Bayesian Symbol-Refined Tree Substitution Grammars for Syntactic Parsing

Hiroyuki Shindo
NTT CS Labs.

Yusuke Miyao
NII

Akinori Fujino
NTT CS Labs.

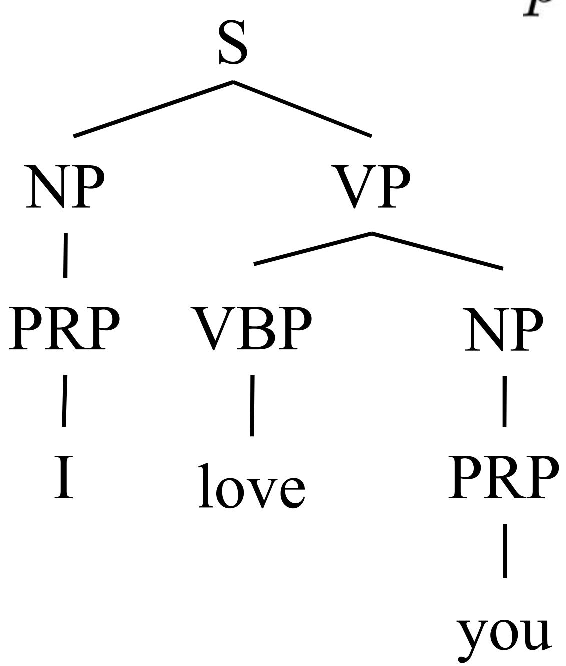
Masaaki Nagata
NTT CS Labs.

July 10
ACL 2012

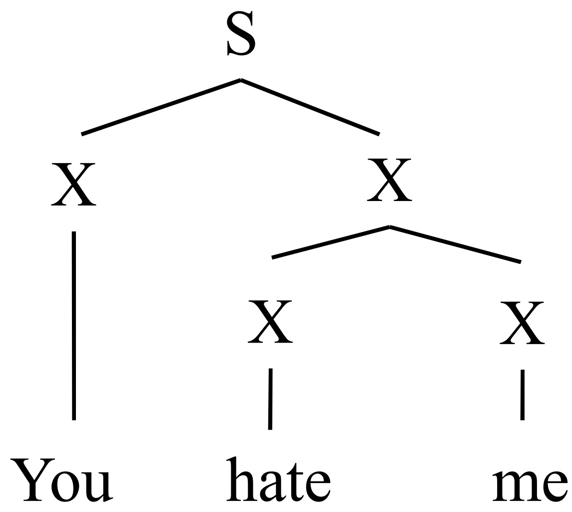
Task: Statistical Constituent Parsing

Training

$p\left(\text{tree}\right)$



Testing



F-score: ?

Previous Work

- Naive CFG-based parser does NOT perform well...
    - Coarse symbol annotations
    - Strong independence assumption

Previous approaches:

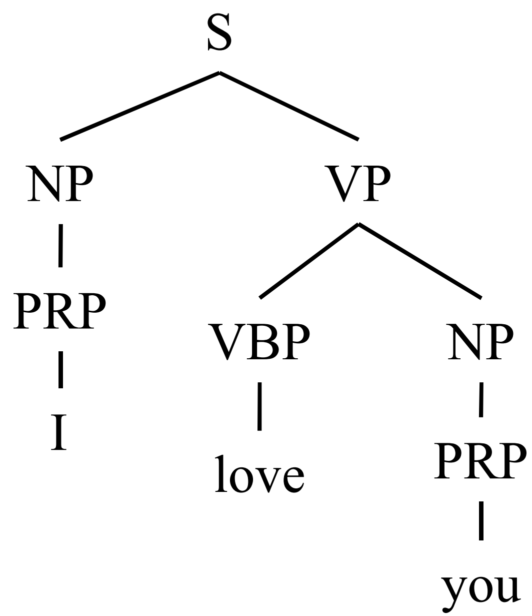
a) CFG with automatic symbol refinement
[Matsuzaki et al. 05, Petrov et al. 06]

b) Tree substitution grammars (TSG)
[Cohn et al. 09, Post et al. 09]

a) CFG with automatic symbol refinement

- Idea: split symbols into subcategories
  (based on the likelihood)

- Inference: EM algorithm



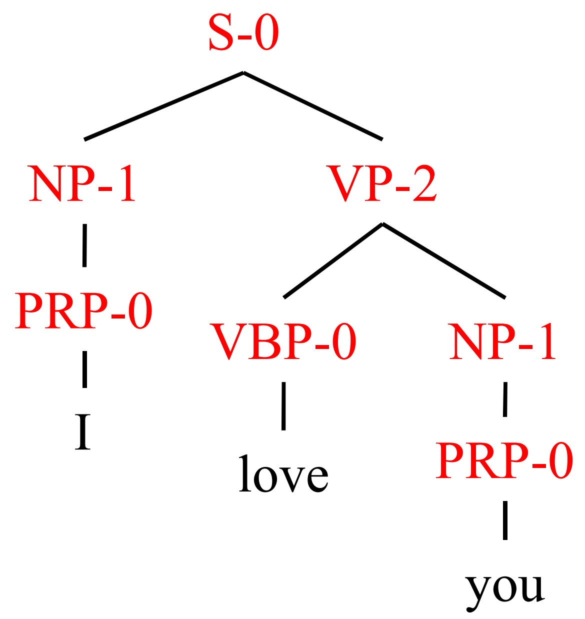
a) CFG with automatic symbol refinement

- Idea: split symbols into subcategories (based on the likelihood)

- Inference: EM algorithm

Refined PCFG rules:

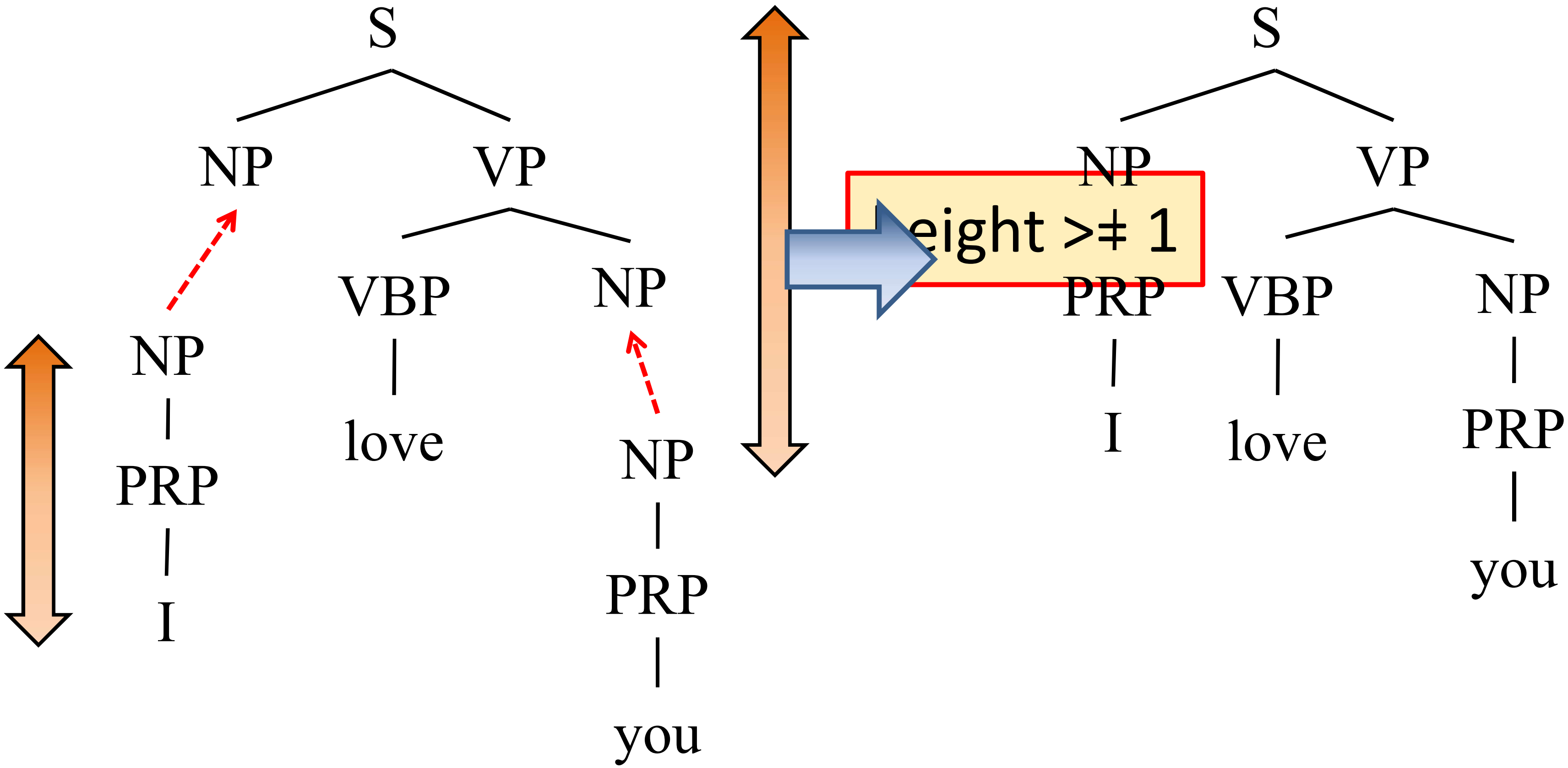
S-0	$\rightarrow$	NP-1 VP-2	p_0
NP-1	$\rightarrow$	PRP-0	p_1
PRP-0	$\rightarrow$	"I"	p_2
VP-2	$\rightarrow$	VBP-0 NP-1	p_3



b) Tree Substitution Grammars (TSG)

- Idea: Allow arbitrarily large tree fragments

- Inference: MCMC sampling



Motivation

Two approaches are complementary

[Cohn et al. 09, Bansal & Klein 10]

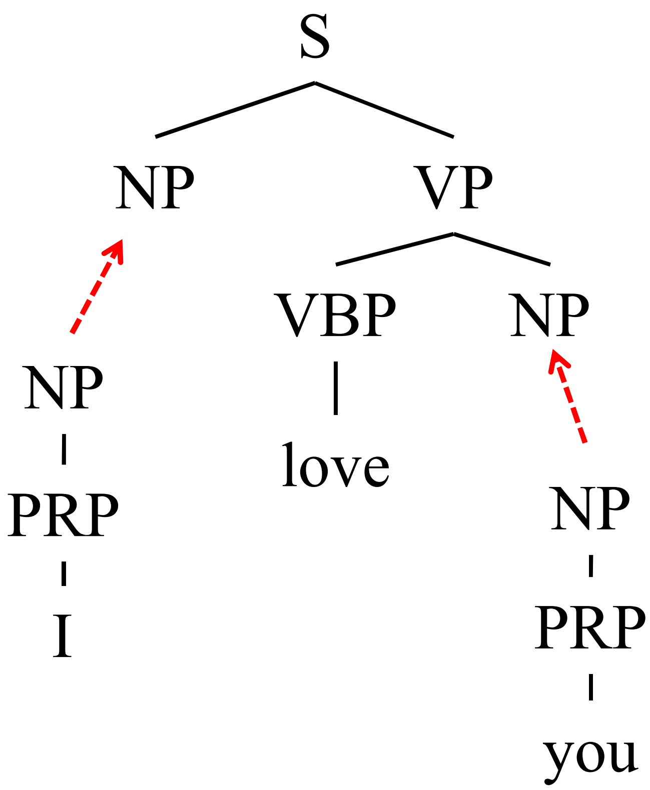
- Clustering context
- Learning structure

Proposal:

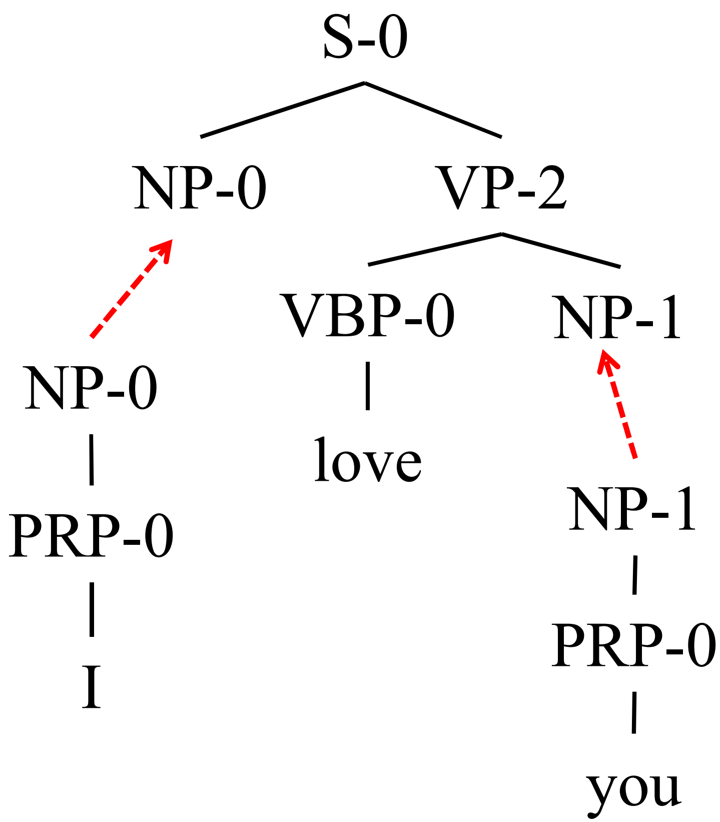
Symbol-Refined TSG (SR-TSG)

SR-TSG

TSG



SR-TSG

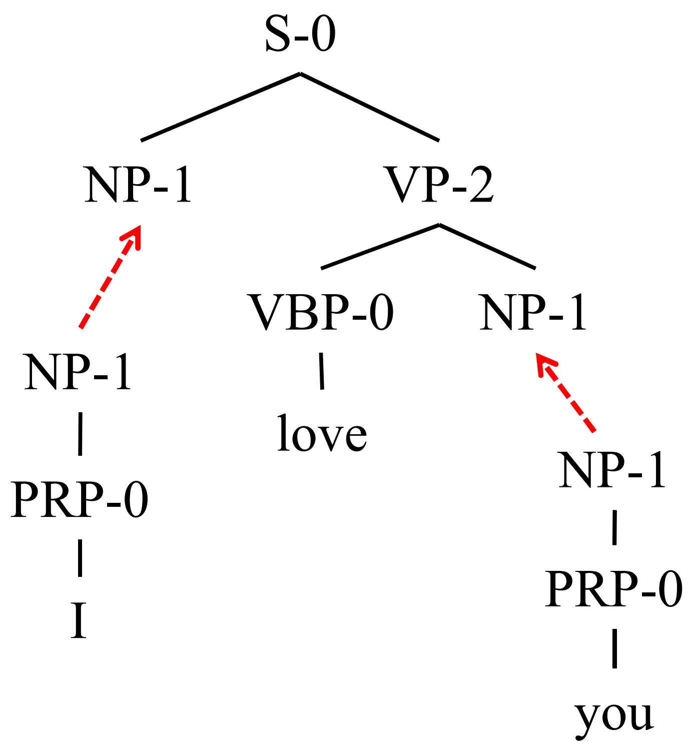


SR-TSG

- Latent variables of SR-TSG:

tree fragments + symbol subcategories

Search space is huge !



Challenge:

<u>Fully automatic learning</u>

⇔ TSG + refinement [Bansal & Klein 10]

Probabilistic model

Inference

Probabilistic Model of SR-TSG

Overview of Probabilistic Model

What we need:

1. Probability distribution over refined tree fragments

   Pitman-Yor process

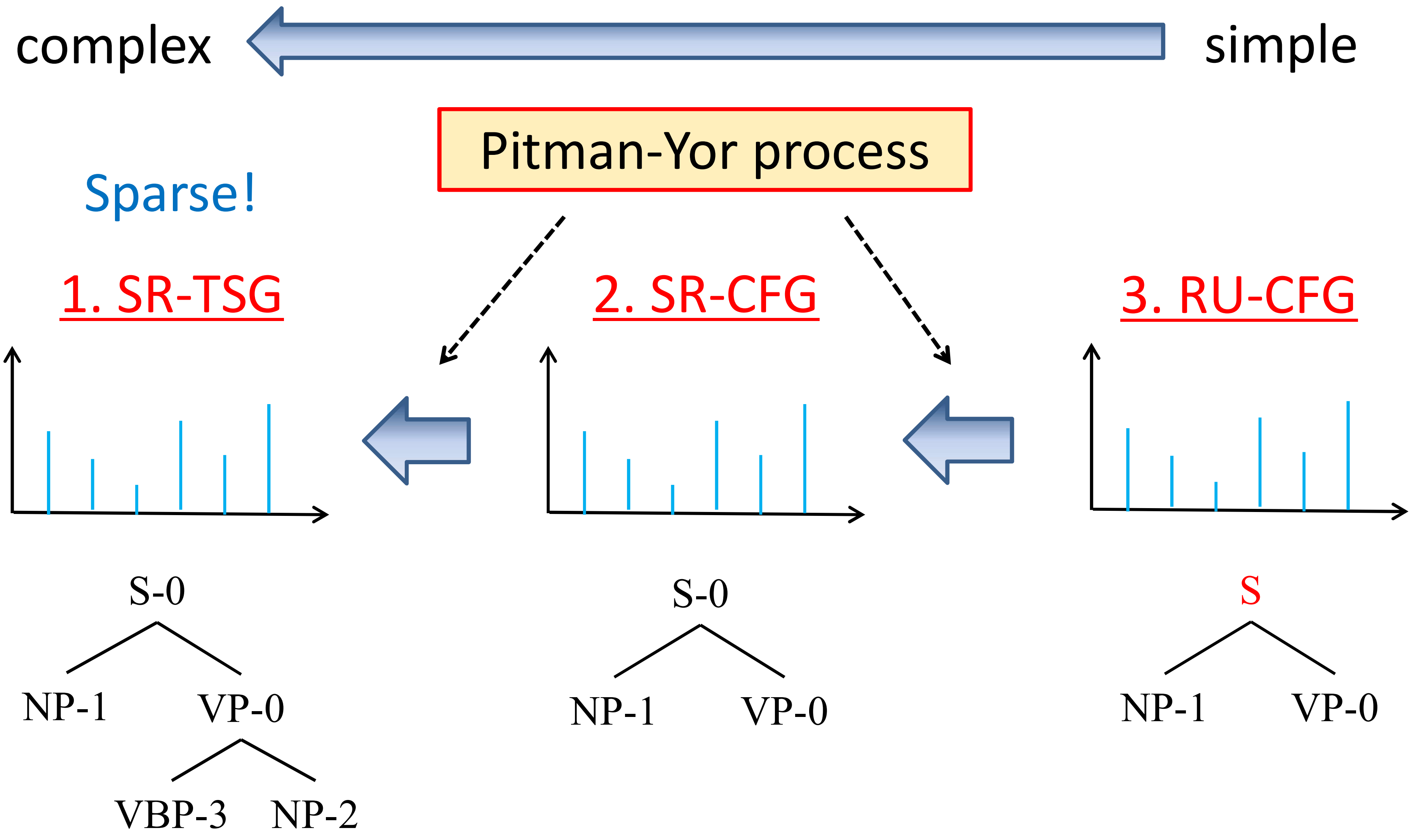
   [Pitman & Yor 96]

2. Framework for back-off smoothing

   Large $\longrightarrow$ small tree fragments

   3-level hierarchy

3-Level Hierarchy



Pitman-Yor Process (PYP)

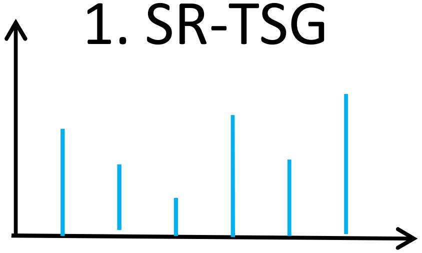
- A prior for non-parametric Bayesian model

- Useful for modeling data with power-law distribution

- Closely related to Chinese Restaurant Process (CRP)

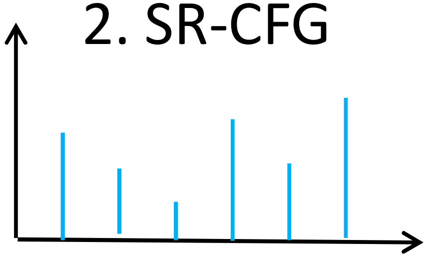
Pitman-Yor Process (PYP)

1. SR-TSG

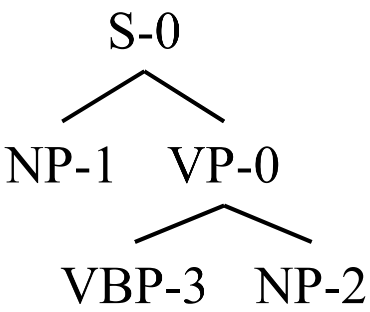


PYP

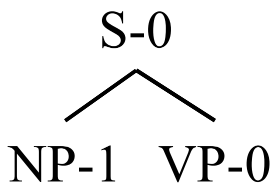
2. SR-CFG



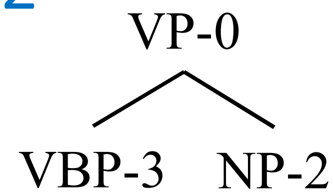
A



A1



A2



Eq. $$\boxed{p\left(A\right) = \alpha \cdot count\left(A\right) + \beta \cdot \underline{P_0\left(A\right)}}$$ Base distr.

"Rich get richer" effect

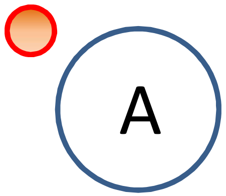
$$\propto p'\left(A1\right) \times p'\left(A2\right)$$

smoothing

Chinese Restaurant Process

Chinese Restaurant Process

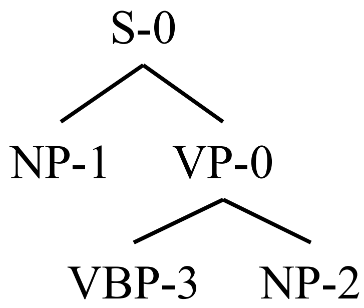
$$\mathrm{PYP}\left(d, \theta, P_0\right)$$



1

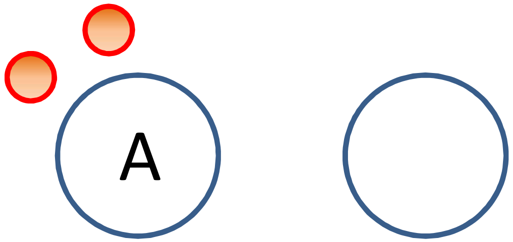
$\times P_0\left(A\right)$

A

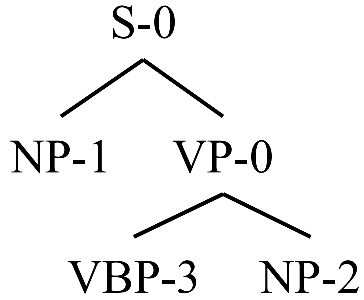


Chinese Restaurant Process

$$\mathrm{PYP}\left(d, \theta, P_0\right)$$

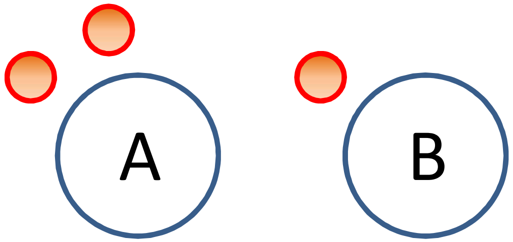


$1 - d$ $\qquad$ $\theta + d$



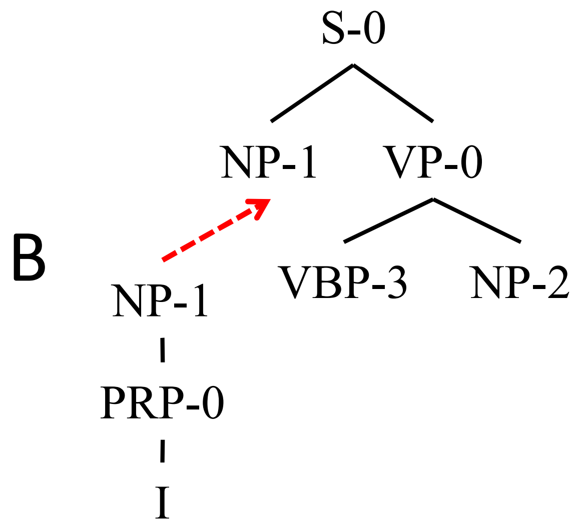
Chinese Restaurant Process

$$\mathrm{PYP}\left(d, \theta, P_0\right)$$



$2 - d$ $\qquad$ $\theta + d$

$\times P_0\left(B\right)$

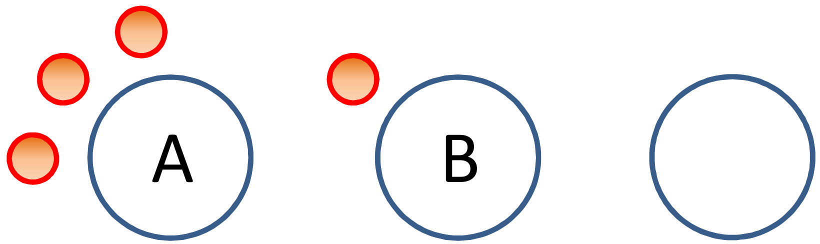


- Prob. of choosing a table $\propto \begin{cases} \mathrm{count}\left(k\right) - d & \text{occupied} \\ \theta + \#\ \mathrm{tables} \cdot d & \text{new table} \end{cases}$

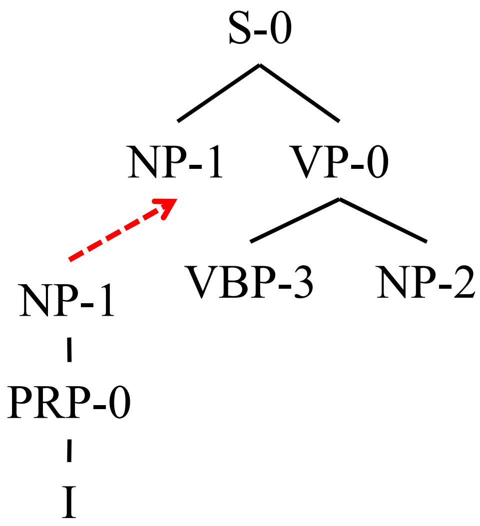
- “Rich get richer” effect

Chinese Restaurant Process

$$\text{PYP}\left(d, \theta, P_0\right)$$



$2 - d$ $\quad 1 - d$ $\quad \theta + 2d$

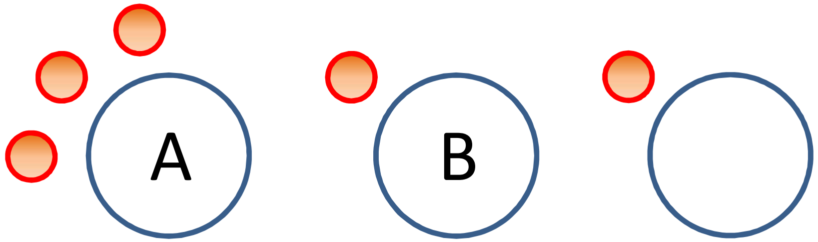


- Prob. of choosing a table $\propto \begin{cases} \text{count}\left(k\right) - d & \text{occupied} \\ \theta + \#\ \text{tables} \cdot d & \text{new table} \end{cases}$

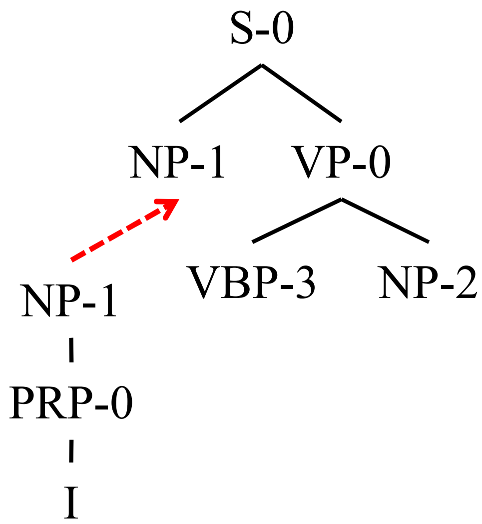
- “Rich get richer” effect

Chinese Restaurant Process

$$\mathrm{PYP}\left(d, \theta, P_0\right)$$



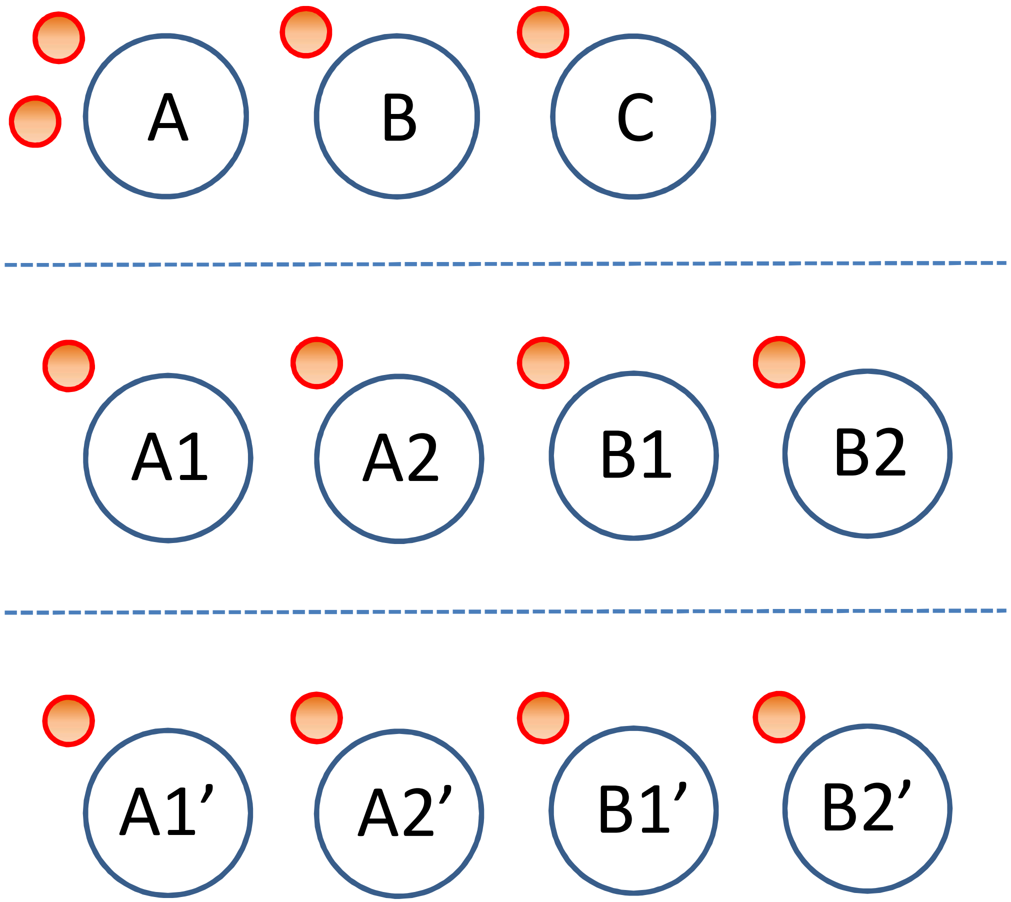
$3 - d$ $\quad 1 - d$ $\quad \theta + 2d$

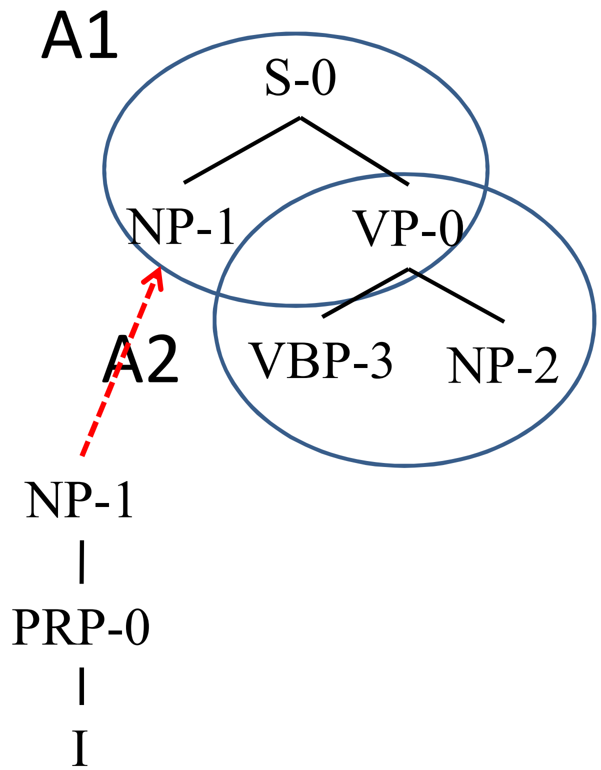


- Prob. of choosing a table $\propto \begin{cases} \mathrm{count}\,(k) - d & \text{occupied} \\ \theta + \#\ \mathrm{tables} \cdot d & \text{new table} \end{cases}$

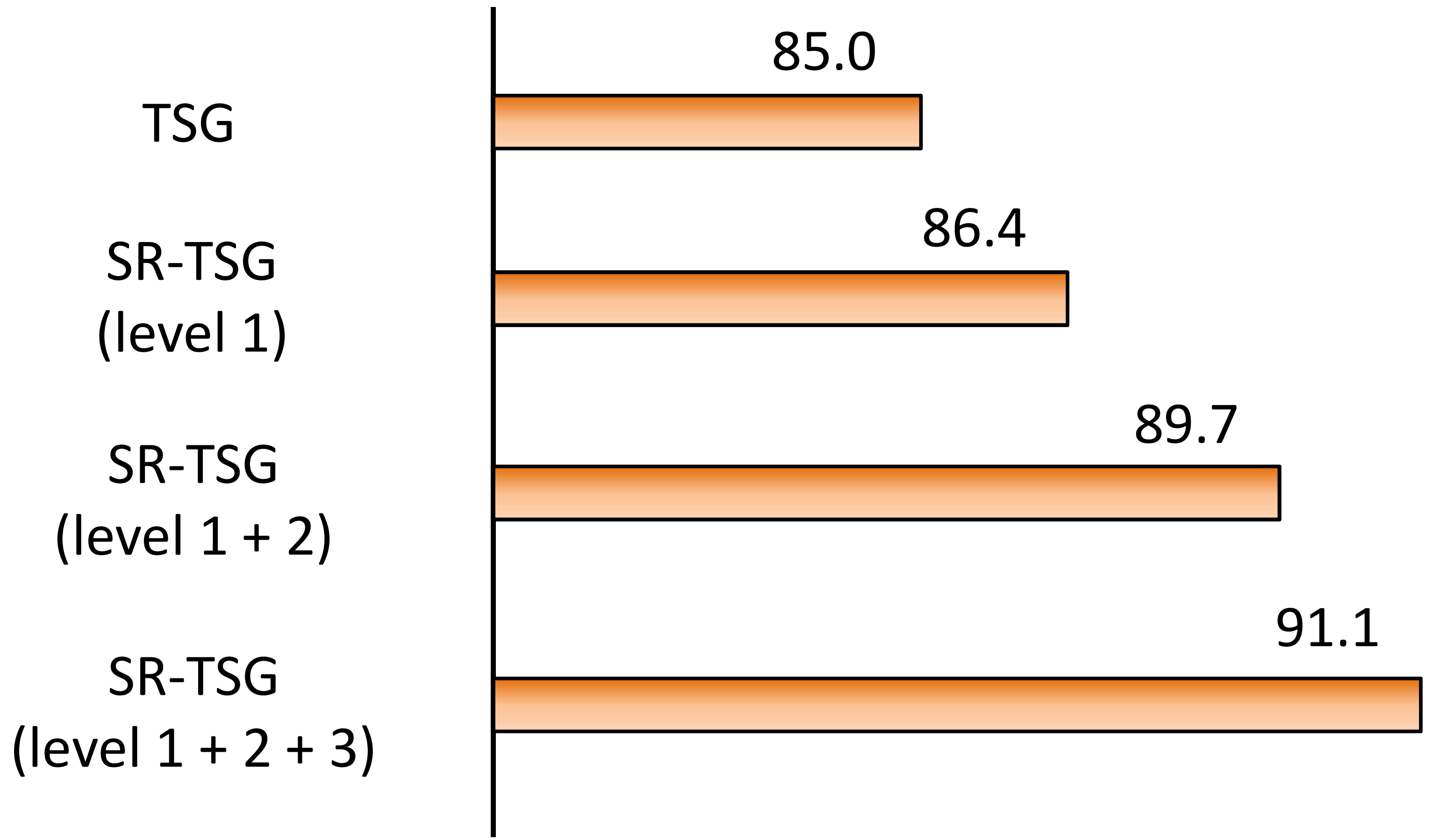
- “Rich get richer” effect

Chinese Restaurant Process





Effect of back-off smoothing



Summary of Probabilistic Model

- Probability distribution over SR-TSG fragments
  - Pitman-Yor process as a prior

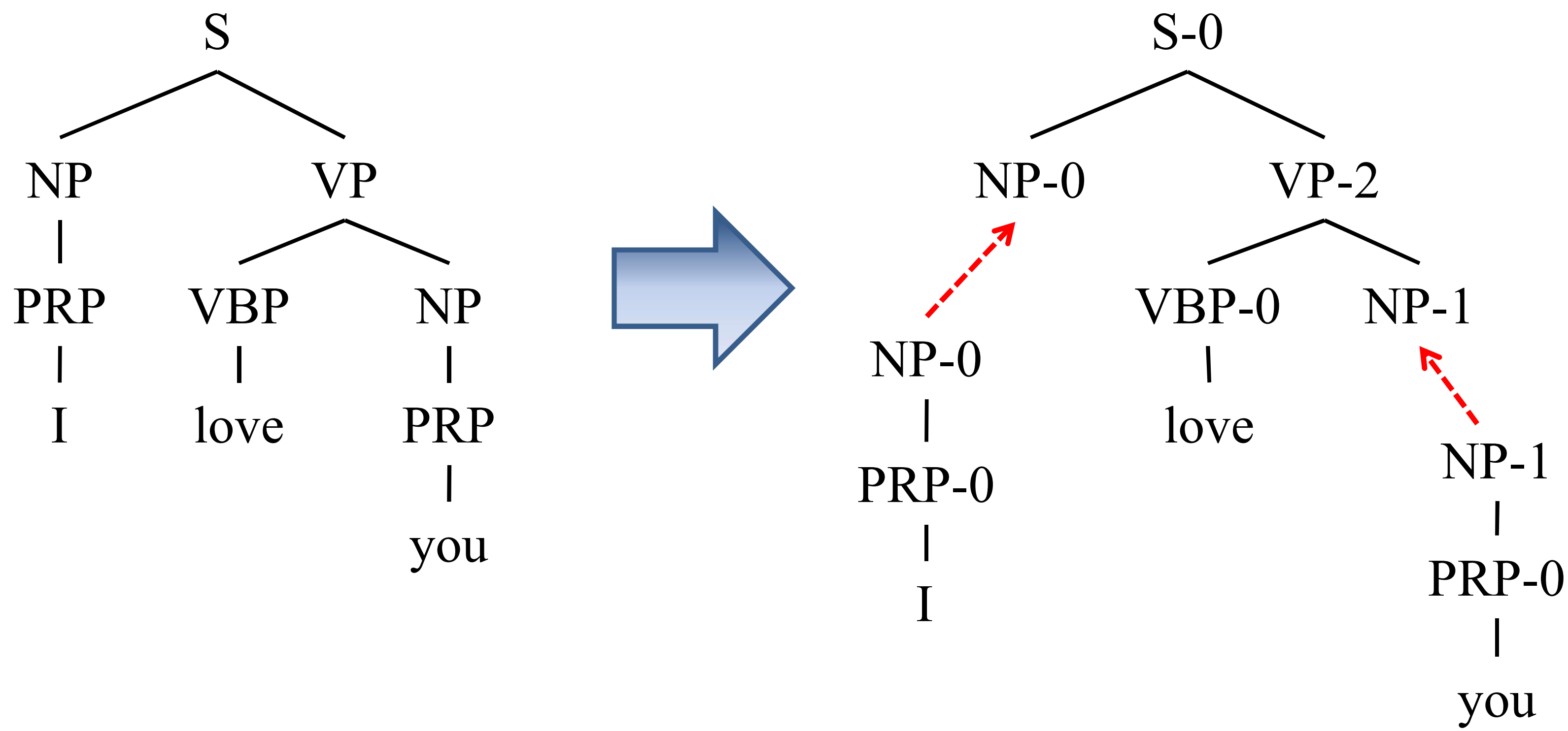
- 3-level hierarchy for back-off smoothing
  - SR-TSG ← SR-CFG ← RU-CFG

- Parsing accuracy(f-score): 91.1% on English PTB

Inference

Overview of Inference

What we want:



Overview of Inference

- MAP estimation: $\underset{\mathbf{Z}}{\text{argmax}}\ p(\mathbf{Z}\,|\,\mathbf{T})$

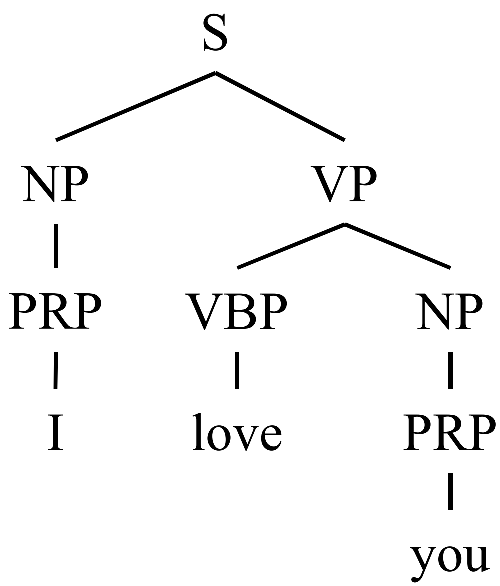
- MCMC sampling for inference

tree fragments	+	symbol subcategories

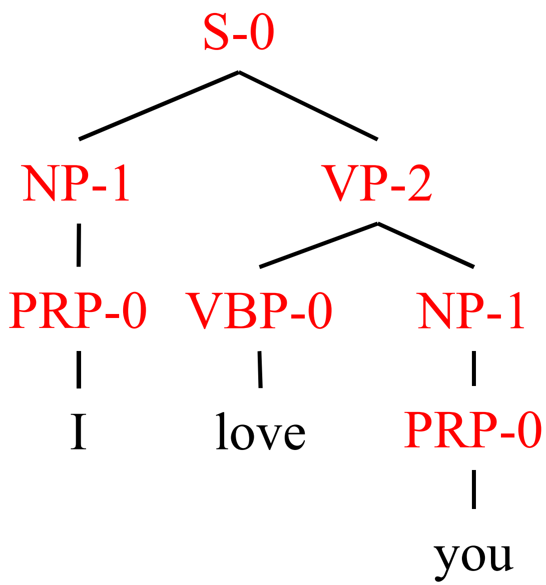
Stepwise training:

	tree fragments	symbol subcategories
1.	Fix	Train
2.	Train	Almost fixed

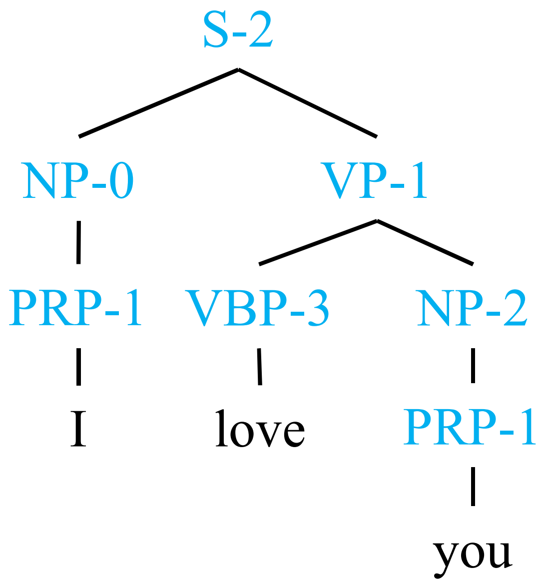
Overview of MCMC Sampling



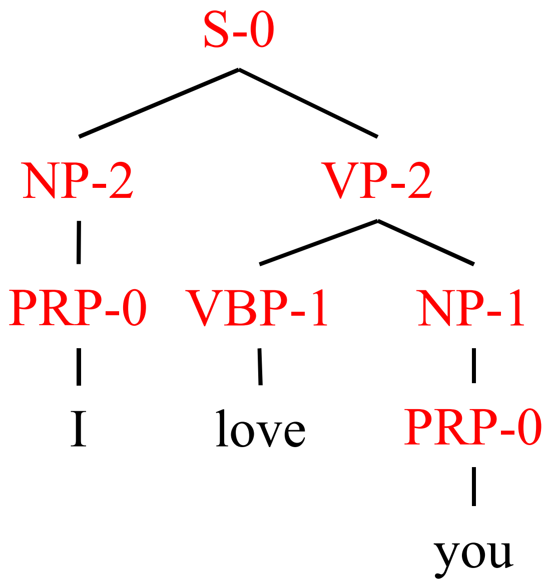
Overview of MCMC Sampling



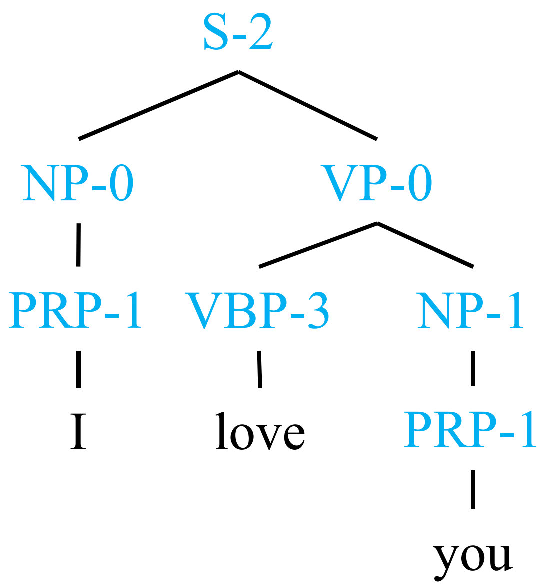
Overview of MCMC Sampling



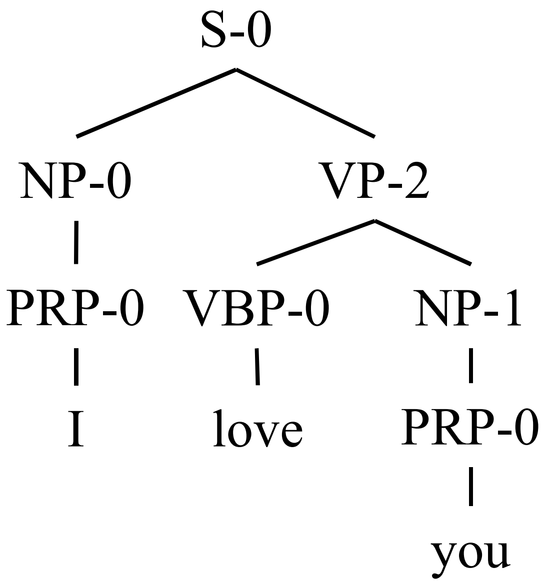
Overview of MCMC Sampling



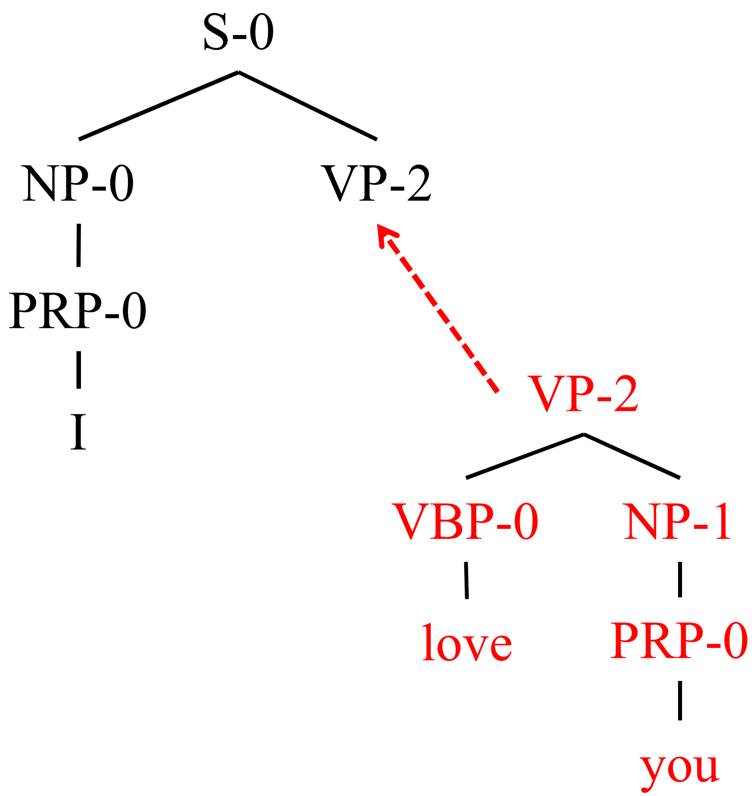
Overview of MCMC Sampling



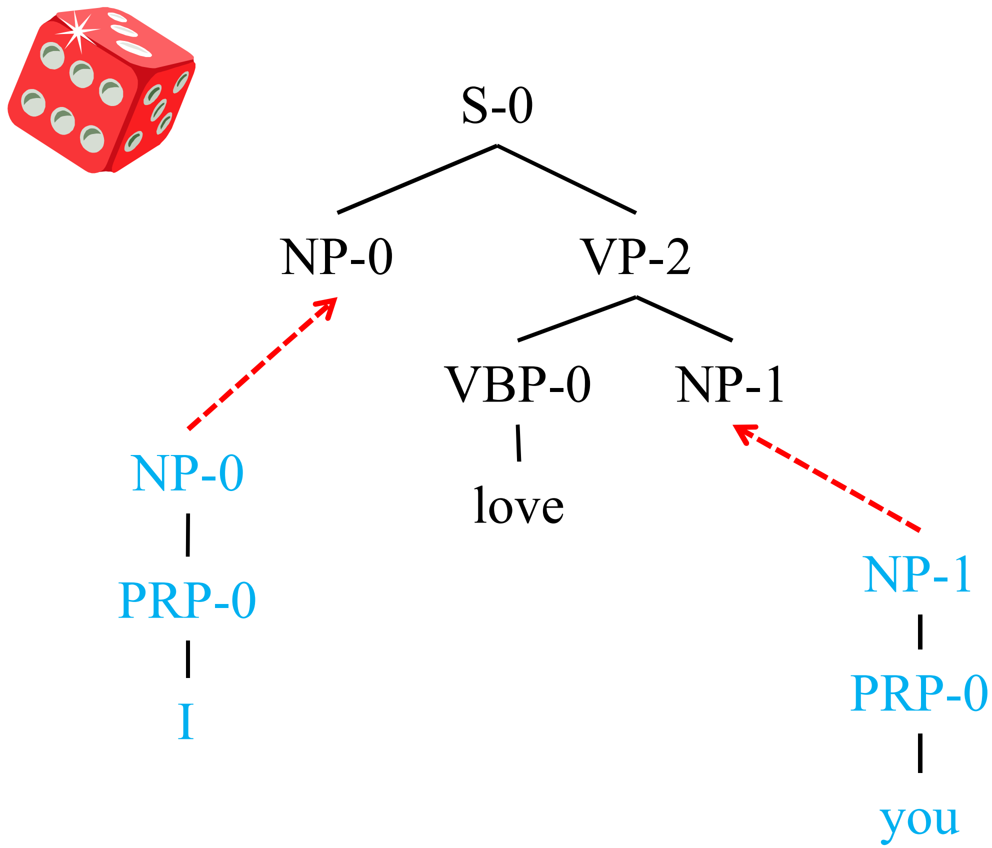
Overview of MCMC Sampling



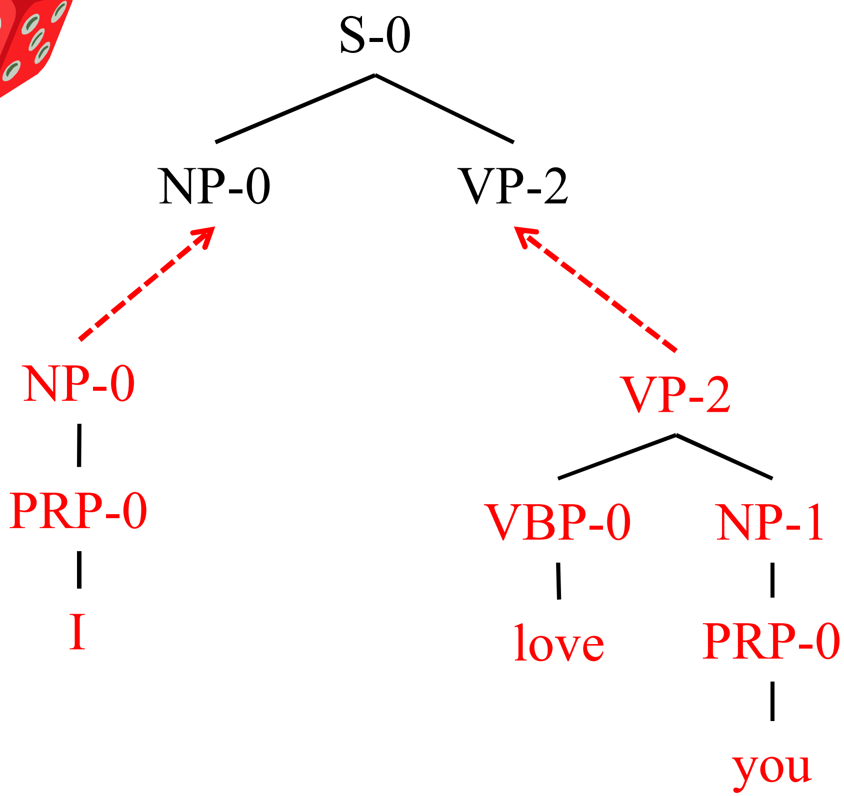
Overview of MCMC Sampling



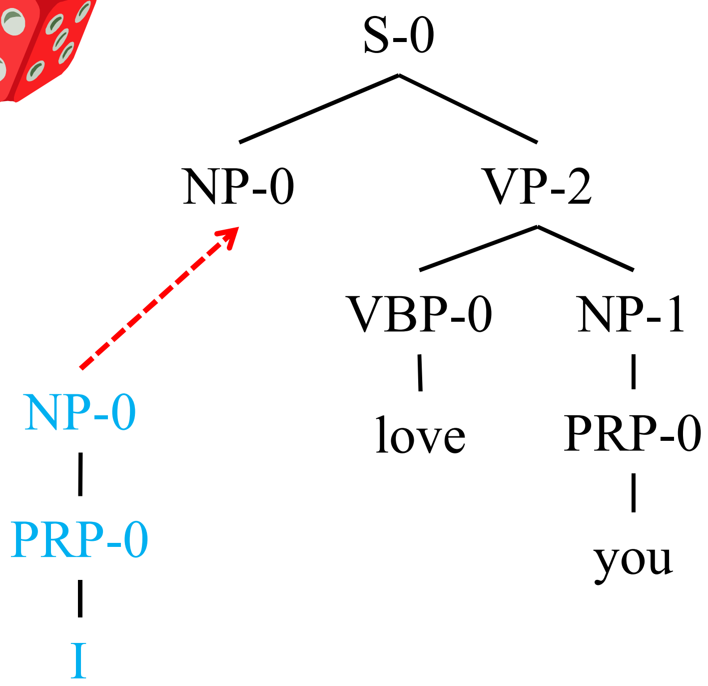
Overview of MCMC Sampling



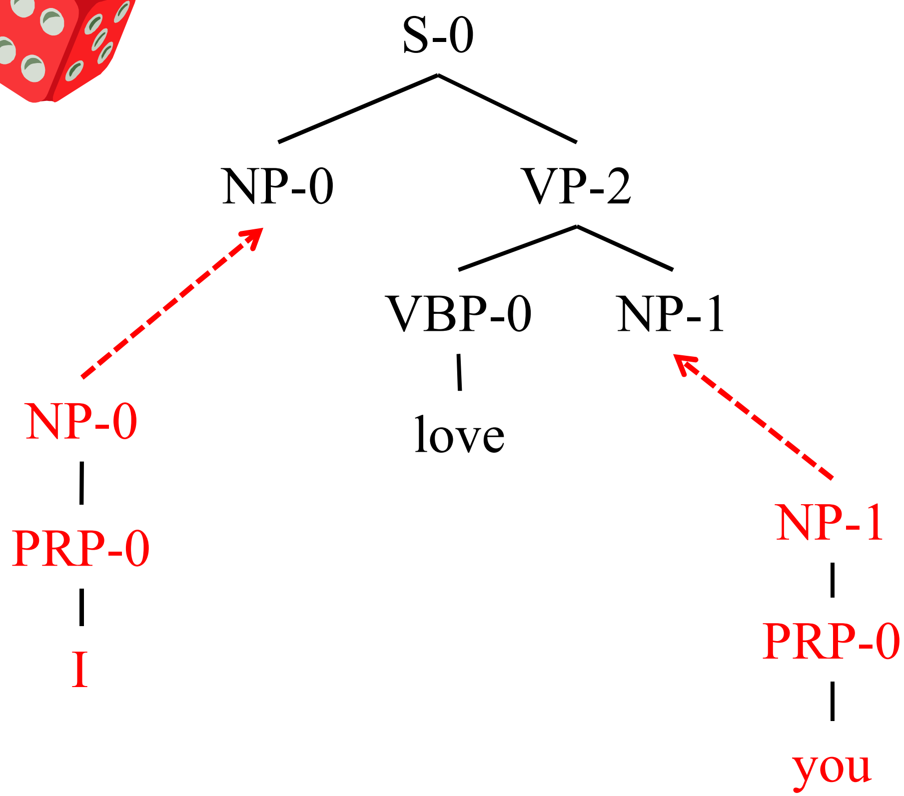
Overview of MCMC Sampling



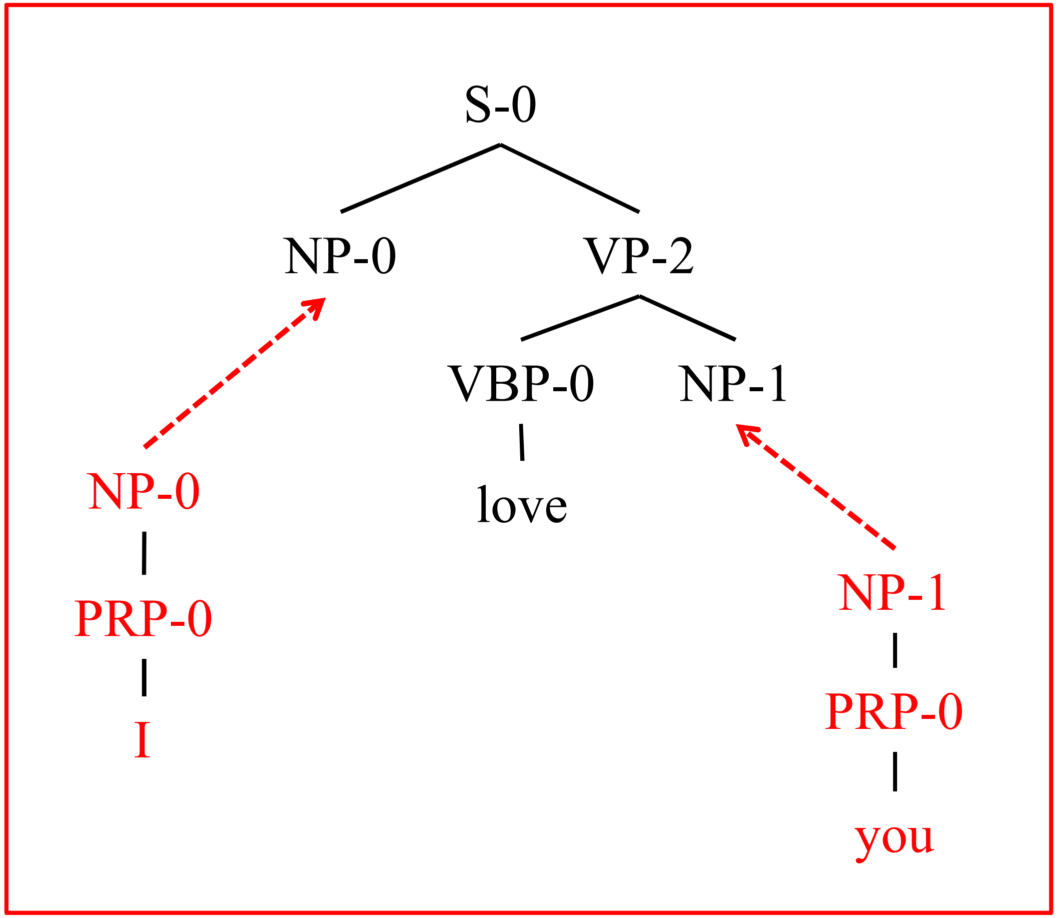
Overview of MCMC Sampling



Overview of MCMC Sampling



Overview of MCMC Sampling

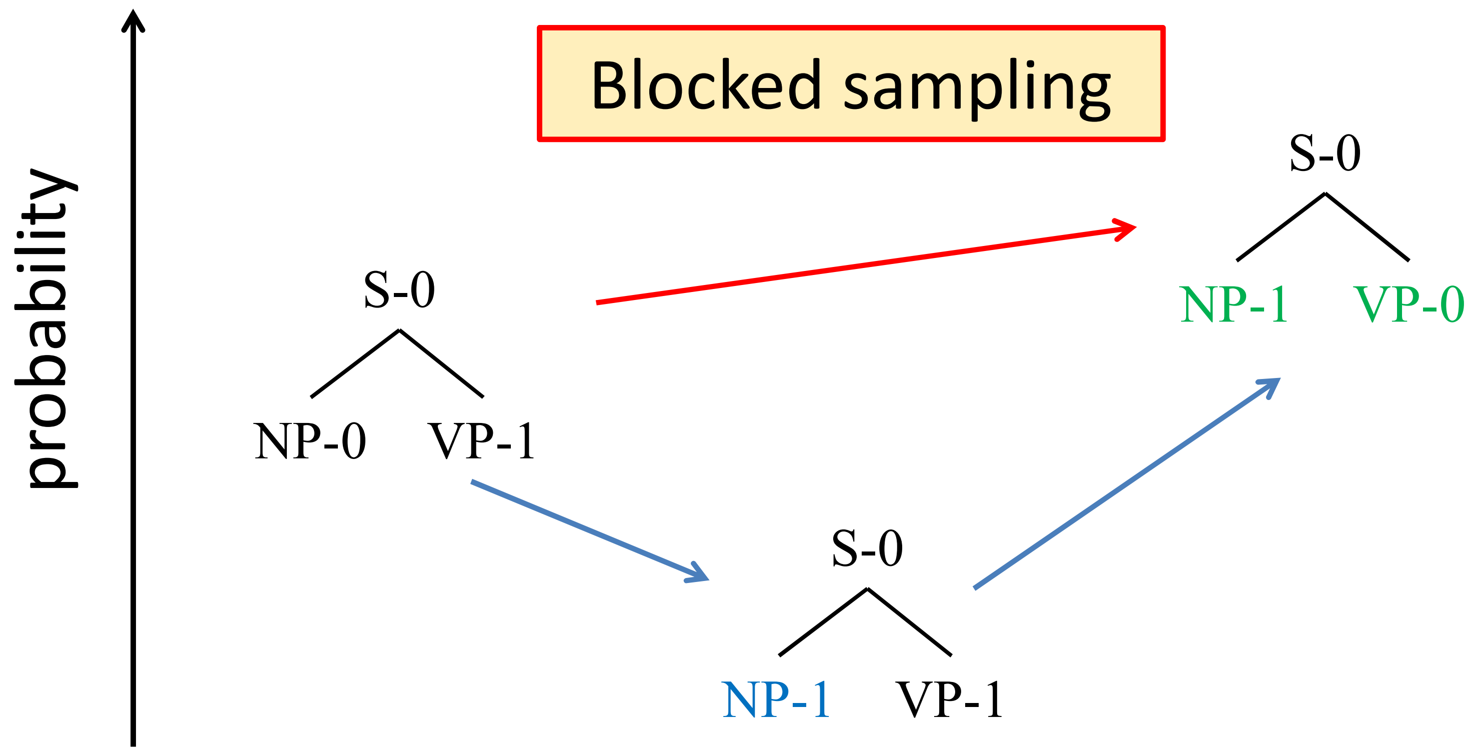


Inference of Symbol Refinement

Problem:

- Gibbs sampler is inefficient

  Update only one variable at a time



Inference of Symbol Refinement

Proposal:

- 3 types of blocked samplers to find better solution

For each MCMC iteration...

a) Run sentence sampler

b) Run subtree sampler

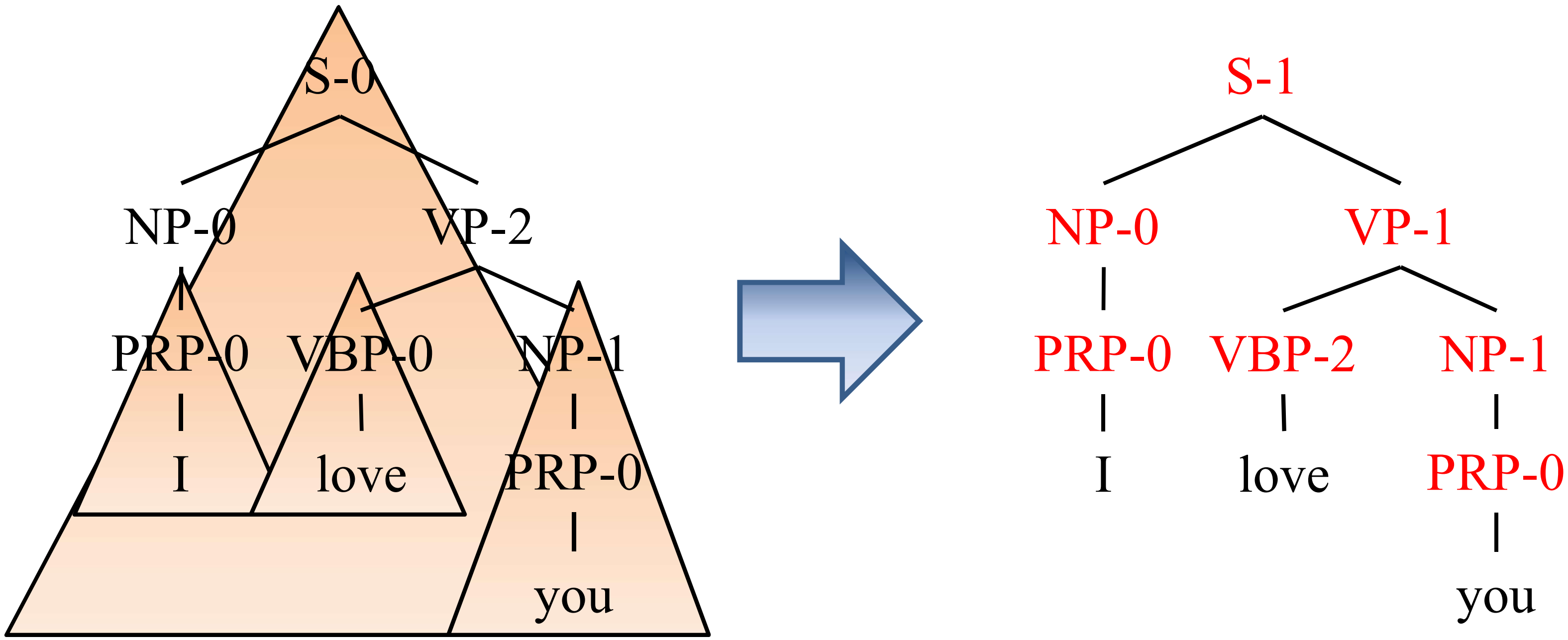
c) Run restaurant sampler

a) Sentence Sampler

- Metropolis-Hastings (MH) algorithm [Johnson et al. 07]

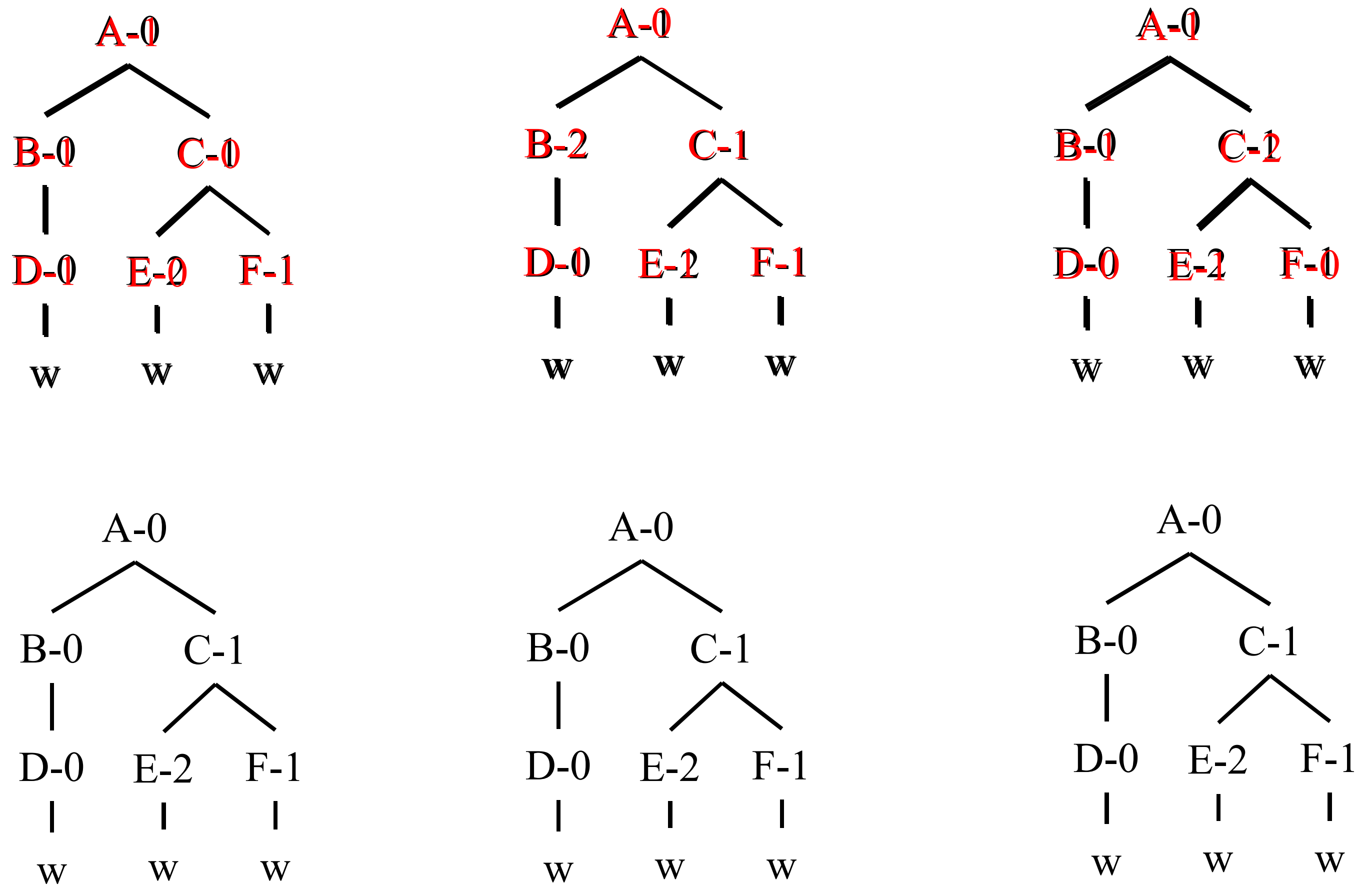
For each sentence...

1. Run inside algorithm
2. Sample a derivation
3. Accept or reject the sample



a) Sentence Sampler

- Update a sentence at a time

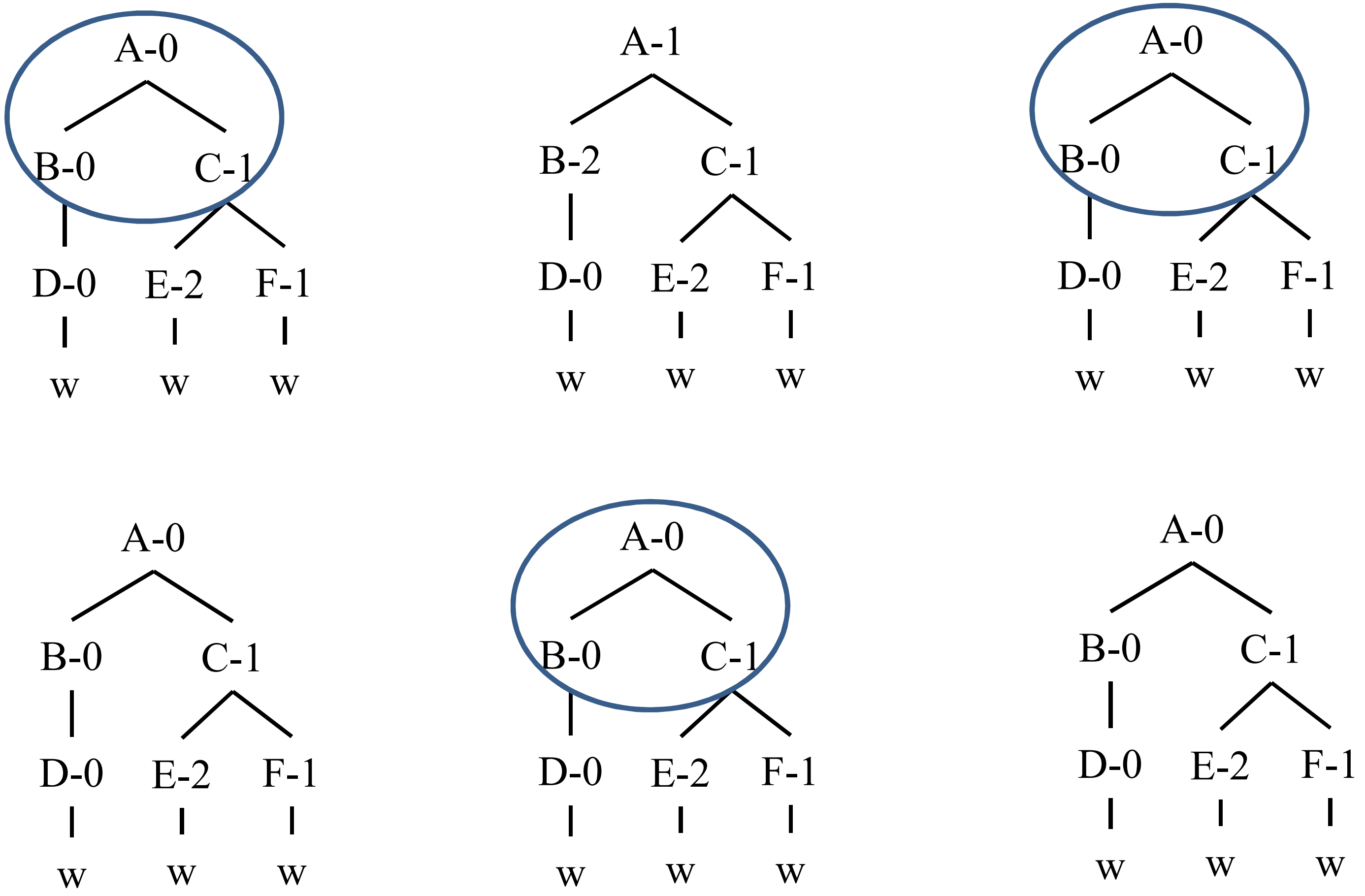


b) Subtree Sampler

- Update the same type of subtree simultaneously

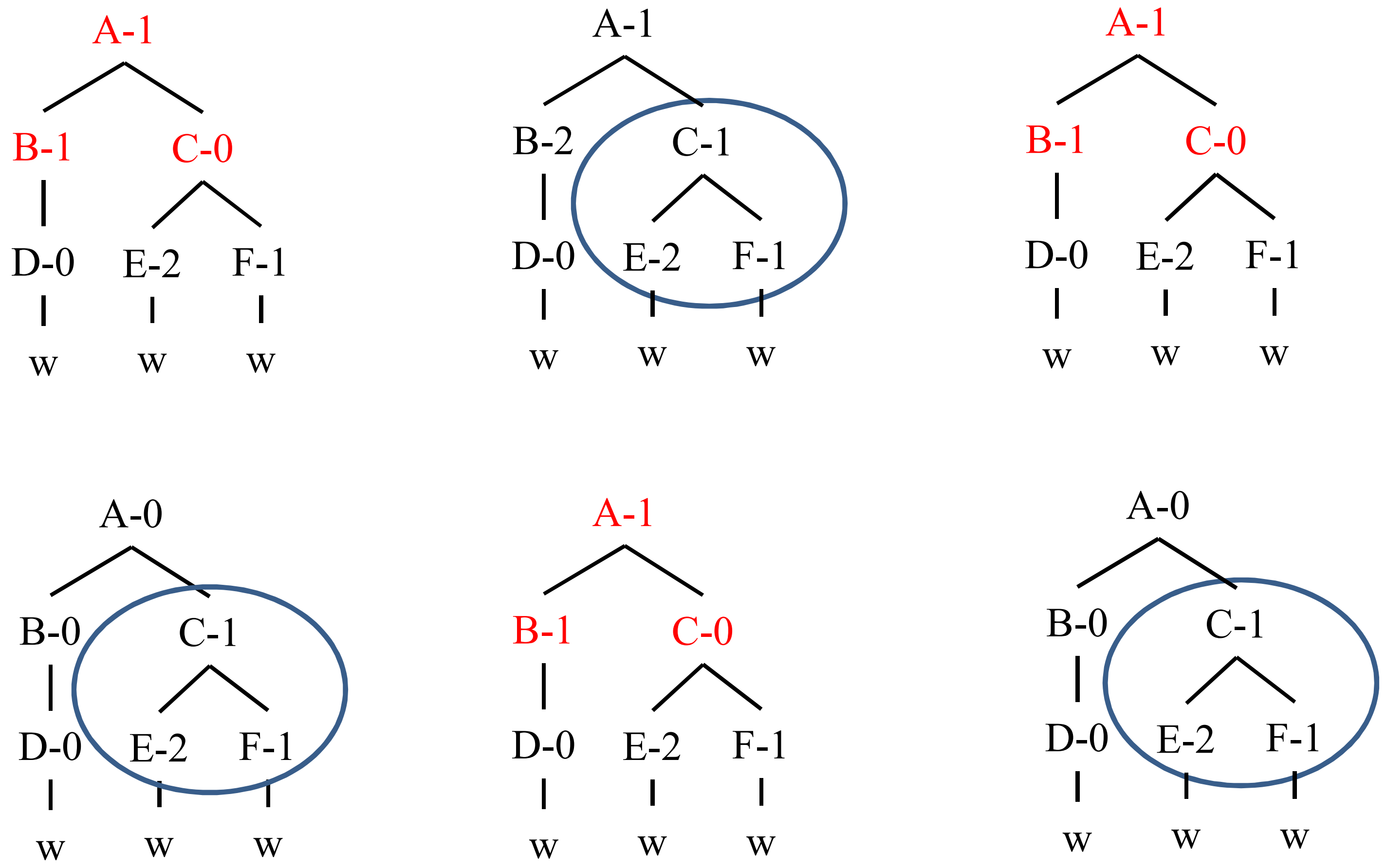
b) Subtree Sampler

- Update the same type of subtree simultaneously



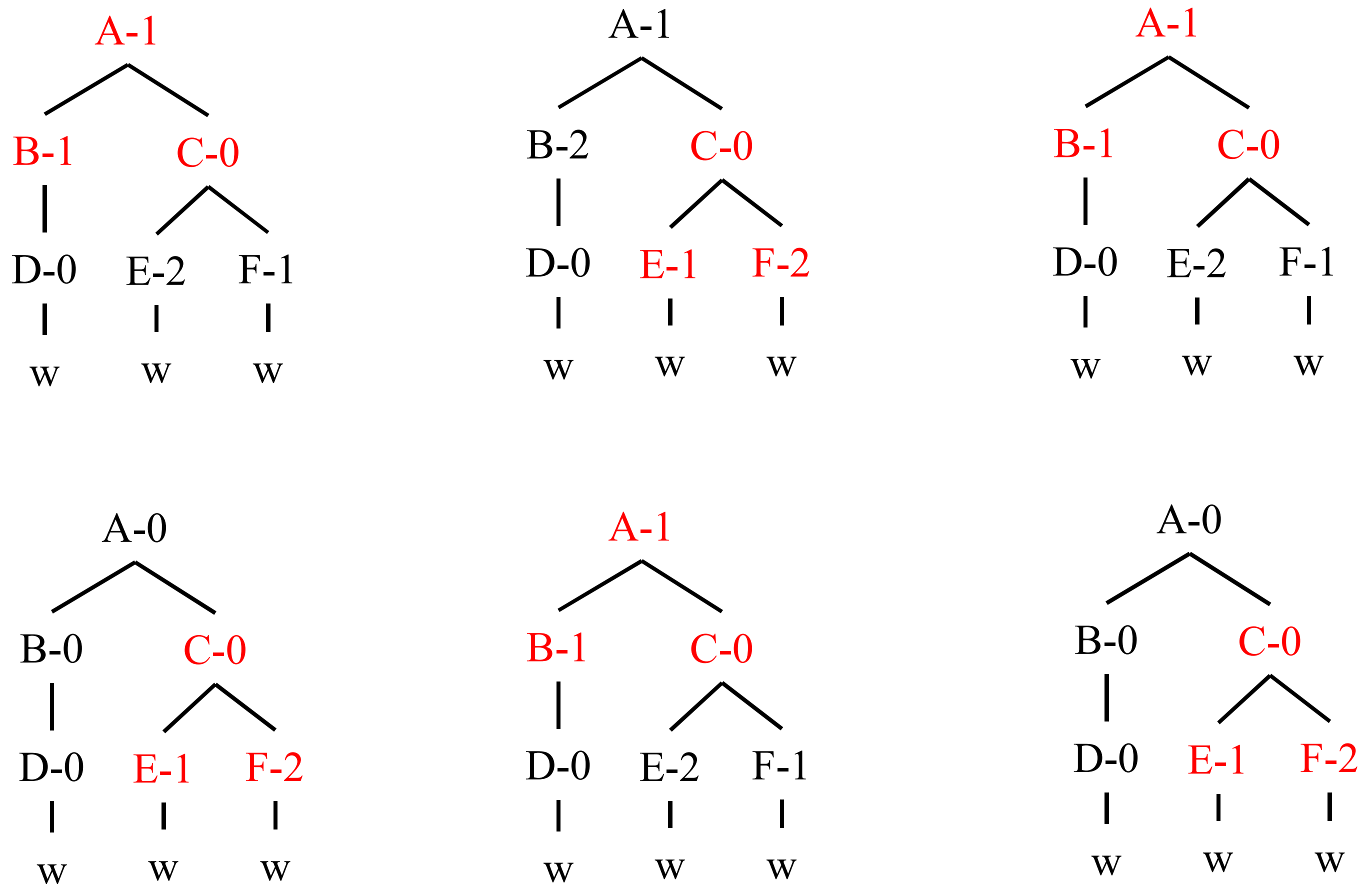
b) Subtree Sampler

- Update the same type of subtree simultaneously



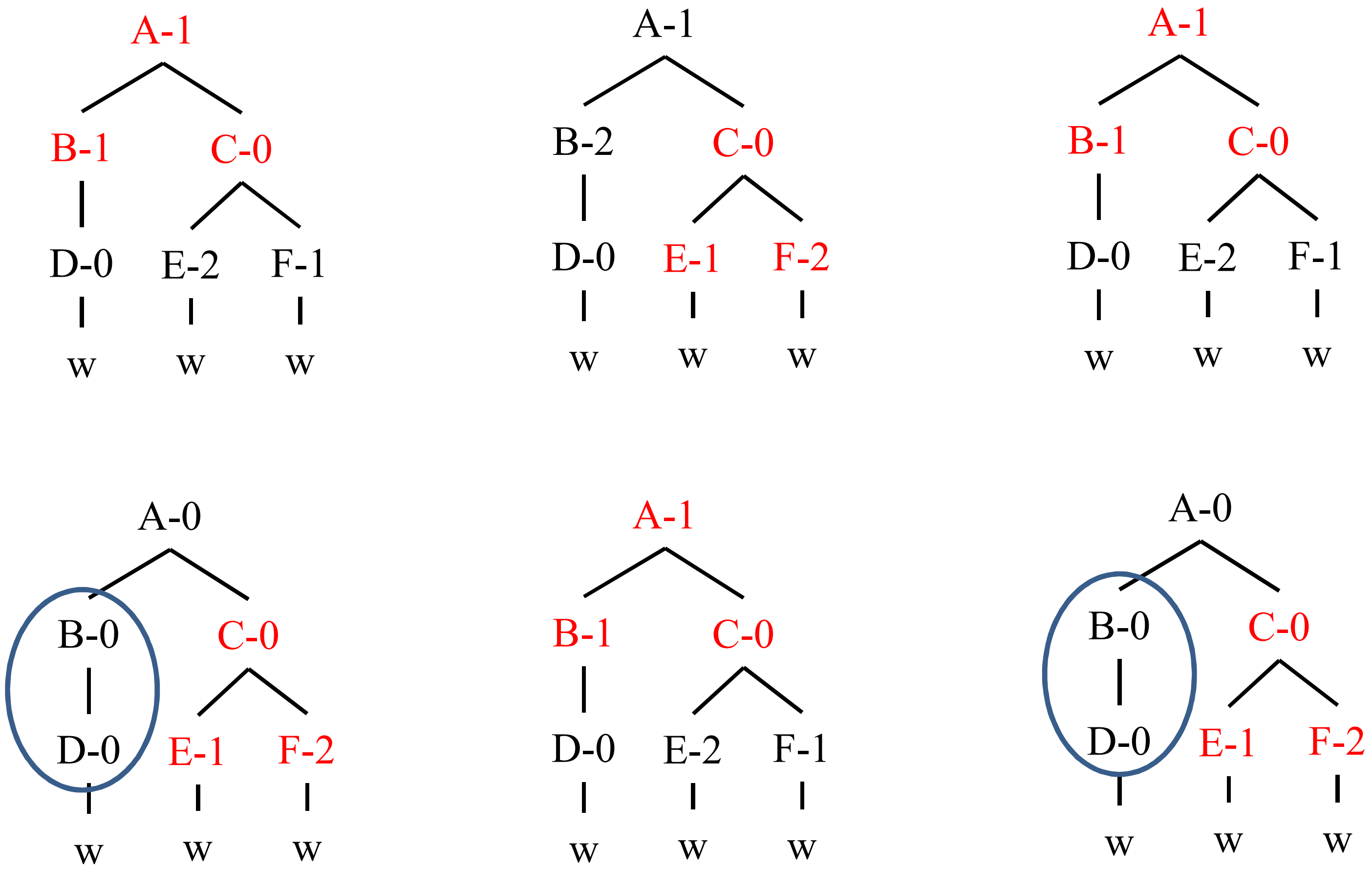
b) Subtree Sampler

- Update the same type of subtree simultaneously



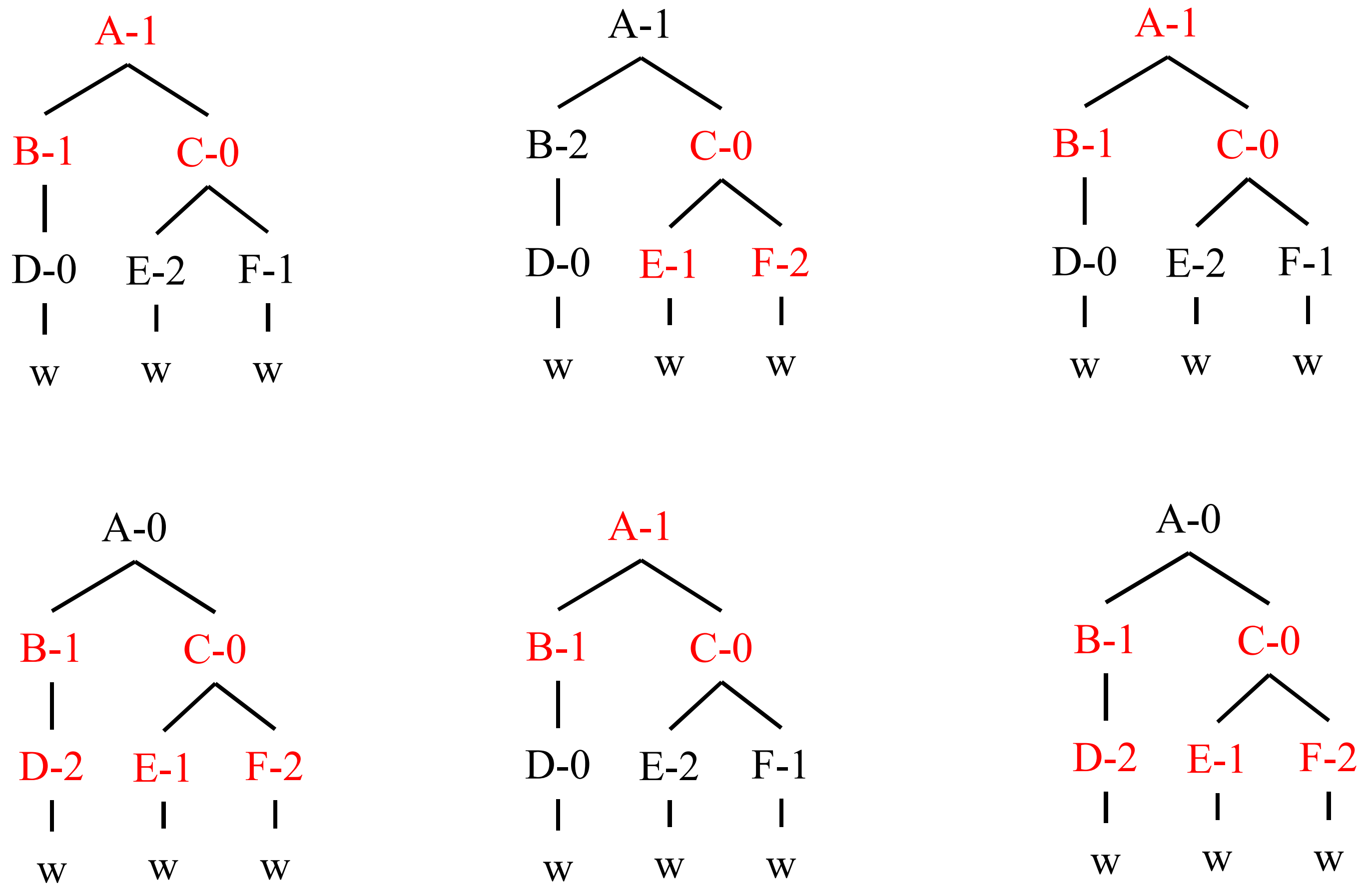
b) Subtree Sampler

- Update the same type of subtree simultaneously



b) Subtree Sampler

- Update the same type of subtree simultaneously

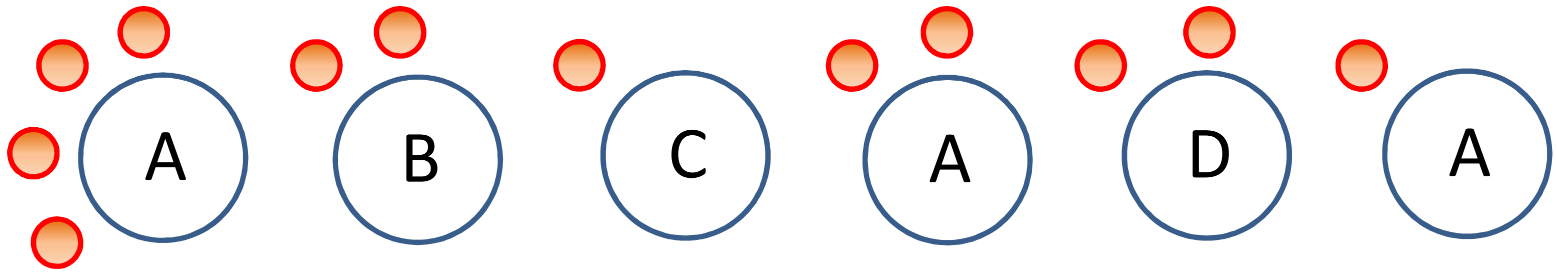


c) Restaurant Sampler

- Update # of tables

c) Restaurant Sampler

- Update # of tables



For each table label ...

- Keep count

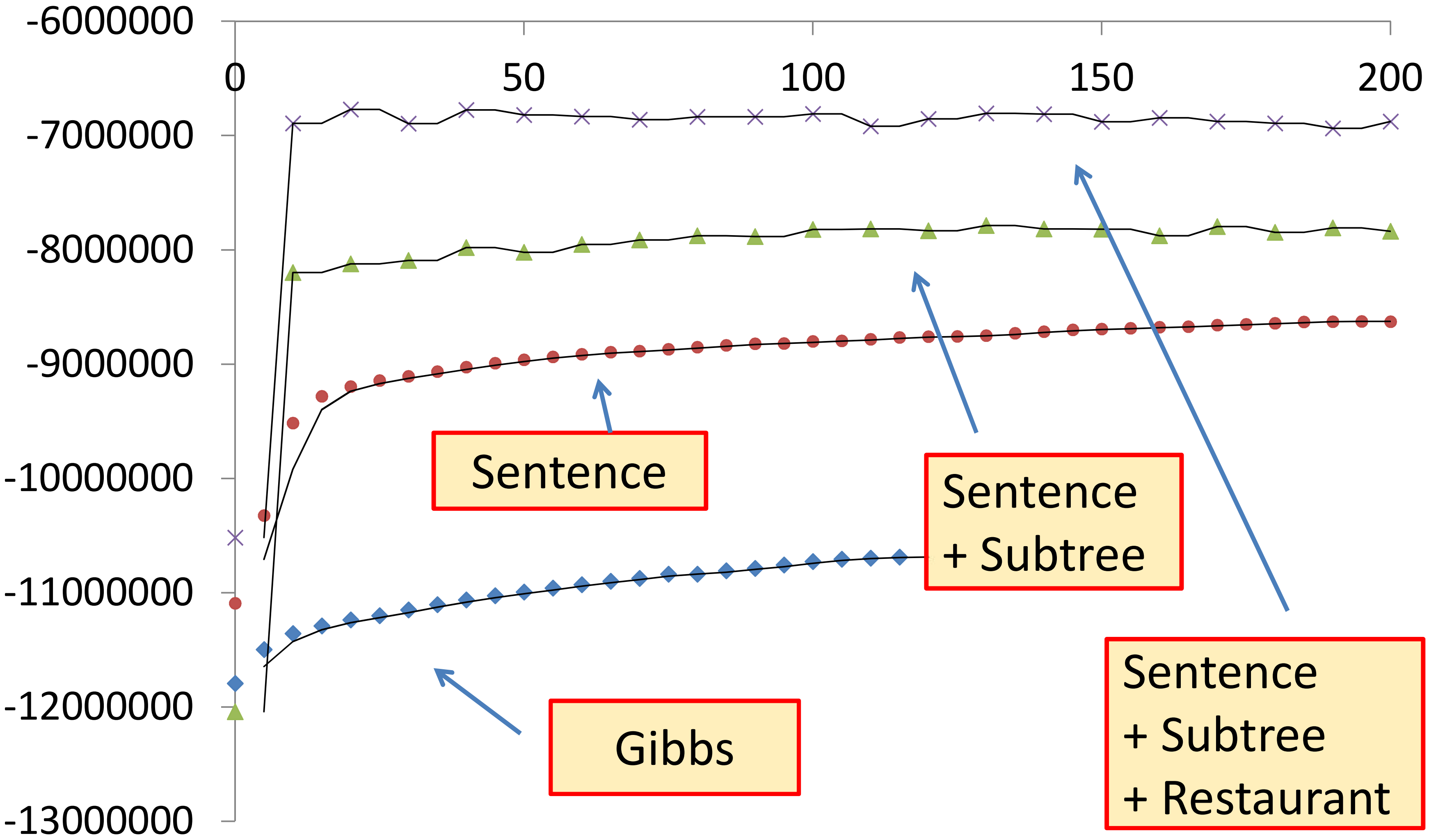
count = 6

- Remove all

- Add one by one

Helpful for optimizing # of tables

Effect of MCMC Samplers



Summary of Inference

- MCMC sampling

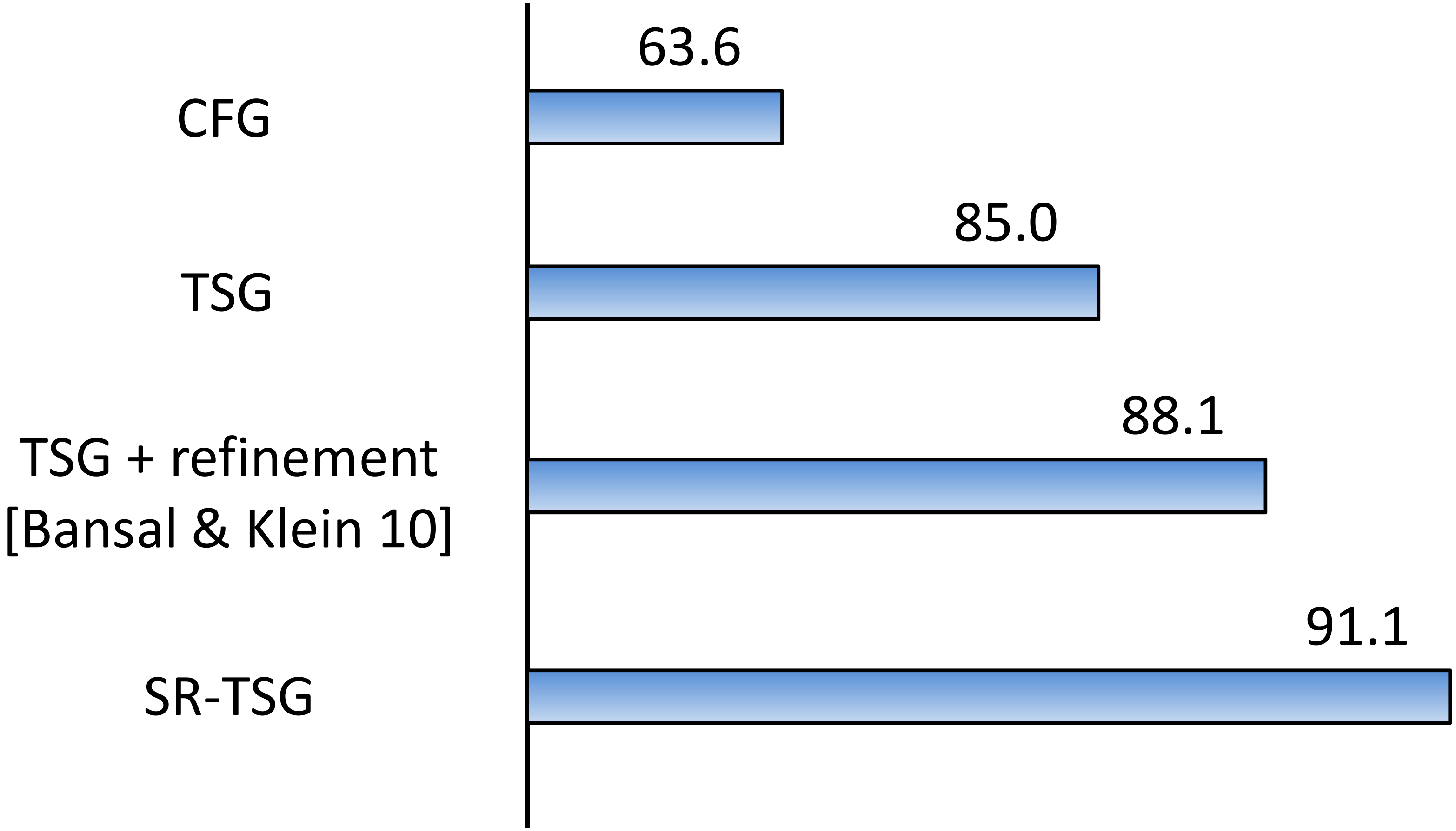
- Stepwise training for the induction of latent variables

- 3 types of blocked samplers

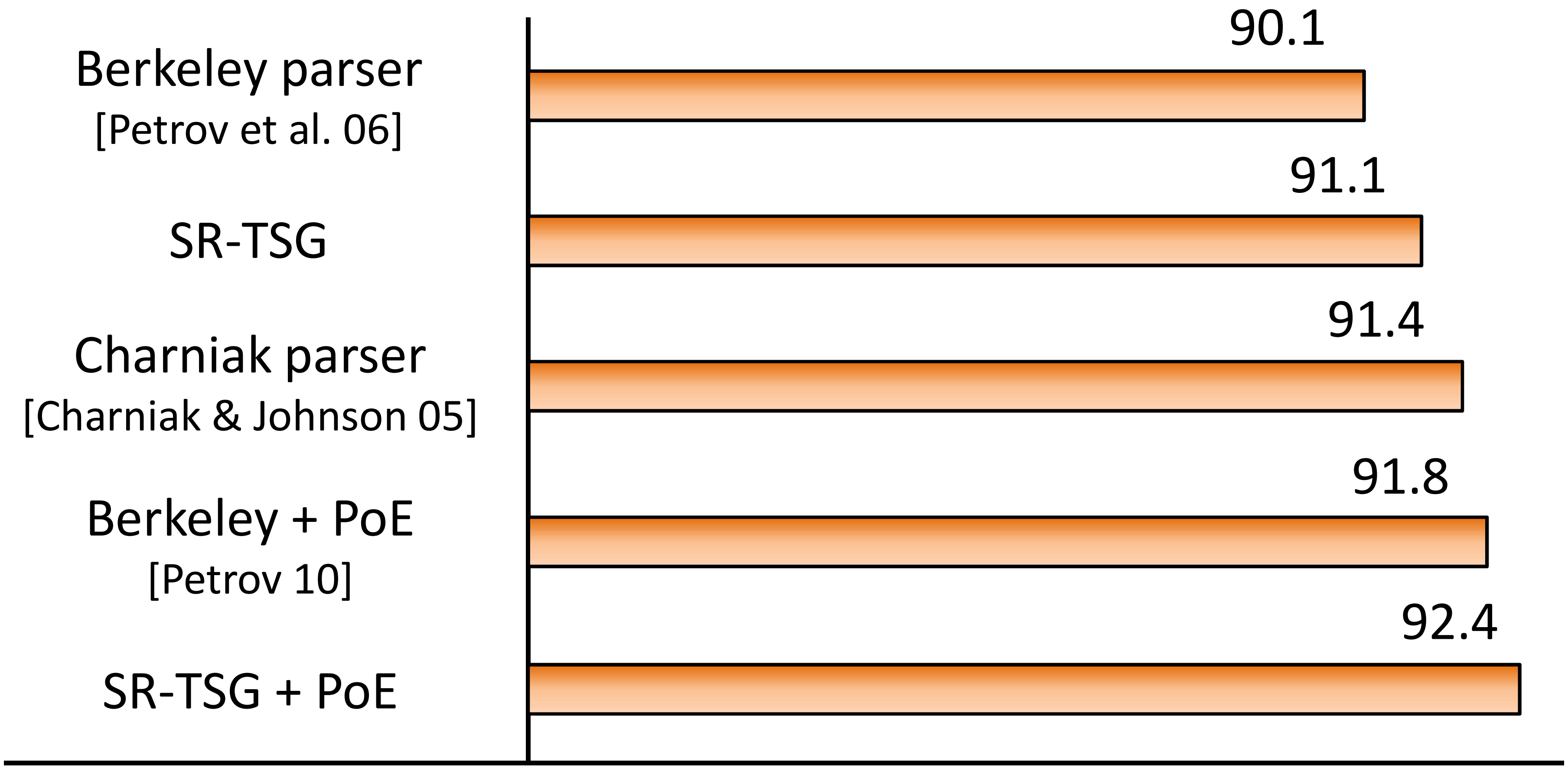
    - Effective for finding better solution

Experiment

Parsing Accuracy



Final Result



- System combination [Zhang et al. 09]
- Self-training [Huang et al. 10]

are better than SR-TSG.

Conclusion

Approach:

SR-TSG = TSG + symbol refinement

- Fully automatic learning

Probabilistic Model:

Pitman-Yor process + 3-level hierarchy

Inference:

3 types of MCMC samplers for efficient training

Result:

State-of-the-art

Thank you.

Internal Subcategory Marginalization

- Encourage the model to find large tree fragments