

NP-hard (Sima'an 1996)

- Viterbi approximation

$$\begin{aligned} y^* &= \arg \max_{y \in \mathrm{Trans}(x)} \max_{d \in \mathrm{D}(x,y)} p(y,d|x) \\ &= \mathrm{Y}(\arg \max_{d \in \mathrm{D}(x)} p(y,d|x)) \end{aligned}$$

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$y^* = \arg\max_{y \in \mathrm{Trans}(x)} p(y|x)$$

NP-hard (Sima'an 1996)

$$= \arg\max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y)} p(y, d|x)$$

- Viterbi approximation

$$y^* = \arg\max_{y \in \mathrm{Trans}(x)} \max_{d \in \mathrm{D}(x,y)} p(y, d|x)$$

$$= \mathrm{Y}(\arg\max_{d \in \mathrm{D}(x)} p(y, d|x))$$

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$y^* = \arg \max_{y \in \mathrm{Trans}(x)} p(y|x)$$

NP-hard (Sima'an 1996)

$$= \arg \max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y)} p(y, d|x)$$

- Viterbi approximation

$$y^* = \arg \max_{y \in \mathrm{Trans}(x)} \max_{d \in \mathrm{D}(x,y)} p(y, d|x)$$

$$= \mathrm{Y}(\arg \max_{d \in \mathrm{D}(x)} p(y, d|x))$$

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$y^* = \arg \max_{y \in \mathrm{Trans}(x)} p(y|x)$$

NP-hard (Sima'an 1996)

$$= \arg \max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y)} p(y, d|x)$$

- Viterbi approximation

$$y^* = \arg \max_{y \in \mathrm{Trans}(x)} \max_{d \in \mathrm{D}(x,y)} p(y, d|x)$$

$$= \mathrm{Y}(\arg \max_{d \in \mathrm{D}(x)} p(y, d|x))$$

- N-best approximation (crunching) (May and Knight 2006)

$$y^* = \arg \max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y) \cap \mathrm{ND}(x)} p(y, d|x)$$

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$y^* = \arg\max_{y \in \mathrm{Trans}(x)} p(y|x)$$

NP-hard (Sima'an 1996)

$$= \arg\max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y)} p(y,d|x)$$

- Viterbi approximation

$$y^* = \arg\max_{y \in \mathrm{Trans}(x)} \max_{d \in \mathrm{D}(x,y)} p(y,d|x)$$

$$= \mathrm{Y}(\arg\max_{d \in \mathrm{D}(x)} p(y,d|x))$$

- N-best approximation (crunching) (May and Knight 2006)

$$y^* = \arg\max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y) \cap \mathrm{ND}(x)} p(y,d|x)$$

MAP, Viterbi and N-best Approximations

- Exact MAP decoding

$$
\begin{aligned}
y^* &= \arg \max_{y \in \mathrm{Trans}(x)} p(y|x) \\
&= \arg \max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y)} p(y,d|x)
\end{aligned}
$$

NP-hard (Sima'an 1996)

- Viterbi approximation

$$
\begin{aligned}
y^* &= \arg \max_{y \in \mathrm{Trans}(x)} \max_{d \in \mathrm{D}(x,y)} p(y,d|x) \\
&= \mathrm{Y}(\arg \max_{d \in \mathrm{D}(x)} p(y,d|x))
\end{aligned}
$$

- N-best approximation (crunching) (May and Knight 2006)

$$
y^* = \arg \max_{y \in \mathrm{Trans}(x)} \sum_{d \in \mathrm{D}(x,y) \cap \mathrm{ND}(x)} p(y,d|x)
$$

MAP vs. Approximations

translation string	MAP	Viterbi	4-best crunching
red translation	0.28	0.16	0.16
blue translation	0.28	0.14	0.28
green translation	0.44	0.13	0.13

derivation	probability
	0.16
	0.14
	0.14
	0.13
	0.12
	0.11
	0.10
	0.10

MAP vs. Approximations

translation string	MAP	Viterbi	4-best crunching
red translation	0.28	0.16	0.16
blue translation	0.28	0.14	0.28
green translation	0.44	0.13	0.13

derivation	probability
	0.16
	0.14
	0.14
	0.13
	0.12
	0.11
	0.10
	0.10

- Exact MAP decoding under spurious ambiguity is intractable on HG

MAP vs. Approximations

translation string	MAP	Viterbi	4-best crunching
red translation	0.28	0.16	0.16
blue translation	0.28	0.14	0.28
green translation	0.44	0.13	0.13

derivation	probability
	0.16
	0.14
	0.14
	0.13
	0.12
	0.11
	0.10
	0.10

- Exact MAP decoding under spurious ambiguity is intractable on HG
- Viterbi and crunching are efficient, but ignore most derivations

MAP vs. Approximations

translation string	MAP	Viterbi	4-best crunching
red translation	0.28	0.16	0.16
blue translation	0.28	0.14	0.28
green translation	0.44	0.13	0.13

derivation	probability
red	0.16
blue	0.14
blue	0.14
green	0.13
red	0.12
green	0.11
green	0.10
green	0.10

- Exact MAP decoding under spurious ambiguity is intractable on HG
- Viterbi and crunching are efficient, but ignore most derivations
- Our goal: develop an approximation that considers all the derivations but still allows tractable decoding

Variational Decoding

Variational Decoding

Decoding using **Variational** approximation

Decoding using a sentence-specific **approximate** distribution

Variational Decoding for MT: an Overview

Variational Decoding for MT: an Overview

Sentence-specific decoding

Variational Decoding for MT: an Overview

Sentence-specific decoding

Three steps:

Variational Decoding for MT: an Overview

Sentence-specific decoding

Three steps:

1 Generate a hypergraph for the foreign sentence

Variational Decoding for MT: an Overview

Sentence-specific decoding

Three steps:

1 Generate a hypergraph for the foreign sentence

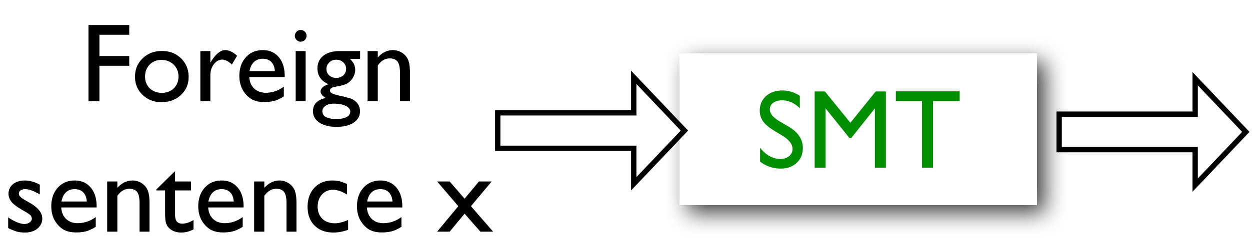
Foreign sentence x

Variational Decoding for MT: an Overview

Sentence-specific decoding

Three steps:

1. Generate a hypergraph for the foreign sentence



Variational Decoding for MT: an Overview

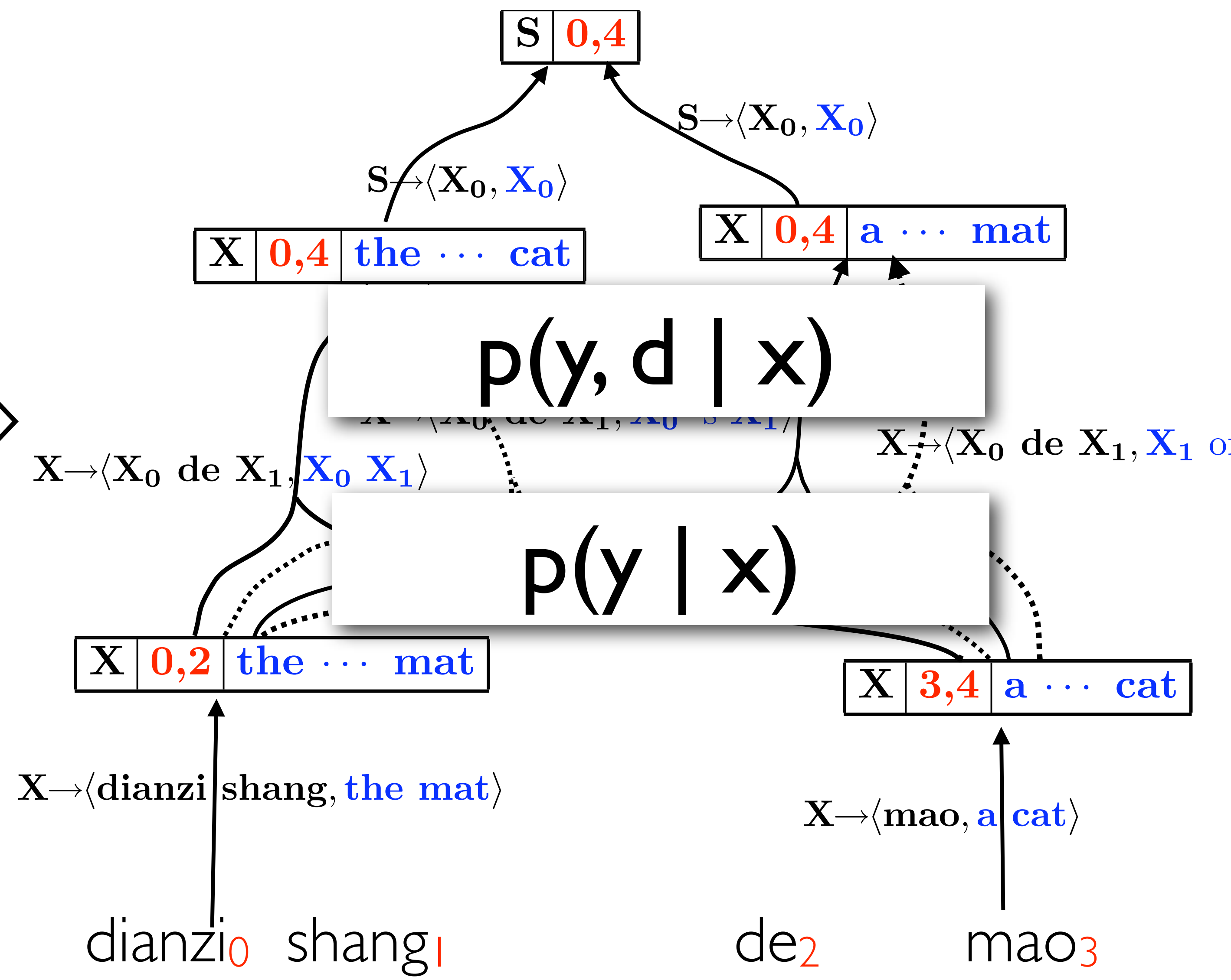
Sentence-specific decoding

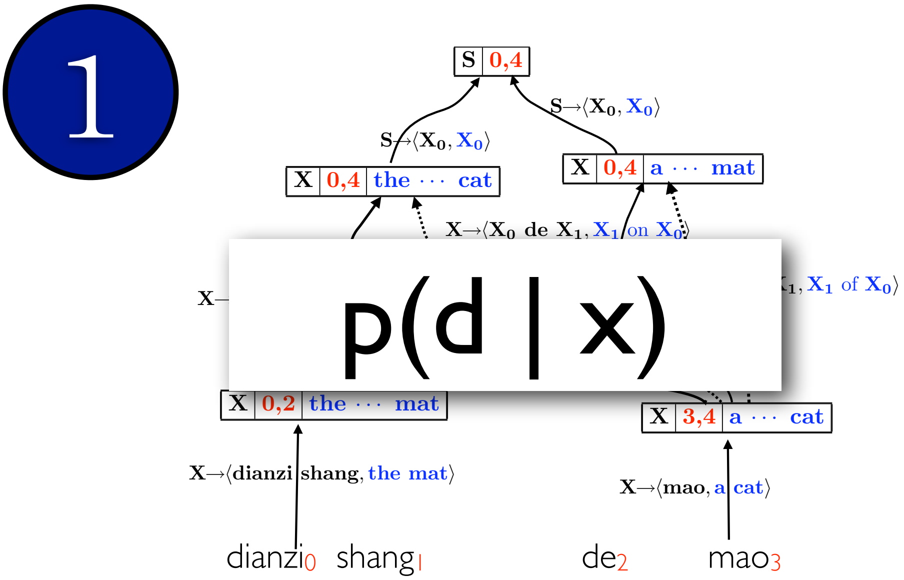
MAP decoding under P is intractable

Three steps:

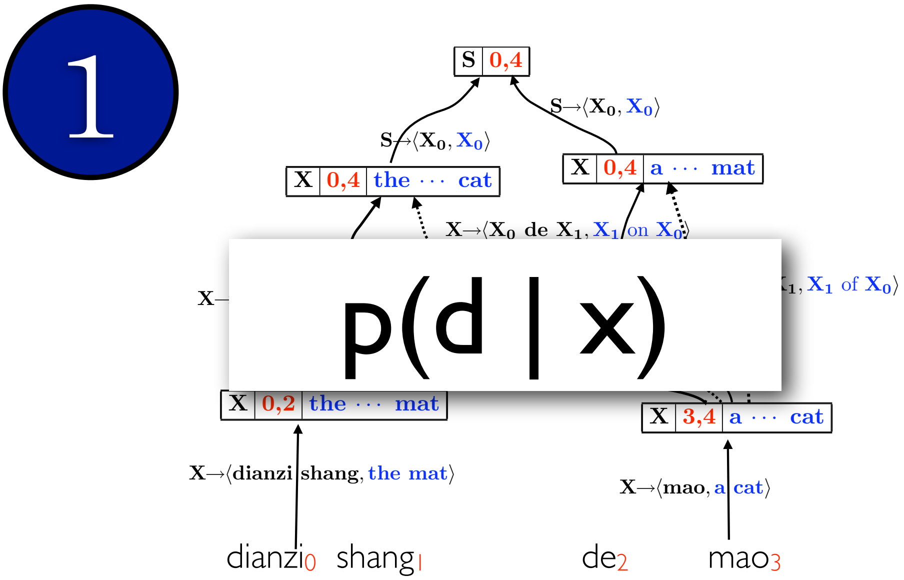
1. Generate a hypergraph for the foreign sentence

Foreign sentence x ⇒ SMT ⇒



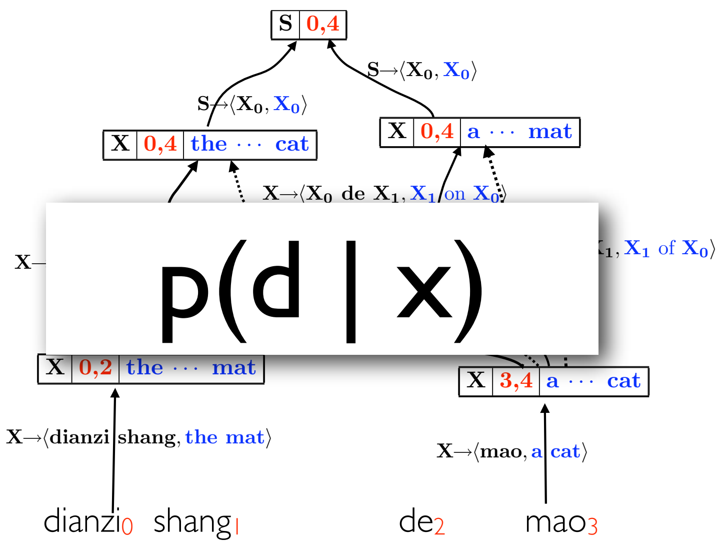


Generate a hypergraph



Generate a hypergraph

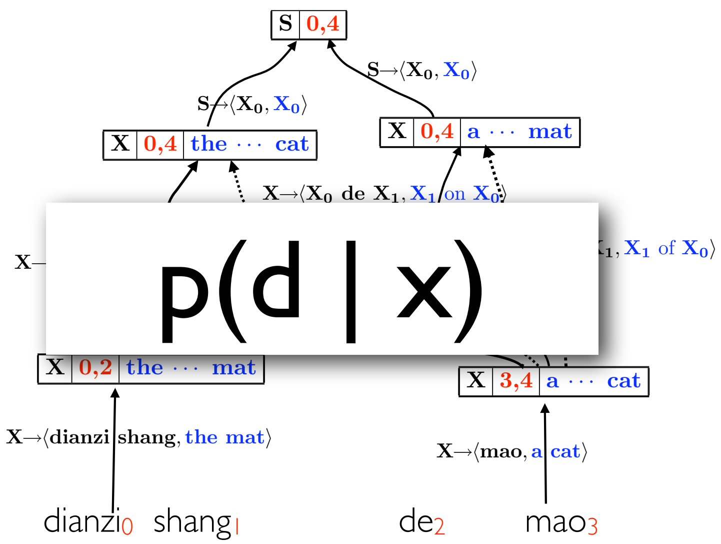
1



Generate a hypergraph

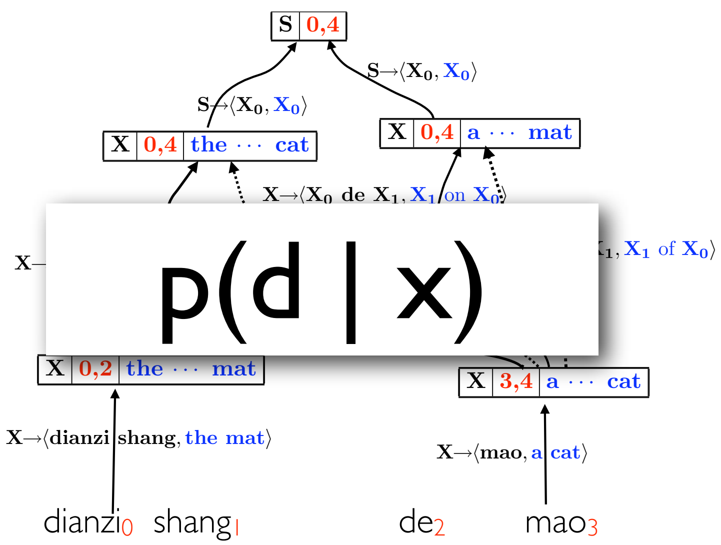
2

1

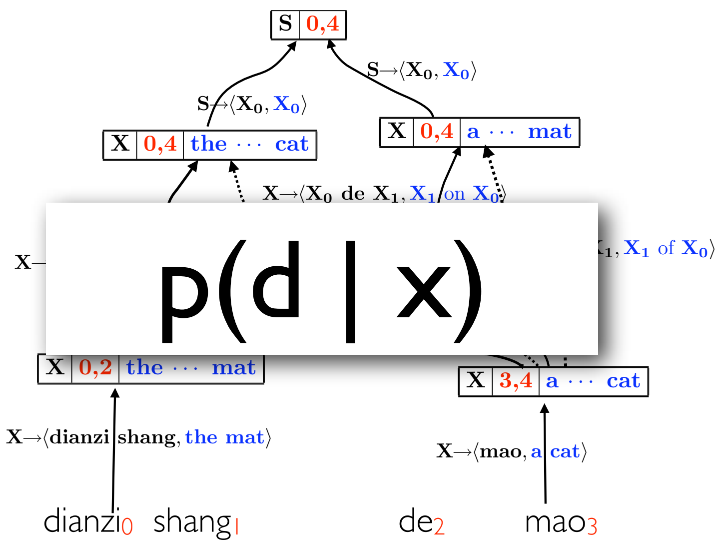


Generate a hypergraph

2

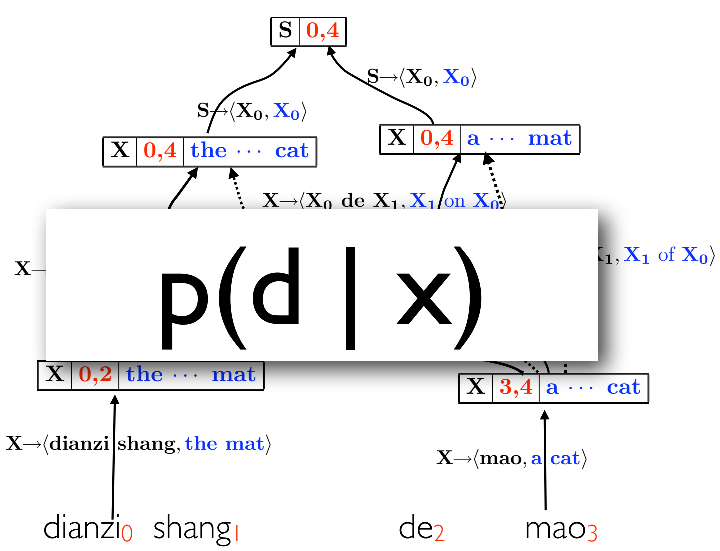


1



Generate a hypergraph

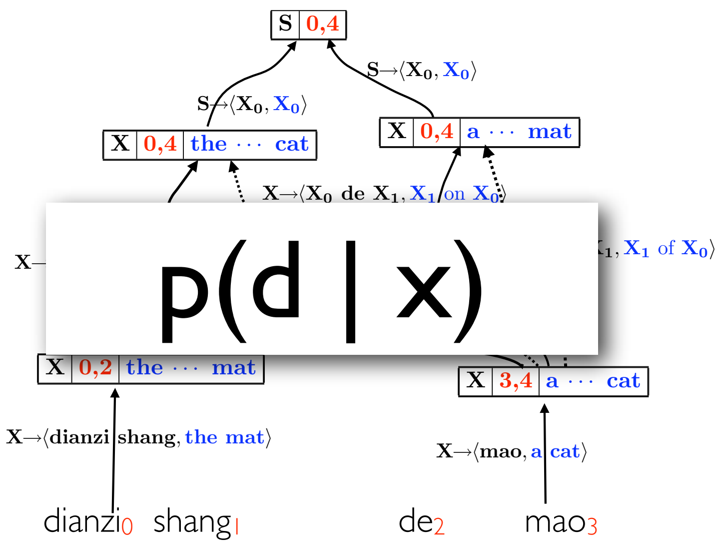
2



Estimate a model from the hypergraph

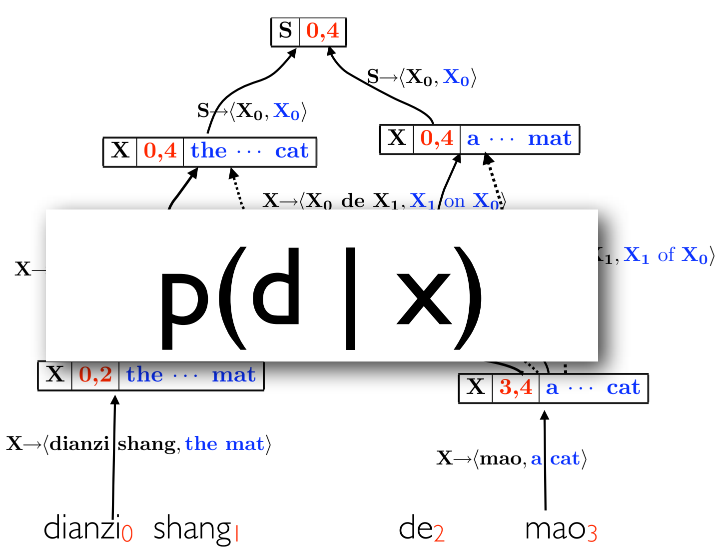
$q^*(y \mid x)$

1



Generate a hypergraph

2

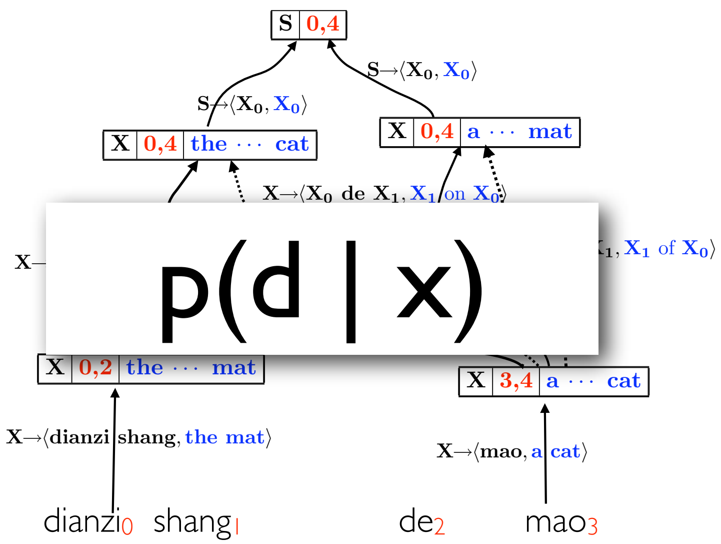


Estimate a model from the hypergraph

q* is an n-gram model over output strings.

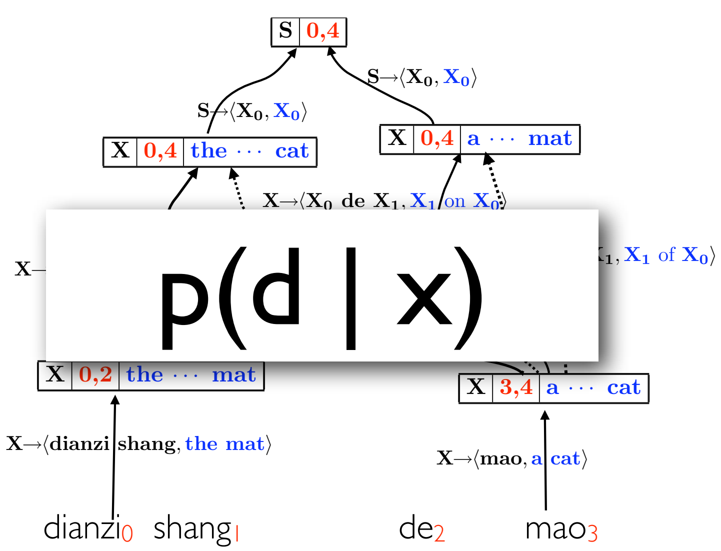
$$q^*(y \mid x)$$

1



Generate a hypergraph

2



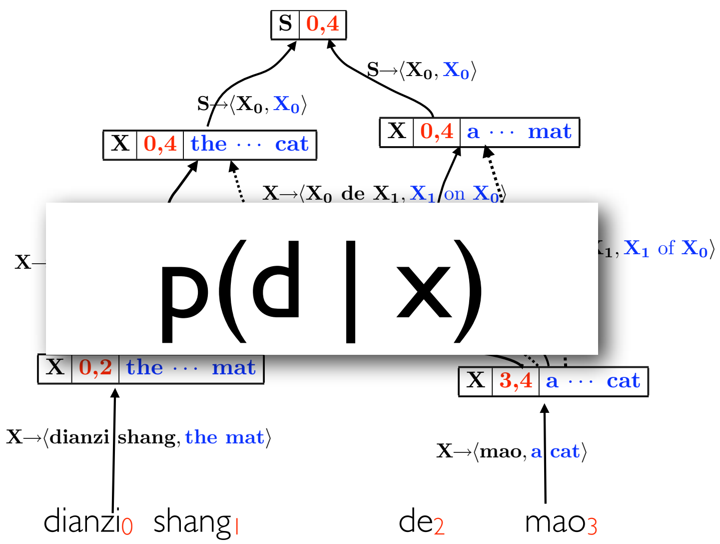
Estimate a model from the hypergraph

q* is an n-gram model over output strings.

$$q^*(y \mid x)$$

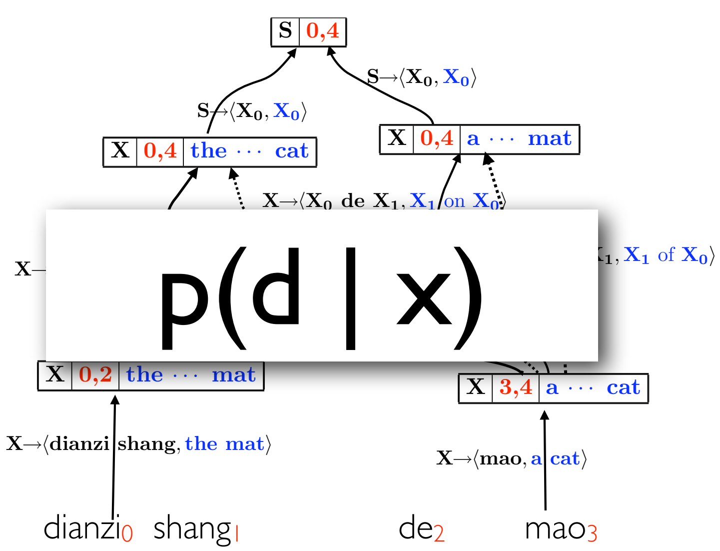
$$\approx \sum_{d \in D(x,y)} p(d|x)$$

1



Generate a hypergraph

2



Estimate a model from the hypergraph ⟹

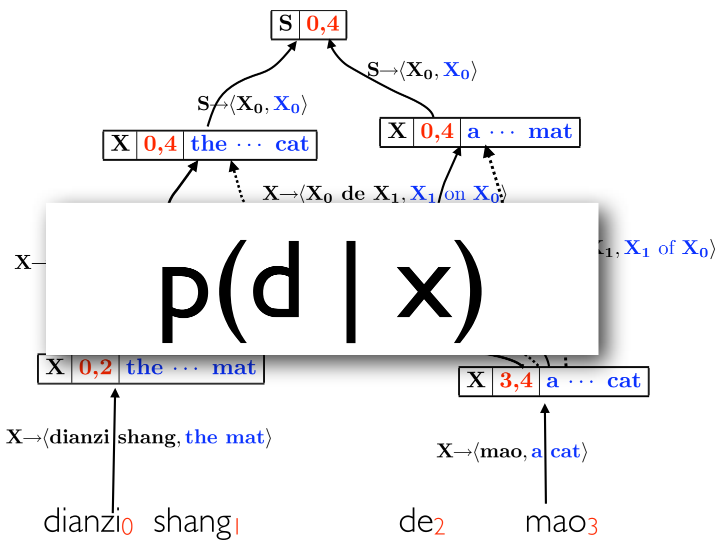
q* is an n-gram model over output strings.

$$q^*(y \mid x)$$

$$\approx \sum_{d \in D(x,y)} p(d|x)$$

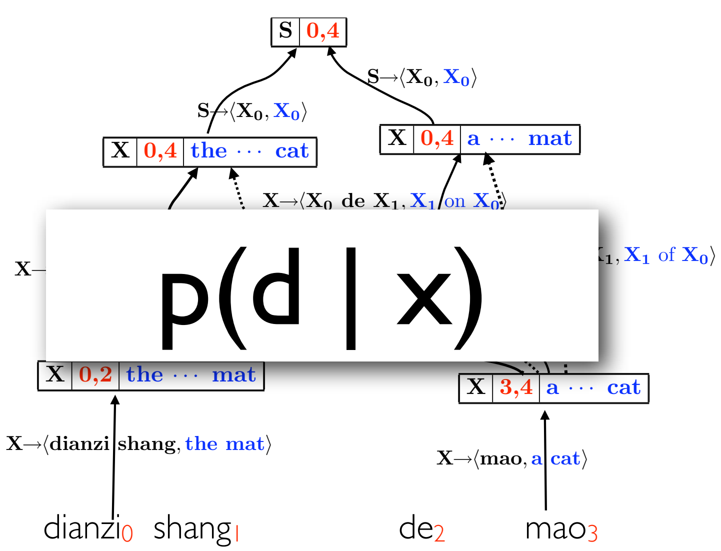
3

1



Generate a hypergraph

2



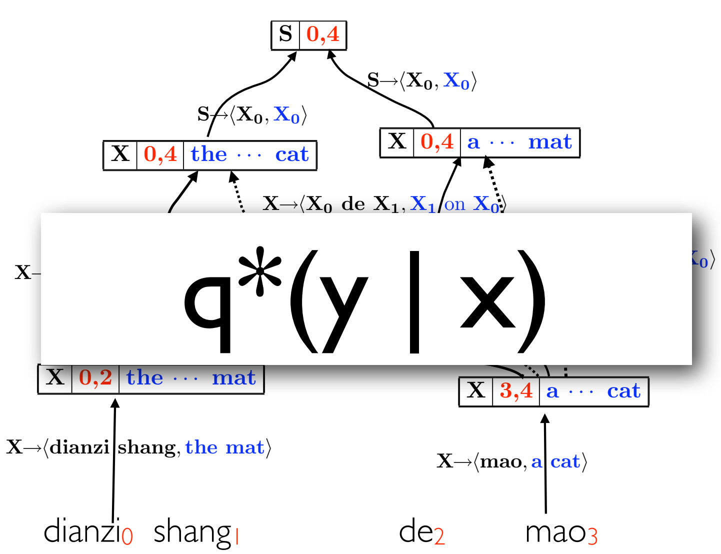
Estimate a model from the hypergraph

q* is an n-gram model over output strings.

$$q^*(y \mid x)$$

$$\approx \sum_{d \in D(x,y)} p(d|x)$$

3



Decode using q* on the hypergraph

Variational Inference

Variational Inference

- We want to do inference under p, but it is intractable

Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$

Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$

- Then, we will use q* as a surrogate for p in inference

Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$

- Then, we will use q* as a surrogate for p in inference

$$y^* = \arg\max_y q^*(y \mid x)$$

Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$

- Then, we will use q* as a surrogate for p in inference

$$y^* = \arg\max_y q^*(y \mid x)$$

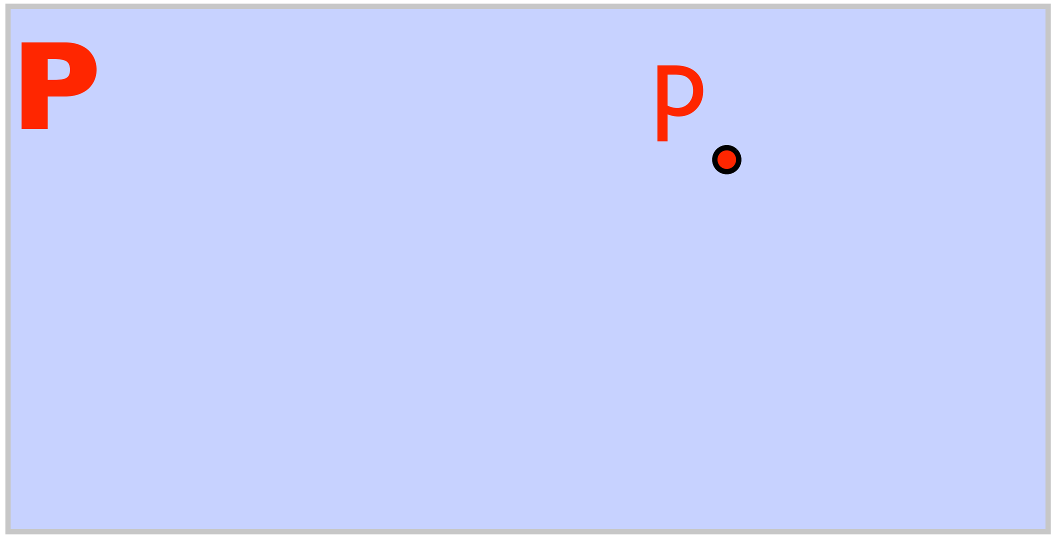
Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \text{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

$$y^* = \arg\max_y q^*(y \mid x)$$

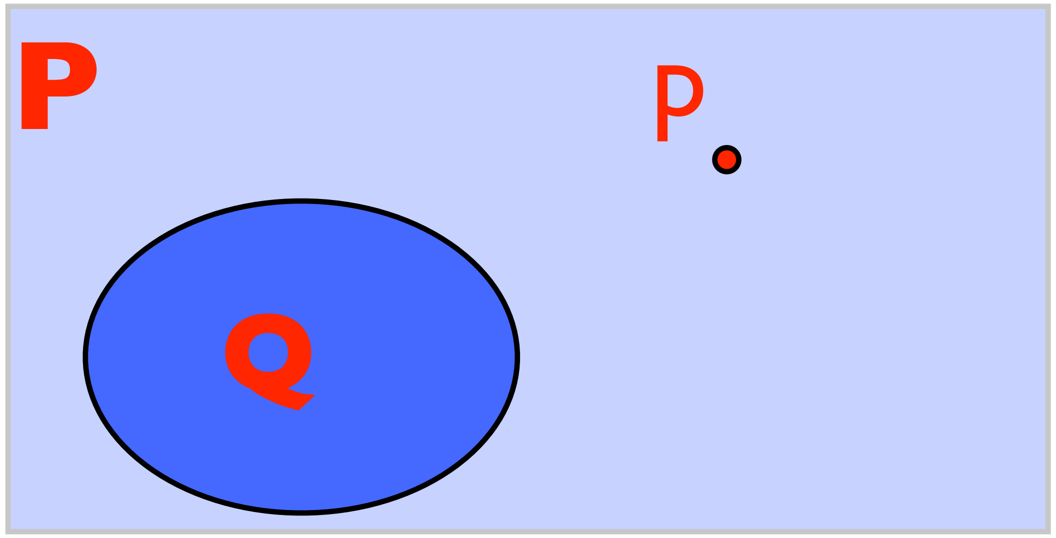
Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \text{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

$$y^* = \arg\max_y q^*(y \mid x)$$

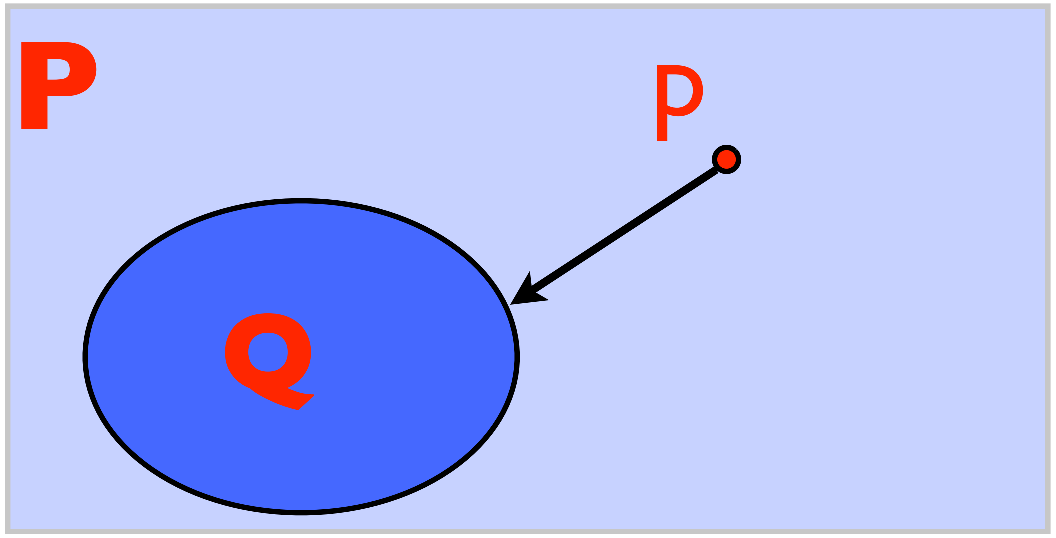
Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

$$y^* = \arg\max_y q^*(y \mid x)$$

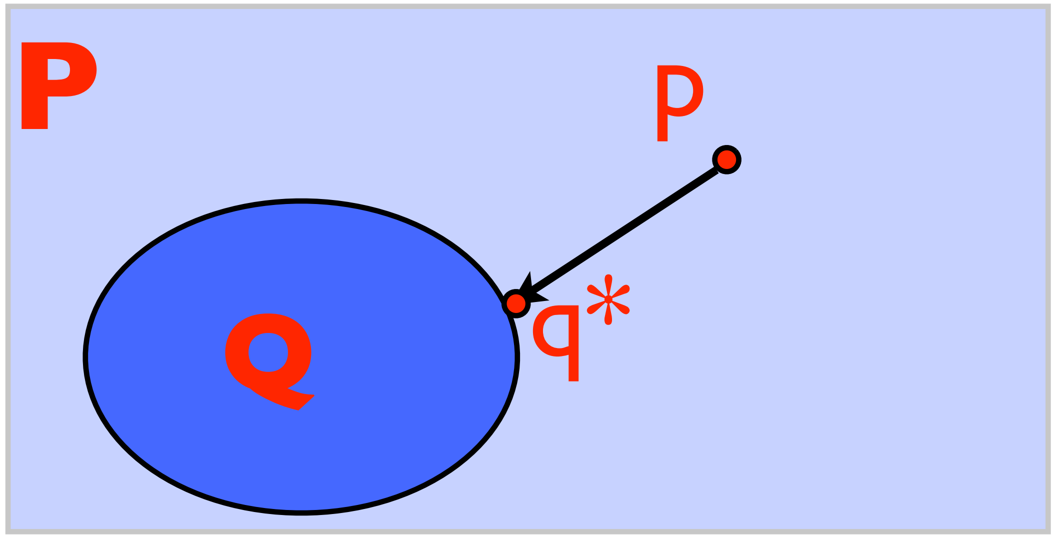
Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

$$y^* = \arg\max_y q^*(y \mid x)$$

Variational Approximation

- q*: an approximation having minimum distance to p

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$

Q → a family of distributions

Variational Approximation

- q*: an approximation having minimum distance to p

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$

→ a family of distributions

$$= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q}$$

Variational Approximation

- q^*: an approximation having minimum distance to p

$$
\begin{aligned}
q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q)
\end{aligned}
$$

Q → a family of distributions

Variational Approximation

- q*: an approximation having minimum distance to p

$$
\begin{aligned}
q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q)
\end{aligned}
$$

a family of distributions

constant

Variational Approximation

- q*: an approximation having minimum distance to p

$$
\begin{aligned}
q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q) \\
&= \arg\max_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log q
\end{aligned}
$$

Q → a family of distributions

$p \log p$ → constant

Variational Approximation

- q*: an approximation having minimum distance to p

$$
\begin{aligned}
q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q) \\
&= \arg\max_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log q
\end{aligned}
$$

Q → a family of distributions

$p \log p$ → constant

- Three questions

Variational Approximation

- q^*: an approximation having minimum distance to p

$$\begin{aligned} q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\ &= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p\log\frac{p}{q} \\ &= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p\log p - p\log q) \\ &= \arg\max_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p\log q \end{aligned}$$

Q: a family of distributions

$p\log p$: constant

- Three questions
  - how to parameterize q?

Variational Approximation

- q*: an approximation having minimum distance to p

$$
\begin{aligned}
q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q) \\
&= \arg\max_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log q
\end{aligned}
$$

Q → a family of distributions

$p \log p$ → constant

- Three questions
  - how to parameterize q?
  - how to estimate q*?

Variational Approximation

- q*: an approximation having minimum distance to p

$$
\begin{aligned}
q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q) \\
&= \arg\max_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log q
\end{aligned}
$$

Q → a family of distributions

$p \log p$ → constant

- Three questions
  - how to parameterize q?
  - how to estimate q*?
  - how to use q* for decoding?

Variational Approximation

- q*: an approximation having minimum distance to p

$$
\begin{aligned}
q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q) \\
&= \arg\max_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log q
\end{aligned}
$$

Q → a family of distributions

$p \log p$ → constant

- Three questions
  - how to parameterize q? — an n-gram model
  - how to estimate q*?
  - how to use q* for decoding?

Variational Approximation

- q*: an approximation having minimum distance to p

$$
\begin{aligned}
q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\
&= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q) \\
&= \arg\max_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log q
\end{aligned}
$$

Q → a family of distributions

$p \log p$ → constant

- Three questions
  - how to parameterize q? — an n-gram model
  - how to estimate q*? — compute expected n-gram counts and normalize
  - how to use q* for decoding?

Variational Approximation

- q*: an approximation having minimum distance to p

$$\begin{aligned} q^* &= \arg\min_{q \in Q} \mathrm{KL}(p||q) \\ &= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log \frac{p}{q} \\ &= \arg\min_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} (p \log p - p \log q) \\ &= \arg\max_{q \in Q} \sum_{y \in \mathrm{Trans}(x)} p \log q \end{aligned}$$

Q → a family of distributions

$p \log p$ → constant

- Three questions
  - how to parameterize q? — an n-gram model
  - how to estimate q*? — compute expected n-gram counts and normalize
  - how to use q* for decoding? — score the hypergraph with the n-gram model

KL divergences under different variational models

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q) = \mathrm{H}(p,q) - \mathrm{H}(p)$$

KL divergences under different variational models

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q) = \mathrm{H}(p,q) - \mathrm{H}(p)$$

Measure bits/word	$\overline{\mathrm{H}}(p)$	$\overline{\mathrm{KL}}(p\|\|\cdot)$ q_1^*	q_2^*	q_3^*	q_4^*
MT'04	1.36	0.97	0.32	0.21	0.17
MT'05	1.37	0.94	0.32	0.21	0.17

KL divergences under different variational models

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q) = \mathrm{H}(p,q) - \mathrm{H}(p)$$

Measure bits/word	$\overline{\mathrm{H}}(p)$	$\overline{\mathrm{KL}}(p\|\|\cdot)$			
		q_1^*	q_2^*	q_3^*	q_4^*
MT'04	1.36	0.97	0.32	0.21	0.17
MT'05	1.37	0.94	0.32	0.21	0.17

- Larger n ==> better approximation q_n ==> smaller KL divergence from p
- The reduction of KL divergence happens mostly when switching from unigram to bigram

BLEU Results on Chinese-English NIST MT 2004 Tasks

	Decoding scheme	BLEU
	Viterbi	35.4
(Kumar and Byrne, 2004)	MBR (K=1000)	35.8
(May and Knight, 2006)	Crunching (N=10000)	35.7
	Crunching+MBR (N=10000)	35.8
New!	Variational (1to4gram+wp+vt)	**36.6**

- variational decoding improves over Viterbi, MBR, and crunching

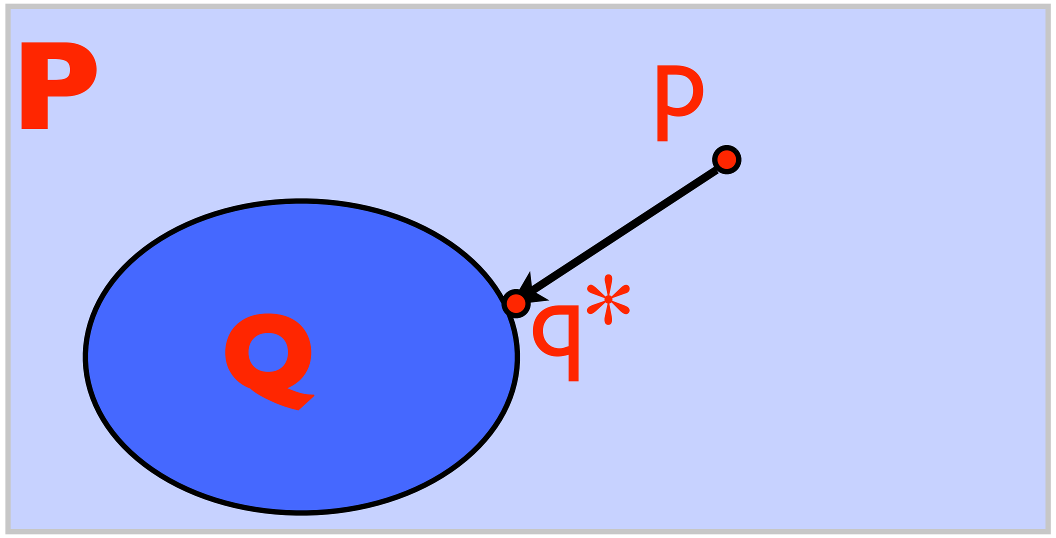
Variational Inference

- We want to do inference under p, but it is intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

$$y^* = \arg\max_y q^*(y \mid x)$$

Variational Inference

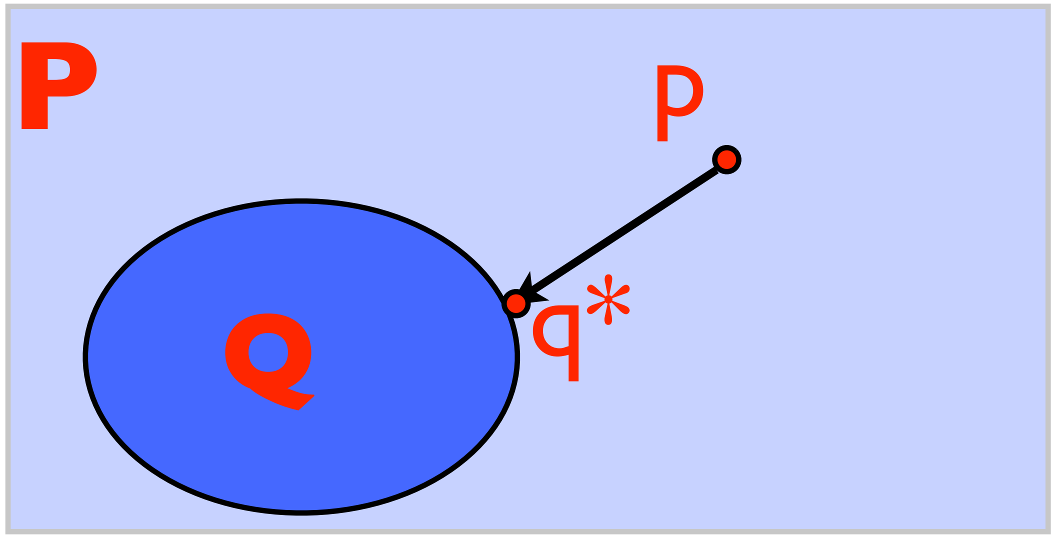
- We want to do inference under p, but it is intractable

intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

$$y^* = \arg\max_y q^*(y \mid x)$$

Variational Inference

- We want to do inference under p, but it is intractable

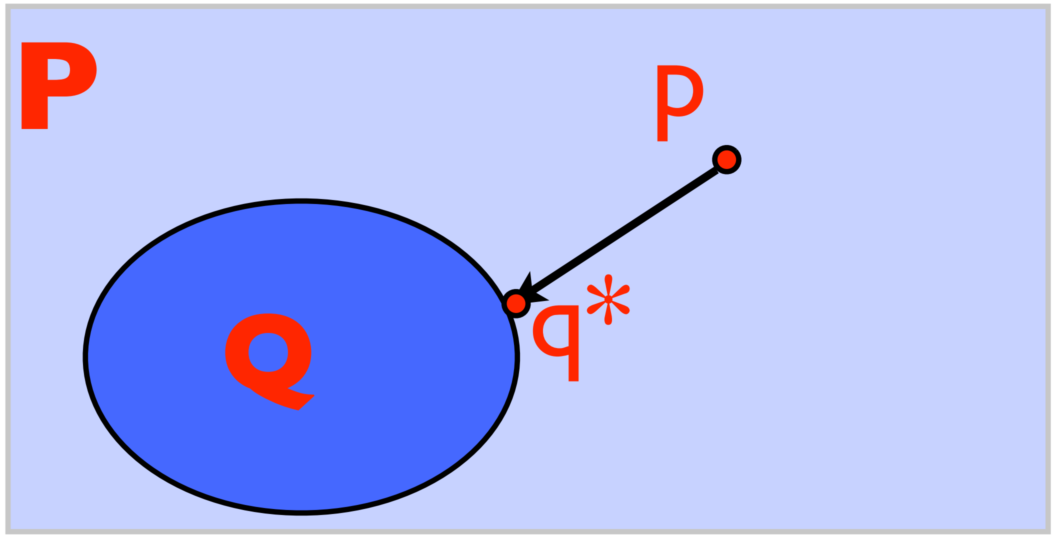
intractable

$$y^* = \arg\max_y p(y|x)$$

- Instead, we derive a simpler distribution q*

tractable

$$q^* = \arg\min_{q \in Q} \mathrm{KL}(p||q)$$



- Then, we will use q* as a surrogate for p in inference

tractable

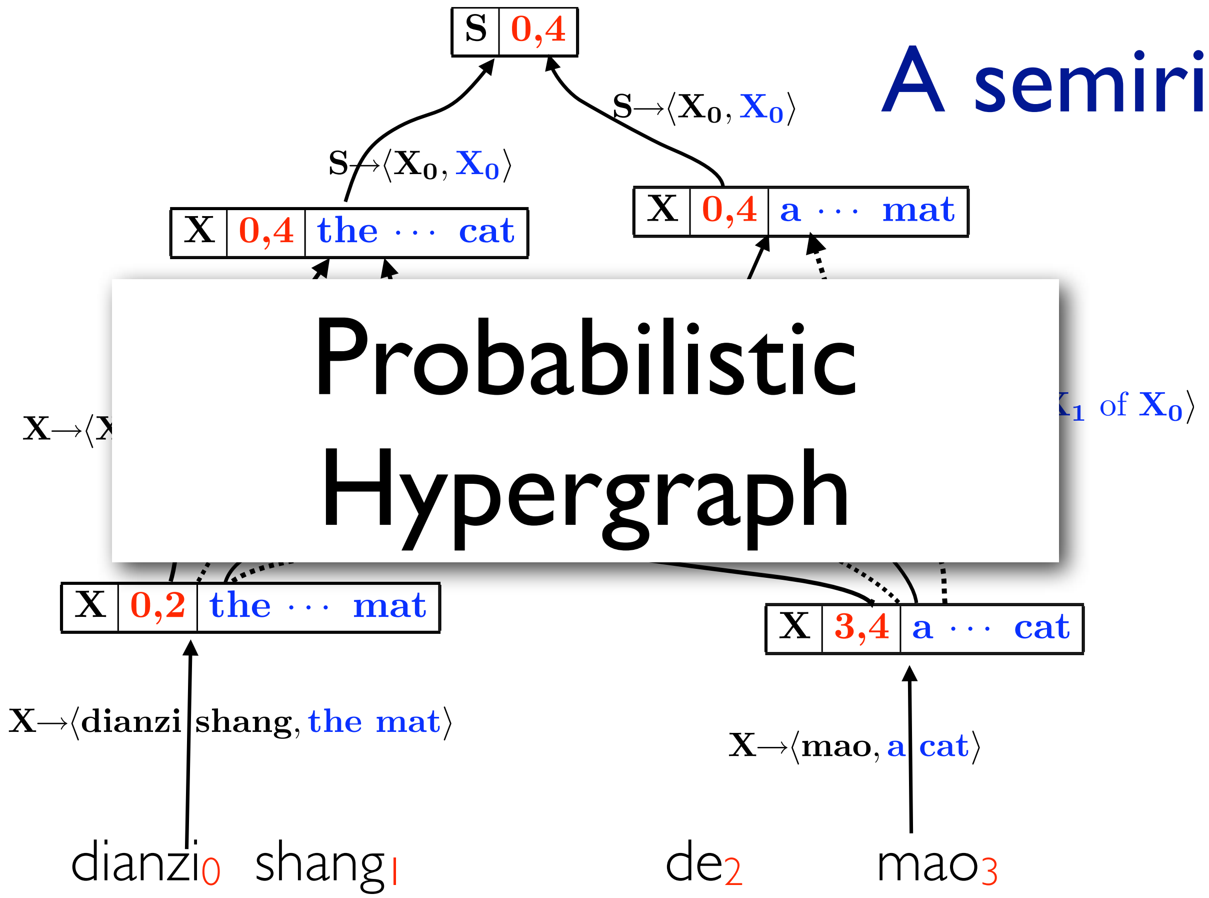
$$y^* = \arg\max_y q^*(y \mid x)$$

Outline

- Hypergraph as Hypothesis Space
- Unsupervised Discriminative Training
  - minimum imputed risk
  - contrastive language model estimation
- Variational Decoding
- First- and Second-order Expectation Semirings

decoding (e.g., mbr)	**training** (e.g., mert)
atomic inference operations (e.g., finding one-best, k-best or expectation, inference can be *exact* or *approximate*)	

A semiring framework to compute all of these



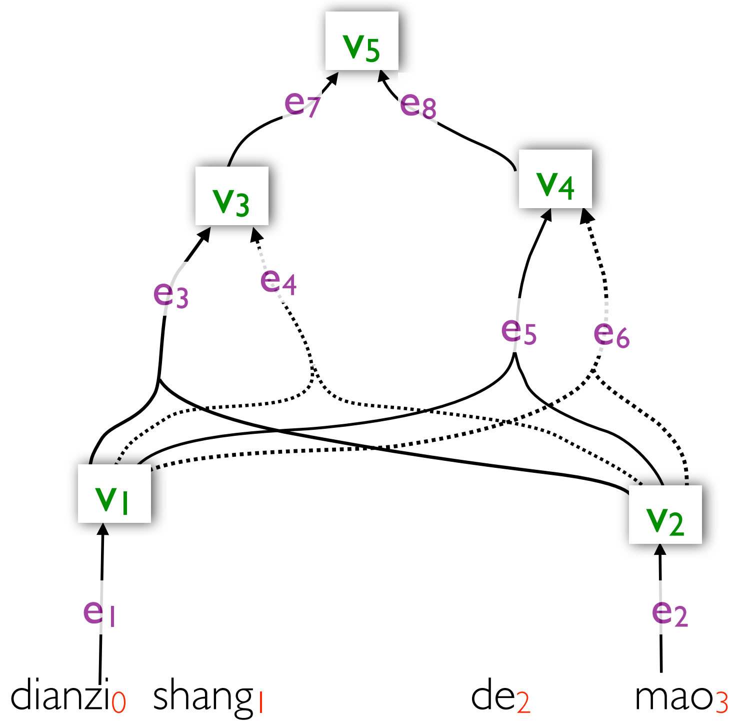
- "decoding" quantities:
  - Viterbi
  - K-best
  - Counting
  - ......

- First-order quantities:
  - expectation
    - entropy
    - Bayes risk
    - cross-entropy
    - KL divergence
    - feature expectations
  - first-order gradient of Z

- Second-order quantities:
  - expectation over product
    - interaction between features
  - Hessian matrix of Z
    - second-order gradient descent
  - gradient of expectation
    - gradient of entropy or Bayes risk

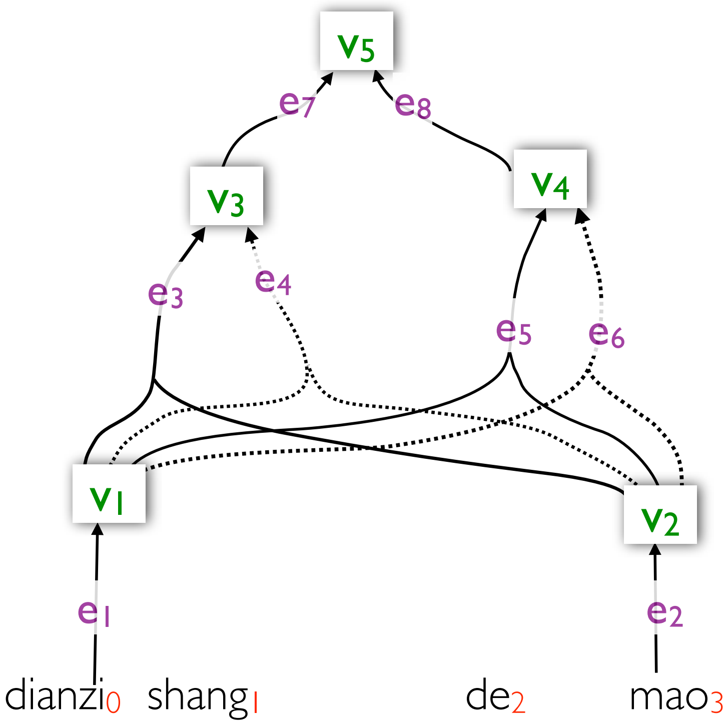
Compute Quantities on a Hypergraph: a Recipe

- Semiring-weighted inside algorithm
  - three steps:



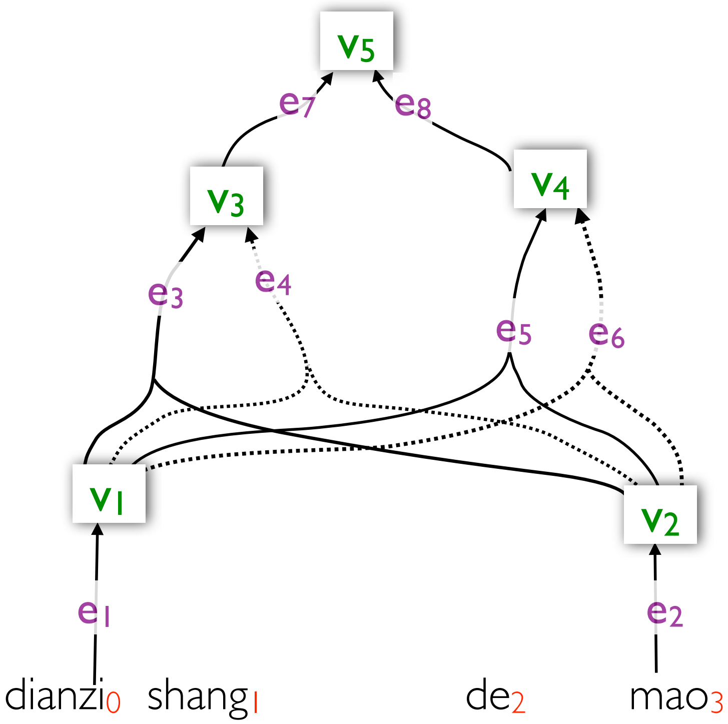
Compute Quantities on a Hypergraph: a Recipe

- Semiring-weighted inside algorithm
  - three steps:
    - choose a semiring



Compute Quantities on a Hypergraph: a Recipe

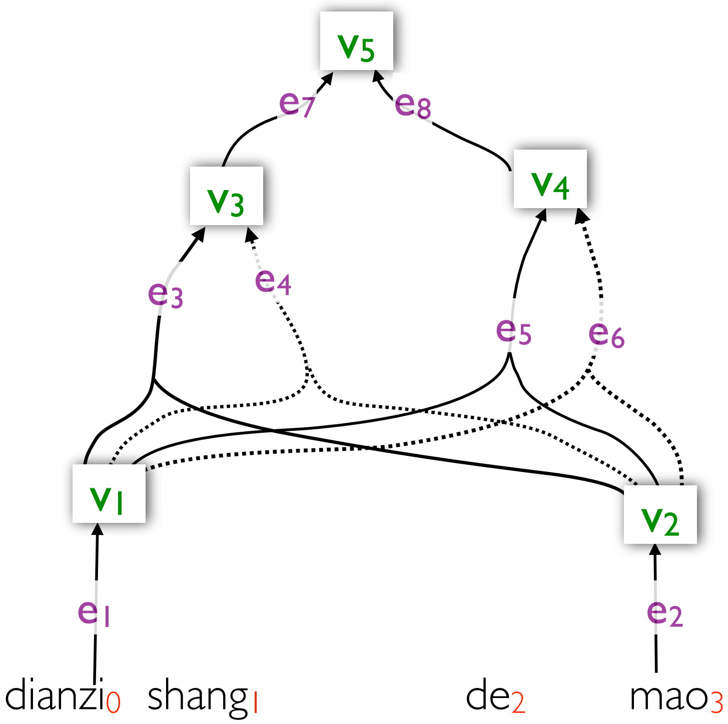
- Semiring-weighted inside algorithm
  - three steps:
    - choose a semiring



    - specify a weight for each hyperedge

Compute Quantities on a Hypergraph: a Recipe

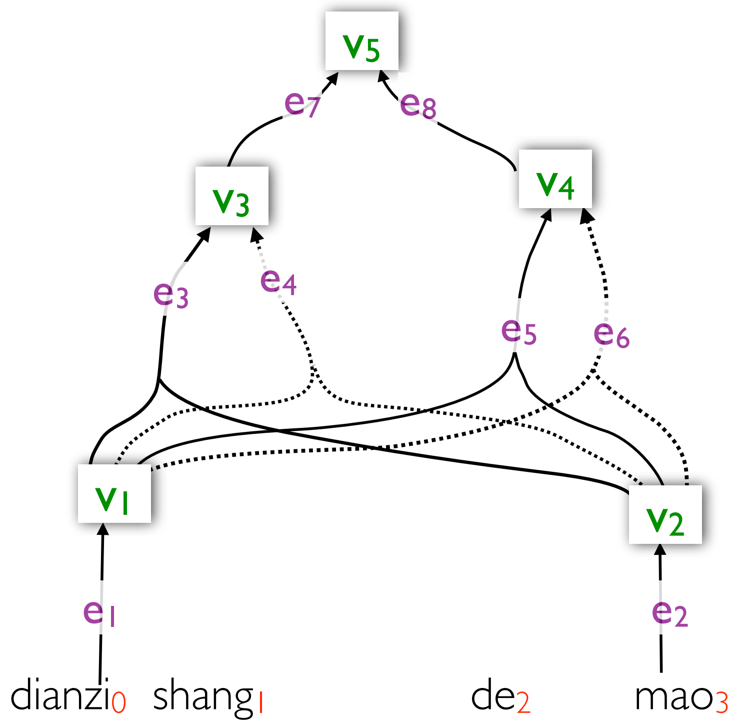
- Semiring-weighted inside algorithm
  - three steps:
    - choose a semiring



    - specify a weight for each hyperedge
    - run the inside algorithm

Compute Quantities on a Hypergraph: a Recipe

- Semiring-weighted inside algorithm
  - three steps:
    - choose a semiring



$$\langle K, \oplus, \otimes \rangle$$

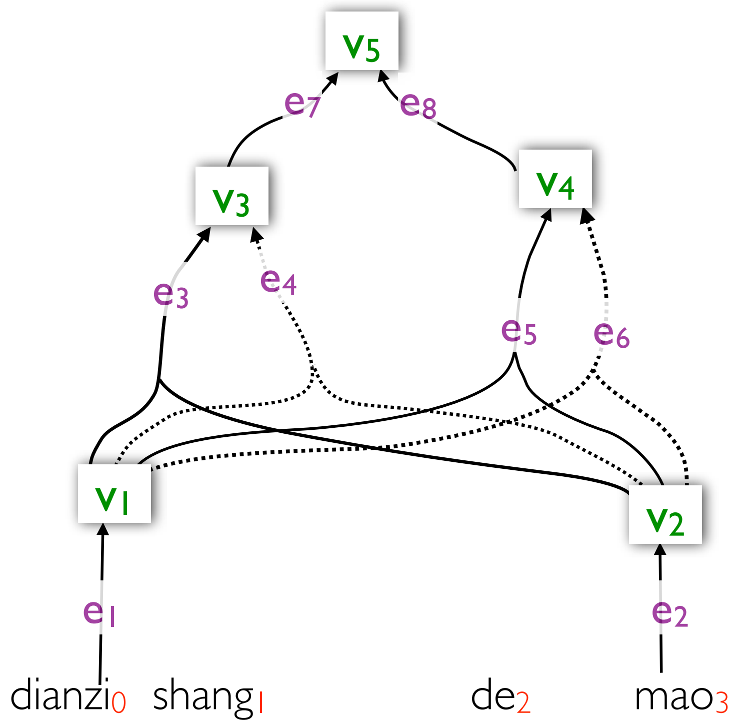
a **set** with **plus** and **times** operations

- specify a weight for each hyperedge

- run the inside algorithm

Compute Quantities on a Hypergraph: a Recipe

- Semiring-weighted inside algorithm
  - three steps:
    - choose a semiring



$$\langle K, \oplus, \otimes \rangle$$

a **set** with **plus** and **times** operations

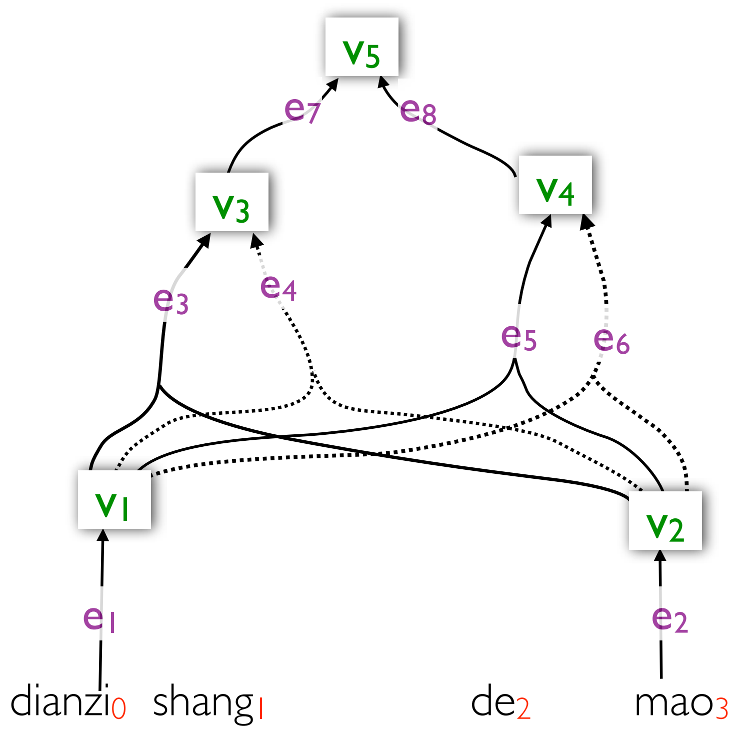
e.g., integer numbers with regular + and ×

    - specify a weight for each hyperedge

    - run the inside algorithm

Compute Quantities on a Hypergraph: a Recipe

- Semiring-weighted inside algorithm
  - three steps:
    - choose a semiring



$$\langle K, \oplus, \otimes \rangle$$

a **set** with **plus** and **times** operations

e.g., integer numbers with regular + and ×

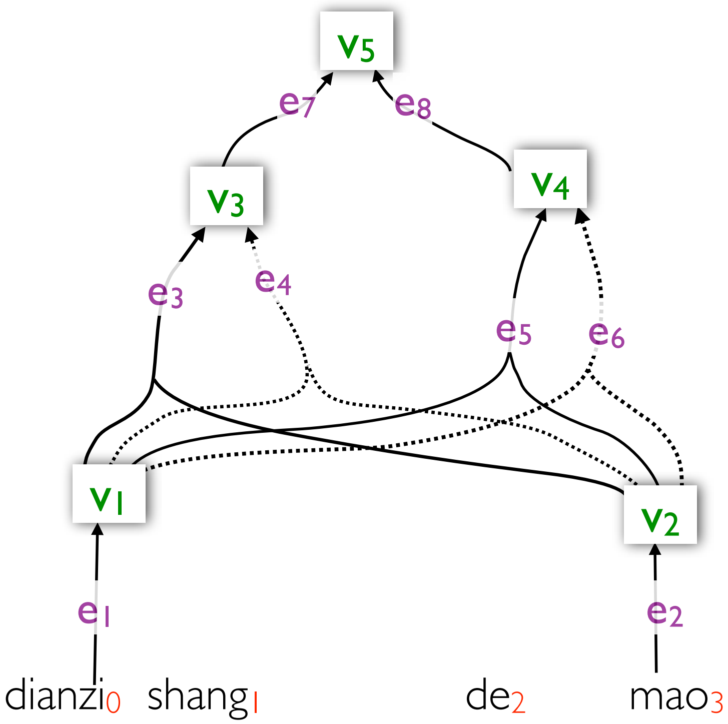
    - specify a weight for each hyperedge

each weight is a semiring member

    - run the inside algorithm

Compute Quantities on a Hypergraph: a Recipe

- Semiring-weighted inside algorithm
  - three steps:



▸ choose a semiring

$$\langle K, \oplus, \otimes \rangle$$

a **set** with **plus** and **times** operations

e.g., integer numbers with regular + and ×

▸ specify a weight for each hyperedge

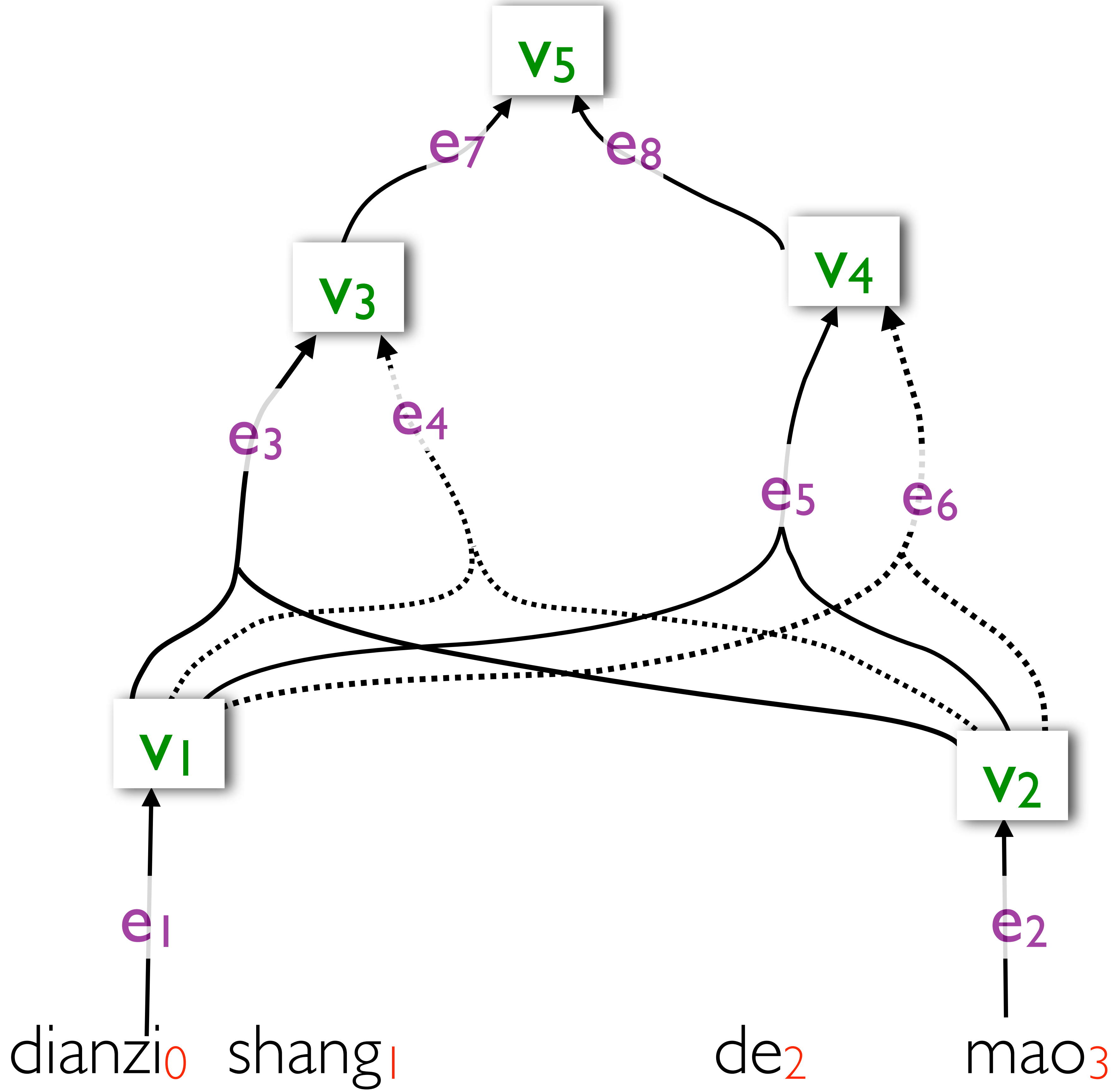
each weight is a semiring member

▸ run the inside algorithm

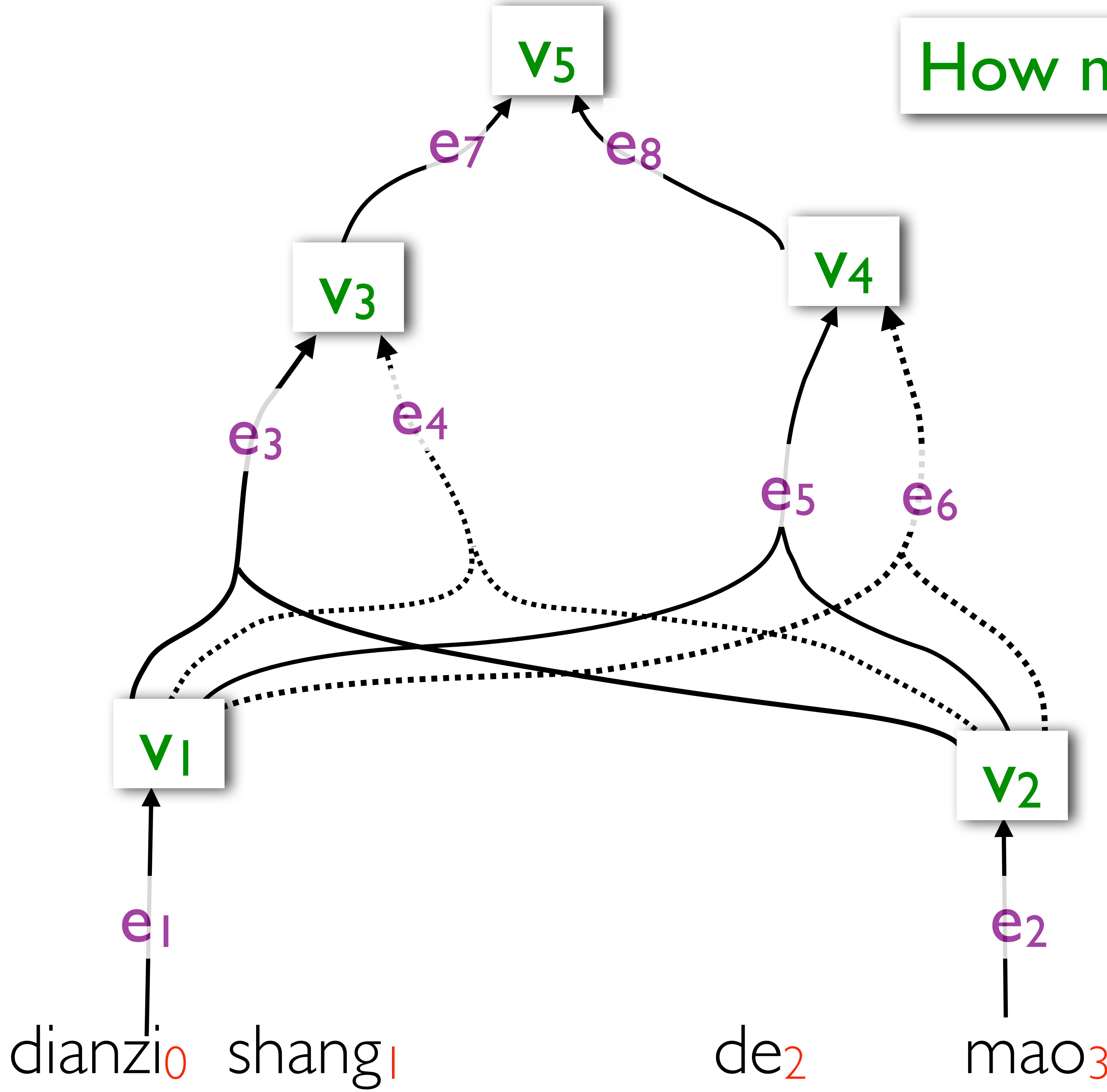
complexity is O(hypergraph size)

Semirings

- "Decoding" time semirings (Goodman, 1999)
  - counting, Viterbi, K-best, etc.
- "Training" time semirings
  - first-order expectation semirings (Eisner, 2002)
  - second-order expectation semirings (new)
- Applications of the Semirings (new)
  - entropy, risk, gradient of them, and many more

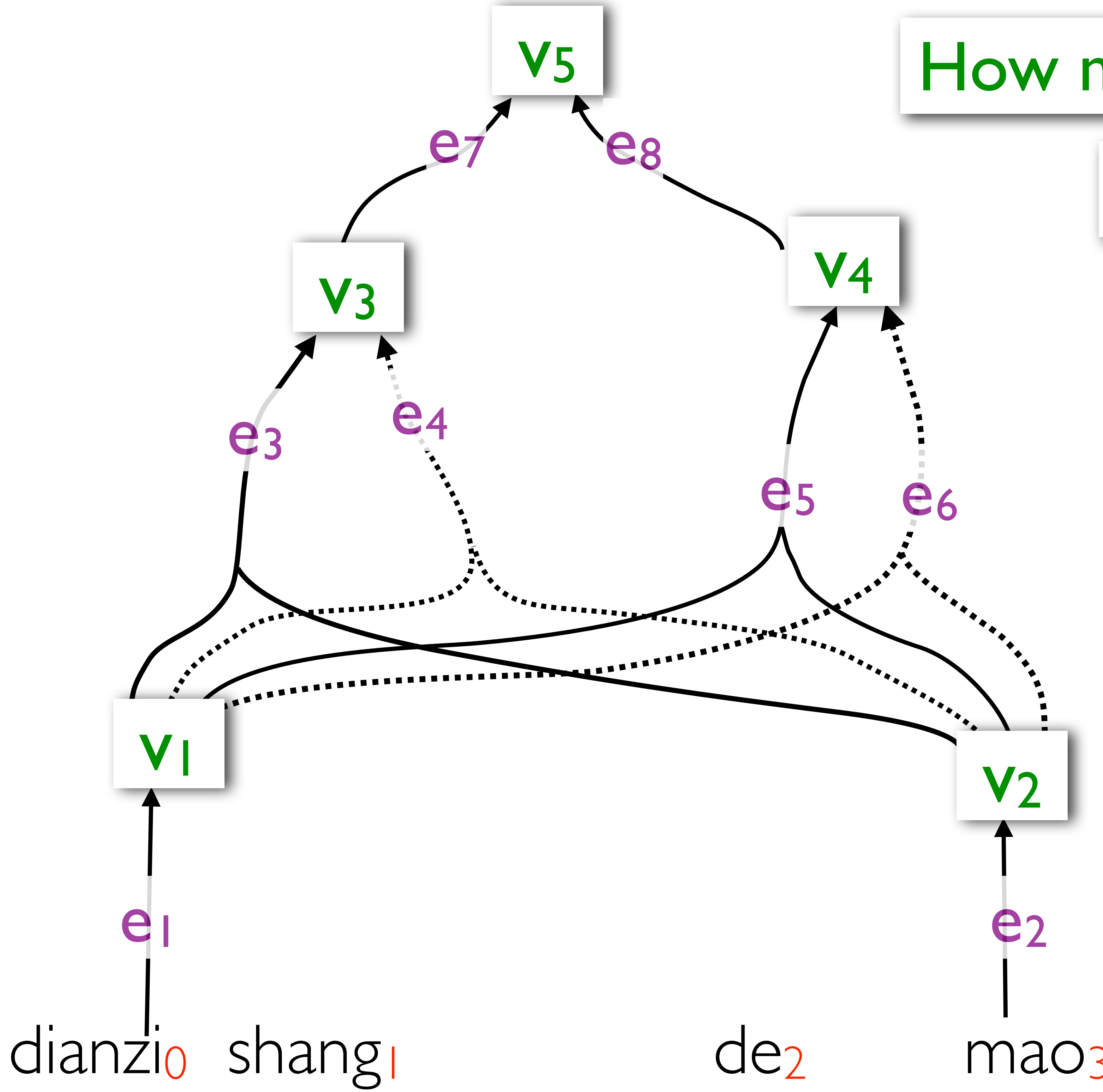

v5
e7
e8
v3
v4
e3
e4
e5
e6
v1
v2
e1
e2
dianzi0
shang1
de2
mao3

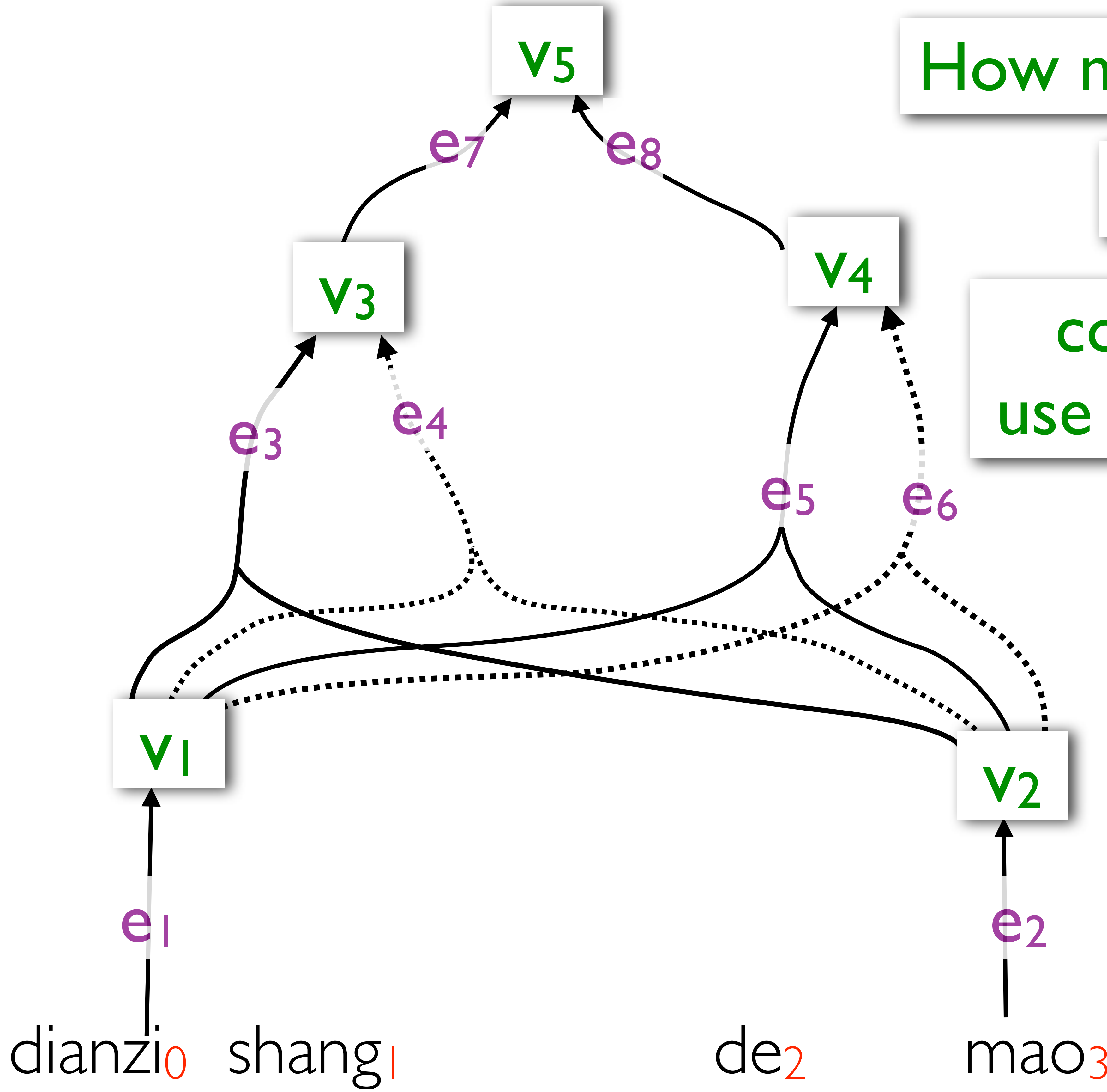
How many trees?



How many trees?

four ☺





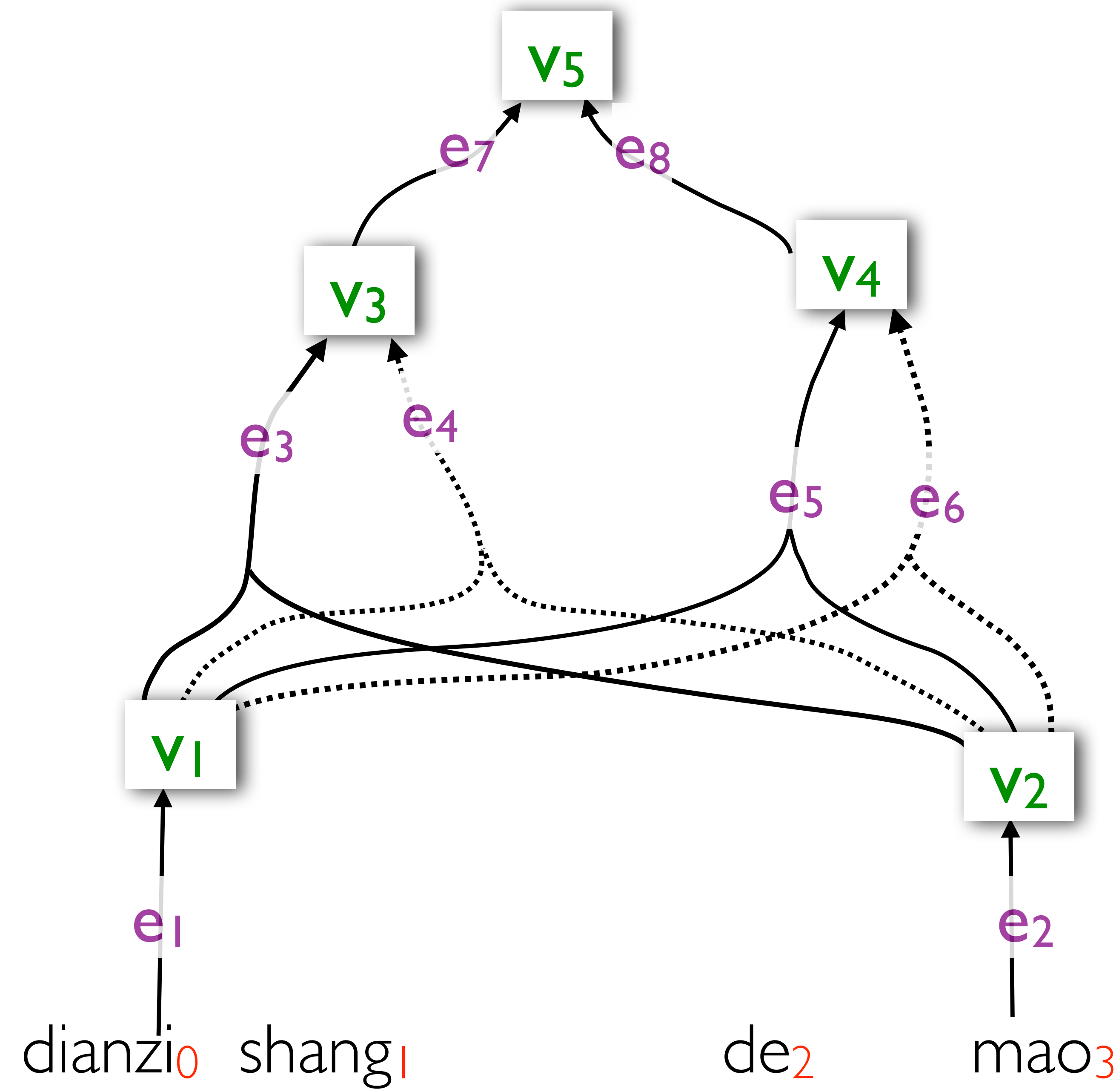
How many trees?

four ☺

compute it
use a semiring?

Compute the Number of Derivation Trees

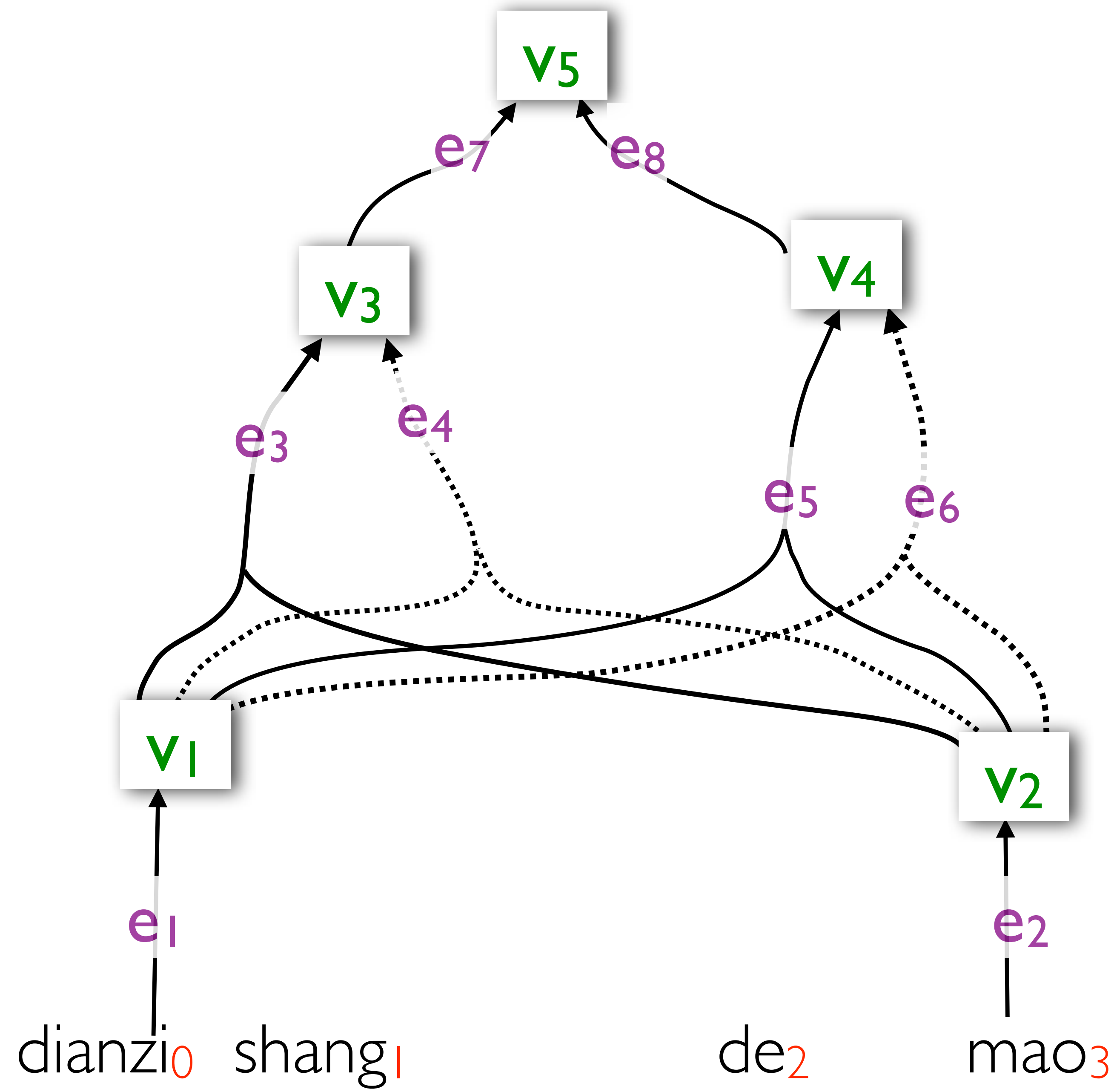
Three steps:



Compute the Number of Derivation Trees

Three steps:

- choose a semiring



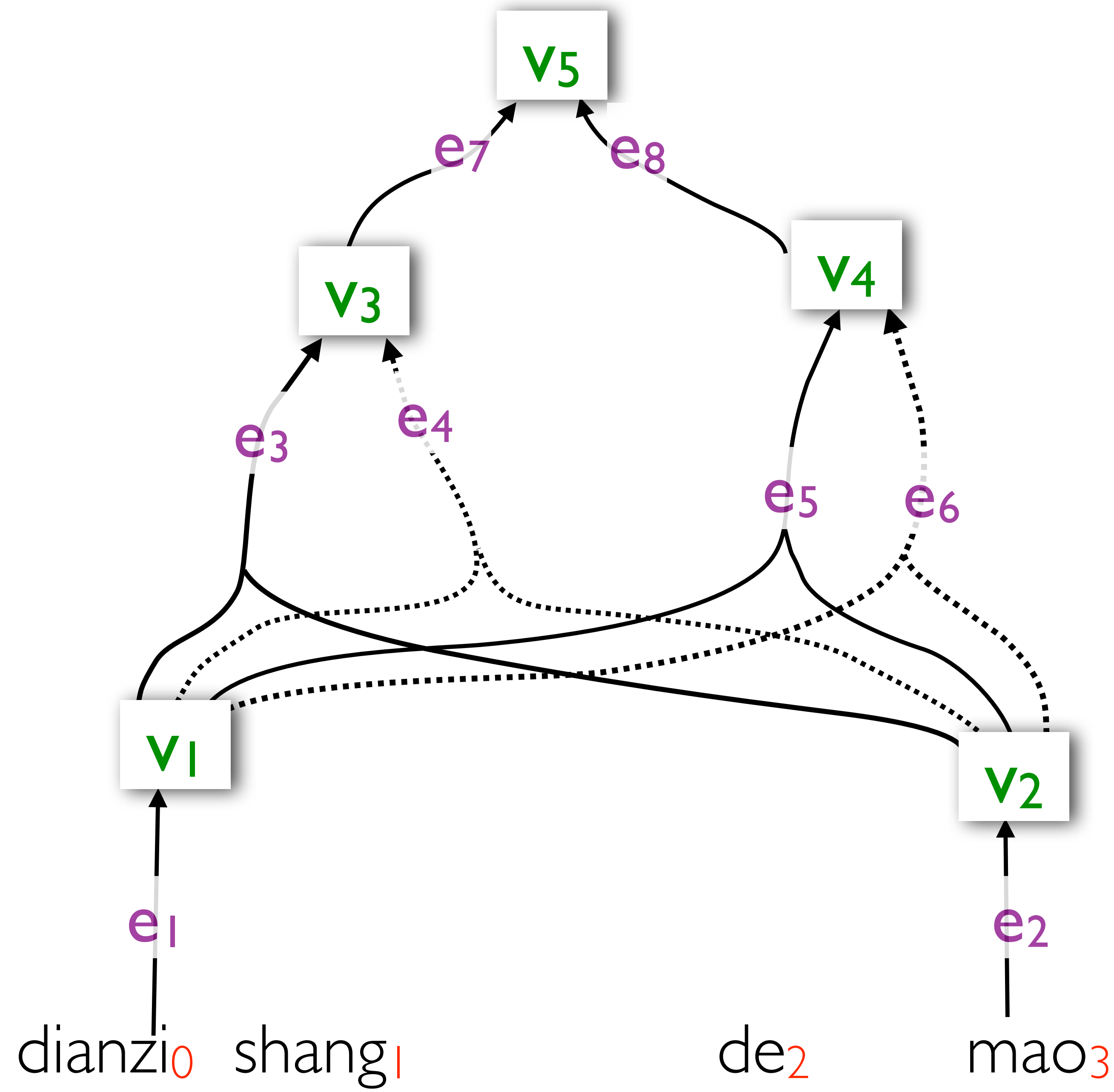
Compute the Number of Derivation Trees

Three steps:

- choose a semiring

  counting semiring:

  ordinary integers with regular + and x



Compute the Number of Derivation Trees

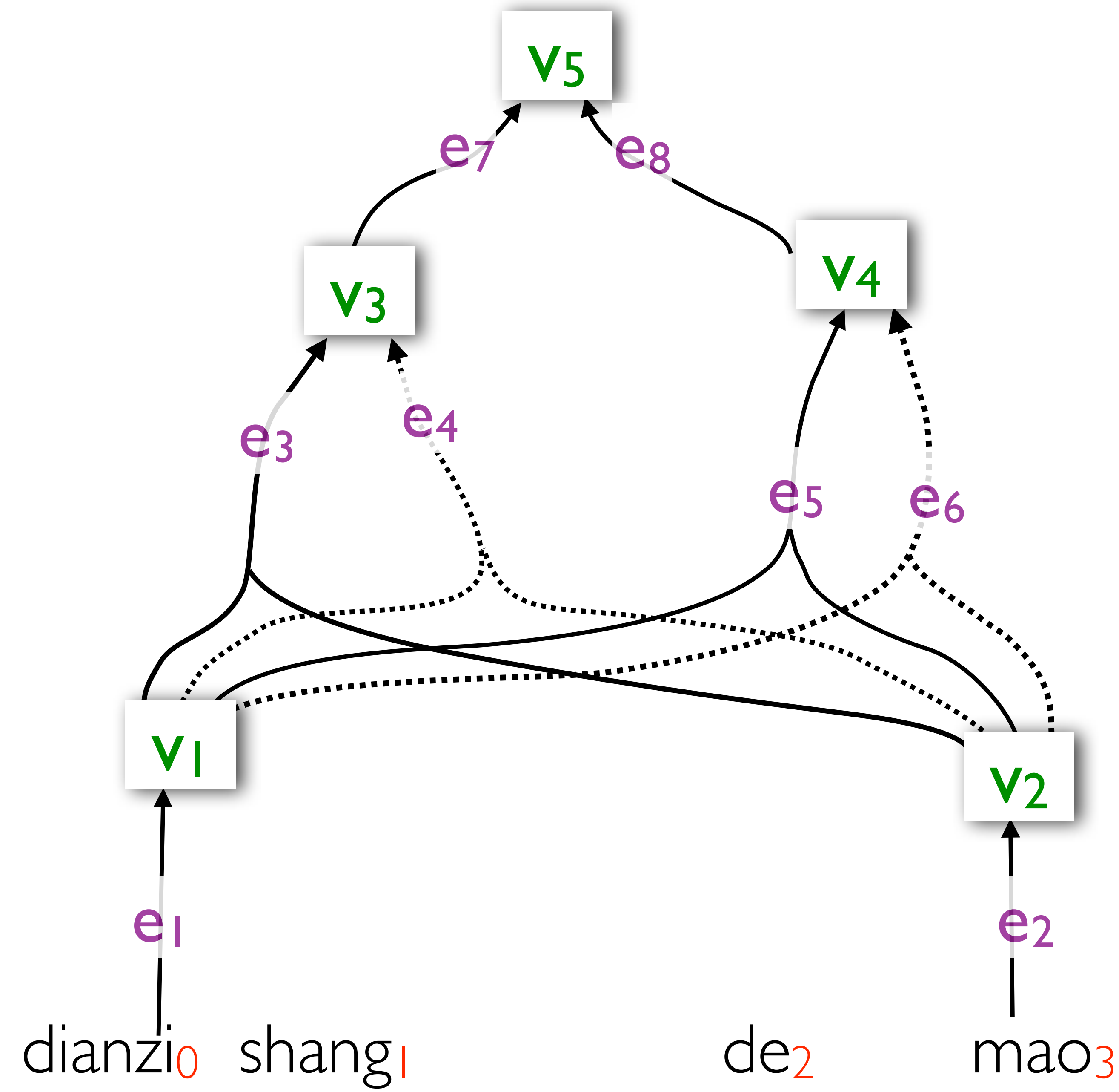
Three steps:

- choose a semiring

  counting semiring:

  ordinary integers with regular + and x

- specify a weight for each hyperedge



Compute the Number of Derivation Trees

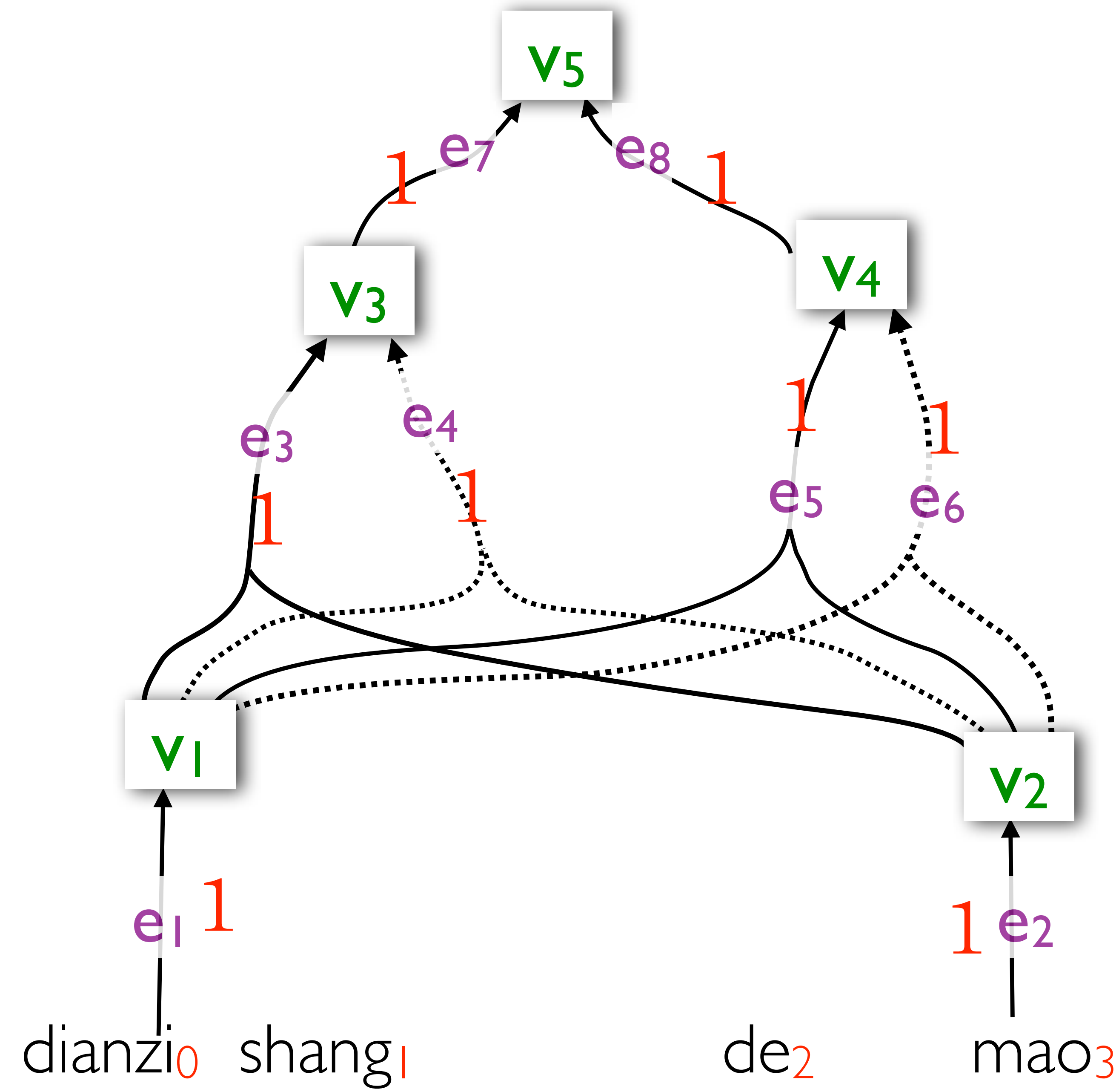
Three steps:

- choose a semiring

  counting semiring:

  ordinary integers with regular + and x

- specify a weight for each hyperedge



Compute the Number of Derivation Trees

Three steps:

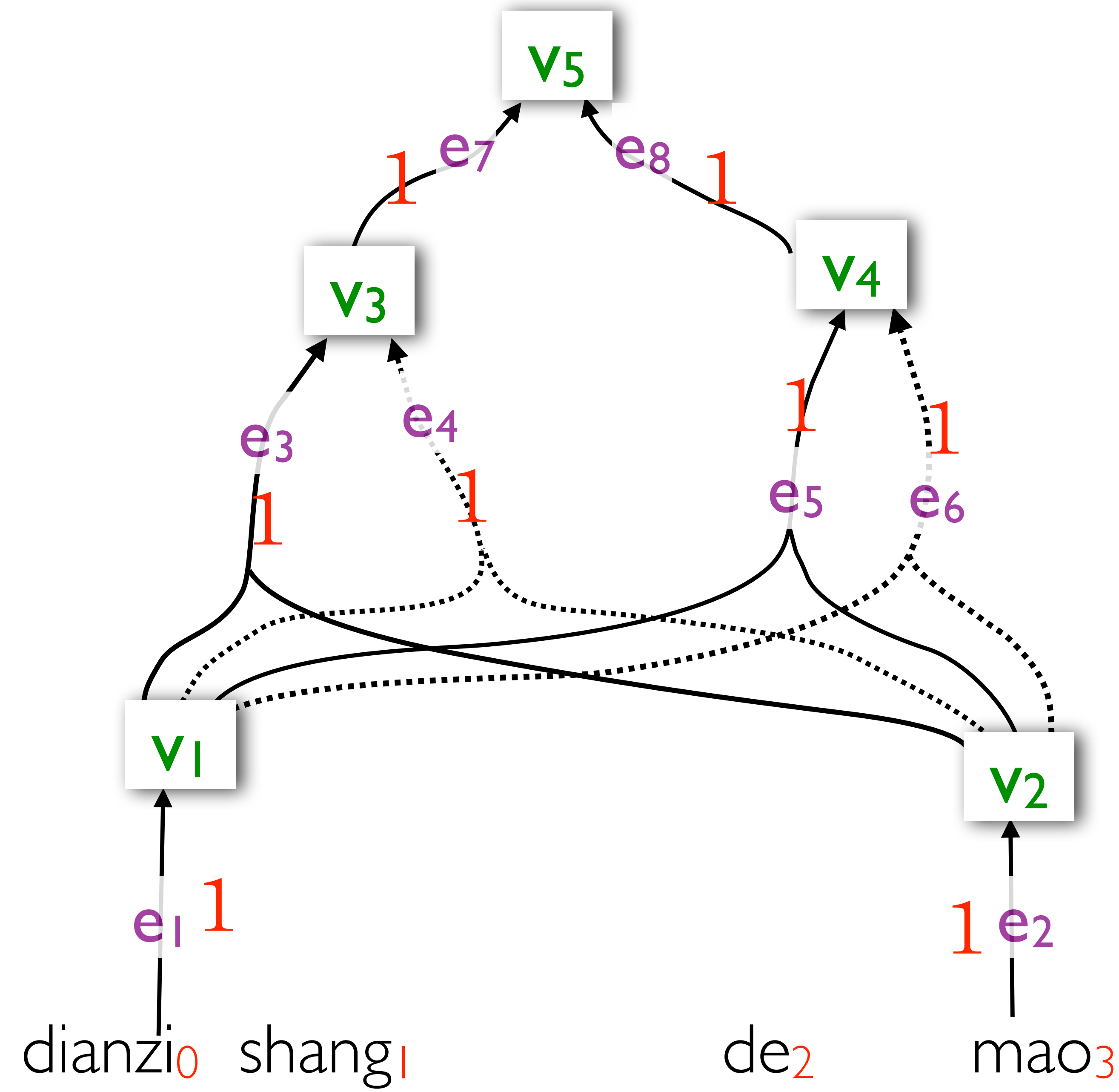
- choose a semiring

  counting semiring:

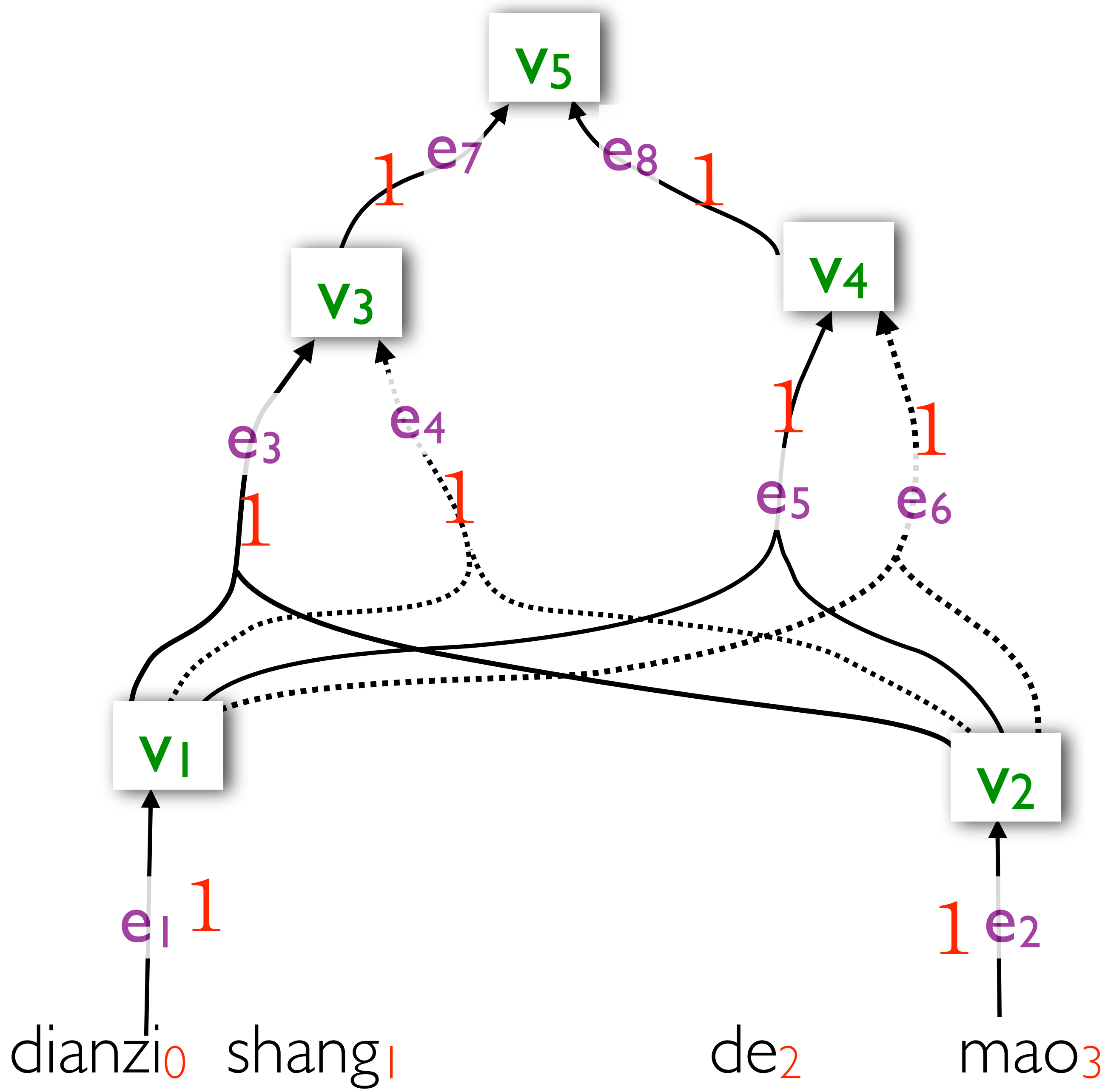
  ordinary integers with regular + and x

- specify a weight for each hyperedge

- run the inside algorithm

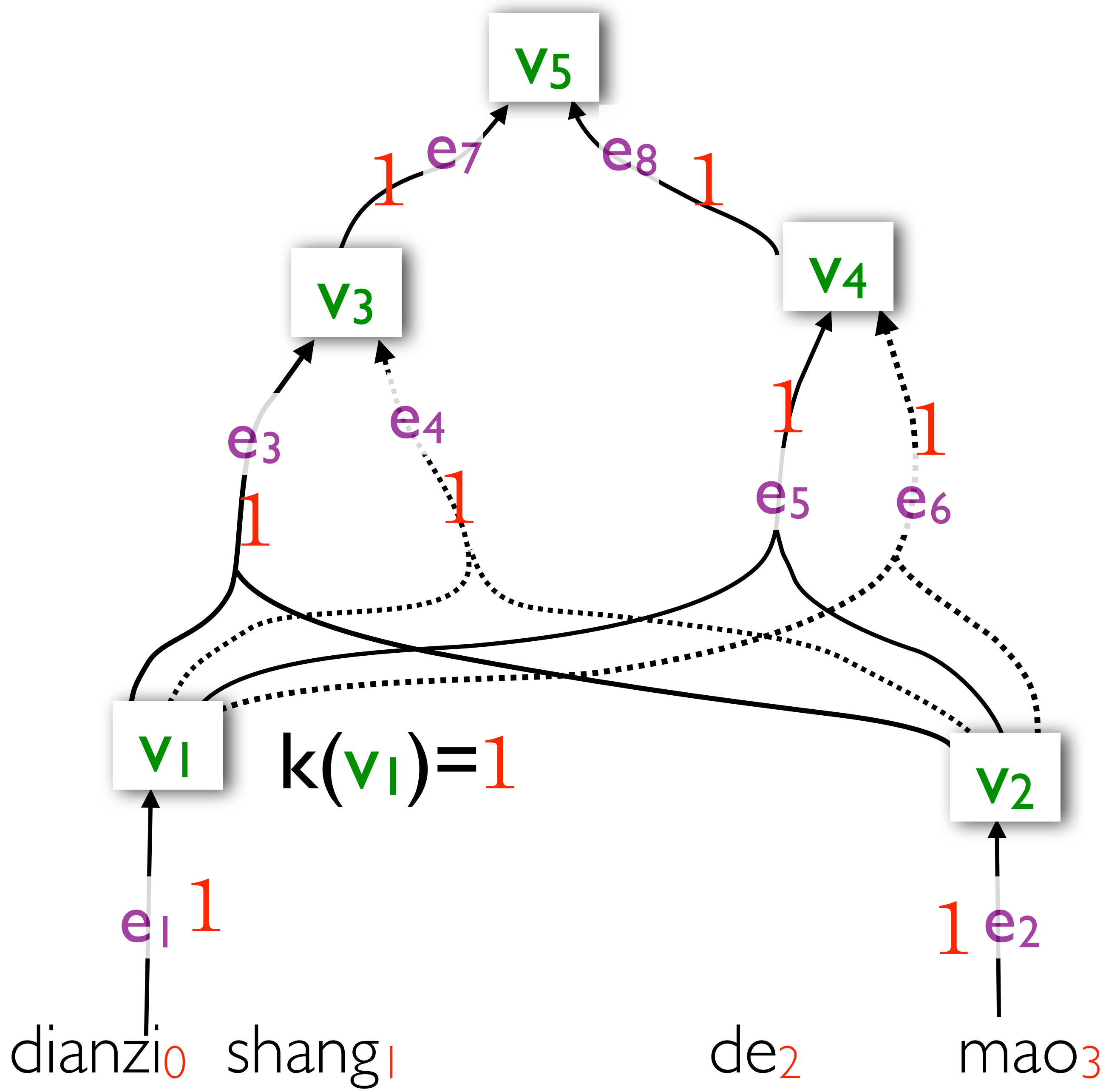


Bottom-up process in computing the number of trees

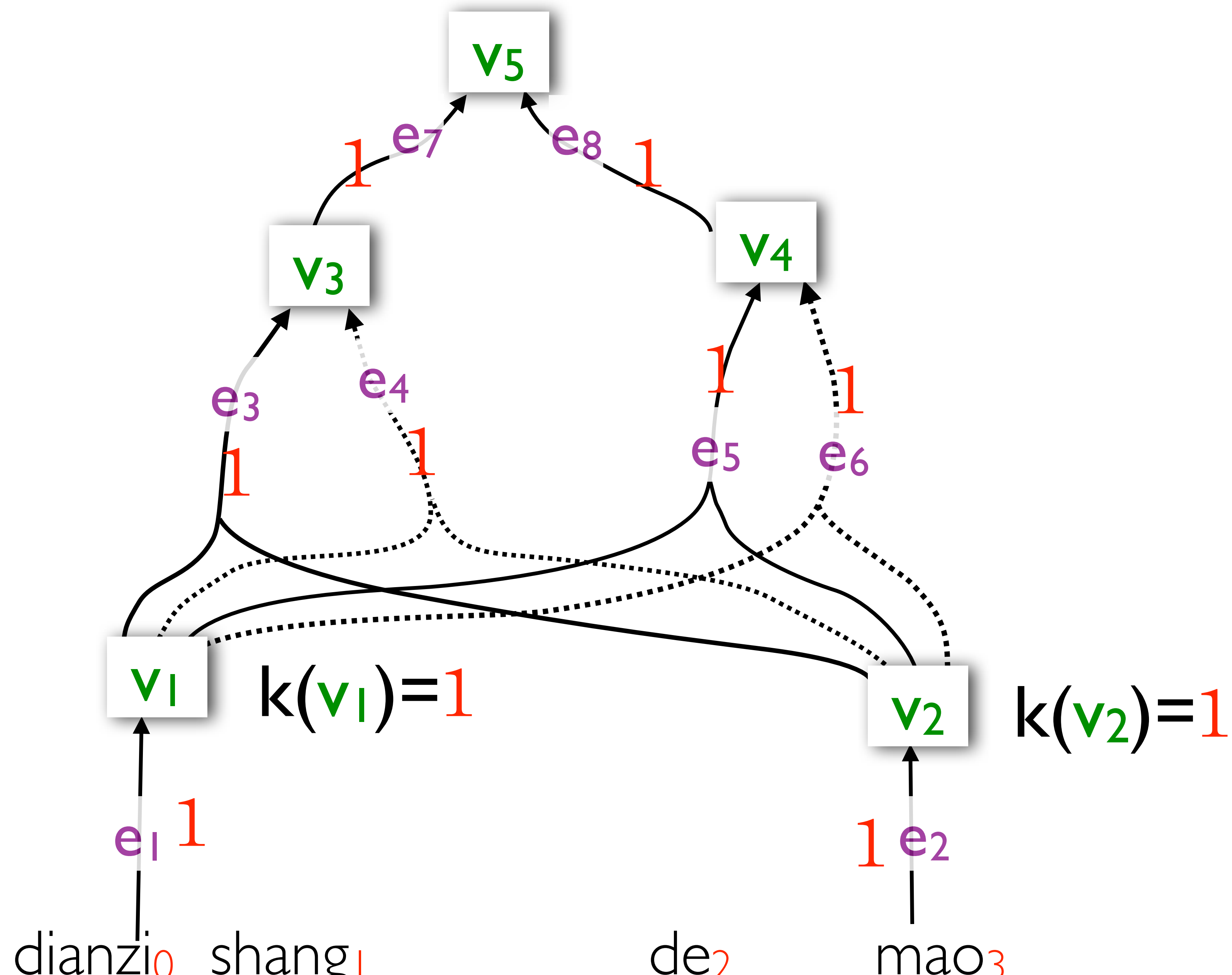


$k(v_1) = k(e_1)$

Bottom-up process in computing the number of trees



$k(v_1) = k(e_1)$ $k(v_2) = k(e_2)$

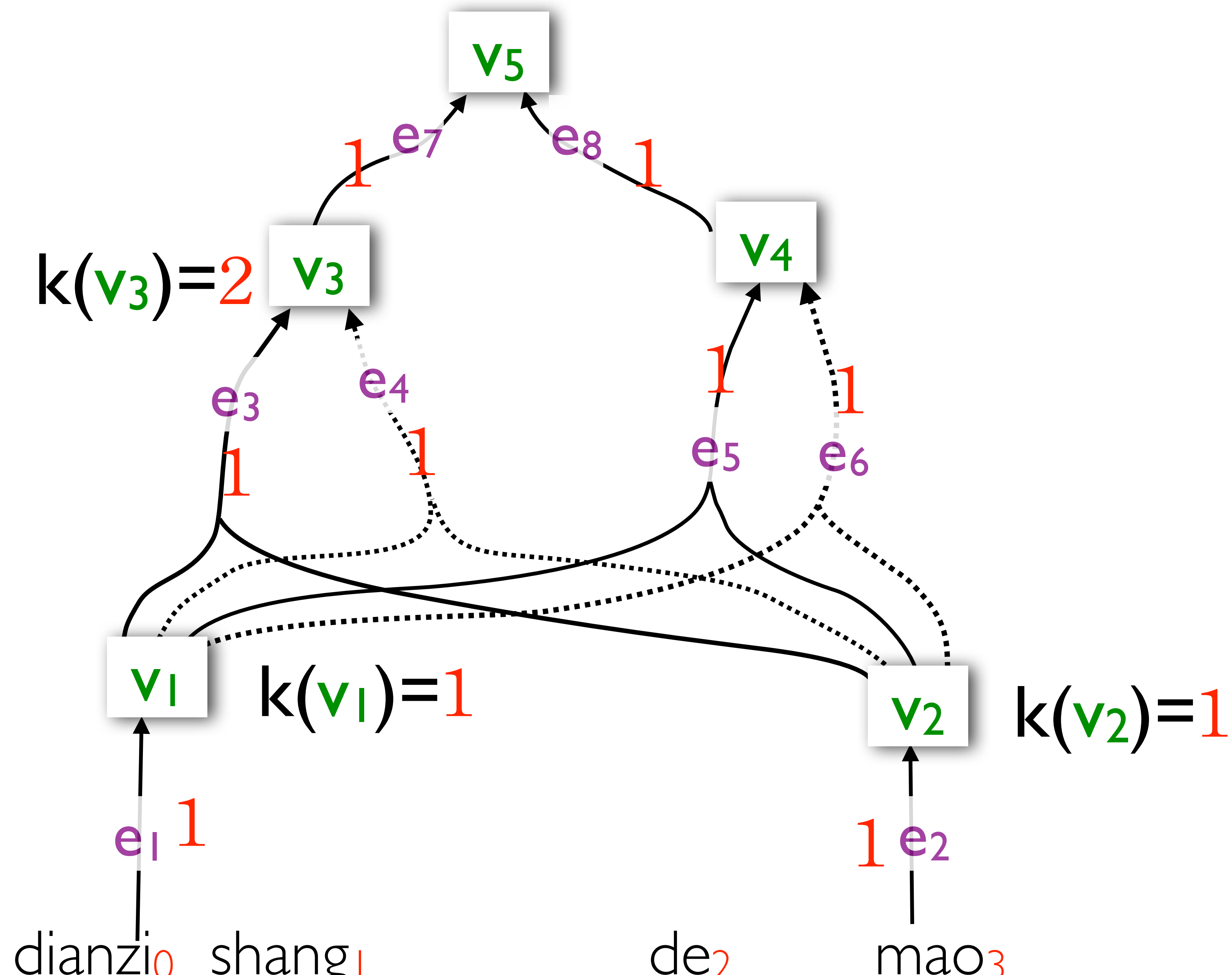


Bottom-up process in computing the number of trees

$k(v_1) = k(e_1)$ $k(v_2) = k(e_2)$

$k(v_3) = k(e_3) \otimes k(v_1) \otimes k(v_2) \oplus k(e_4) \otimes k(v_1) \otimes k(v_2)$

Bottom-up process in computing the number of trees

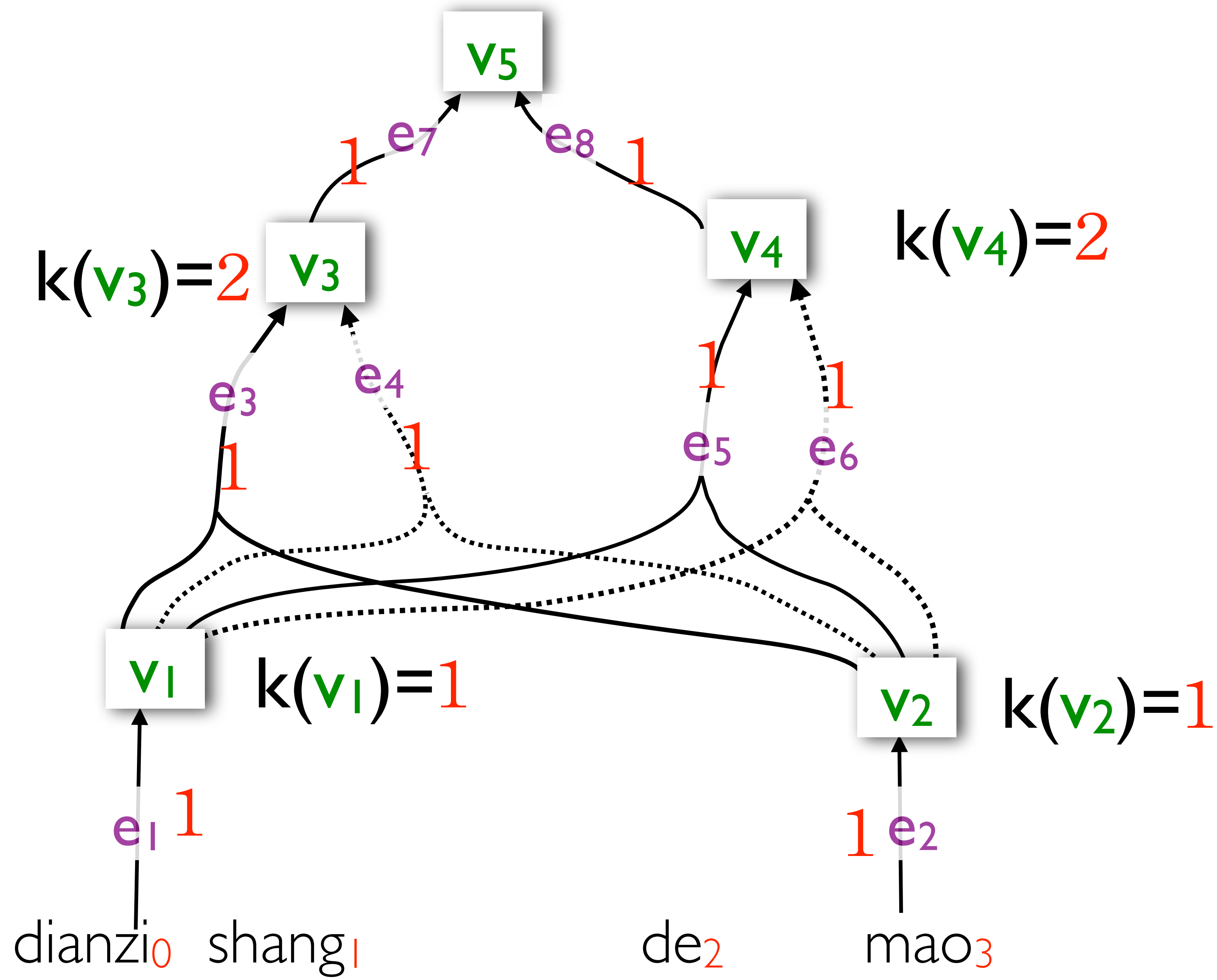


$k(v_1) = k(e_1)$ $k(v_2) = k(e_2)$

$k(v_3) = k(e_3) \otimes k(v_1) \otimes k(v_2) \oplus k(e_4) \otimes k(v_1) \otimes k(v_2)$

$k(v_4) = k(e_5) \otimes k(v_1) \otimes k(v_2) \oplus k(e_6) \otimes k(v_1) \otimes k(v_2)$

Bottom-up process in computing the number of trees

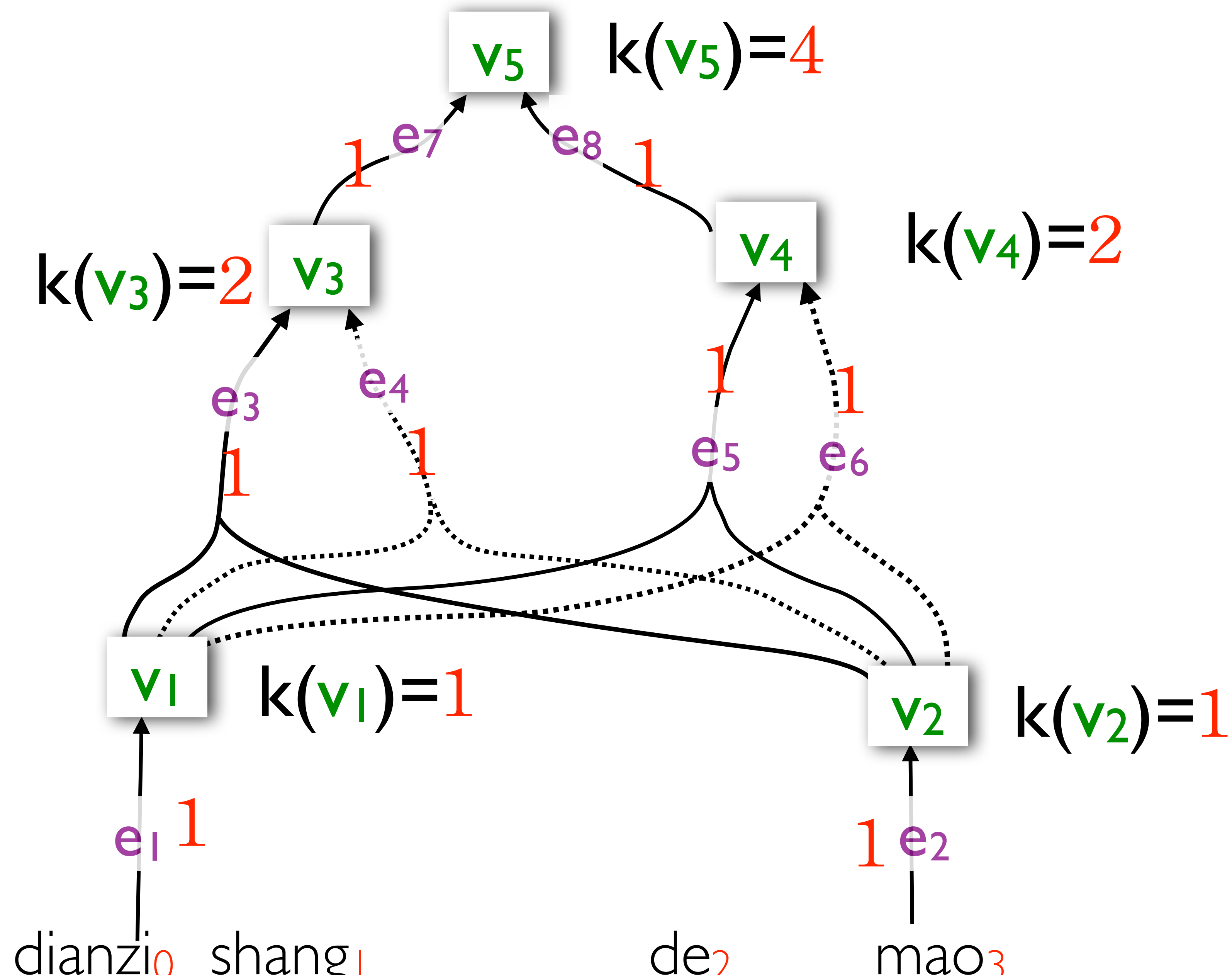


$k(v_1) = k(e_1)$ $\quad$ $k(v_2) = k(e_2)$

$k(v_3) = k(e_3) \otimes k(v_1) \otimes k(v_2) \oplus k(e_4) \otimes k(v_1) \otimes k(v_2)$

$k(v_4) = k(e_5) \otimes k(v_1) \otimes k(v_2) \oplus k(e_6) \otimes k(v_1) \otimes k(v_2)$

$k(v_5) = k(e_7) \otimes k(v_3) \oplus k(e_8) \otimes k(v_4)$



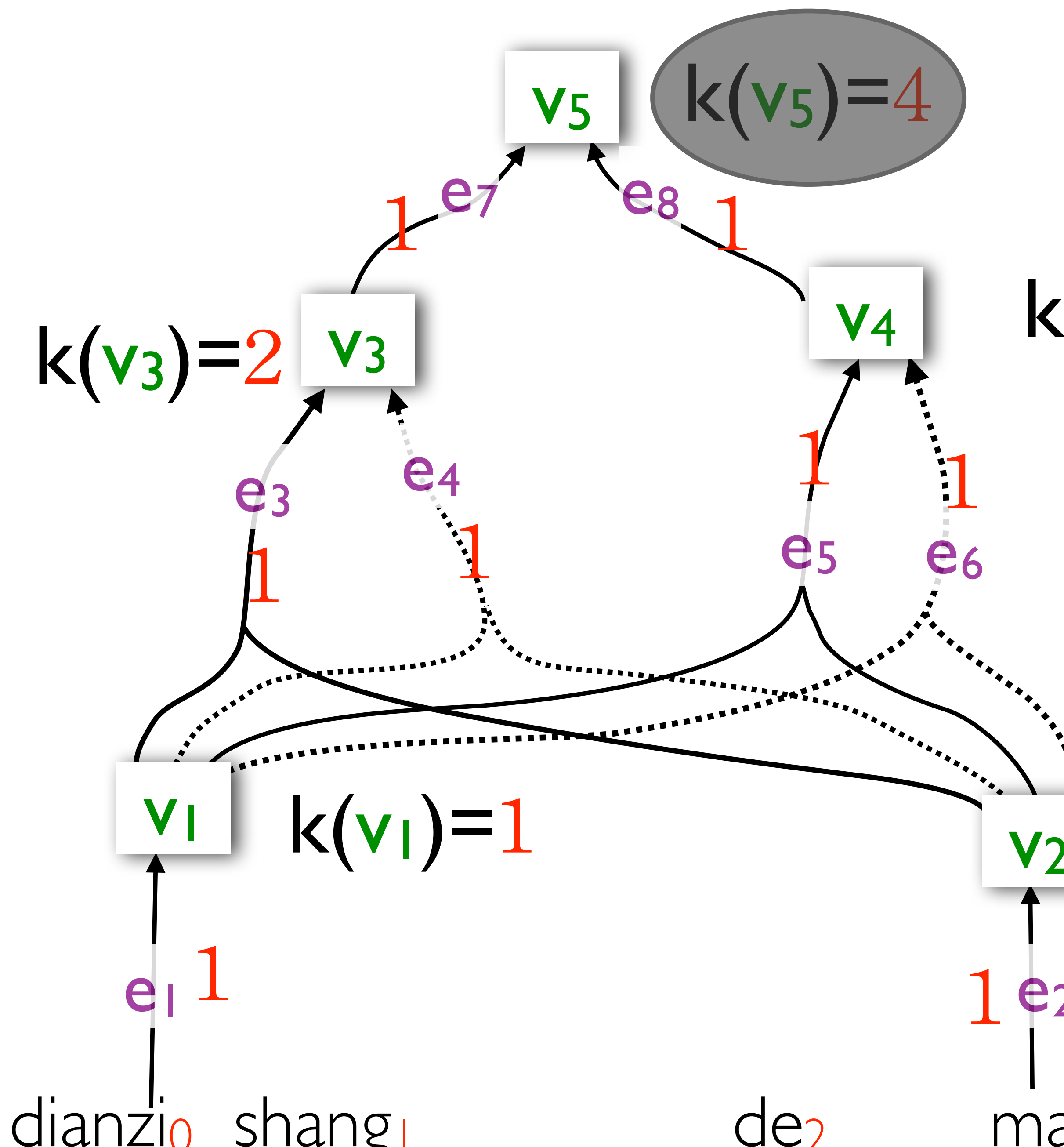
Bottom-up process in computing the number of trees

$k(v_1) = k(e_1)$ $\qquad$ $k(v_2) = k(e_2)$

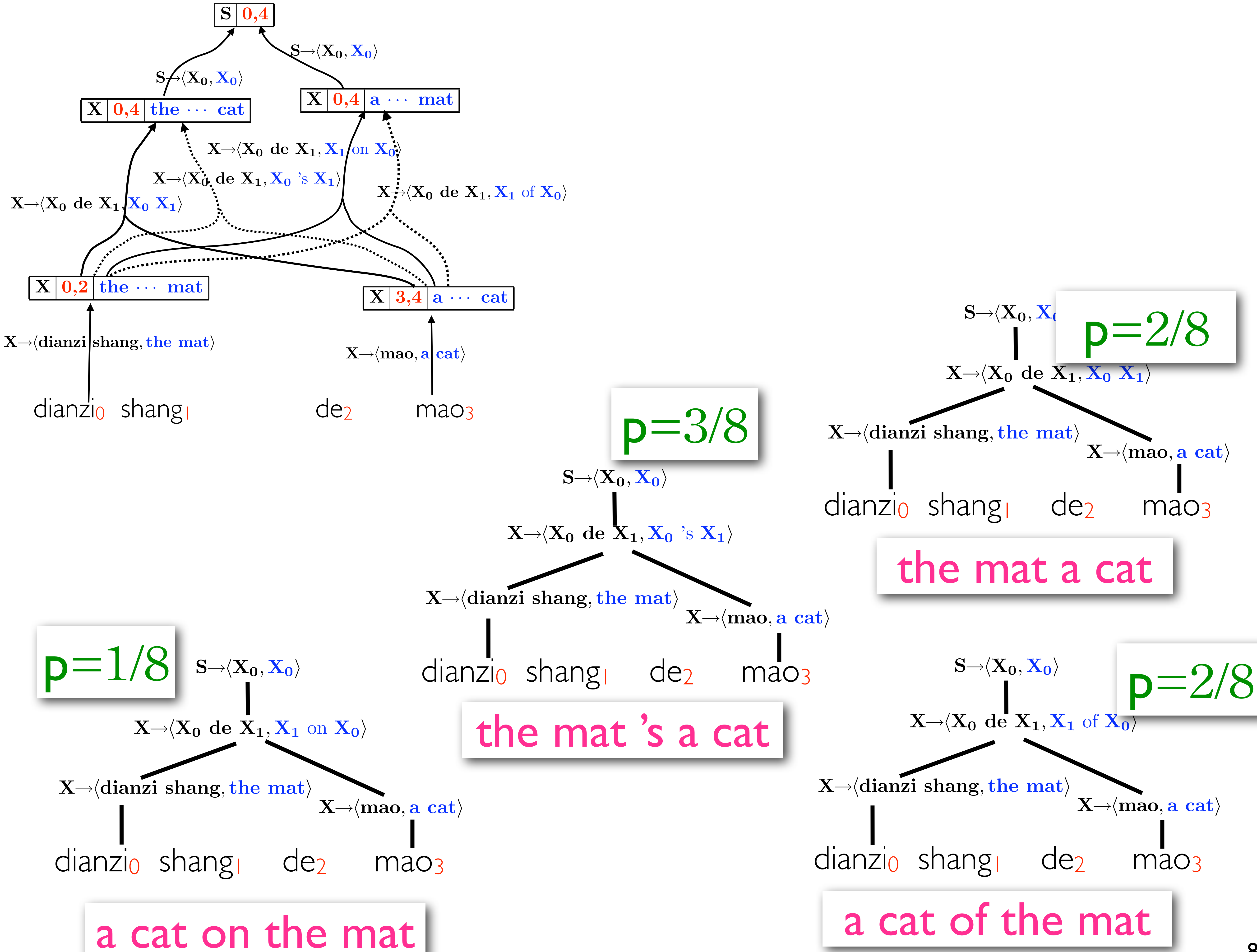
$k(v_3) = k(e_3) \otimes k(v_1) \otimes k(v_2) \oplus k(e_4) \otimes k(v_1) \otimes k(v_2)$

$k(v_4) = k(e_5) \otimes k(v_1) \otimes k(v_2) \oplus k(e_6) \otimes k(v_1) \otimes k(v_2)$

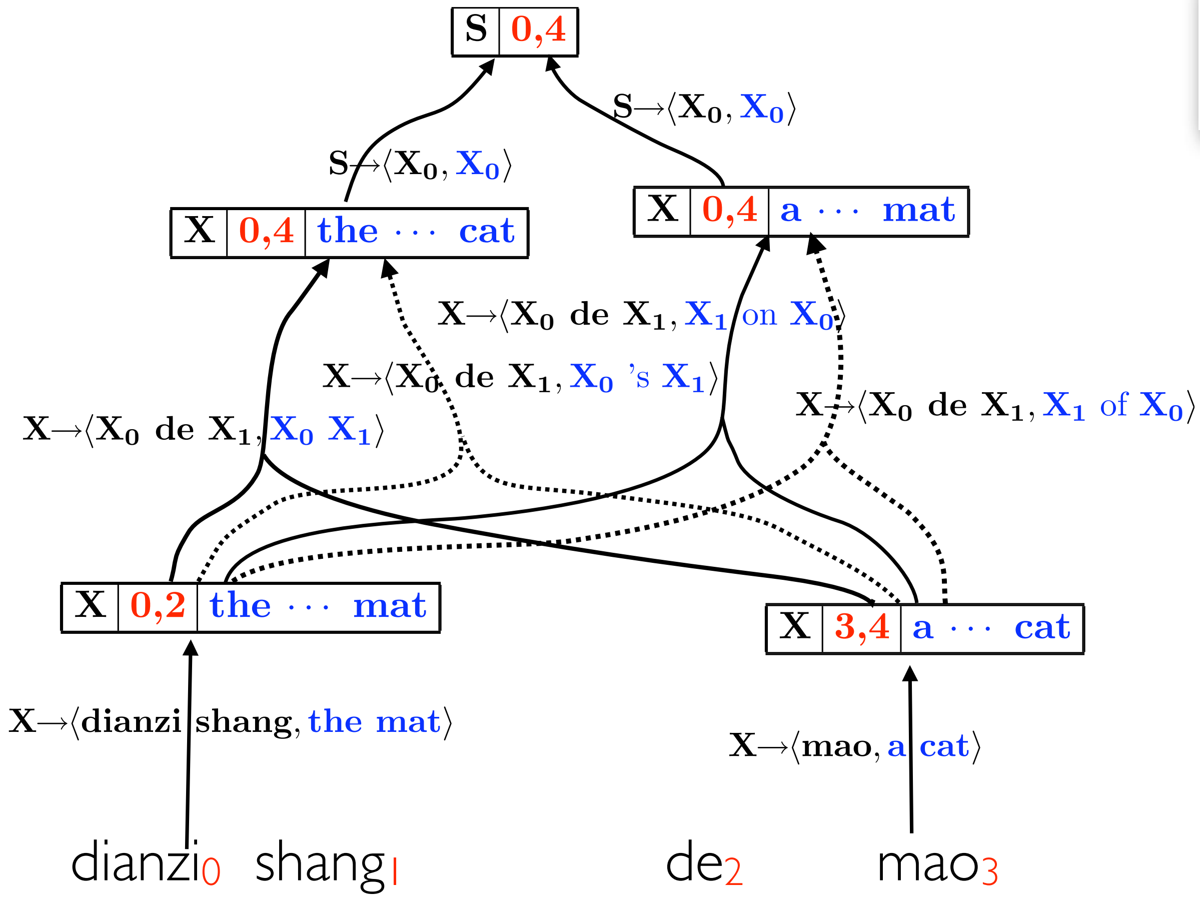
$k(v_5) = k(e_7) \otimes k(v_3) \oplus k(e_8) \otimes k(v_4)$

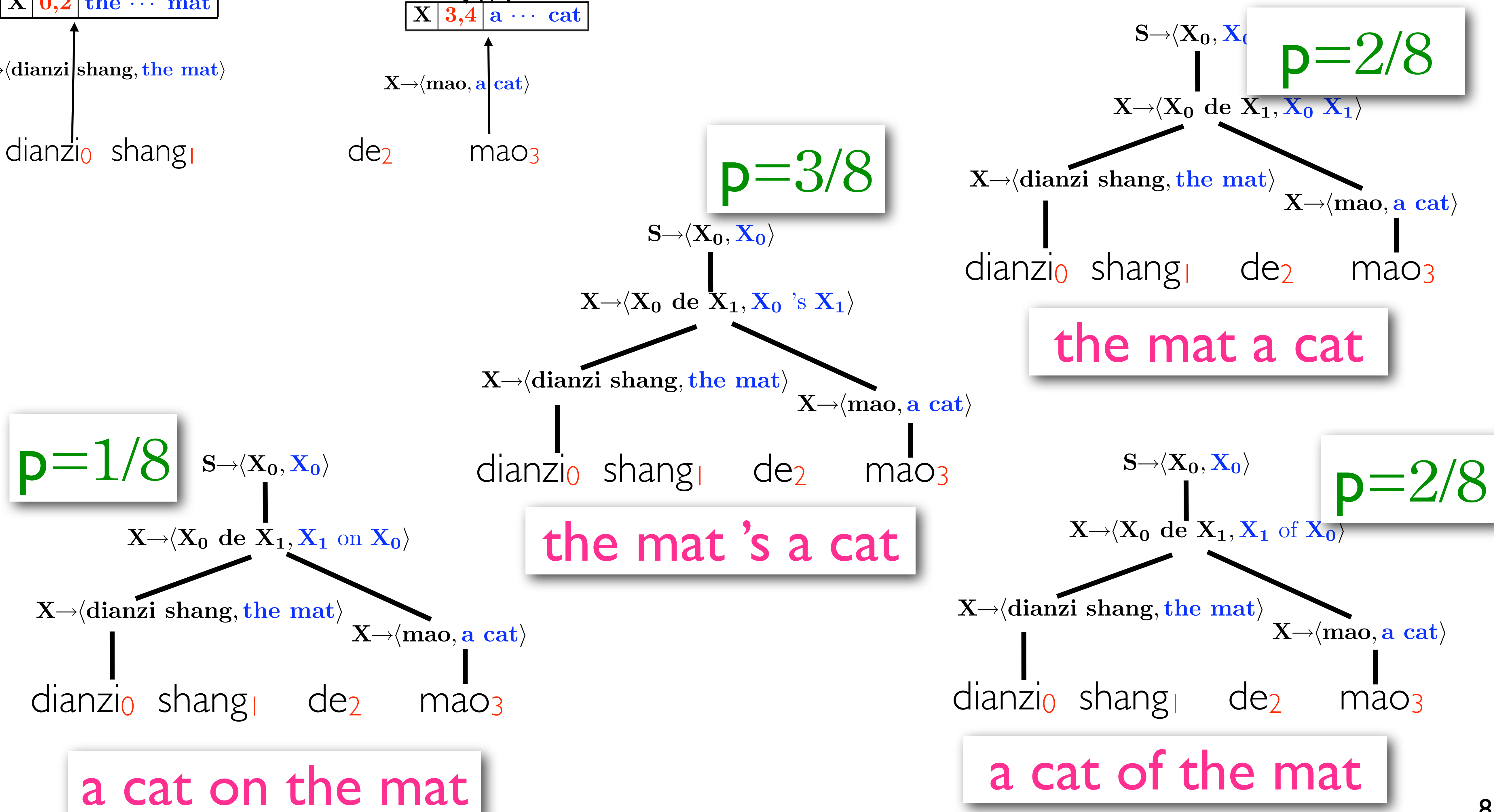


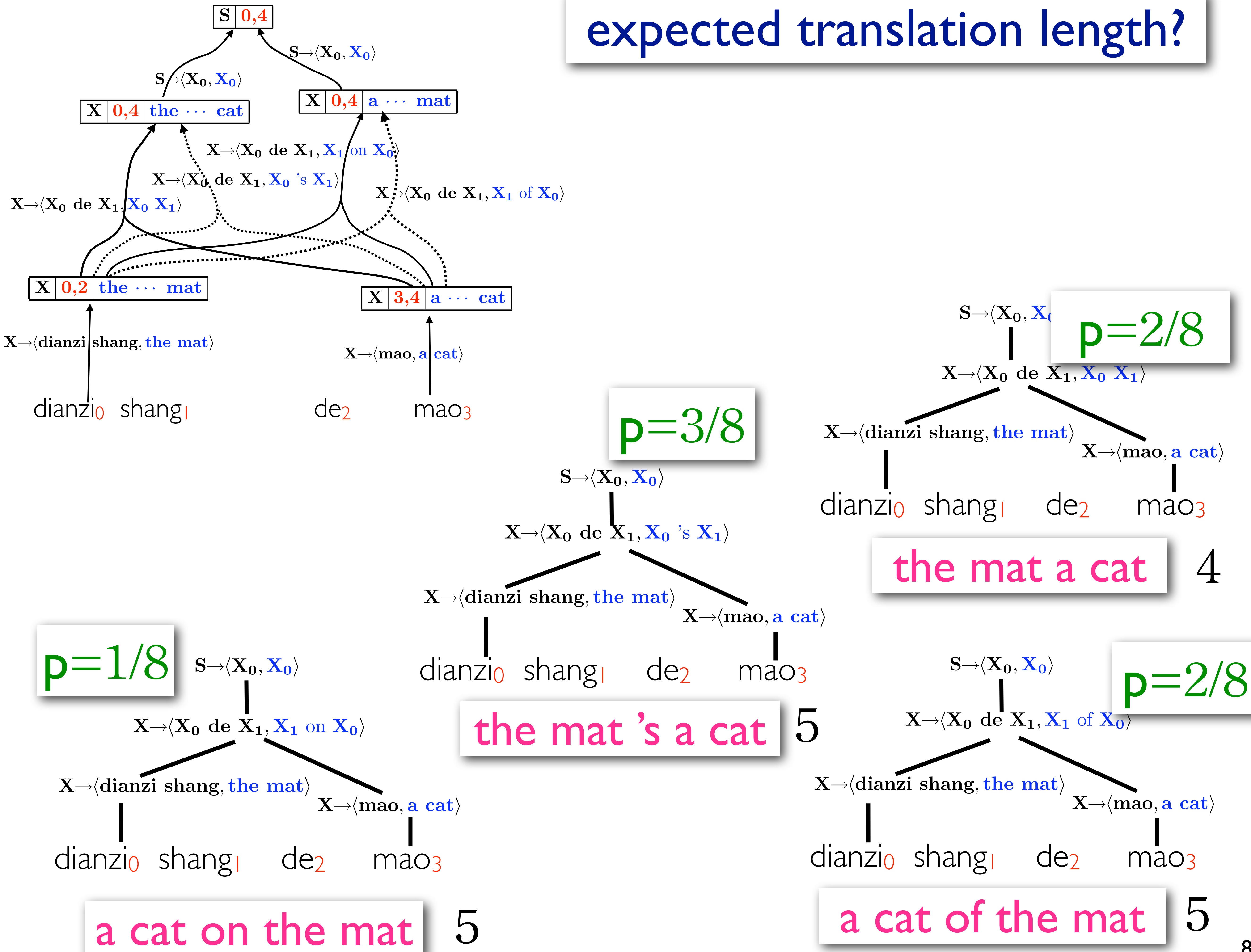
Bottom-up process in computing the number of trees


S | 0,4
S→⟨X0, X0⟩
S→⟨X0, X0⟩
X | 0,4 | the ··· cat
X | 0,4 | a ··· mat
X→⟨X0 de X1, X1 on X0⟩
X→⟨X0 de X1, X0 's X1⟩
X→⟨X0 de X1, X0 X1⟩
X→⟨X0 de X1, X1 of X0⟩
X | 0,2 | the ··· mat
X | 3,4 | a ··· cat
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
p=2/8
S→⟨X0, X0⟩
X→⟨X0 de X1, X0 X1⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
the mat a cat
p=3/8
S→⟨X0, X0⟩
X→⟨X0 de X1, X0 's X1⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
the mat 's a cat
p=1/8
S→⟨X0, X0⟩
X→⟨X0 de X1, X1 on X0⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
a cat on the mat
p=2/8
S→⟨X0, X0⟩
X→⟨X0 de X1, X1 of X0⟩
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
a cat of the mat

expected translation length?

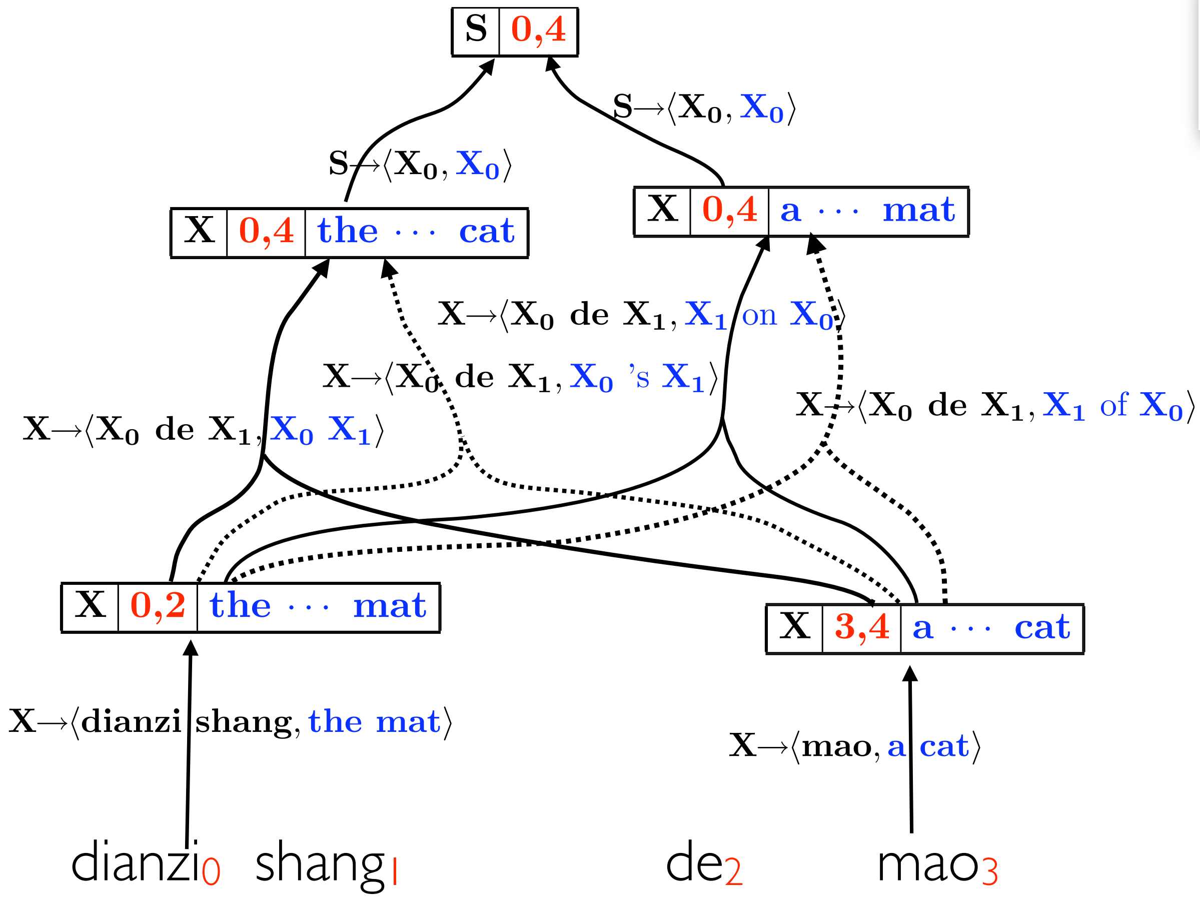


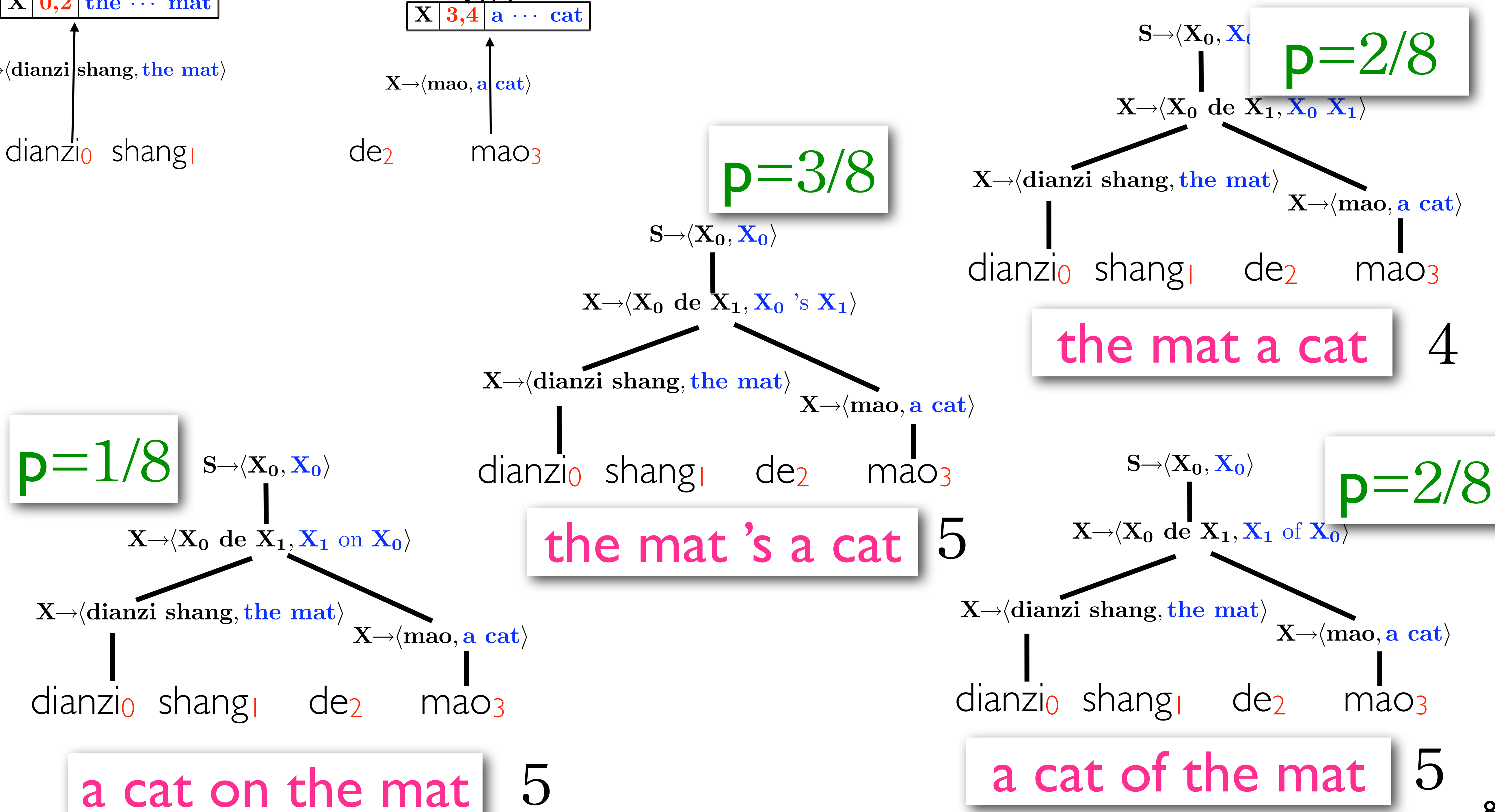



expected translation length?
S 0,4
S→⟨X0, X0⟩
S→⟨X0, X0⟩
X 0,4 the ··· cat
X 0,4 a ··· mat
X→⟨X0 de X1, X1 on X0⟩
X→⟨X0 de X1, X0 's X1⟩
X→⟨X0 de X1, X1 of X0⟩
X→⟨X0 de X1, X0 X1⟩
X 0,2 the ··· mat
X 3,4 a ··· cat
X→⟨dianzi shang, the mat⟩
X→⟨mao, a cat⟩
dianzi0 shang1 de2 mao3
p=2/8
the mat a cat
4
p=3/8
the mat 's a cat
5
p=1/8
a cat on the mat
5
p=2/8
a cat of the mat
5

expected translation length?

$2/8 \times 4 + 6/8 \times 5 = 4.75$

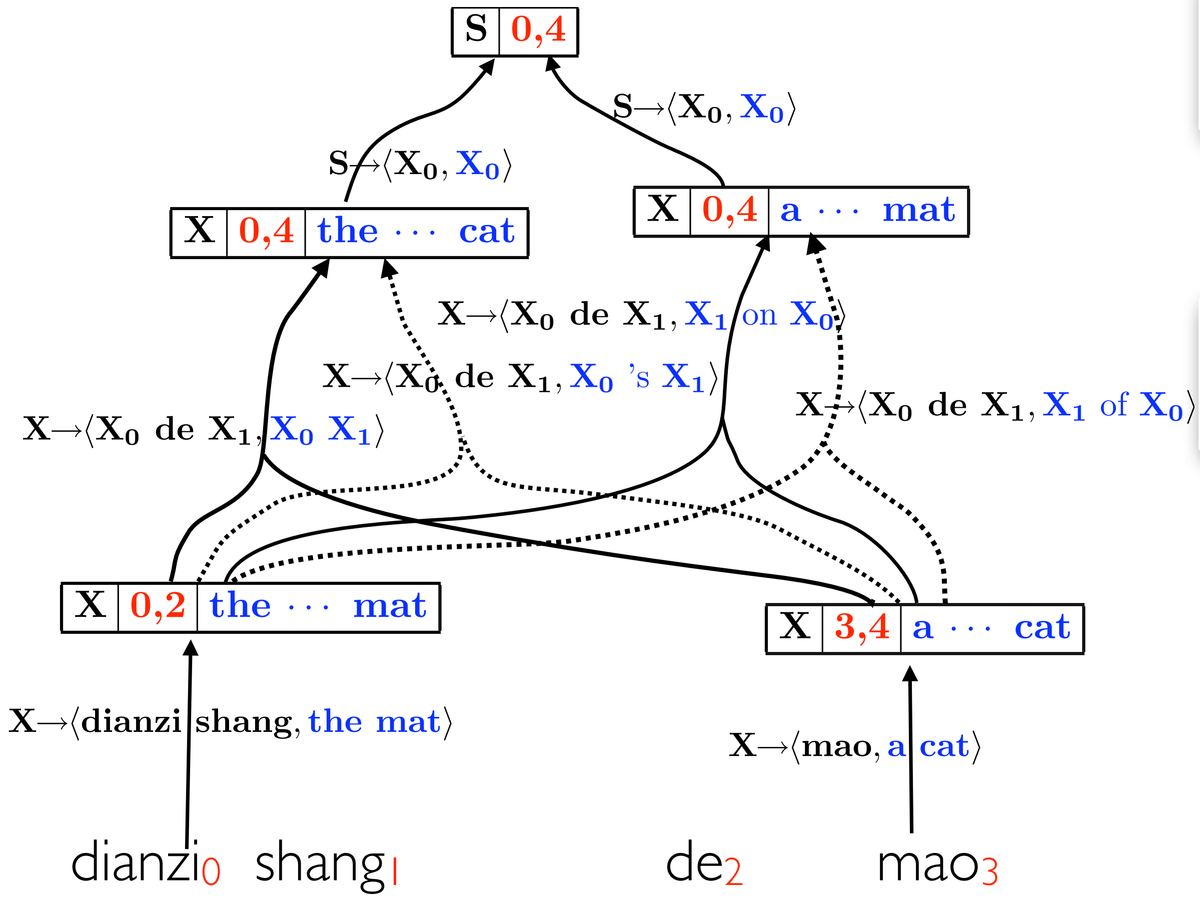




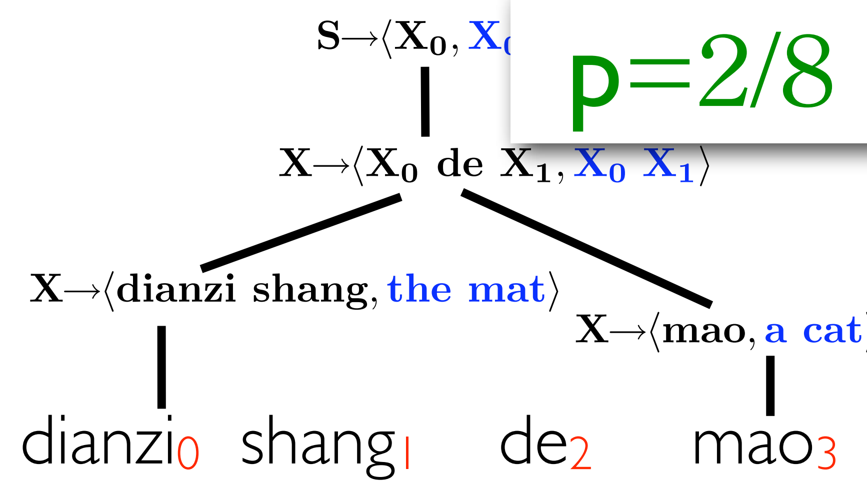
expected translation length?

$2/8 \times 4 + 6/8 \times 5 = 4.75$

variance?

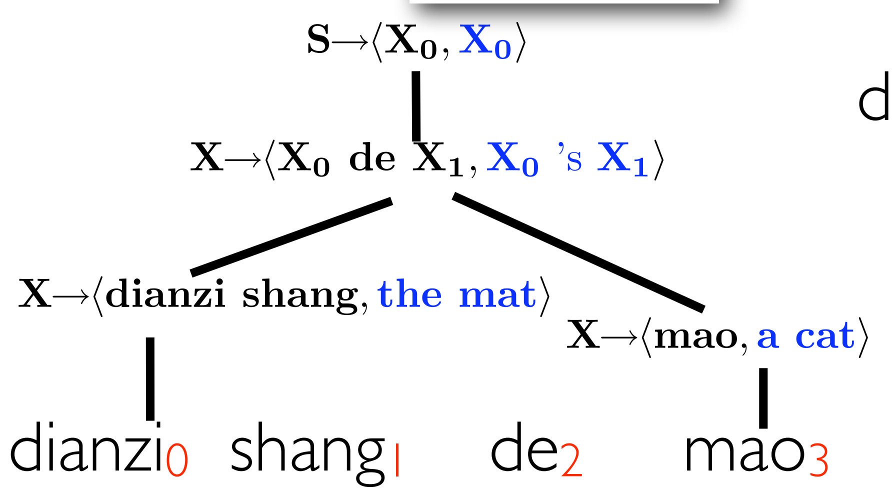


p=2/8



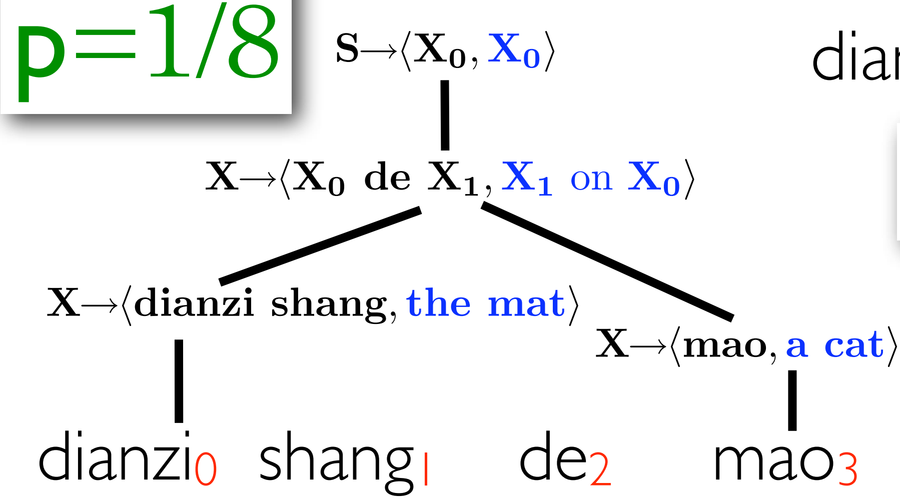
the mat a cat 4

p=3/8



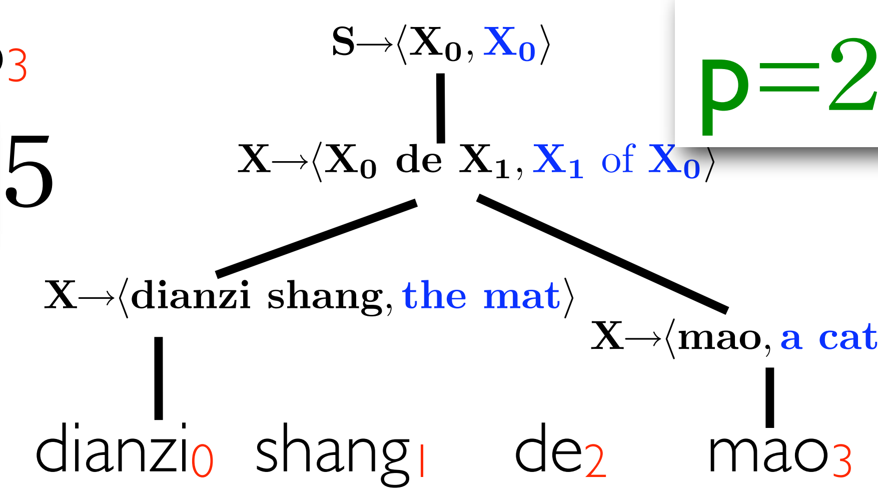
the mat 's a cat 5

p=1/8

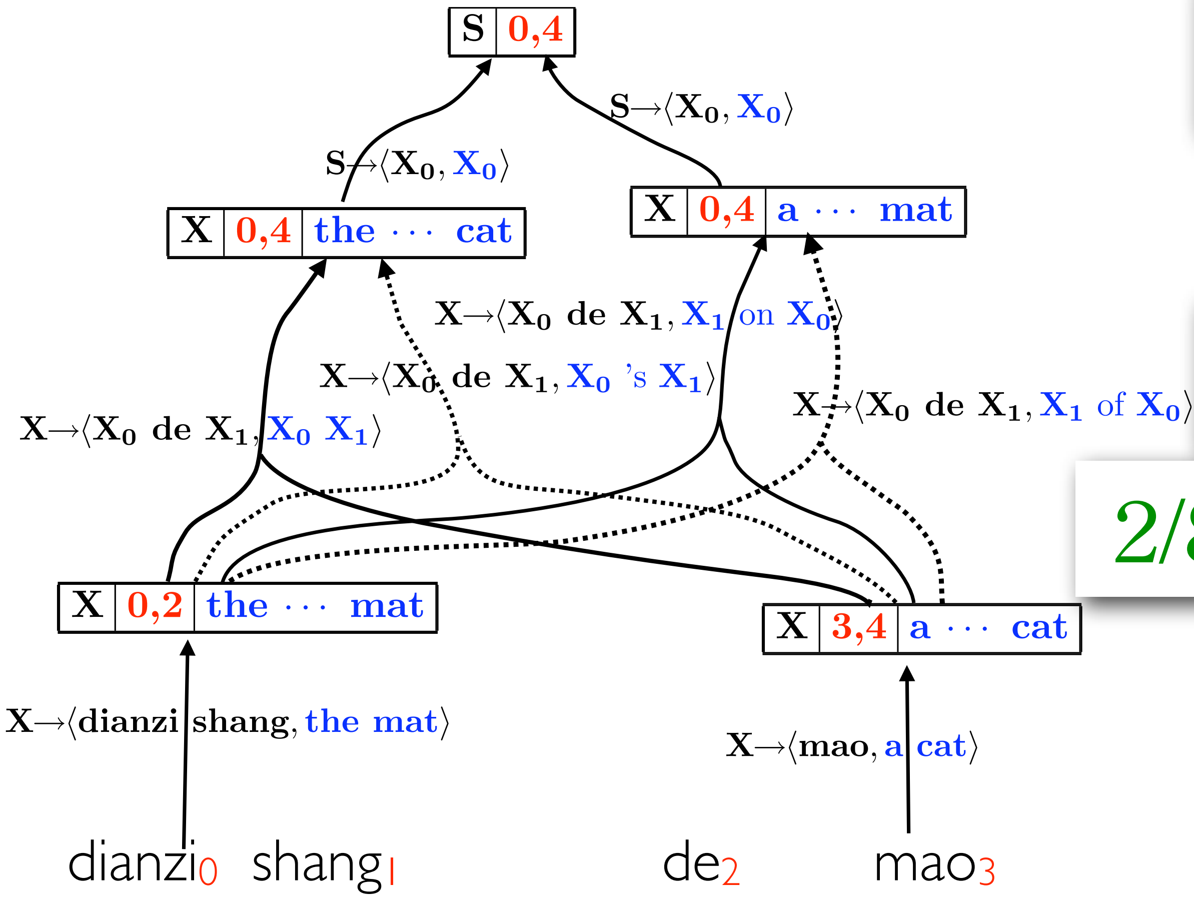


a cat on the mat 5

p=2/8



a cat of the mat 5



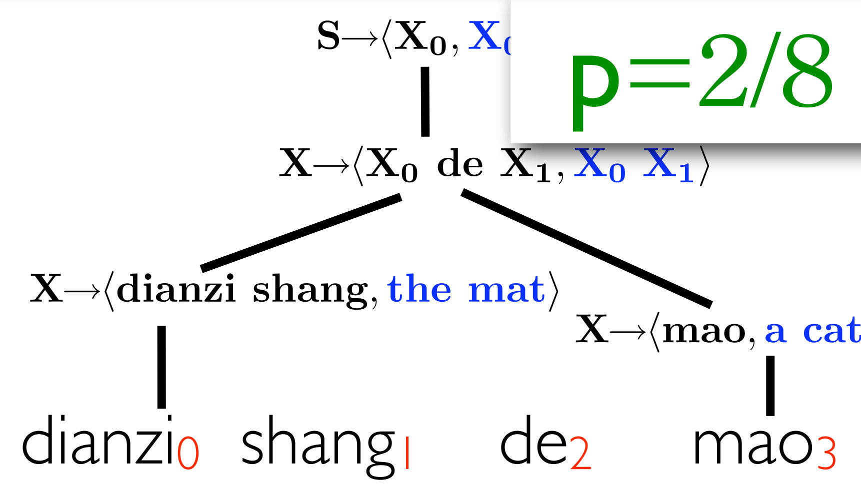
expected translation length?

$2/8 \times 4 + 6/8 \times 5 = 4.75$

variance?

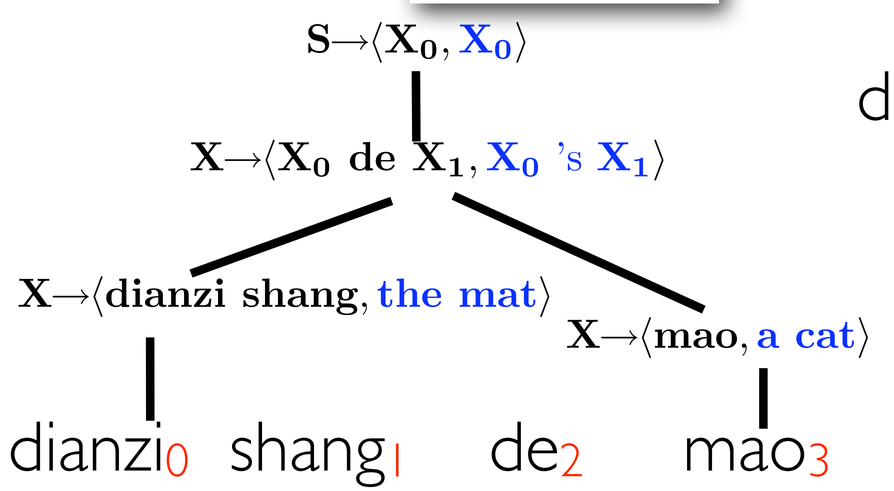
$2/8 \times (4-4.75)^2 + 6/8 \times (5-4.75)^2 \approx 0.19$

p=2/8



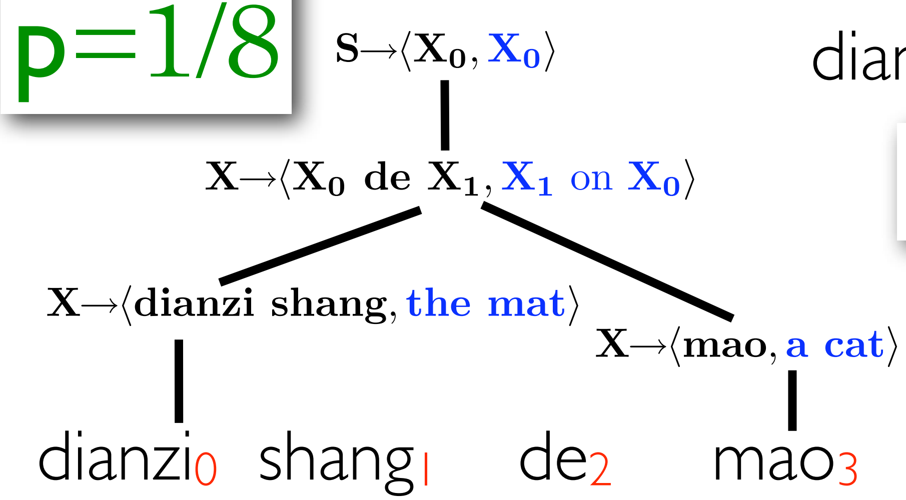
the mat a cat 4

p=3/8



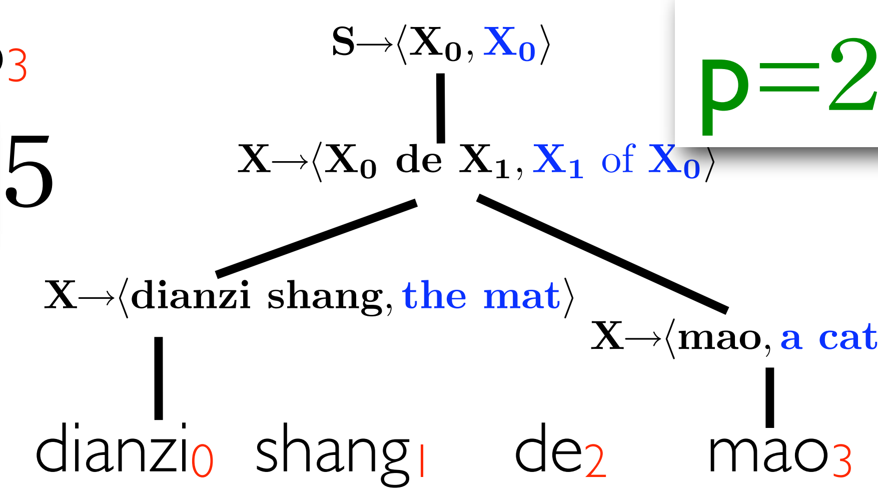
the mat 's a cat 5

p=1/8



a cat on the mat 5

p=2/8



a cat of the mat 5

First- and Second-order Expectation Semirings

First-order:

(Eisner, 2002)

- each member is a 2-tuple: $\langle p, r \rangle$

$\langle p_1, r_1 \rangle \otimes \langle p_2, r_2 \rangle$	$\langle p_1 p_2,\ p_1 r_2 + p_2 r_1 \rangle$
$\langle p_1, r_1 \rangle \oplus \langle p_2, r_2 \rangle$	$\langle p_1 + p_2,\ r_1 + r_2 \rangle$

Second-order:

- each member is a 4-tuple: $\langle p, r, s, t \rangle$

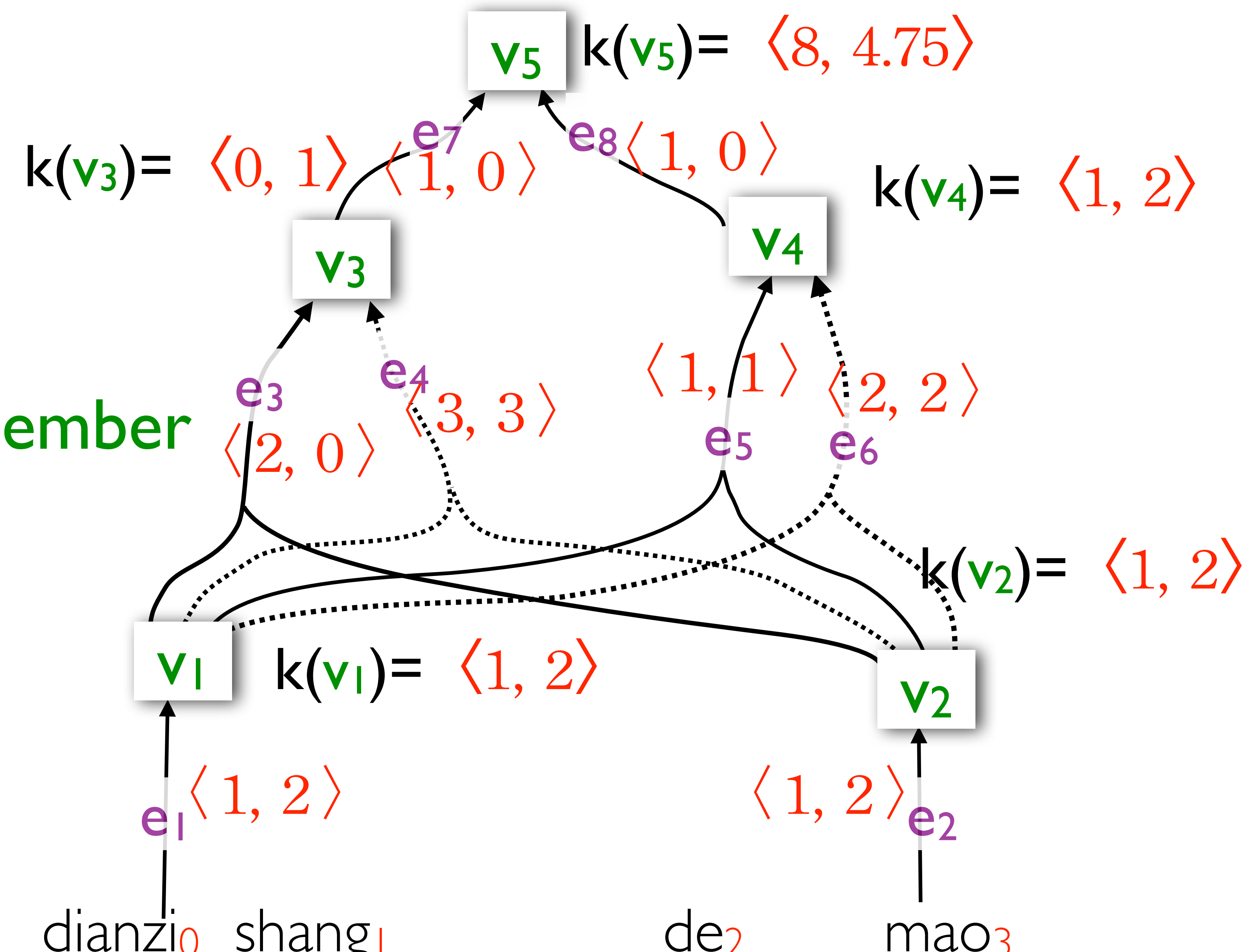
$\langle p_1, r_1, s_1, t_1 \rangle \otimes \langle p_2, r_2, s_2, t_2 \rangle$	$\langle p_1 p_2,\ p_1 r_2 + p_2 r_1,\ p_1 s_2 + p_2 s_1,$ $p_1 t_2 + p_2 t_1 + r_1 s_2 + r_2 s_1 \rangle$
$\langle p_1, r_1, s_1, t_1 \rangle \oplus \langle p_2, r_2, s_2, t_2 \rangle$	$\langle p_1 + p_2,\ r_1 + r_2,\ s_1 + s_2,\ t_1 + t_2 \rangle$

$k(v_1) = k(e_1)$ $\quad$ $k(v_2) = k(e_2)$

$k(v_3) = k(e_3) \otimes k(v_1) \otimes k(v_2) \oplus k(e_4) \otimes k(v_1) \otimes k(v_2)$

$k(v_4) = k(e_5) \otimes k(v_1) \otimes k(v_2) \oplus k(e_6) \otimes k(v_1) \otimes k(v_2)$

$k(v_5) = k(e_7) \otimes k(v_3) \oplus k(e_8) \otimes k(v_4)$



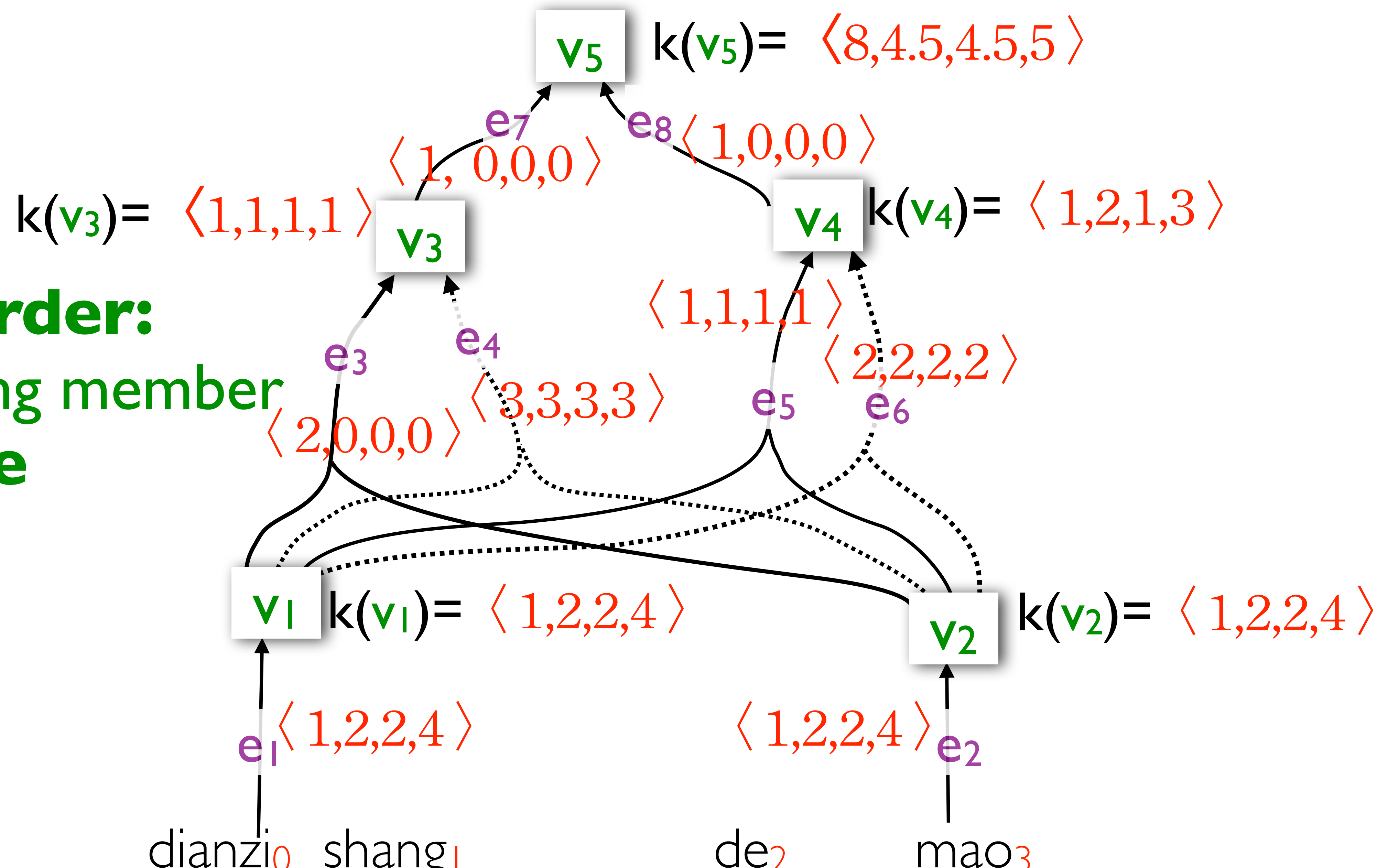
First-order:
each semiring member is a **2-tuple**

$$k(v_1) = k(e_1) \qquad k(v_2) = k(e_2)$$

$$k(v_3) = k(e_3) \otimes k(v_1) \otimes k(v_2) \oplus k(e_4) \otimes k(v_1) \otimes k(v_2)$$

$$k(v_4) = k(e_5) \otimes k(v_1) \otimes k(v_2) \oplus k(e_6) \otimes k(v_1) \otimes k(v_2)$$

$$k(v_5) = k(e_7) \otimes k(v_3) \oplus k(e_8) \otimes k(v_4)$$



Second-order:

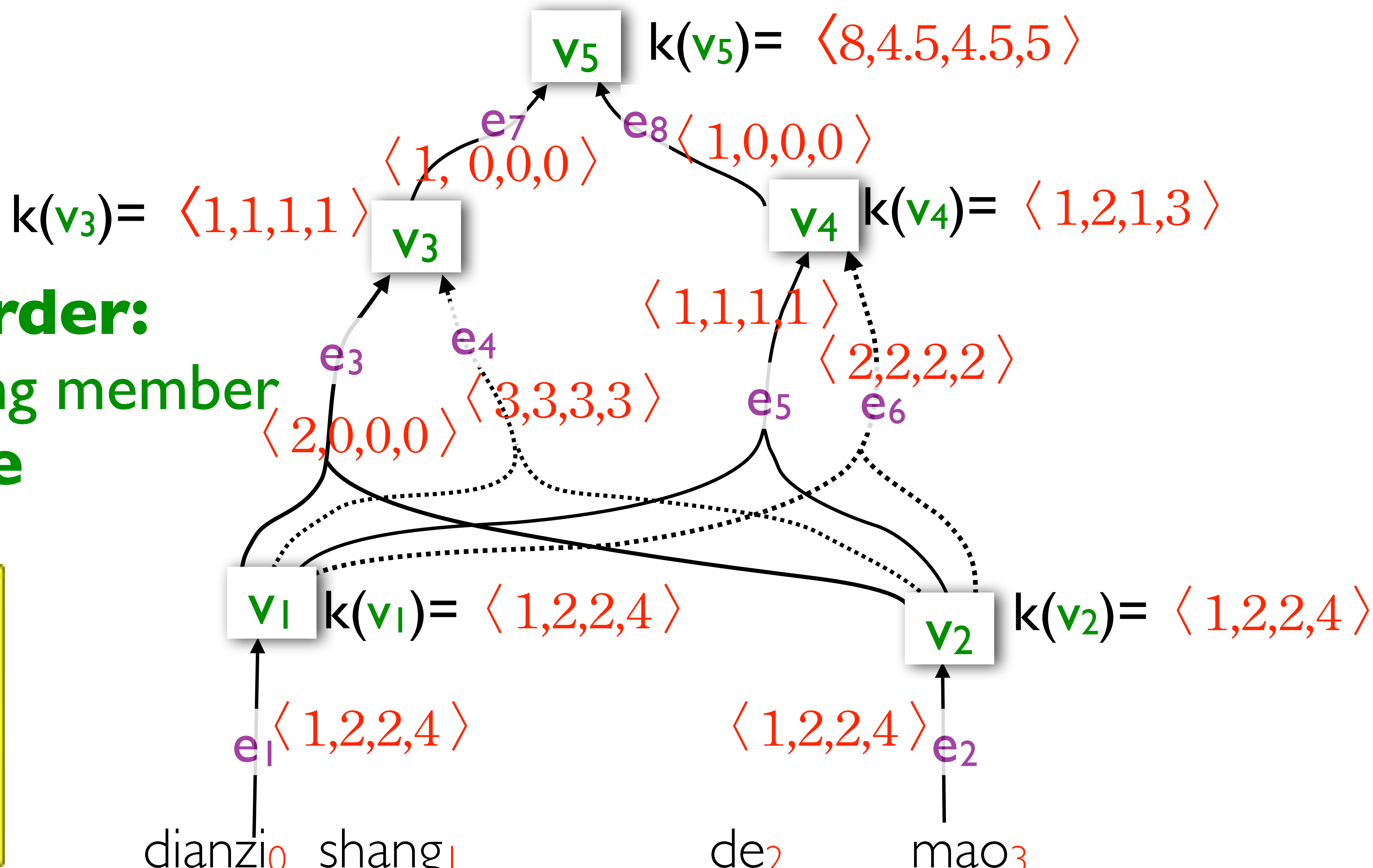
each semiring member is a **4-tuple**

$$k(v_1) = k(e_1) \qquad k(v_2) = k(e_2)$$

$$k(v_3) = k(e_3) \otimes k(v_1) \otimes k(v_2) \oplus k(e_4) \otimes k(v_1) \otimes k(v_2)$$

$$k(v_4) = k(e_5) \otimes k(v_1) \otimes k(v_2) \oplus k(e_6) \otimes k(v_1) \otimes k(v_2)$$

$$k(v_5) = k(e_7) \otimes k(v_3) \oplus k(e_8) \otimes k(v_4)$$



Second-order:

each semiring member

is a **4-tuple**

Expectations on Hypergraphs

Expectations on Hypergraphs

- Expectation over a hypergraph

Expectations on Hypergraphs

- Expectation over a hypergraph

$$\overline{r} \stackrel{\text{def}}{=} \mathbb{E}_p[r] = \sum_{d \in \text{HG}} p(d) r(d)$$

- *r*(*d*) is a function over a derivation *d*

  e.g., the length of the translation yielded by *d*

Expectations on Hypergraphs

- Expectation over a hypergraph

$$\bar{r} \stackrel{\text{def}}{=} \mathbb{E}_p[r] = \sum_{d \in \text{HG}} p(d) r(d)$$

exponential size

- *r*(*d*) is a function over a derivation *d*

  e.g., the length of the translation yielded by *d*

Expectations on Hypergraphs

- Expectation over a hypergraph

$$\bar{r} \stackrel{\text{def}}{=} \mathbb{E}_p[r] = \sum_{d \in \mathrm{HG}} p(d) r(d)$$

exponential size

- *r*(*d*) is a function over a derivation *d*

  e.g., the length of the translation yielded by *d*

- *r*(*d*) is additively decomposed

$$r(d) \stackrel{\text{def}}{=} \sum_{e \in d} r_e$$

e.g., translation length is additively decomposed!

Second-order Expectations on Hypergraphs

- Expectation of **products** over a hypergraph

$$\bar{t} \stackrel{\text{def}}{=} \mathbb{E}_p[r \cdot s] = \sum_{d \in \text{HG}} p(d) r(d) s(d)$$

exponential size

- r and s are additively decomposed

$$r(d) \stackrel{\text{def}}{=} \sum_{e \in d} r_e$$

$$s(d) \stackrel{\text{def}}{=} \sum_{e \in d} s_e$$

r and s can be identical or different functions.

Compute expectation using expectation semiring:

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy:

$$r_e \stackrel{\text{def}}{=} \log p_e$$

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy:

$$r_e \stackrel{\text{def}}{=} \log p_e$$

Why?

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy:

$$r_e \stackrel{\text{def}}{=} \log p_e$$

Why?

entropy is an **expectation**

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy:

$$r_e \stackrel{\text{def}}{=} \log p_e$$

Why?

entropy is an **expectation**

$$\mathrm{H}(p) = \mathbb{E}_p[-\log p] = -\sum_{d \in \mathrm{HG}} p(d) \log p(d)$$

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy:

$$r_e \stackrel{\text{def}}{=} \log p_e$$

Why?

entropy is an **expectation**

$$\mathrm{H}(p) = \mathbb{E}_p[-\log p] = -\sum_{d \in \mathrm{HG}} p(d) \log p(d)$$

log $p(d)$ is additively decomposed!

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy: $$r_e \stackrel{\text{def}}{=} \log p_e$$

Cross-entropy: $$r_e \stackrel{\text{def}}{=} \log q_e$$

Why?

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy: $$r_e \stackrel{\text{def}}{=} \log p_e$$

Cross-entropy: $$r_e \stackrel{\text{def}}{=} \log q_e$$

Why? cross-entropy is an **expectation**

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy: $$r_e \stackrel{\text{def}}{=} \log p_e$$

Cross-entropy: $$r_e \stackrel{\text{def}}{=} \log q_e$$

Why?

cross-entropy is an **expectation**

$$\mathrm{H}(p, q) = \mathbb{E}_p(-\log q) = -\sum_{d \in \mathrm{HG}} p(d) \log q(d)$$

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy: $r_e \stackrel{\text{def}}{=} \log p_e$

Cross-entropy: $r_e \stackrel{\text{def}}{=} \log q_e$

Why?

cross-entropy is an **expectation**

$$\mathrm{H}(p, q) = \mathbb{E}_p(-\log q) = -\sum_{d \in \mathrm{HG}} p(d) \log q(d)$$

log *q*(*d*) is additively decomposed!

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy:

$$r_e \stackrel{\text{def}}{=} \log p_e$$

Cross-entropy:

$$r_e \stackrel{\text{def}}{=} \log q_e$$

Bayes risk:

$$r_e \stackrel{\text{def}}{=} \text{loss at edge } e$$

Why?

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy: $r_e \stackrel{\text{def}}{=} \log p_e$

Cross-entropy: $r_e \stackrel{\text{def}}{=} \log q_e$

Bayes risk: $r_e \stackrel{\text{def}}{=} \text{loss at edge } e$

Why? Bayes risk is an **expectation**

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy: $r_e \stackrel{\text{def}}{=} \log p_e$

Cross-entropy: $r_e \stackrel{\text{def}}{=} \log q_e$

Bayes risk: $r_e \stackrel{\text{def}}{=} \text{loss at edge } e$

Why?

Bayes risk is an **expectation**

$$\text{Risk} = \mathbb{E}_p(L) = -\sum_{d \in \text{HG}} p(d) \cdot L(Y(d))$$

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy: $r_e \stackrel{\text{def}}{=} \log p_e$

Cross-entropy: $r_e \stackrel{\text{def}}{=} \log q_e$

Bayes risk: $r_e \stackrel{\text{def}}{=} \text{loss at edge } e$

Why?

Bayes risk is an **expectation**

$$\text{Risk} = \mathbb{E}_p(L) = -\sum_{d \in \text{HG}} p(d) \cdot L(Y(d))$$

L(*Y*(*d*)) is additively decomposed!

Compute expectation using expectation semiring:

$$k_e \stackrel{\text{def}}{=} \langle p_e, p_e r_e \rangle$$

p_e : transition probability or log-linear score at edge e

r_e ?

Entropy: $r_e \stackrel{\text{def}}{=} \log p_e$

Cross-entropy: $r_e \stackrel{\text{def}}{=} \log q_e$

Bayes risk: $r_e \stackrel{\text{def}}{=} \text{loss at edge } e$

Why?

Bayes risk is an **expectation**

$$\text{Risk} = \mathbb{E}_p(L) = -\sum_{d \in \text{HG}} p(d) \cdot L(Y(d))$$

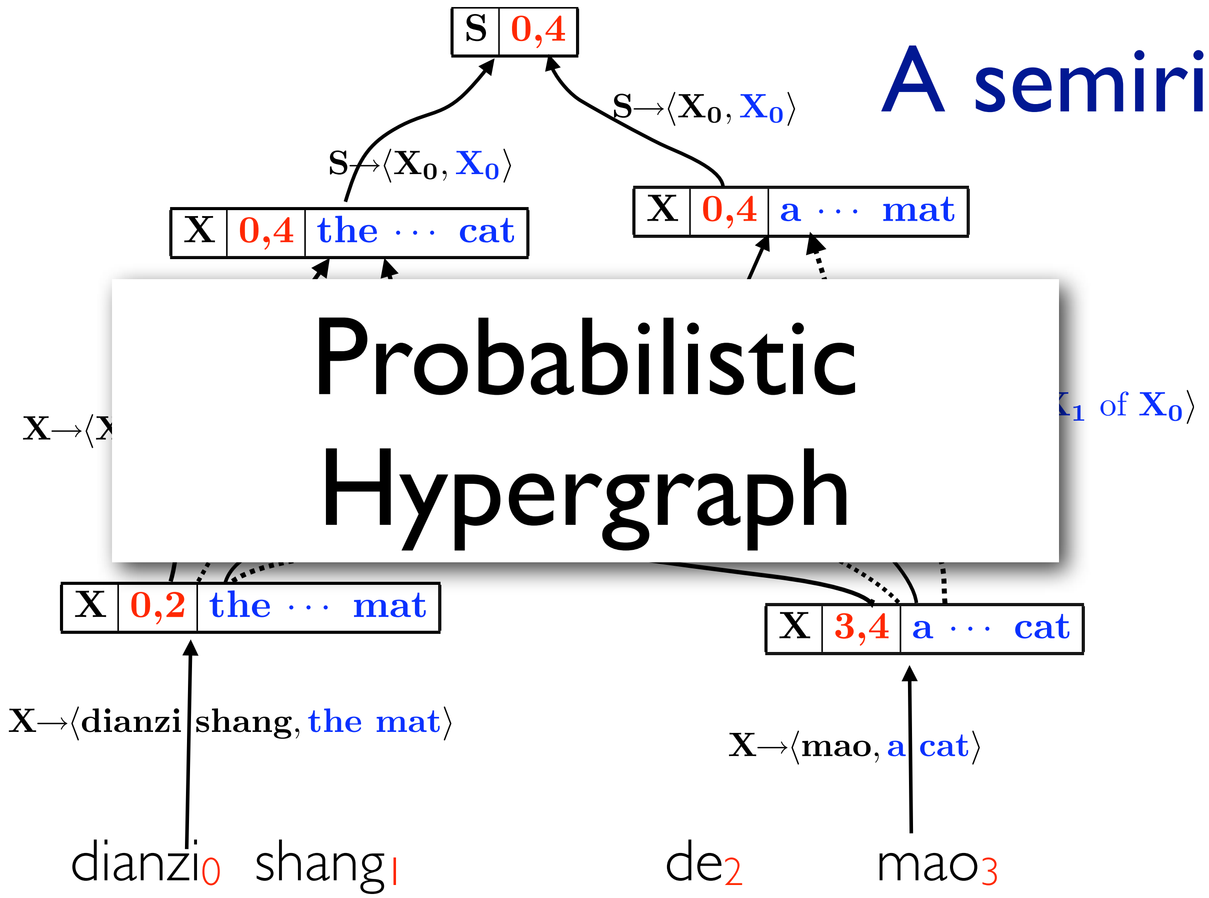
L(*Y*(*d*)) is additively decomposed!

(Tromble et al. 2008)

Applications of Expectation Semirings: a Summary

Quantity	k_e	$k_{\textbf{root}}$	**Final**
Expectation	$\langle p_e, p_e r_e \rangle$	$\langle Z, \bar{r} \rangle$	$\bar{r}/Z$
Entropy	$r_e \stackrel{\text{def}}{=} \log p_e$, so $k_e = \langle p_e, p_e \log p_e \rangle$	$\langle Z, \bar{r} \rangle$	$\log Z - \bar{r}/Z$
Cross-entropy	$\langle q_e \rangle$ $r_e \stackrel{\text{def}}{=} \log q_e$, so $k_e = \langle p_e, p_e \log q_e \rangle$	$\langle Z_q \rangle$ $\langle Z_p, \bar{r} \rangle$	$\log Z_q - \bar{r}/Z_p$
Bayes risk	$r_e \stackrel{\text{def}}{=} \mathrm{L}_e$, so $k_e = \langle p_e, p_e \mathrm{L}_e \rangle$	$\langle Z, \bar{r} \rangle$	$\bar{r}/Z$
First-order gradient	$\langle p_e, \nabla p_e \rangle$	$\langle Z, \nabla Z \rangle$	∇Z
Covariance matrix	$\langle p_e, p_e r_e, p_e s_e, p_e r_e s_e \rangle$	$\langle Z, \bar{r}, \bar{s}, \bar{t} \rangle$	$\frac{\bar{t}}{Z} - \frac{\bar{r}\,\bar{s}^{\mathbf{T}}}{Z^2}$
Hessian matrix	$\langle p_e, \nabla p_e, \nabla p_e, \nabla^2 p_e \rangle$	$\langle Z, \nabla Z, \nabla Z, \nabla^2 Z \rangle$	$\nabla^2 Z$
Gradient of expectation	$\langle p_e, p_e r_e, \nabla p_e, (\nabla p_e) r_e + p_e (\nabla r_e) \rangle$	$\langle Z, \bar{r}, \nabla Z, \nabla \bar{r} \rangle$	$\frac{Z \nabla \bar{r} - \bar{r} \nabla Z}{Z^2}$
Gradient of entropy	$\langle p_e, p_e \log p_e, \nabla p_e, (1 + \log p_e) \nabla p_e \rangle$	$\langle Z, \bar{r}, \nabla Z, \nabla \bar{r} \rangle$	$\frac{\nabla Z}{Z} - \frac{Z \nabla \bar{r} - \bar{r} \nabla Z}{Z^2}$
Gradient of risk	$\langle p_e, p_e \mathrm{L}_e, \nabla p_e, \mathrm{L}_e \nabla p_e \rangle$	$\langle Z, \bar{r}, \nabla Z, \nabla \bar{r} \rangle$	$\frac{Z \nabla \bar{r} - \bar{r} \nabla Z}{Z^2}$

A semiring framework to compute all of these



- "decoding" quantities:
  - Viterbi
  - K-best
  - Counting
  - ......

- First-order quantities:
  - expectation
    - entropy
    - Bayes risk
    - cross-entropy
    - KL divergence
    - feature expectations
  - first-order gradient of Z

- Second-order quantities:
  - Expectation over product
    - interaction between features
  - Hessian matrix of Z
    - second-order gradient descent
  - gradient of expectation
    - gradient of entropy or Bayes risk