Variational Inference for Adaptor Gramars

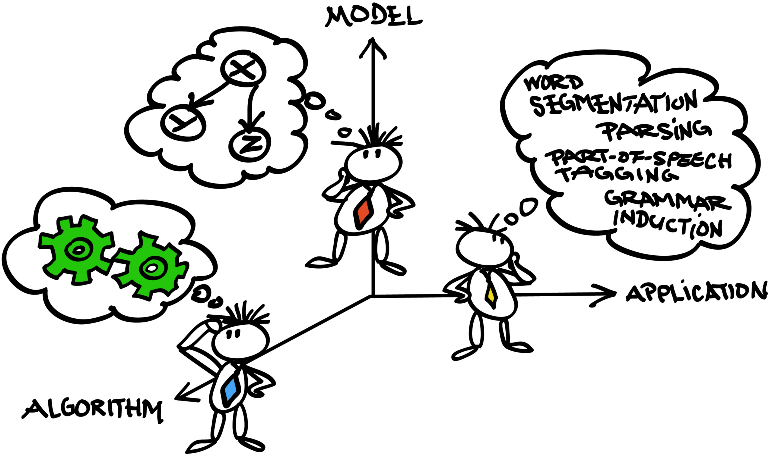
Shay Cohen
School of Computer Science
Carnegie Mellon University

David Blei
Computer Science Department
Princeton University

Noah Smith
School of Computer Science
Carnegie Mellon University

Outline

The lifecycle of unsupervised learning:



Outline

We give a new representation to an existing model (adaptor grammars)

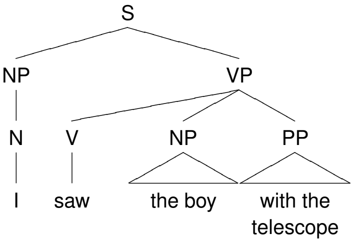
This representation leads to a new variational inference algorithm for adaptor grammars

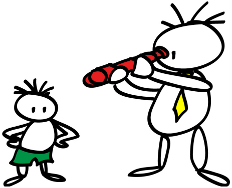
We do a sanity check on word segmentation, comparing to state-of-the-art results

Our inference algorithm permits to do dependency unsupervised parsing with adaptor grammars

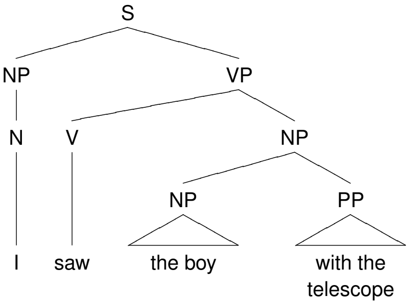
Problem 1 - PP Attachment

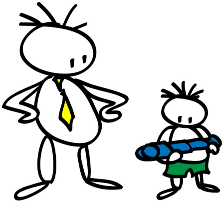
I saw the boy with the telescope





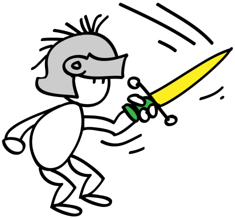
I saw the boy with the telescope





Problem 2 - Word Segmentation

Matthewslikeswordfighting
Matthews like sword fighting



Matthewslikeswordfighting
Matthews likes word fighting

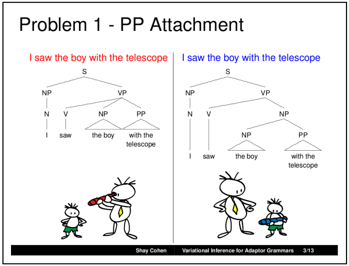
What is missing?

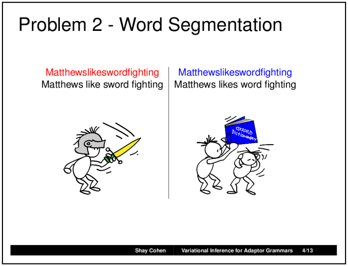
Context could resolve this ambiguity

But we want unsupervised learning...

Where do we get the context?

. . .



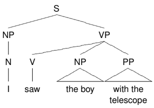


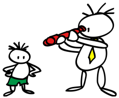
. . .

Problem 1 - PP Attachment

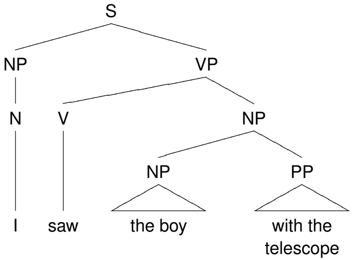
(S (NP The boy with the telescope) (V entered) (NP the park))

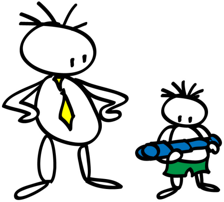
I saw the boy with the telescope





I saw the boy with the telescope



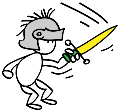


Problem 2 - Word Segmentation

Word fighting is the new hobby of computational linguists.
Mr. Matthews is a computational linguist.

Matthewslikeswordfighting

Matthews like sword fighting



Matthewslikeswordfighting

Matthews likes word fighting

Dreaming Up Patterns

Context helps. Where do we get it? Adaptor grammars (Johnson et al. 2006)

Define a distribution over trees

New samples depend on the history - “rich get richer” dynamics

Dream up “patterns” as we go along

Adaptor Grammars

Use the Pitman-Yor process with PCFGs as base distribution

To make it fully Bayesian, we also have a Dirichlet prior over the PCFG rules

Originally represented using the Chinese restaurant process (CRP)

CRP is convenient for sampling – not for variational inference

Variational Inference in a Nutshell

"Posterior inference" requires that we find parse trees $z_1, ..., z_n$ given raw sentences $x_1, ..., x_n$

Mean-field approximation: take all hidden variables: $z_1, ..., z_n$ and parameters θ.
Find a posterior of the form $q(z_1, ..., z_n, \theta) = q(\theta) \prod_{i=1}^{n} q(z_i)$ (makes inference tractable)

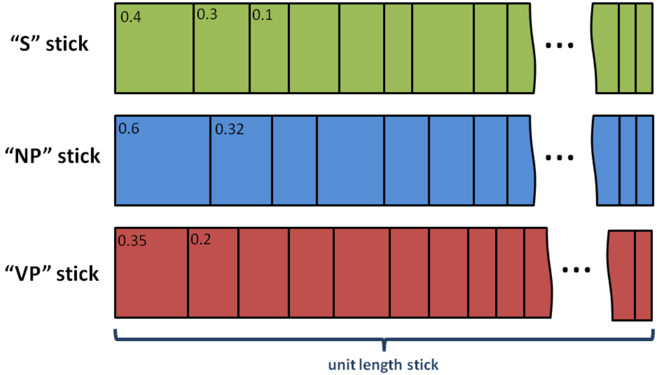
Makes independence assumptions in the posterior

That's all! Almost. We need a manageable representation for $z_1, ..., z_n$ and θ

Sampling vs. Variational Inference

	MCMC sampling	variational inference
convergence	guaranteed	local maximum
speed	slow	fast
algorithm	randomized	objective optimization
parallelization	non-trivial	easy

Stick Breaking Representation

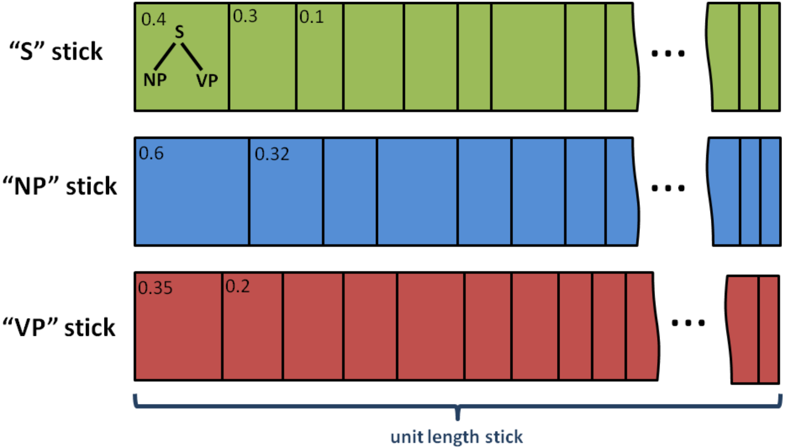


S	$\longrightarrow$	NP VP	*1.0*
NP	$\longrightarrow$	the boy	*0.5*
NP	$\longrightarrow$	an apple	*0.2*
NP	$\longrightarrow$	a mango	*0.3*

VP	$\longrightarrow$	ate NP	*0.6*
VP	$\longrightarrow$	met NP	*0.4*

Sticks are sampled from the GEM distribution
Everything which is a number in this slide, belongs to θ

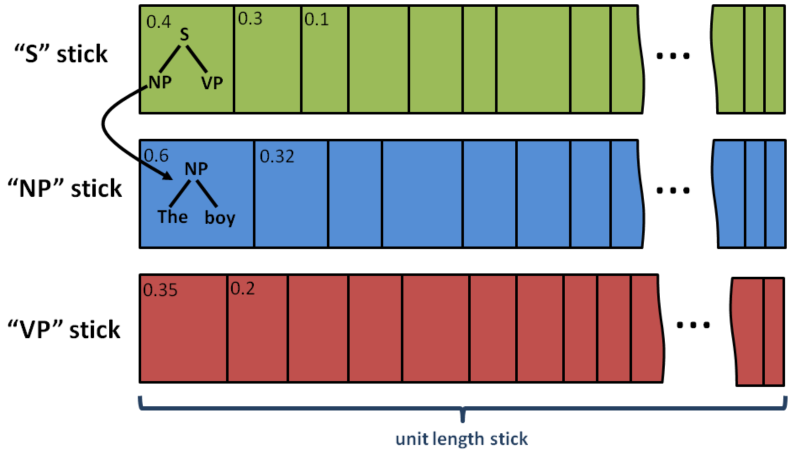
Stick Breaking Representation



S	$\longrightarrow$	NP VP	*1.0*
NP	$\longrightarrow$	the boy	*0.5*
NP	$\longrightarrow$	an apple	*0.2*
NP	$\longrightarrow$	a mango	*0.3*

VP	$\longrightarrow$	ate NP	*0.6*
VP	$\longrightarrow$	met NP	*0.4*

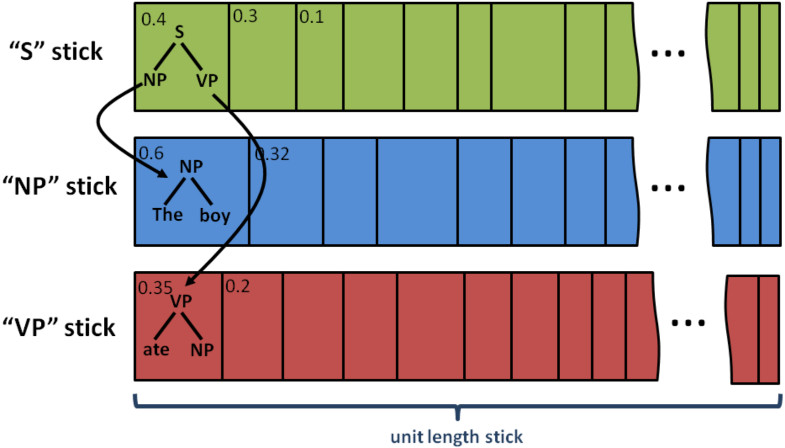
Stick Breaking Representation



S	$\longrightarrow$	NP VP	*1.0*
NP	$\longrightarrow$	the boy	*0.5*
NP	$\longrightarrow$	an apple	*0.2*
NP	$\longrightarrow$	a mango	*0.3*

VP	$\longrightarrow$	ate NP	*0.6*
VP	$\longrightarrow$	met NP	*0.4*

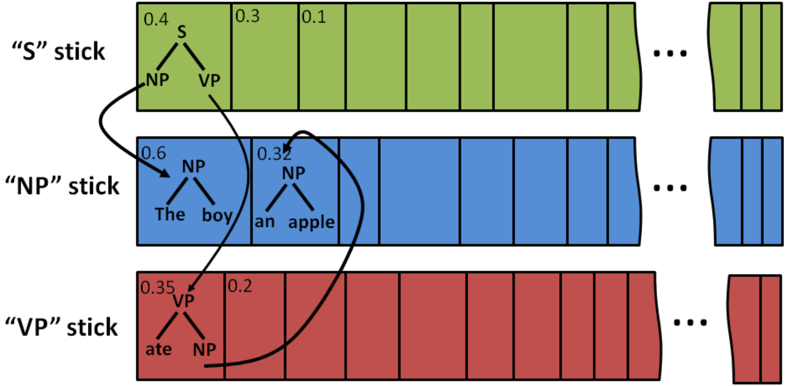
Stick Breaking Representation



S	$\longrightarrow$	NP VP	*1.0*
NP	$\longrightarrow$	the boy	*0.5*
NP	$\longrightarrow$	an apple	*0.2*
NP	$\longrightarrow$	a mango	*0.3*

VP	$\longrightarrow$	ate NP	*0.6*
VP	$\longrightarrow$	met NP	*0.4*

Stick Breaking Representation

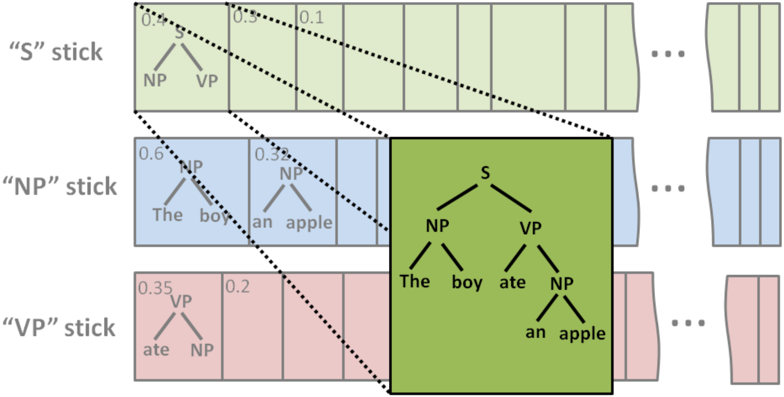


Total probability of "the boy ate an apple" is 0.4!

Probabilities for sampling trees are chosen based on the stick probabilities only, as if independently from the PCFG!

The PCFG determines the trees the sticks are populated with

Stick Breaking Representation



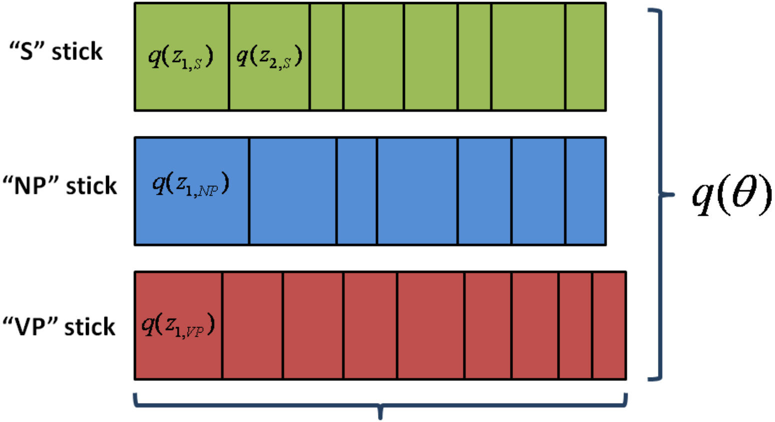
Total probability of "the boy ate an apple" is 0.4!

Probabilities for sampling trees are chosen based on the stick probabilities only, as if independently from the PCFG!

⟶

The PCFG determines the trees the sticks are populated with

Truncated Stick Approximation



truncated stick approximation (Blei and Jordan, 2004)

S $\longrightarrow$	NP VP	*1.0*	VP $\longrightarrow$	ate NP	*0.6*
NP $\longrightarrow$	the boy	*0.5*	VP $\longrightarrow$	met NP	*0.4*
NP $\longrightarrow$	an apple	*0.2*			
NP $\longrightarrow$	a mango	*0.3*			

Sanity Check - Word Segmentation

Task is to segment a sequence of phonemes into words

- Example: yuwanttulUk&tDIs → yu want tu lUk &t DIs

Models language acquisition in children (using the corpus from Brent and Cartwright, 1996)

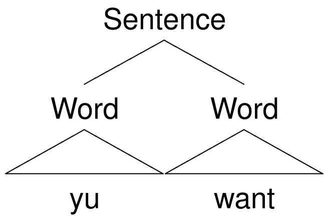
The corpus includes 9,790 utterances

Has been used before with adaptor grammars with three grammars

Baseline: Sampling method from Johnson and Goldwater, 2009

Word Segmentation - Grammars

Unigram Grammar
Sentence $\rightarrow$ Word$^+$
Word $\rightarrow$ Char$^+$



- "Word" is adapted (hence, if something was a Word constituent previously, it is more likely to appear again)
- There are additional grammars: collocation grammar and syllable grammar (take into account more information about language)
- Words are segmented according to "Word" constituents
- All grammars are not recursive
- Used in Johnson and Goldwater (2009)

Word Segmentation - Results

grammar	our paper	J&G 2009
$\boldsymbol{G}_{\text{Unigram}}$	0.84	0.81
$\boldsymbol{G}_{\text{Colloc}}$	0.86	0.86
$\boldsymbol{G}_{\text{Syllable}}$	0.83	0.89

J&G 2009 - Johnson and Goldwater (2009) – best result

Scores reported are F_1 measure

Variants

Model:
Pitman-Yor Process vs. Dirichlet Process (did not have much effect)

Inference:
Fixed Stick vs. Dynamic Stick Expansion (fixed stick is better)

Decoding:
Minimum Bayes Risk vs. Viterbi (MBR does better)

See paper for details!

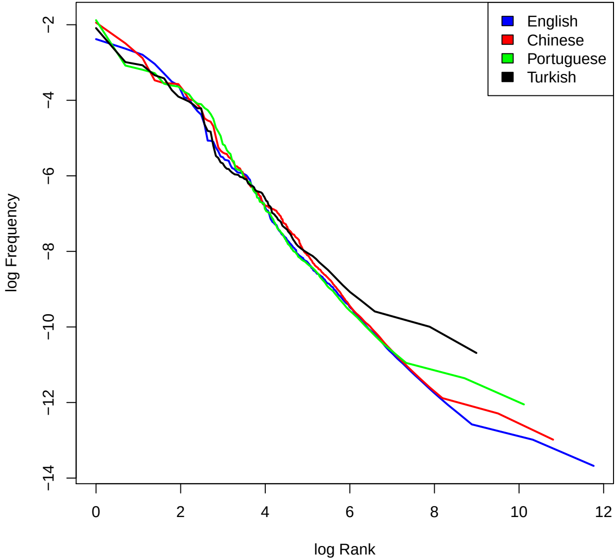
Running Time

Running time (clock time) of the sampler and variational inference is approximately the same (note that implementations are different)

However, variational inference can be parallelized

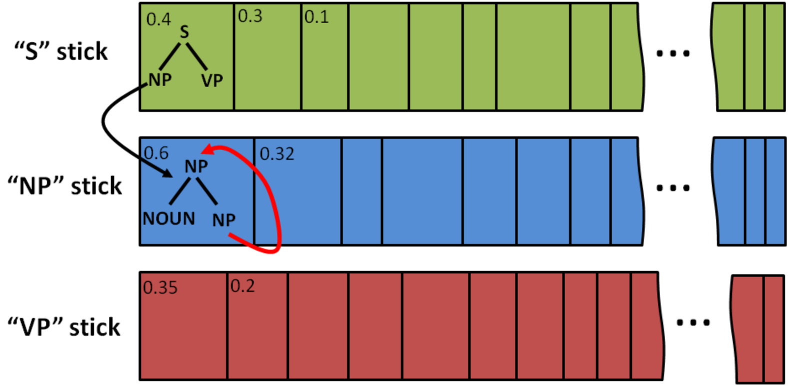
Reduction in clock time by factor of 2.8 when parallelizing on 20 weaker CPUs

Syntax and Power Law



Motivating adaptor grammars for unsupervised parsing, a plot of log rank of constituents vs. their log frequency

Recursive Grammars



S	$\longrightarrow$	NP VP	*1.0*
NP	$\longrightarrow$	the boy	*0.5*
NP	$\longrightarrow$	an apple	*0.2*
NP	$\longrightarrow$	a mango	*0.2*
VP	$\longrightarrow$	ate NP	*0.6*
VP	$\longrightarrow$	met NP	*0.4*
NP	$\longrightarrow$	NOUN NP	0.1

Recursive Grammars - Solution

Our finite approximation of the stick zeros all “bad” events in the variational distribution

Equivalent to inference when assuming the model is:

$$p'(x,z) = \frac{p(x,z) I(x,z \notin bad)}{\sum_{(x,z) \notin bad} p(x,z)}$$

where p is the original adaptor grammar model that gives non-zero probability to bad events and I is an 0/1 indicator

Unsupervised Parsing Setting

Experiments on the English Penn Treebank

Stripped off punctuation and kept only part-of-speech tags

Used adaptor grammars with dependency model with valence (Klein and Manning, 2004)

DMV has a PCFG representation (Smith, 2006)

We “adapt” the nonterminals that head noun constituents

Unsupervised Parsing - Results

model	Viterbi	MBR
non-Bayesian	45.8	46.1
Dirichlet prior	45.9	46.1
with adaptor grammars	48.3	**50.2**

Results are attachment-accuracy - fraction of parents correctly identified

A gain over vanilla Dirichlet, which is the prior used with adaptor grammars on the PCFG rules

Other priors (instead of Dirichlet prior) give performance $\sim$60. Can use it with adaptor grammars - future work

Summary

We described a variational inference algorithm for adaptor grammars

We showed it can lead to improvement in performance for various grammars

We showed it can be faster than sampling when parallelization is used

We applied adaptor grammars to dependency unsupervised parsing

Thanks! Questions?

Sampling vs. Variational Inference