Bayesian Tools for Natural Language Learning

Yee Whye Teh
Gatsby Computational Neuroscience Unit
UCL

Bayesian Learning of Probabilistic Models

- Potential outcomes/observations X.
- Unobserved latent variables Y.
- Joint distribution over X and Y:

$$P(x \in X, y \in Y|\theta)$$

- Parameters of the model θ.
- Inference:

$$P(y \in Y|x \in X, \theta) = \frac{P(y, x|\theta)}{P(x|\theta)}$$

- Learning:

$$P(\text{training data}|\theta)$$

- Bayesian learning:

$$P(\theta|\text{training data}) = \frac{P(\text{training data}|\theta)P(\theta)}{Z}$$

Why Bayesian Learning?

- Less worry about overfitting.
- Nonparametric Bayes mitigates issues of model selection.

- Separation of modeling assumptions and algorithmic concerns.
- Explicit statement of assumptions made.

- *Allows inclusion of domain knowledge into model via the structure and form of Bayesian priors.*
  - *Power law properties via Pitman-Yor processes.*
  - *Information sharing via hierarchical Bayes.*

Hierarchical Bayes, Pitman-Yor Processes, and N-gram Language Models

N-gram Language Models

- Probabilistic models for sequences of words and characters, e.g.

south, parks, road

s, o, u, t, h, _, p, a, r, k, s, _, r, o, a, d

- (N-1)th order Markov model:

$$P(\text{sentence}) = \prod_i P(\text{word}_i | \text{word}_{i-N+1} \dots \text{word}_{i-1})$$

Sparsity and Smoothing

$$P(\text{sentence}) = \prod_i P(\text{word}_i | \text{word}_{i-N+1} \ldots \text{word}_{i-1})$$

- Large vocabulary size means naïvely estimating parameters of this model from data counts is problematic for N>2.

$$P^{\text{ML}}(\text{word}_i | \text{word}_{i-N+1} \ldots \text{word}_{i-1}) = \frac{C(\text{word}_{i-N+1} \ldots \text{word}_i)}{C(\text{word}_{i-N+1} \ldots \text{word}_{i-1})}$$

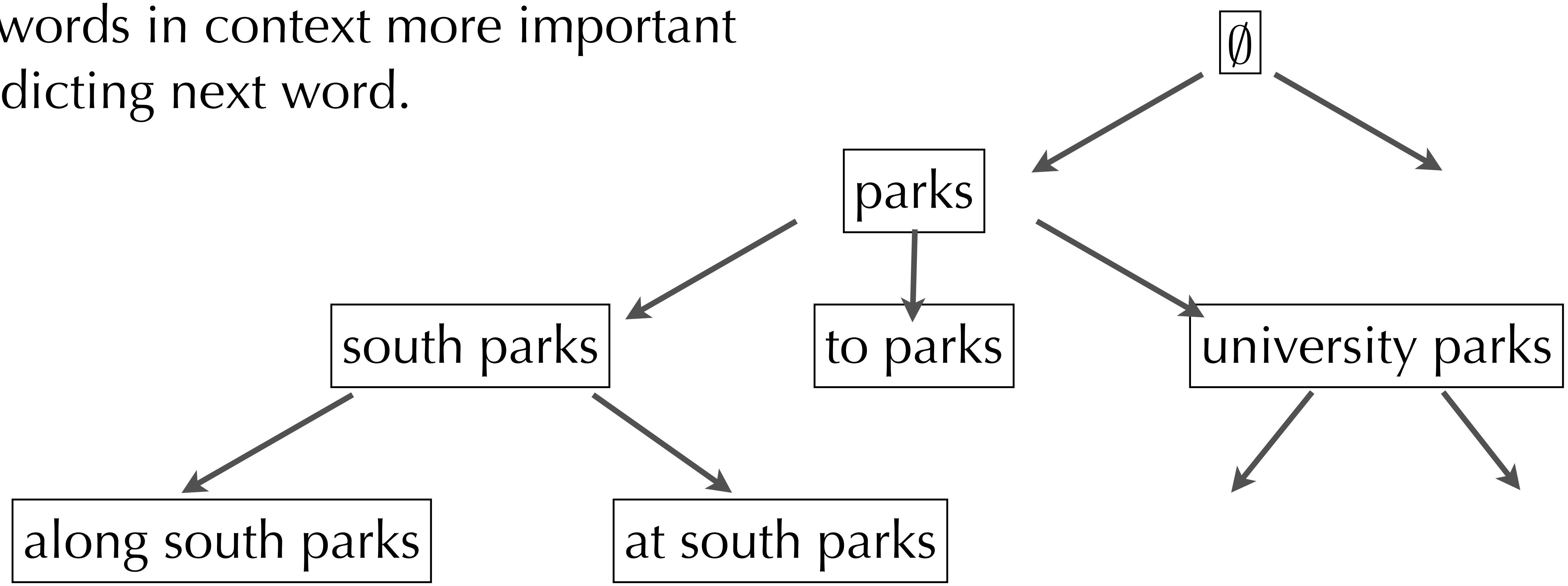
- Naïve priors/regularization fail as well: most parameters have *no* associated data.
  - Smoothing.

Smoothing on Context Tree

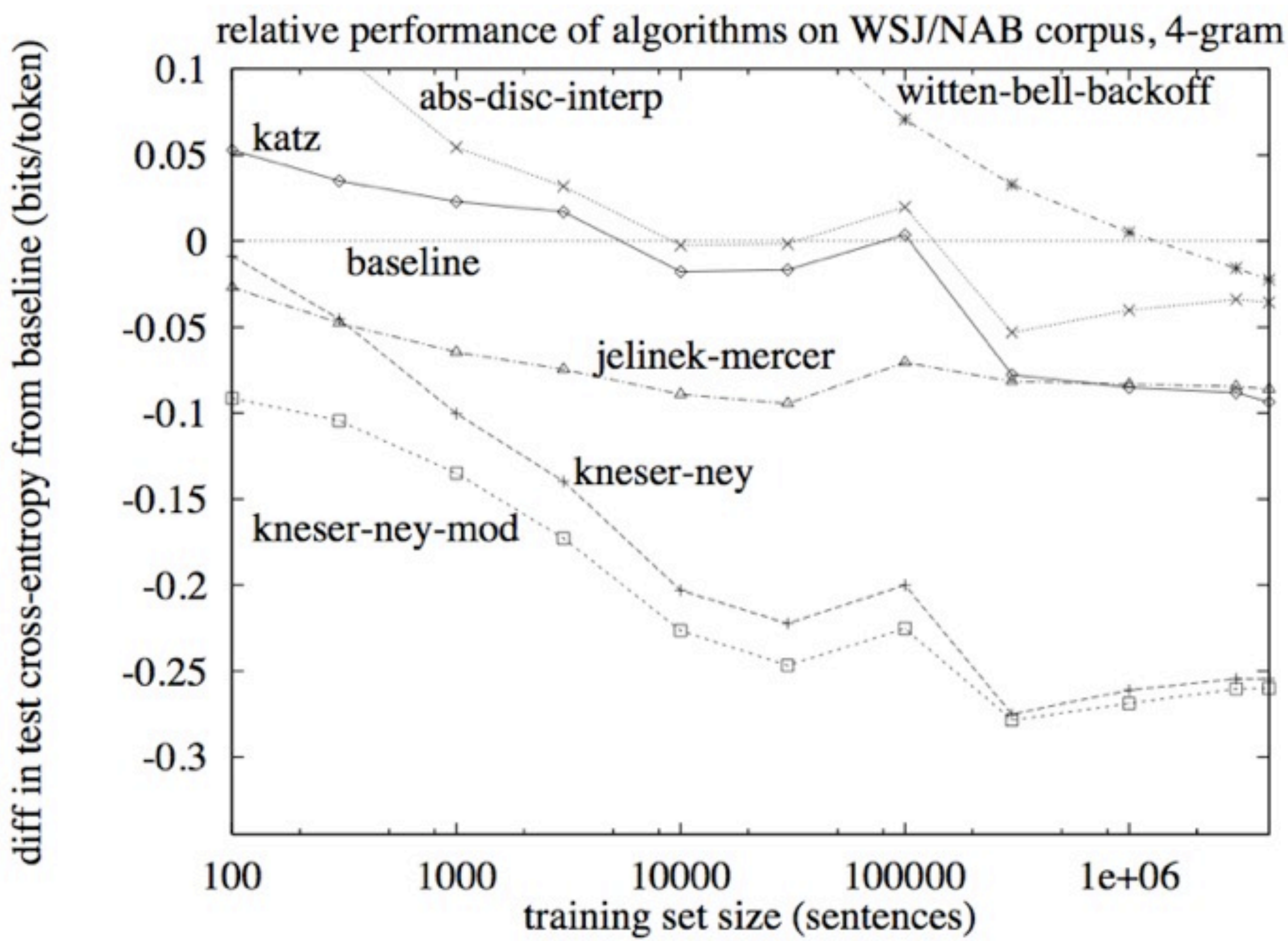
- *Context* of conditional probabilities naturally organized using a tree.
- Smoothing makes conditional probabilities of neighbouring contexts more similar.

$$
\begin{aligned}
& P^{\text{smooth}}(\text{road}|\text{south parks}) \\
= \; & \lambda(3)Q_3(\text{road}|\text{south parks}) + \\
& \lambda(2)Q_2(\text{road}|\text{parks}) + \\
& \lambda(1)Q_1(\text{road}|\emptyset)
\end{aligned}
$$

- Later words in context more important in predicting next word.



Smoothing in Language Models



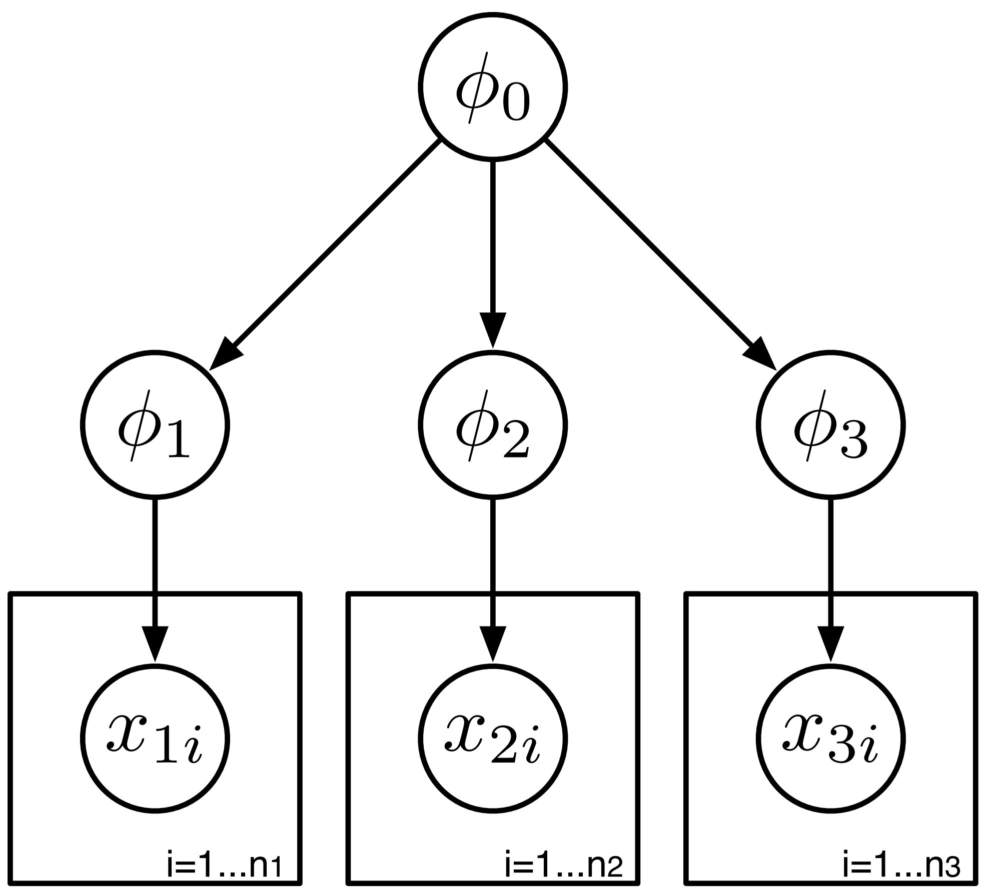
- Interpolated and modified Kneser-Ney are best under virtually all circumstances.

[Chen and Goodman 1998]

Hierarchical Bayesian Models

- Hierarchical modelling an important overarching theme in modern statistics.

- In machine learning, have been used for multitask learning, transfer learning, learning-to-learn and domain adaptation.



[Gelman et al, 1995, James & Stein 1961]

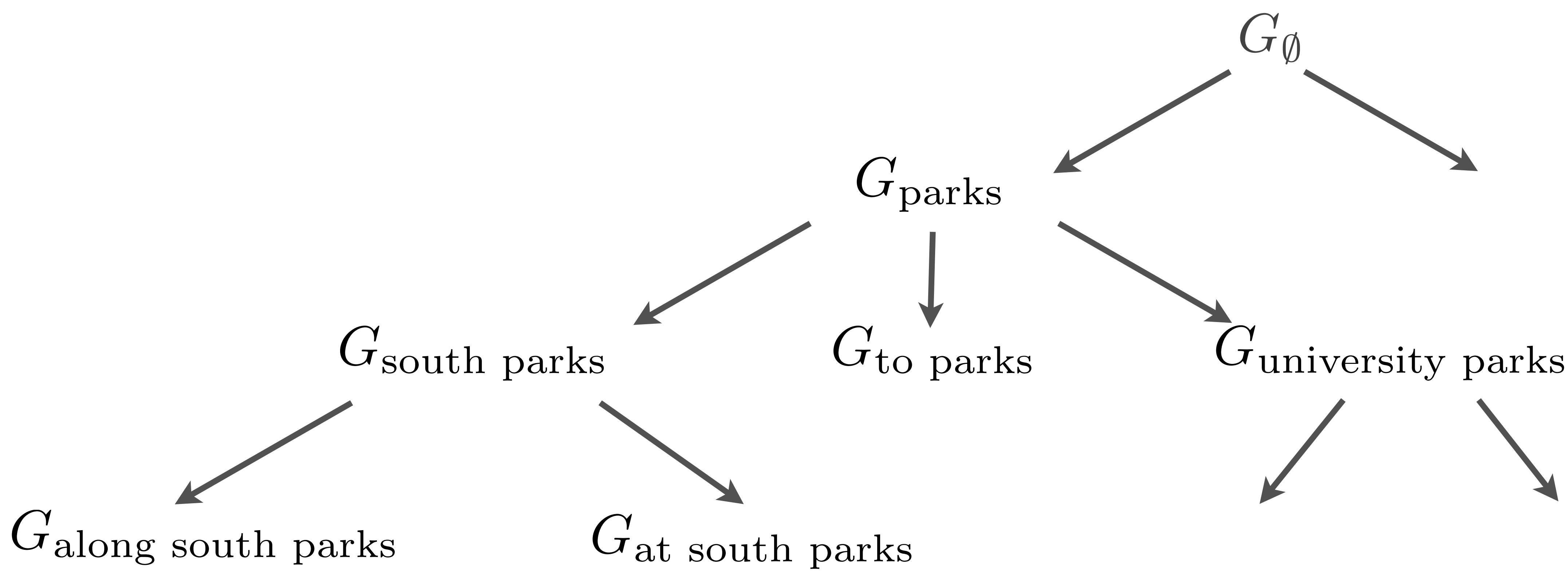
Hierarchical Bayesian Models on Context Tree

- Parametrize the conditional probabilities of Markov model:

$$P(\text{word}_i = w | \text{word}_{i-N+1}^{i-1} = u) = G_u(w)$$

$$G_u = [G_u(w)]_{w \in \text{vocabulary}}$$

- G_u is a probability vector associated with context u.



[MacKay and Peto 1994]

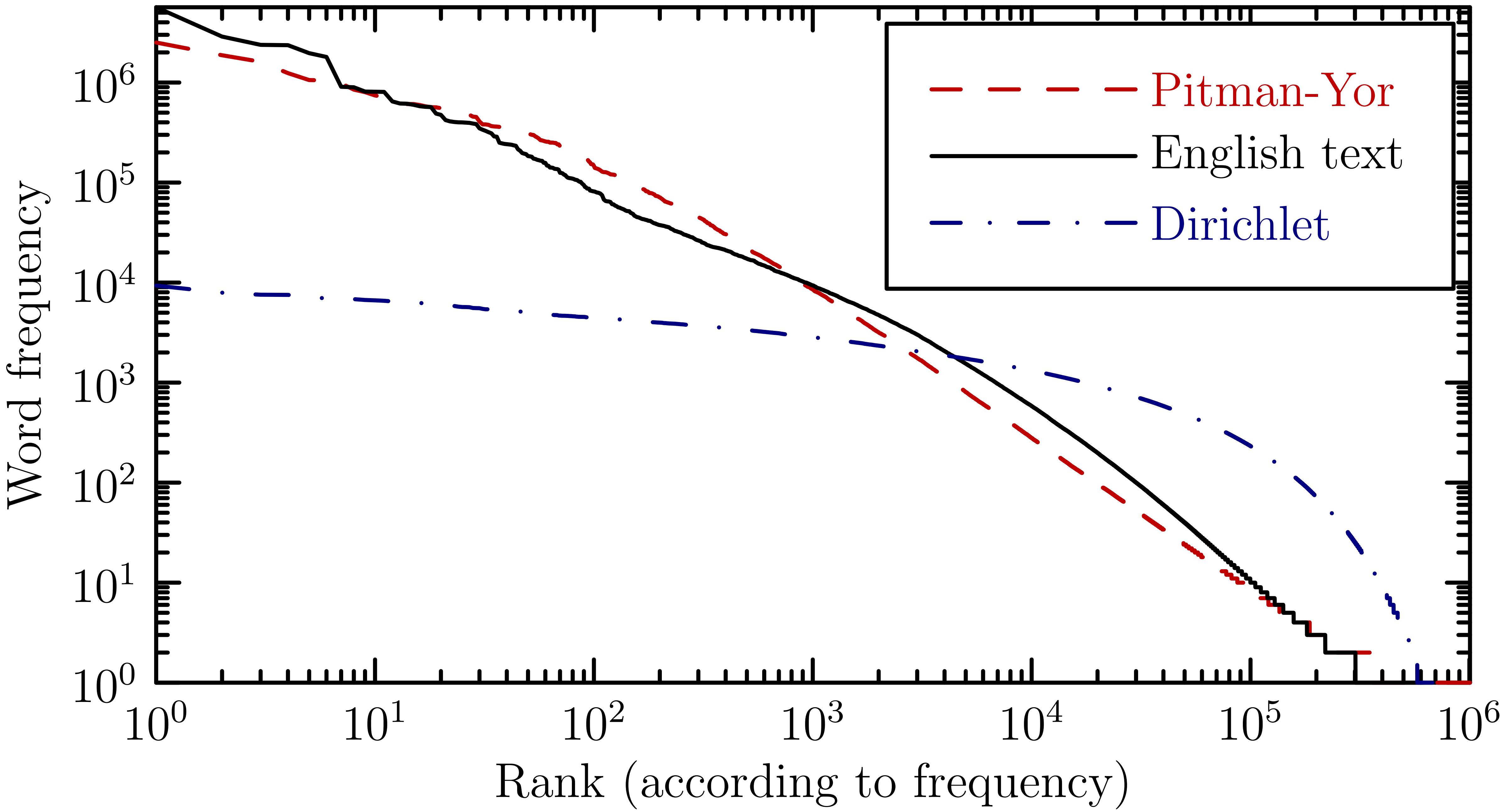
Hierarchical Dirichlet Language Models

- What is $P(G_u | G_{\mathrm{pa}(u)})$? [MacKay and Peto 1994] proposed using the standard Dirichlet distribution over probability vectors.

T	N-1	IKN	MKN	HDLM
2×10^6	2	148.8	144.1	191.2
4×10^6	2	137.1	132.7	172.7
6×10^6	2	130.6	126.7	162.3
8×10^6	2	125.9	122.3	154.7
10×10^6	2	122.0	118.6	148.7
12×10^6	2	119.0	115.8	144.0
14×10^6	2	116.7	113.6	140.5
14×10^6	1	169.9	169.2	180.6
14×10^6	3	106.1	102.4	136.6

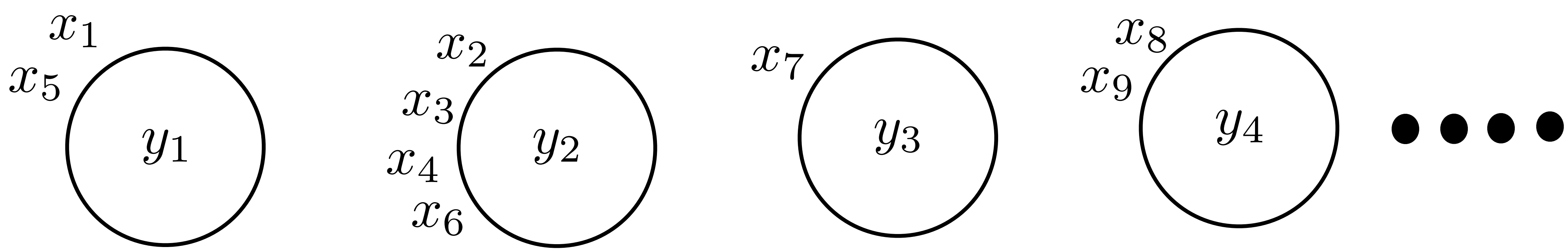
- We will use Pitman-Yor processes instead.

Power Laws in English



Chinese Restaurant Processes

- Generative Process:



$$p(\text{sit at table } k) = \frac{c_k - d}{\theta + \sum_{j=1}^{K} c_j}$$

$$p(\text{sit at new table}) = \frac{\theta + dK}{\theta + \sum_{j=1}^{K} c_j} \qquad p(\text{table serves dish } y) = H(y)$$

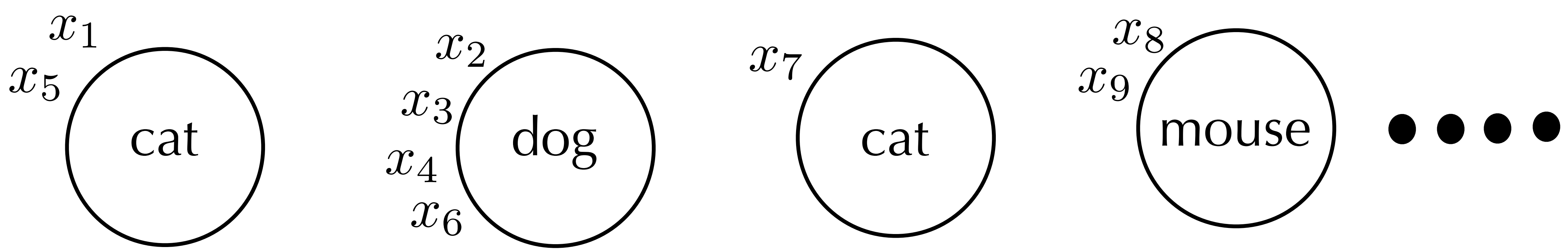
- Defines an exchangeable stochastic process over sequences x_1, x_2, ...
- The de Finetti measure is the Pitman-Yor process,

$$\begin{aligned} G &\sim \mathrm{PY}(\theta, d, H) \\ x_i &\sim G \qquad i = 1, 2, \ldots \end{aligned}$$

[Perman, Pitman & Yor 1992, Pitman & Yor 1997, Ishwaran & James 2001]

Chinese Restaurant Processes

- customers = word tokens.
- H = dictionary.
- tables = dictionary lookup.



- Dictionary look-up sequence:

cat, dog, cat, mouse

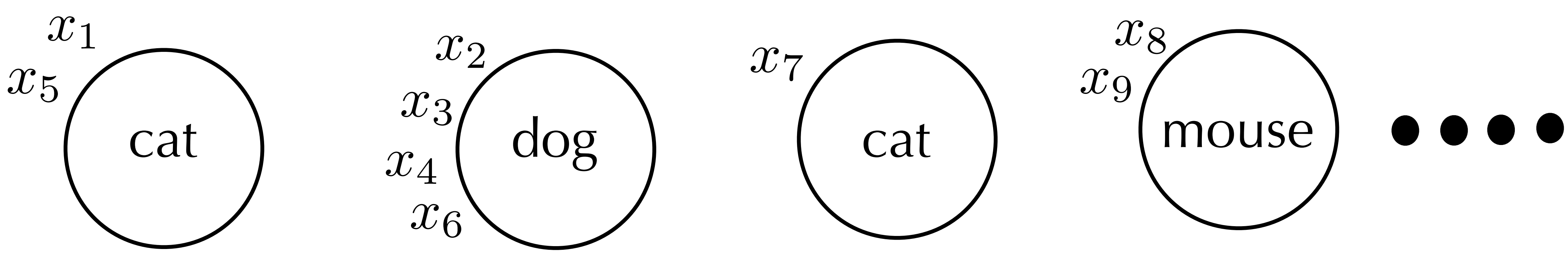
- Word token sequence:

cat, dog, dog, dog, cat, dog, cat, mouse, mouse

Stochastic Programming Perspective

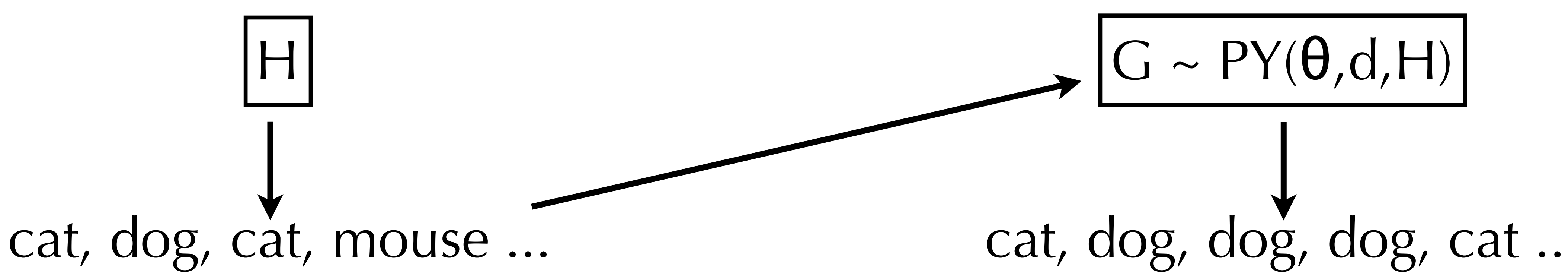
- G ~ PY(θ,d,H)

cat, dog, cat, mouse ... ~ H iid



cat dog dog dog cat dog cat mouse mouse ~ G iid

- A stochastic program producing a random sequence of words.



[Goodman et al 2008]

Power Law Properties of Pitman-Yor Processes

- Chinese restaurant process:

$$\begin{aligned} p(\text{sit at table } k) &\propto c_k - d \\ p(\text{sit at new table}) &\propto \theta + dK \end{aligned}$$

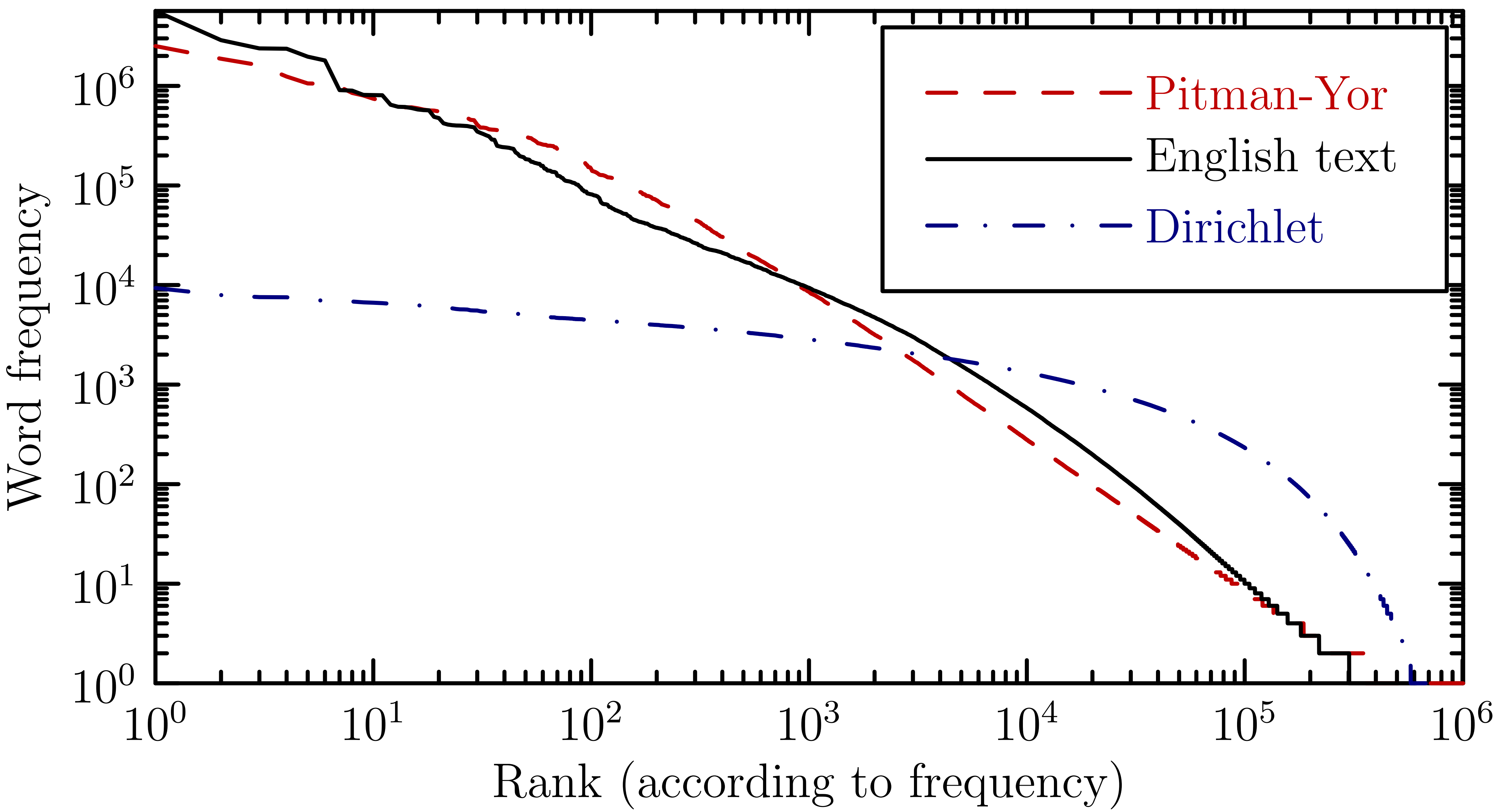
- Pitman-Yor processes produce distributions over words given by a power law distribution with index 1+d.
  - Small number of common word types;
  - Large number of rare word types.

- This is more suitable for languages than Dirichlet distributions.

- [Goldwater et al 2006] investigated the Pitman-Yor process from this perspective.

[Goldwater et al 2006]

Power Law Properties of Pitman-Yor Processes



Hierarchical Pitman-Yor Language Models

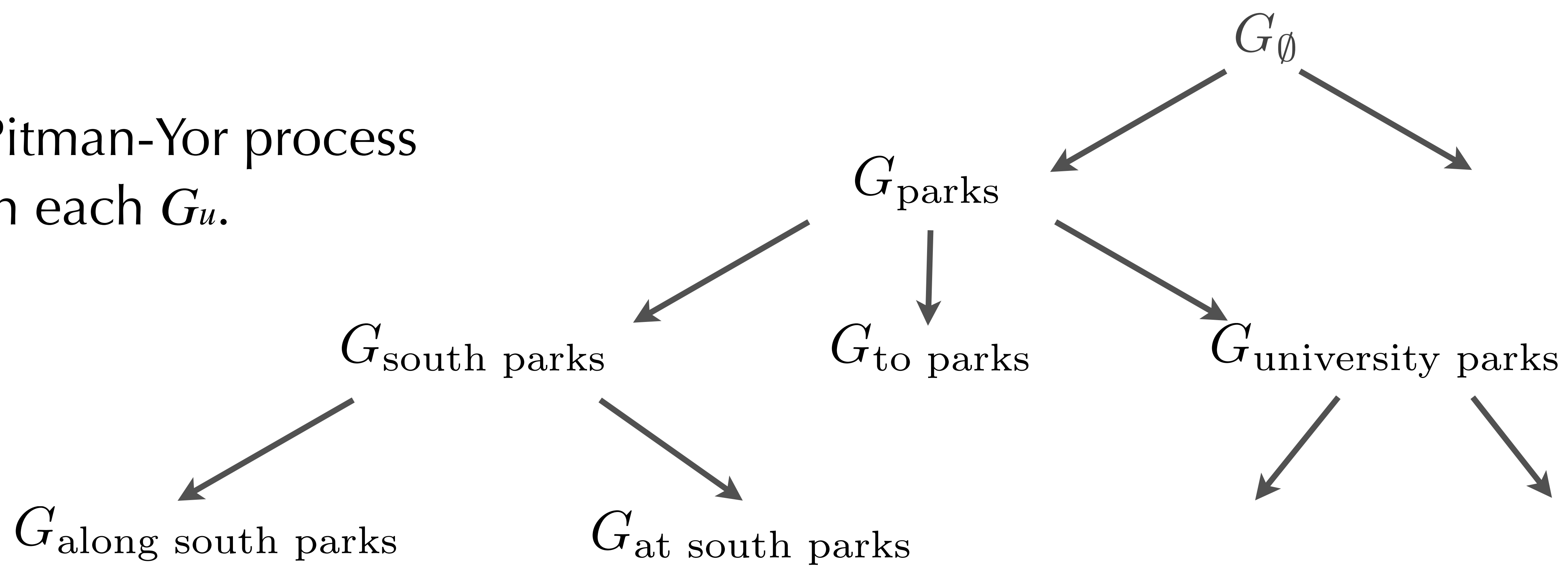
- Parametrize the conditional probabilities of Markov model:

$$P(\text{word}_i = w | \text{word}_{i-N+1}^{i-1} = u) = G_u(w)$$

$$G_u = [G_u(w)]_{w \in \text{vocabulary}}$$

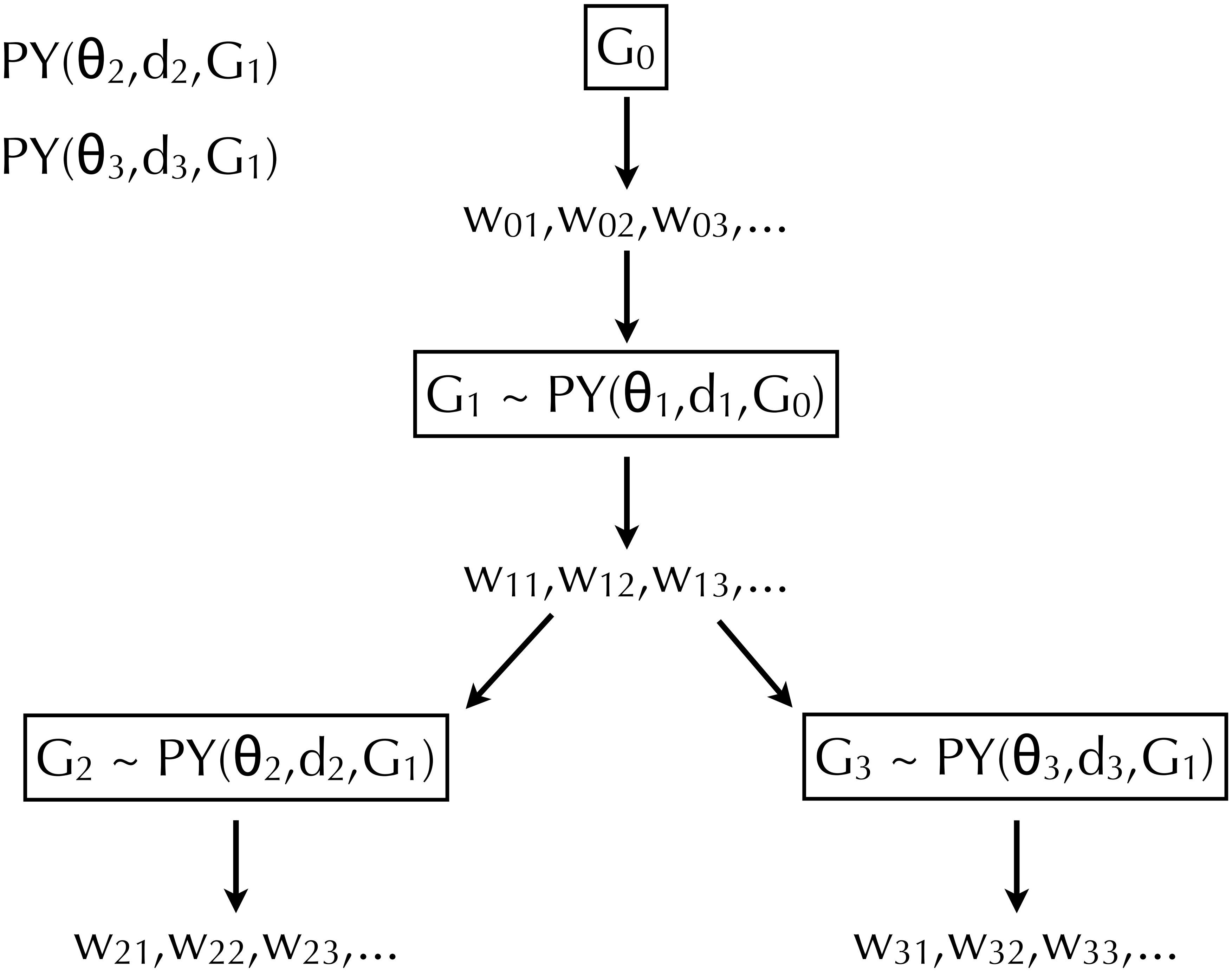
- G_u is a probability vector associated with context u.

- Place Pitman-Yor process prior on each G_u.



Stochastic Programming Perspective

- $G_1 \sim PY(\theta_1, d_1, G_0)$
- $G_2 \mid G_1 \sim PY(\theta_2, d_2, G_1)$
- $G_3 \mid G_1 \sim PY(\theta_3, d_3, G_1)$



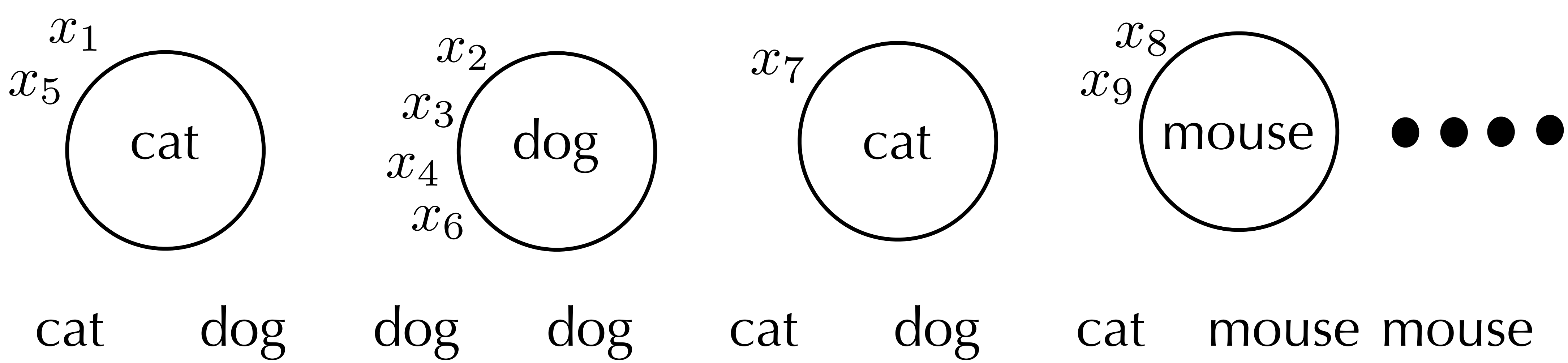
Hierarchical Pitman-Yor Language Models

- Significantly improved on the hierarchical Dirichlet language model.
- Results better Kneser-Ney smoothing, state-of-the-art language models.

T	N-1	IKN	MKN	HDLM	HPYLM
2×10^6	2	148.8	**144.1**	191.2	144.3
4×10^6	2	137.1	**132.7**	172.7	**132.7**
6×10^6	2	130.6	126.7	162.3	**126.4**
8×10^6	2	125.9	122.3	154.7	**121.9**
10×10^6	2	122.0	118.6	148.7	**118.2**
12×10^6	2	119.0	115.8	144.0	**115.4**
14×10^6	2	116.7	113.6	140.5	**113.2**
14×10^6	1	169.9	**169.2**	180.6	169.3
14×10^6	3	106.1	102.4	136.6	**101.9**

Pitman-Yor and Kneser-Ney

- Interpolated Kneser-Ney can be derived as a particular approximate inference method in a hierarchical Pitman-Yor language model.



$$P(x_{10} = \text{dog}) = \frac{4-d}{\theta+9} + \frac{\theta+4d}{\theta+9} H(\text{dog})$$

absolute discounting

interpolation with H(dog)

- *Pitman-Yor processes can be used in place of Kneser-Ney.*

∞-gram Language Models and Computational Advantages

Markov Language Models

- Usually makes a Markov assumption to simplify model:

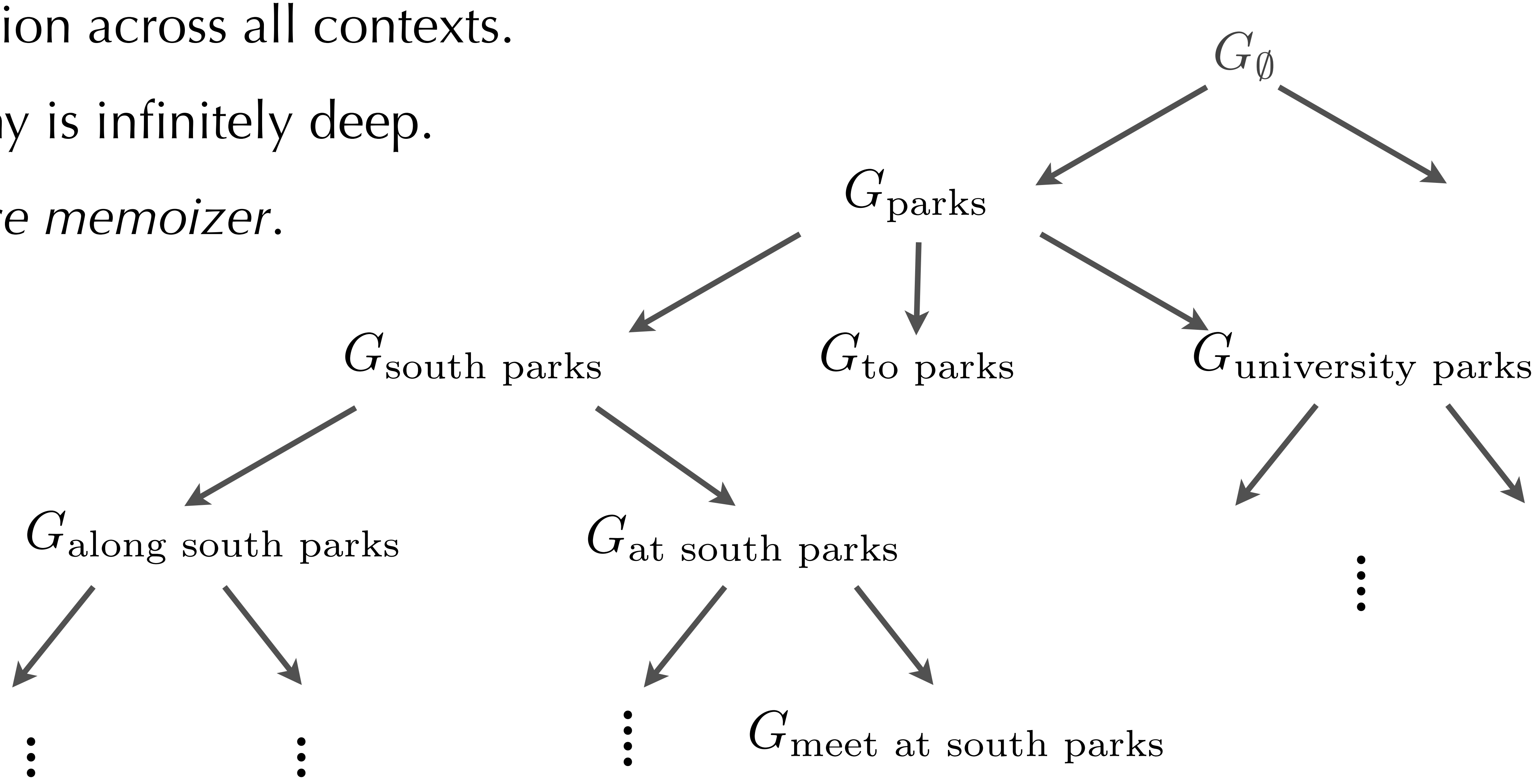
$$\begin{gathered} P(\text{south parks road}) \sim \\ P(\text{south})* \\ P(\text{parks} \mid \text{south})* \\ P(\text{road} \mid \text{parks}) \end{gathered}$$

- Language models: usually Markov models of order 2-4 (3-5-grams).
- How do we determine the order of our Markov models?
- Is the Markov assumption a reasonable assumption?
  - Be nonparametric about Markov order...

Non-Markov Language Models

- Model the conditional probabilities of each possible word occurring after each possible context (of unbounded length).

- Use hierarchical Pitman-Yor process prior to share information across all contexts.
- Hierarchy is infinitely deep.
- *Sequence memoizer*.

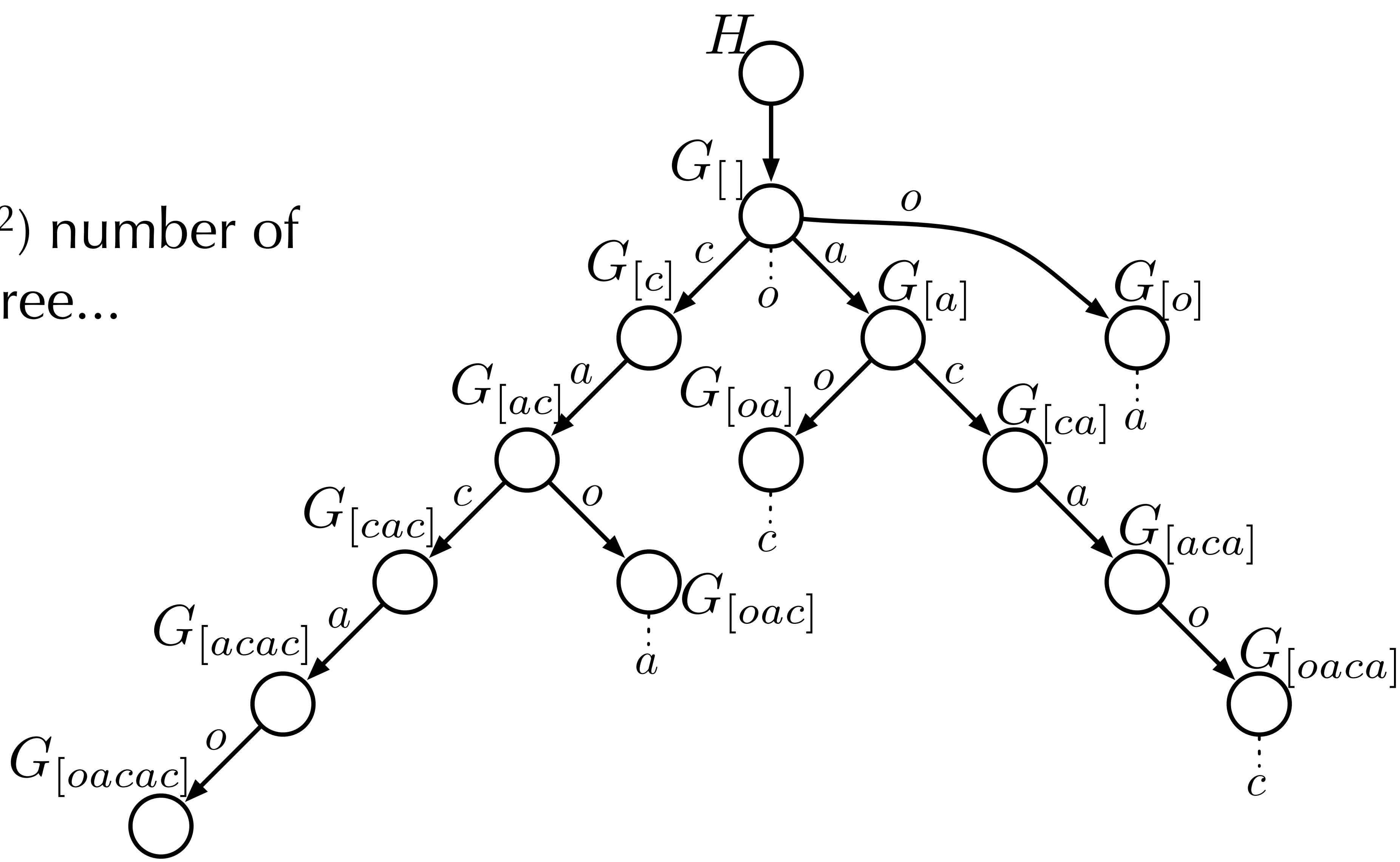


Model Size: Infinite -> O(T^2)

- The sequence memoizer model is very large (actually, infinite).

- Given a training sequence (e.g.: o,a,c,a,c), most of the model can be ignored (integrated out), leaving a finite number of nodes in context tree.

- But there are still O(T^2) number of nodes in the context tree...

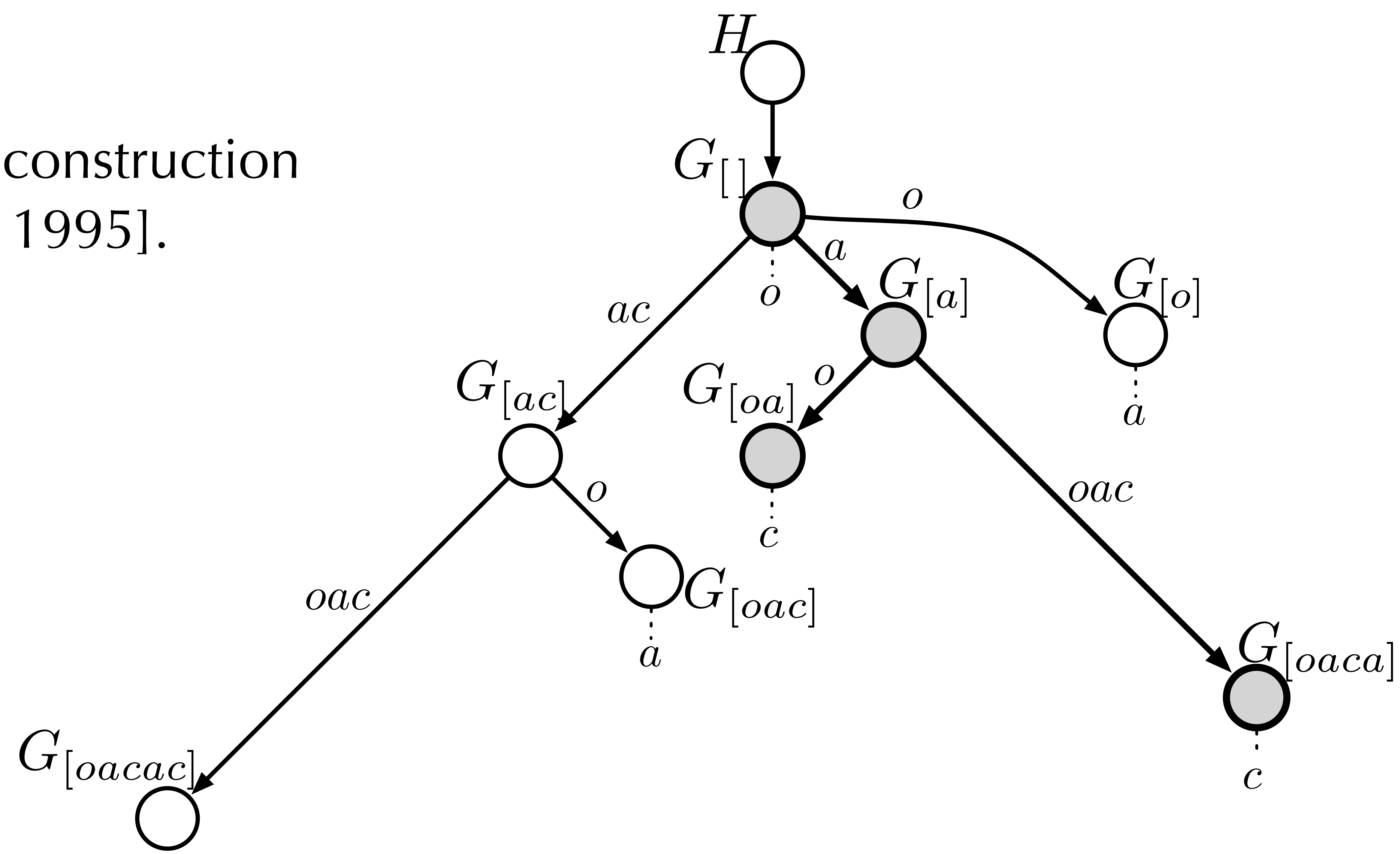


Model Size: Infinite -> O(T²) -> 2T

- Idea: integrate out non-branching, non-leaf nodes of the context tree.

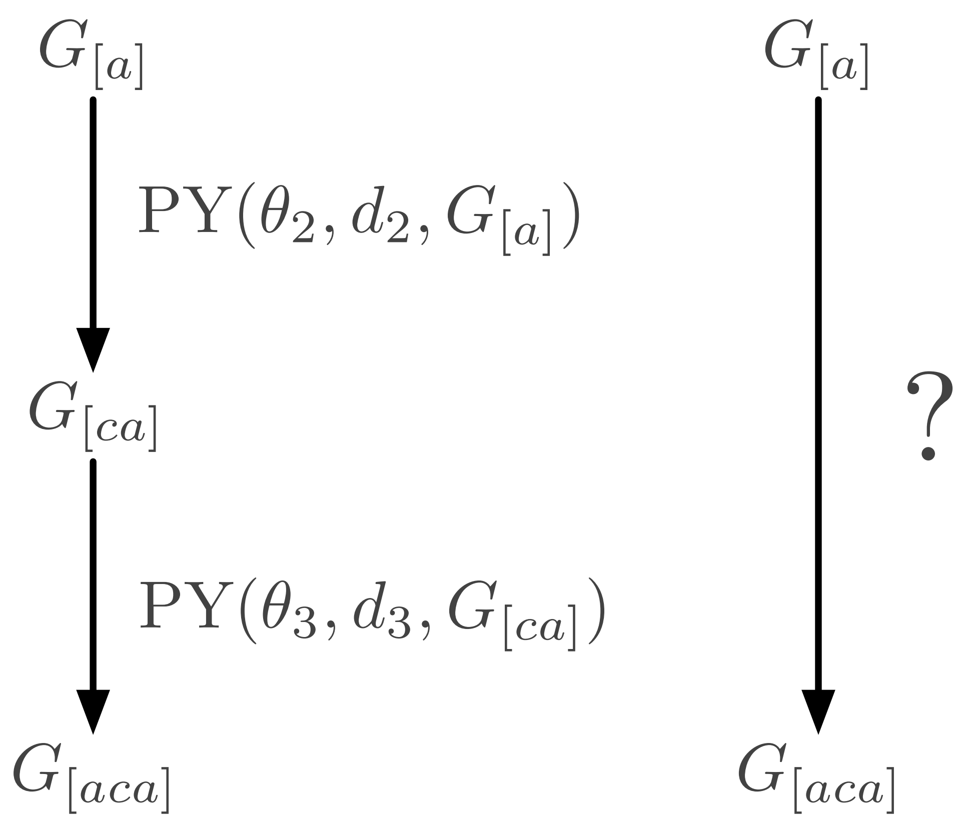
- Resulting tree is related to a suffix tree data structure, and has at most 2T nodes.

- There are linear time construction algorithms [Ukkonen 1995].



Closure under Marginalization

- In marginalizing out non-branching interior nodes, need to ensure that resulting conditional distributions are still tractable.



- E.g.: If each conditional is Dirichlet, resulting conditional is not of known analytic form.

Closure under Marginalization

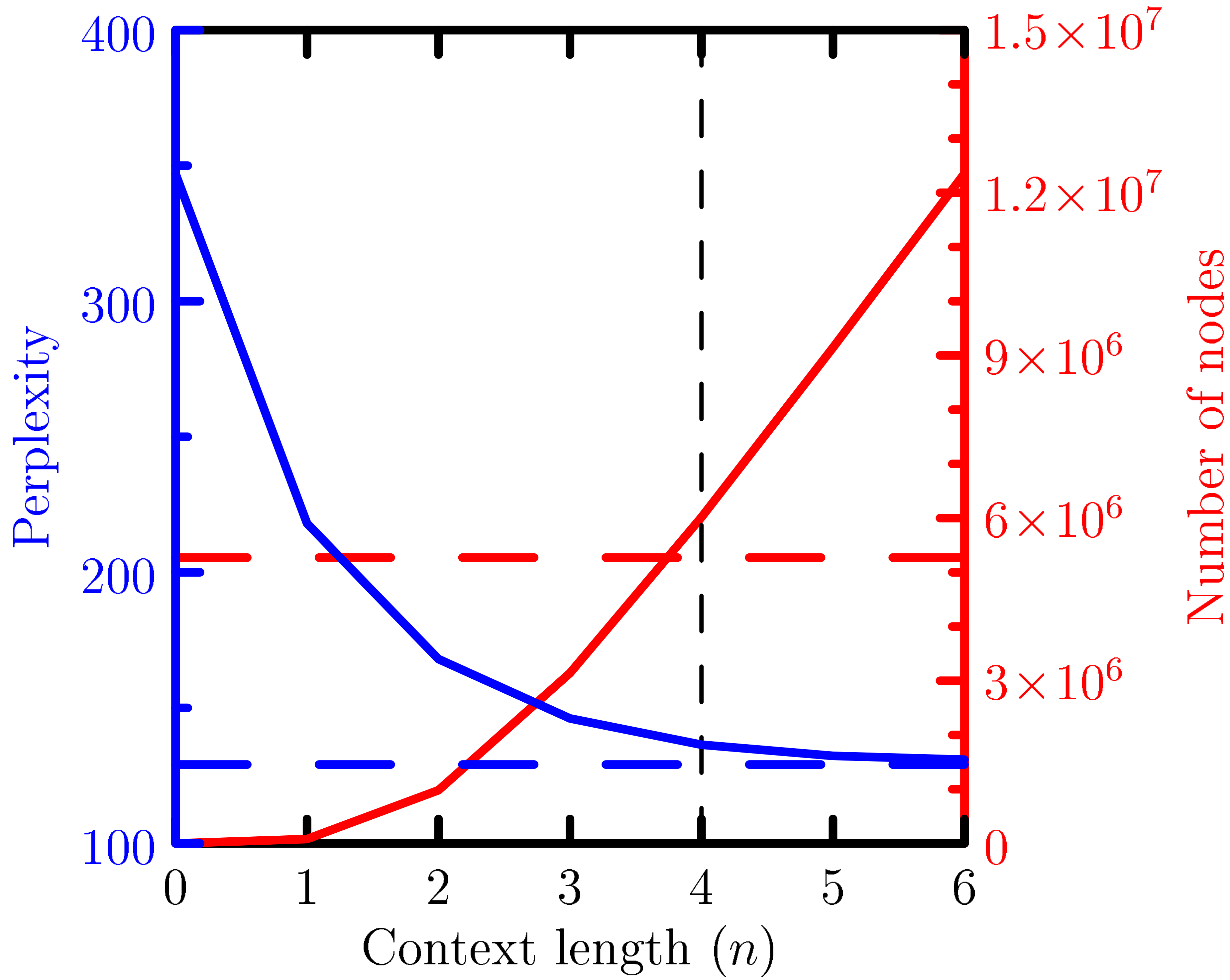
- In marginalizing out non-branching interior nodes, need to ensure that resulting conditional distributions are still tractable.

$$
\begin{tikzcd}
G_{[a]} \arrow[d, "\mathrm{PY}(\theta_2, d_2, G_{[a]})"] \\
G_{[ca]} \arrow[d, "\mathrm{PY}(\theta_2 d_3, d_3, G_{[ca]})"] \\
G_{[aca]}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
G_{[a]} \arrow[d, "\mathrm{PY}(\theta_2 d_3, d_2 d_3, G_{[a]})"] \\
G_{[aca]}
\end{tikzcd}
$$

- For certain parameter settings, Pitman-Yor processes are closed under marginalization!

[Pitman 1999]

Comparison to Finite Order HPYLM



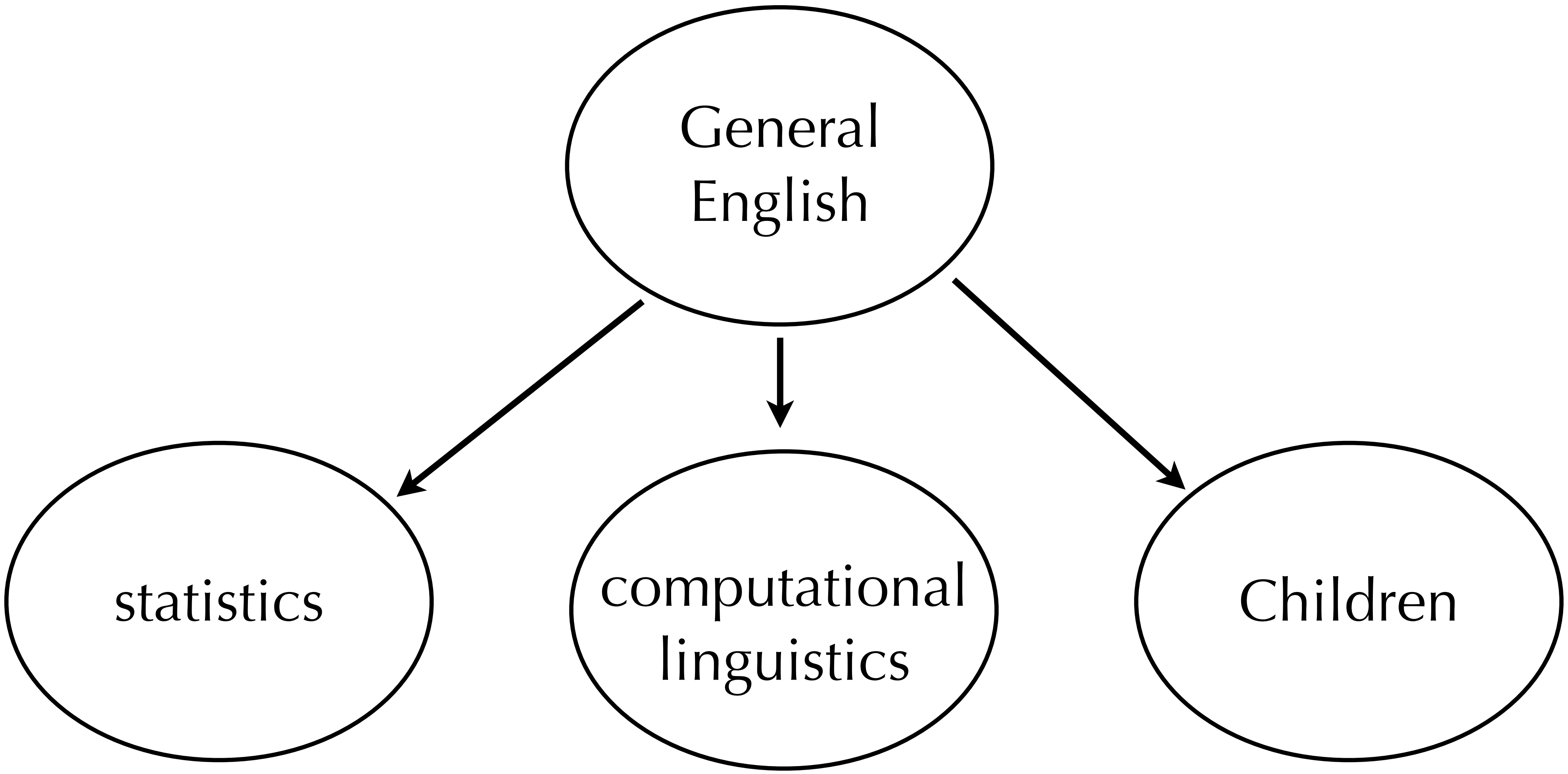
Compression Results

Model	Average bits/byte
gzip	2.61
bzip2	2.11
CTW	1.99
PPM	1.93
Sequence Memoizer	1.89

Calgary corpus
SM inference: particle filter
PPM: Prediction by Partial Matching
CTW: Context Tree Weigting
Online inference, entropic coding.

Hierarchical Bayes and Domain Adaptation

Domain Adaptation



Graphical Pitman-Yor Process

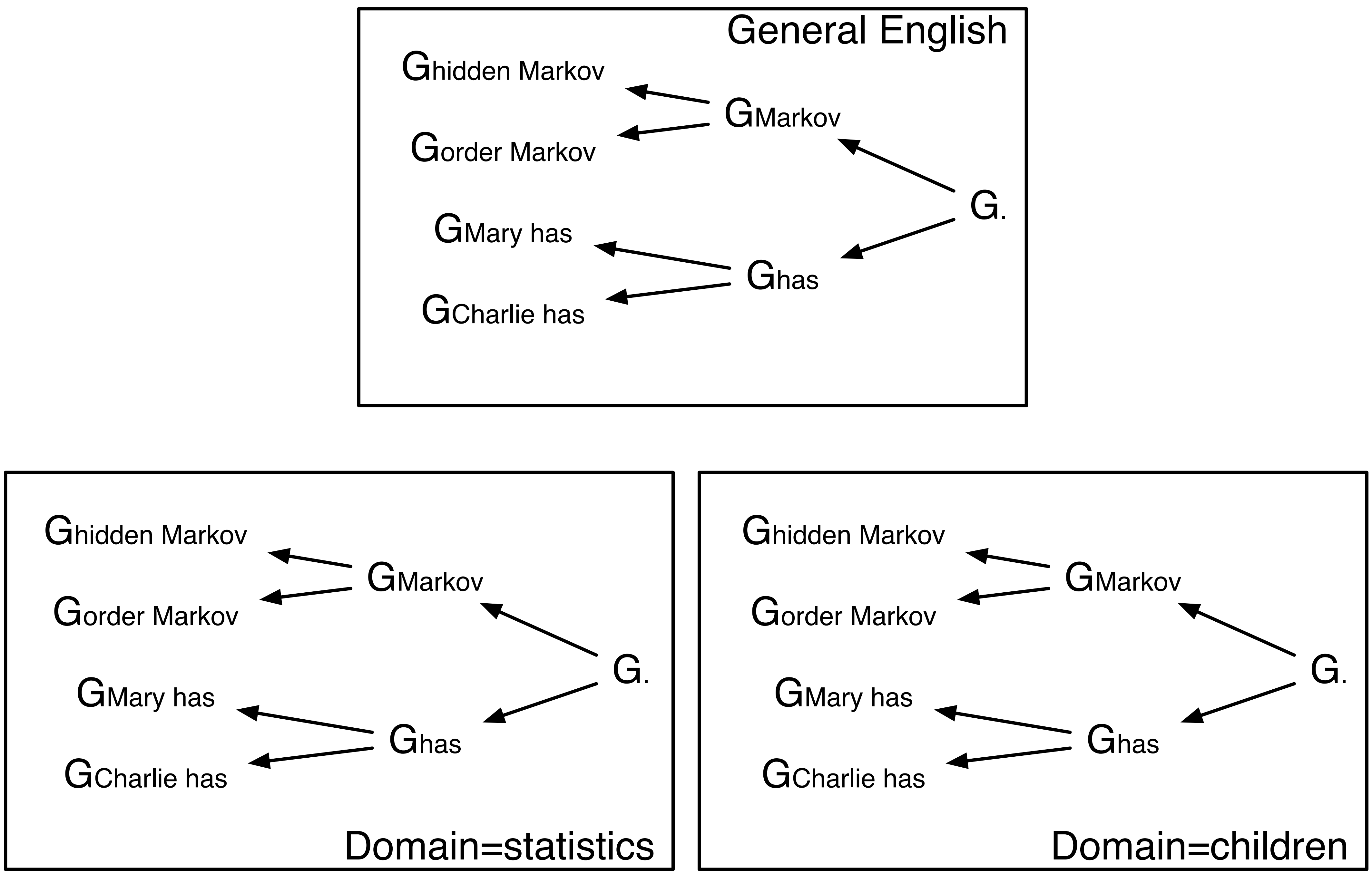
- Each conditional probability vector given context u=(w_1,w_2) and in domain has prior:

$$
\begin{aligned}
& G^{\text{domain}}_{w_1,w_2} \mid G^{\text{domain}}_{w_2}, G^{\text{general}}_{w_1,w_2} \\
\sim \quad & \mathrm{PY}(\theta, d, \pi G^{\text{domain}}_{w_2} + (1-\pi) G^{\text{general}}_{w_1,w_2})
\end{aligned}
$$

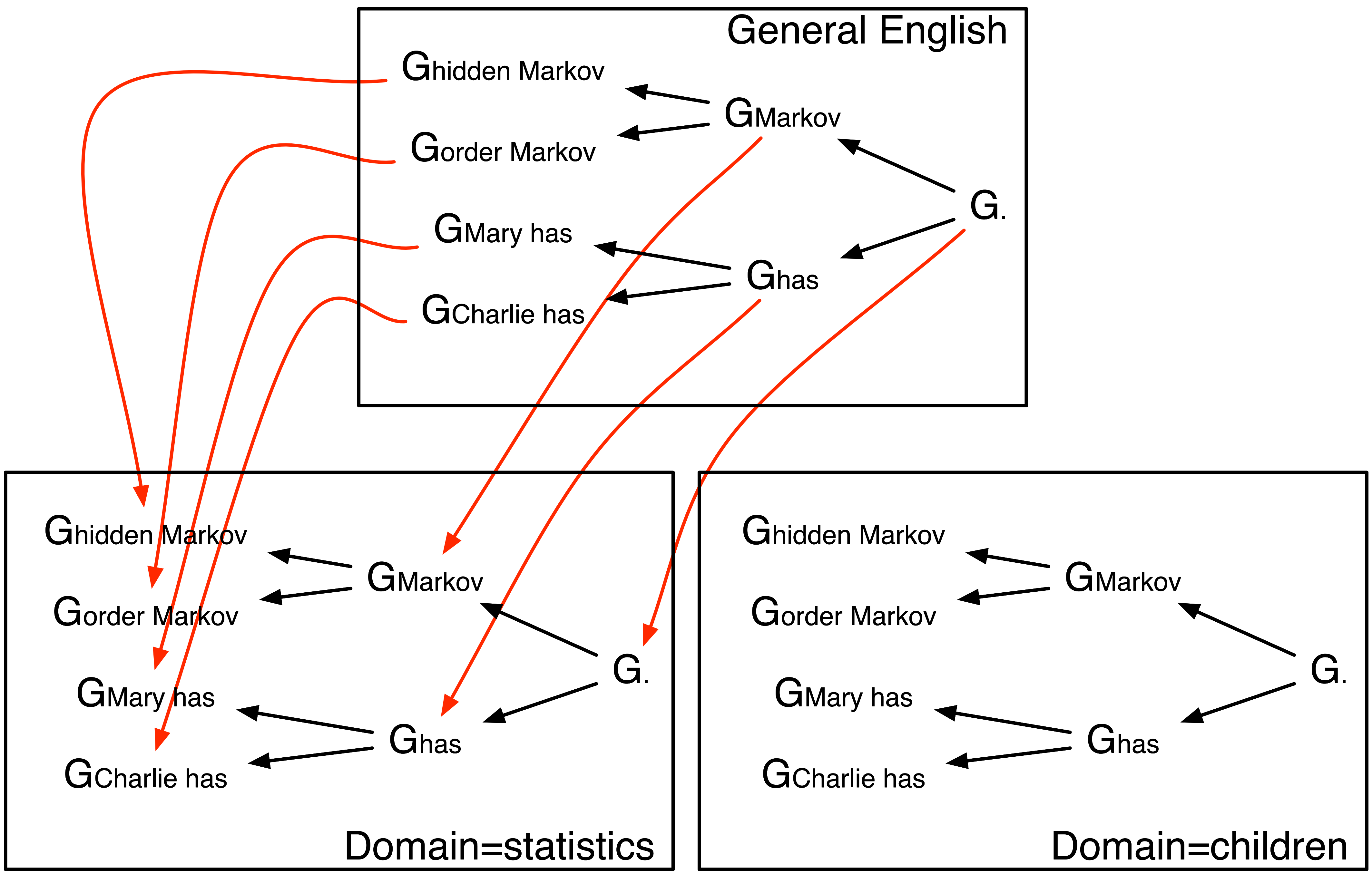
- Back-off in two different ways.
- More flexible than a straight mixture of the two base distributions.

- An example of a graphical Pitman-Yor process.

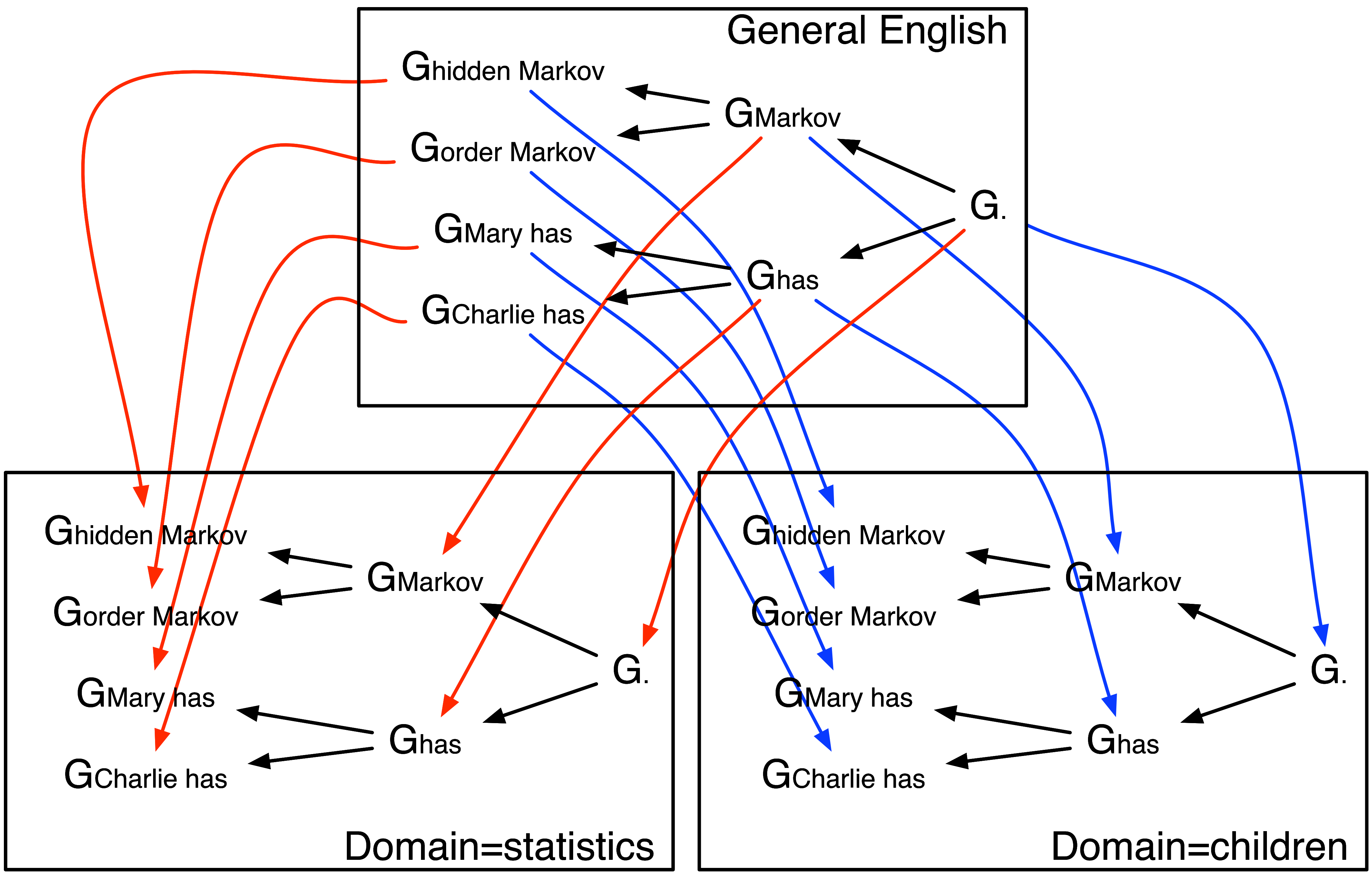
Graphical Pitman-Yor Process



Graphical Pitman-Yor Process

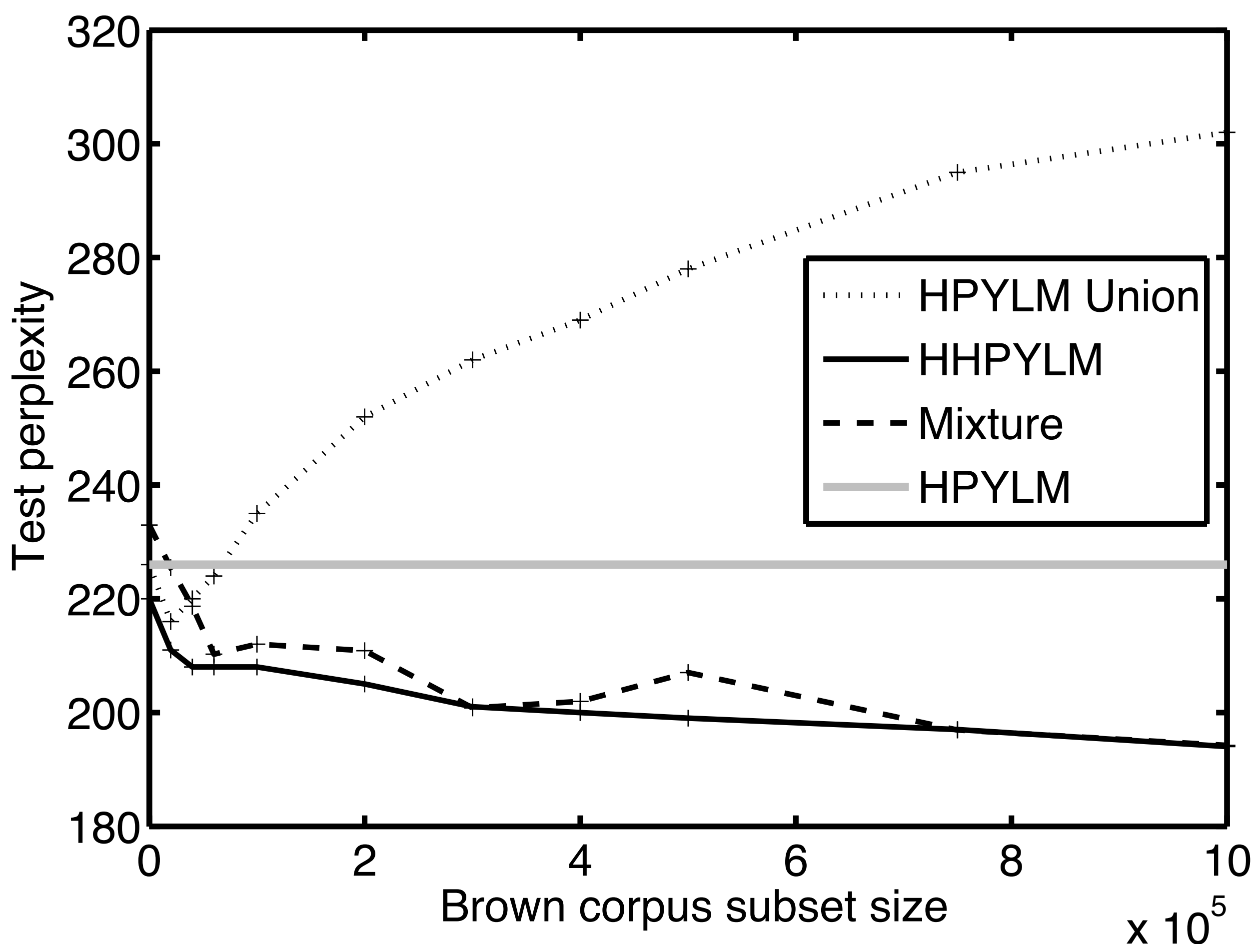


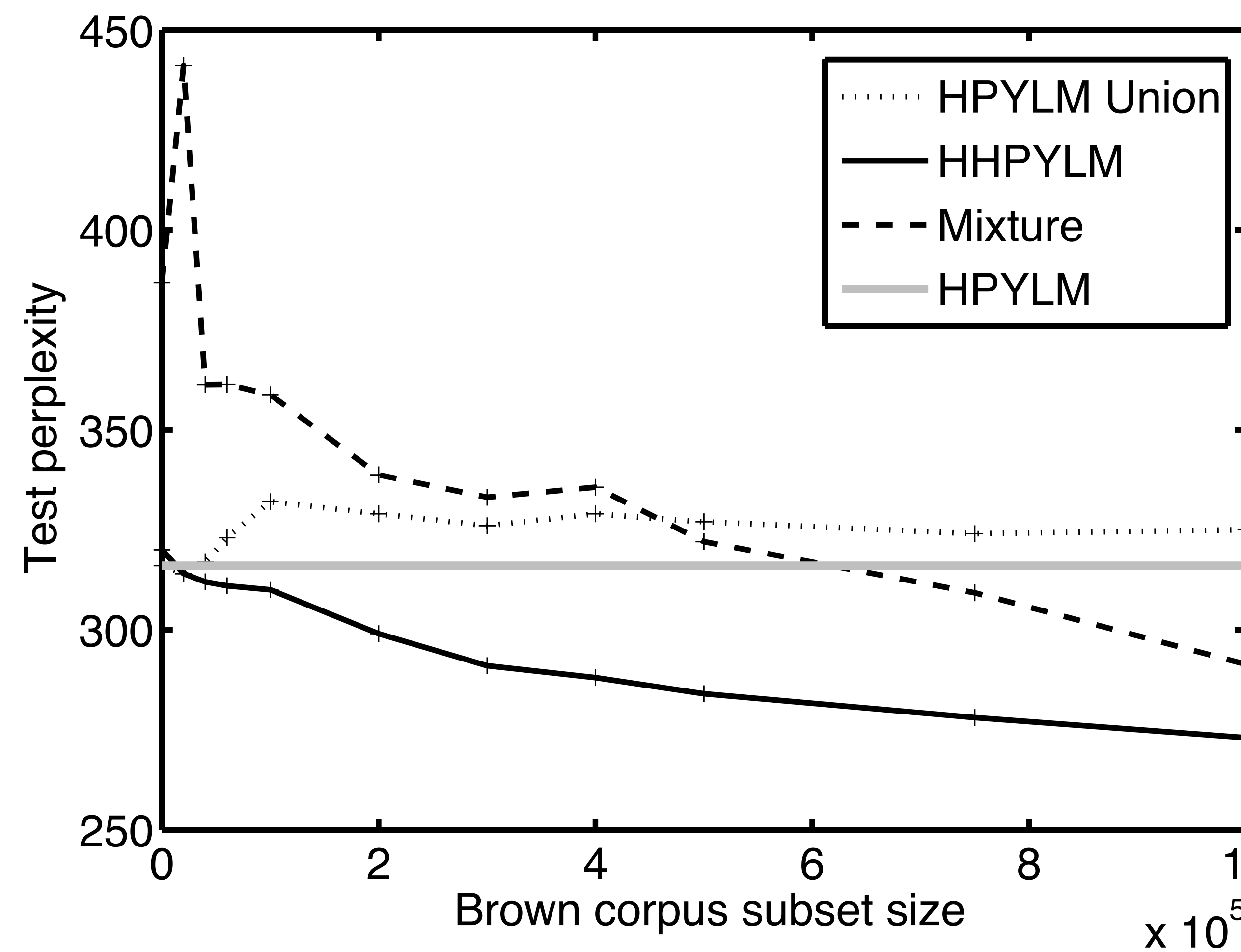
Graphical Pitman-Yor Process



Domain Adaptation Results

- Compared a graphical Pitman-Yor domain adapted language model to:
  - no additional domain.
  - naively including additional domain.
  - mixture model.

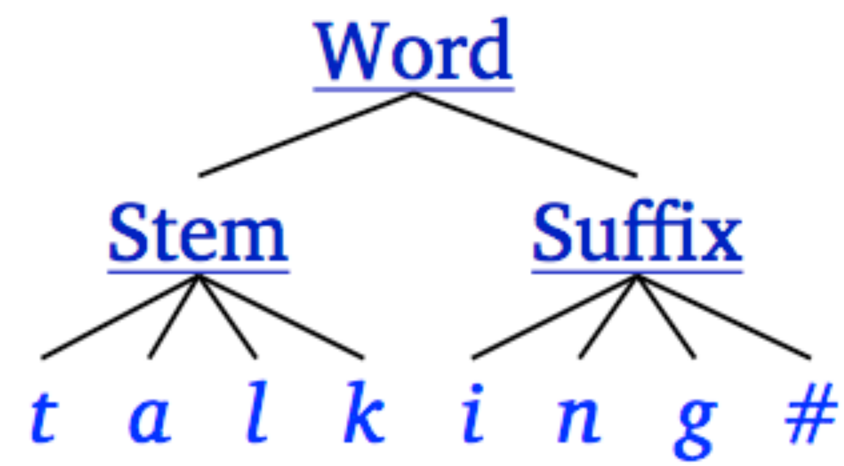




Related Works

Adaptor Grammars

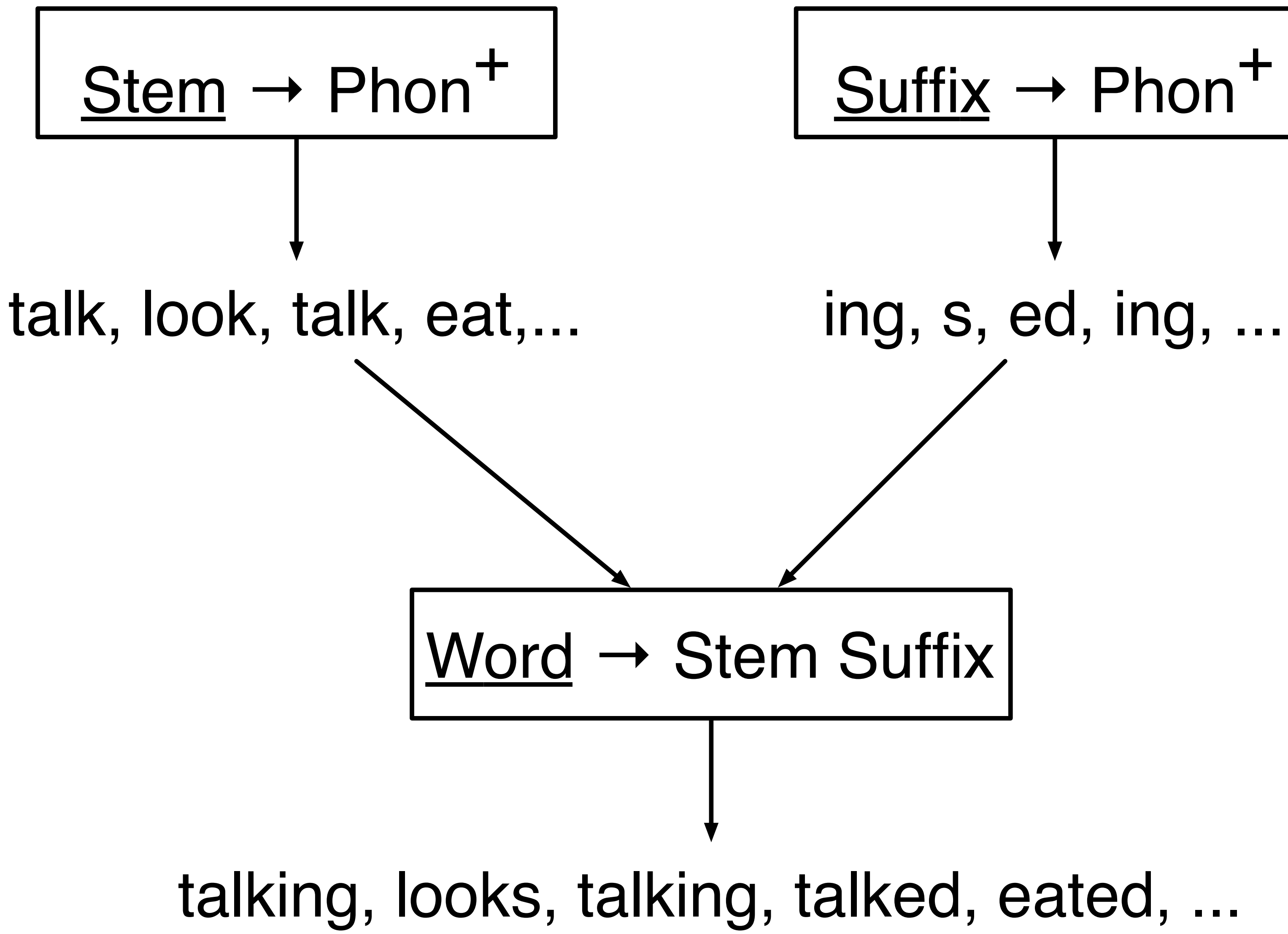
$$\underline{\text{Word}} \rightarrow \text{Stem Suffix}$$
$$\underline{\text{Stem}} \rightarrow \text{Phon}^{+}$$
$$\underline{\text{Suffix}} \rightarrow \text{Phon}^{+}$$



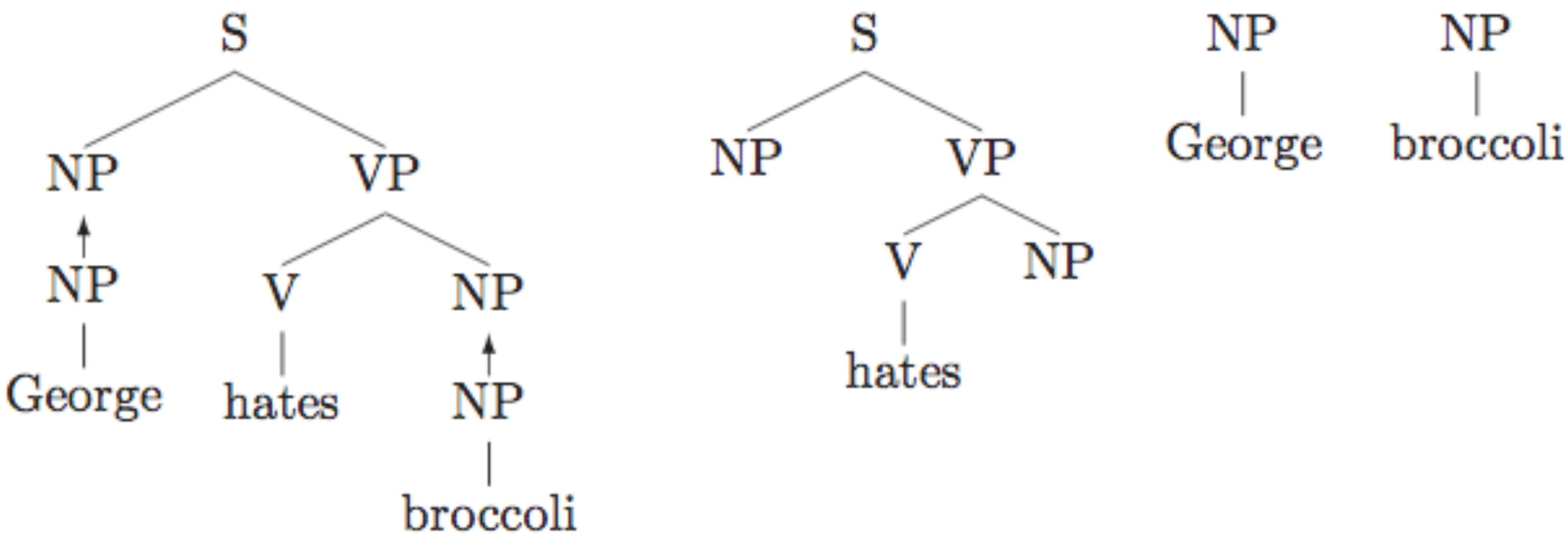
- Reuse fully expanded subtrees of PCFG using Chinese restaurant processes.
- Flexible framework and software to make use of hierarchical Pitman-Yor process technology.
- Applied to unsupervised word segmentation, morphological analysis etc.

[Johnson, Griffiths, Goldwater *]

Adaptor Grammars



Tree Substitution Grammars



- Multiple level hierarchy of adapted by Pitman-Yor processes:
  - tree fragments
  - PCFG productions
  - lexicalization
  - heads of CFG rules

[Goldwater, Blunsom & Cohn *] also [Post & Gildea 2009, O'Donnell et al 2009]

Other Related Works

- Infinite PCFGs [Finkel et al 2007, Liang et al 2007]
- Infinite Markov model [Mochihashi & Sumita 2008]
- Nested Pitman-Yor language models [Mochihashi et al 2009]
- POS induction [Blunsom & Cohn 2011]

Concluding Remarks

Conclusions

- Bayesian methods are powerful approaches to computational linguistics.
  - Hierarchical Bayesian models for encoding generalization capabilities.
  - Pitman-Yor processes for encoding power law properties.
  - Nonparametric Bayesian models for sidestepping model selection.
- Hurdles to future progress:
  - Scaling up Bayesian inference to large datasets and large models.
  - Better exploration of combinatorial spaces.
- Fruitful cross-pollination of ideas across machine learning, statistics and computational linguistics.

Thank You!

Sharon Goldwater, Chris Manning & CoNLL committee

Acknowledgements:
Frank Wood, Jan Gasthaus,
Cedric Archambeau, Lancelot James

Lee Kuan Yew Foundation
Gatsby Charitable Foundation