From Probablistic Context-Free Grammars to Adaptor Grammars and Beyond

Mark Johnson

Joint work with Ben Börschinger, Wray Buntine, Katherine Demuth, Lan Du, Michael Frank, Sharon Goldwater, Tom Griffiths and Bevan Jones

Macquarie University
Sydney, Australia

May 2013

Research motivation and strategy

- *How are human languages acquired?*
  - *Empiricist explanation:* languages are learnt from exposure to linguistic data
  - *Rationalist explanation:* the “essential” structure of language is innate
- Obviously both are correct to varying degrees

⇒ Start with aspects of language everyone agrees are learned:
  - the pronunciations of words
  - the mapping between words and meanings
- Even these learning problems are very hard!
- The inference methods we develop have other practical applications
  - the same techniques used to learn words and their referents can be used to learn *topical collocations for information extraction and document summarisation*

Outline

MACQUARIE UNIVERSITY

Probabilistic context-free grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - choosing a rule expanding that nonterminal, and
  - recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

Probability θ_r	*Rule r*
1	S $\rightarrow$ NP VP
0.7	NP $\rightarrow$ *Sam*
0.3	NP $\rightarrow$ *Sandy*
1	VP $\rightarrow$ V NP
0.8	V $\rightarrow$ *likes*
0.2	V $\rightarrow$ *hates*

S

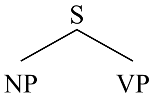
$$\mathrm{P}(\text{Tree}) \quad = $$

MACQUARIE UNIVERSITY

Probabilistic context-free grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - choosing a rule expanding that nonterminal, and
  - recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

Probability θ_r	*Rule r*
1	S $\rightarrow$ NP VP
0.7	NP $\rightarrow$ *Sam*
0.3	NP $\rightarrow$ *Sandy*
1	VP $\rightarrow$ V NP
0.8	V $\rightarrow$ *likes*
0.2	V $\rightarrow$ *hates*

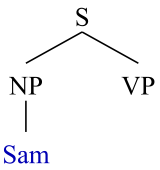


$$\mathrm{P}(\text{Tree}) = 1 \times$$

Probabilistic context-free grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - choosing a rule expanding that nonterminal, and
  - recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

Probability θ_r	*Rule r*
1	S → NP VP
0.7	NP → *Sam*
0.3	NP → *Sandy*
1	VP → V NP
0.8	V → *likes*
0.2	V → *hates*



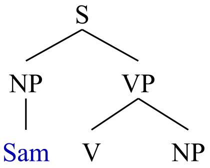
$$\mathrm{P}(\text{Tree}) = 1 \times 0.7 \times$$

MACQUARIE UNIVERSITY

Probabilistic context-free grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - choosing a rule expanding that nonterminal, and
  - recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

Probability θ_r	*Rule* r
1	S $\rightarrow$ NP VP
0.7	NP $\rightarrow$ *Sam*
0.3	NP $\rightarrow$ *Sandy*
1	VP $\rightarrow$ V NP
0.8	V $\rightarrow$ *likes*
0.2	V $\rightarrow$ *hates*



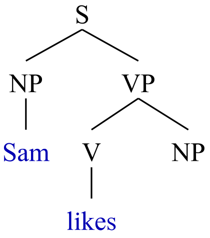
$$\mathrm{P}(\text{Tree}) \;=\; 1 \times 0.7 \times 1 \times$$

MACQUARIE UNIVERSITY

Probabilistic context-free grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - choosing a rule expanding that nonterminal, and
  - recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

Probability θ_r	*Rule* r
1	S $\rightarrow$ NP VP
0.7	NP $\rightarrow$ *Sam*
0.3	NP $\rightarrow$ *Sandy*
1	VP $\rightarrow$ V NP
0.8	V $\rightarrow$ *likes*
0.2	V $\rightarrow$ *hates*



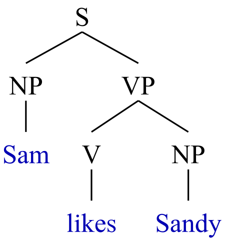
$$\mathrm{P}(\text{Tree}) \;=\; 1 \times 0.7 \times 1 \times 0.8 \times$$

MACQUARIE UNIVERSITY

Probabilistic context-free grammars

- Probabilistic context-free grammars (PCFGs) define *probability distributions over trees*
- Each *nonterminal node* expands by
  - choosing a rule expanding that nonterminal, and
  - recursively expanding any nonterminal children it contains
- Probability of tree is *product of probabilities of rules* used to construct it

Probability θ_r	*Rule* r
1	S $\rightarrow$ NP VP
0.7	NP $\rightarrow$ *Sam*
0.3	NP $\rightarrow$ *Sandy*
1	VP $\rightarrow$ V NP
0.8	V $\rightarrow$ *likes*
0.2	V $\rightarrow$ *hates*



$$\mathrm{P}(\text{Tree}) \;=\; 1 \times 0.7 \times 1 \times 0.8 \times 0.3$$

MACQUARIE UNIVERSITY

PCFGs as models of natural language syntax

- Simple PCFGs are *not very good models of natural language syntax*
  - PCFGs aren't good parameterisations of natural language
  - accurate PCFGs need thousands of nonterminal symbols and hundreds of thousands of rules

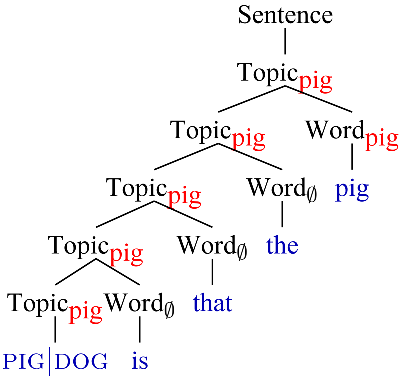
  $\Rightarrow$ smoothing is an essential "black art"
  - unsupervised estimators of PCFGs perform very poorly *even when initialised with correct parses*
- But PCFGs can model many other interesting things!

Learning the mapping from words to referents

- Input to learner:
  - word sequence: *Is that the pig?*
  - objects in nonlinguistic context: DOG, PIG
- Learning objectives:
  - identify utterance topic: PIG
  - identify word-topic mapping: *pig* $\rightsquigarrow$ PIG

A PCFG for learning word referents

- Prefix sentences with *possible topic marker*, e.g., PIG|DOG
- PCFG rules *choose a topic* from topic marker and *propagate it through sentence*
- Each word is either generated from sentence topic or null topic $\emptyset$



- Input grammar contains all possible rules of form $\text{Word}_t \rightarrow w$ for each topic t and word w
- PCFG inference procedure learns which words are associated with each topic

MACQUARIE UNIVERSITY

Modelling social cues in word learning

- Everyone agrees social interactions are important for children's early language acquisition
  - e.g. children who engage in more joint attention with caregivers (e.g., looking at toys together) learn words faster (Carpenter 1998)
- *Can computational models exploit social cues?*
  - we show this by building models that can exploit social cues, and show they *learns better on data with social cues than on data with social cues removed*
- Many different social cues could be relevant: *can our models learn the importance of different social cues?*
  - our models estimate *probability of each cue occuring with "topical objects"* and *probability of each cue occuring with "non-topical objects"*
  - they do this in an unsupervised way, i.e., they are not told which objects are topical

Exploiting social cues for learning word referents

- Frank et al (2012) corpus of 4,763 utterances with the following information:
  - the orthographic words uttered by the care-giver,
  - a set of *available topics* (i.e., objects in the non-linguistic objects),
  - the values of the social cues, and
  - a set of *intended topics*, which the care-giver refers to.
- Social cues annotated in corpus:

Social cue	**Value**
child.eyes	objects child is looking at
child.hands	objects child is touching
mom.eyes	objects care-giver is looking at
mom.hands	objects care-giver is touching
mom.point	objects care-giver is pointing to

Example utterance and its encoding as a string

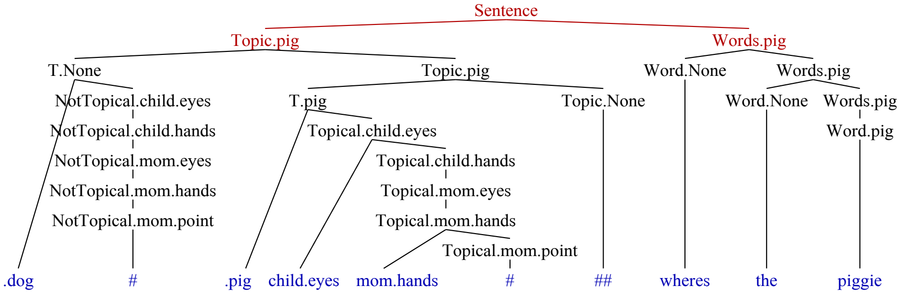
Input to learner:

.dog # .pig child.eyes mom.eyes mom.hands # ## wheres the piggie

Intended topic: *.pig*

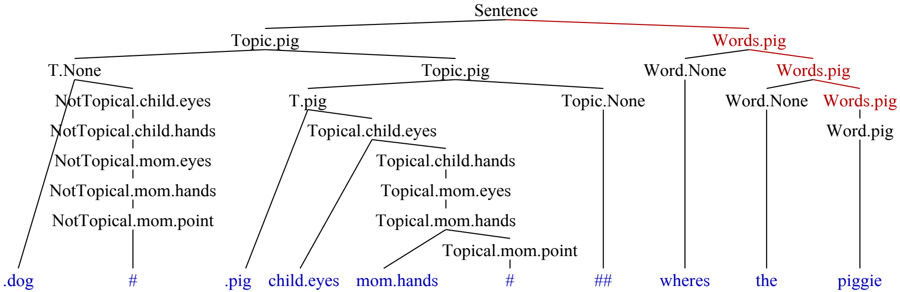
Word-topic associations: *piggie* $\rightsquigarrow$ *.pig*

Nondeterministically generating a topic

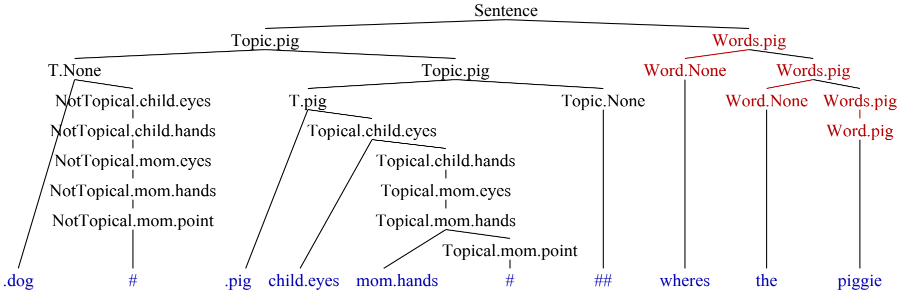


MACQUARIE UNIVERSITY

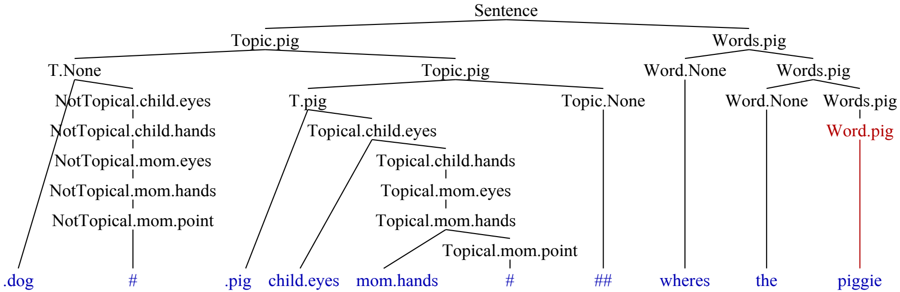
Propagating topic through utterance



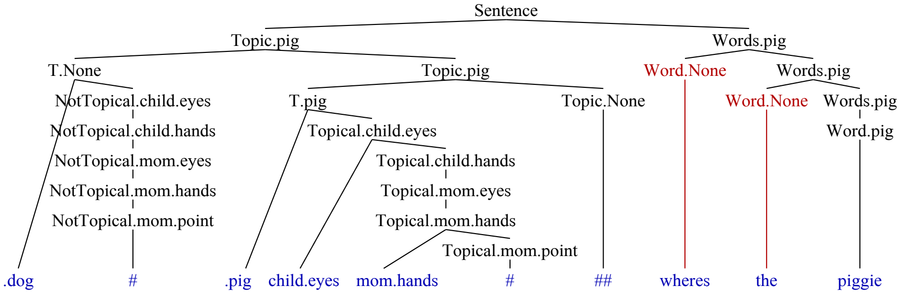
Choosing which words are topical



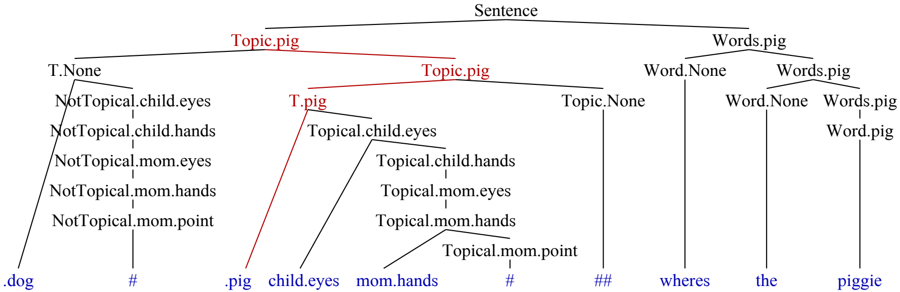
Generating topical words



Generating non-topical words

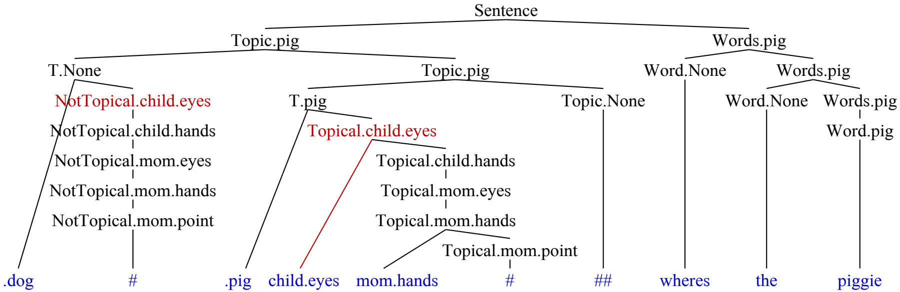


Checking topic is a possible topic



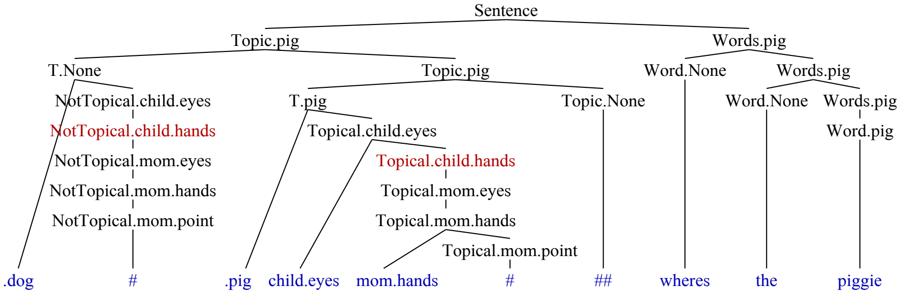
MACQUARIE UNIVERSITY

Generating social cues (child.eyes)



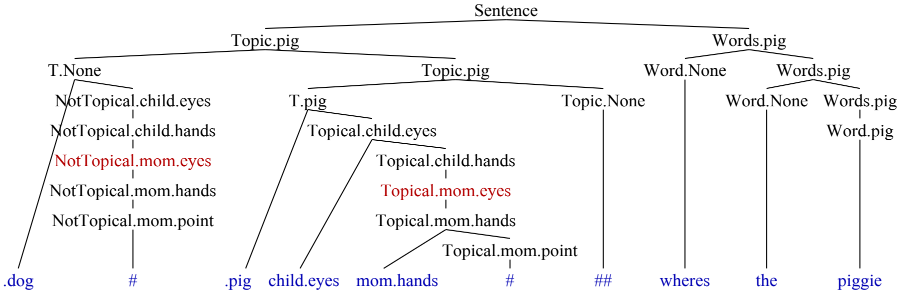
MACQUARIE UNIVERSITY

Generating social cues (child.hands)

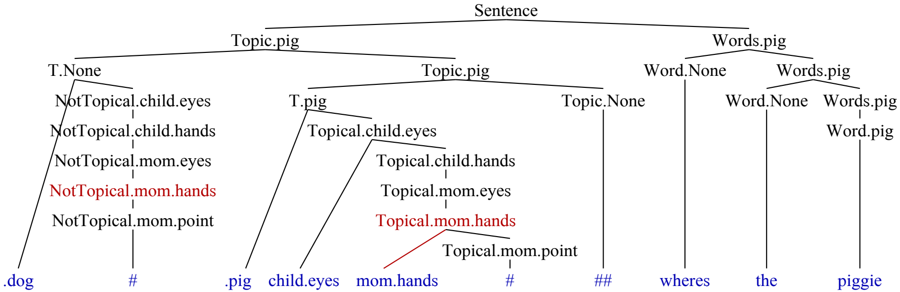


MACQUARIE UNIVERSITY

Generating social cues (mom.eyes)

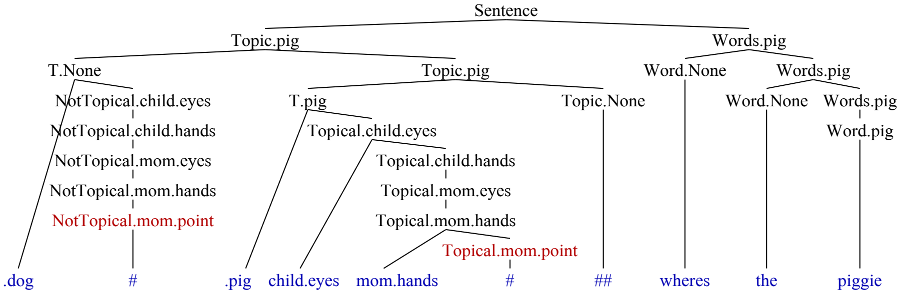


Generating social cues (mom.hands)



MACQUARIE UNIVERSITY

Generating social cues (mom.point)



Results for learning social cues

- Because all our models are implemented in the same framework, comparing their performance lets us *study the contributions of different information sources*
- In the four different models we tried, *social cues* improved the accuracy of:
  - recovering the *utterance topic*
  - identifying the *word(s) referring to the topic*, and
  - *learning a lexicon* (word $\rightsquigarrow$ topic mapping)
- *kideyes* was the most important social cue for each of these tasks in all of the models
- We've extended this model to account for *inter-sentential topic dependencies*
  - this required new PCFG parsing and inference algorithms that can parse entire discourses

Outline

Probabilistic Context-Free Grammars

Topic models as PCFGs

Adaptor grammars

Learning word pronunciations

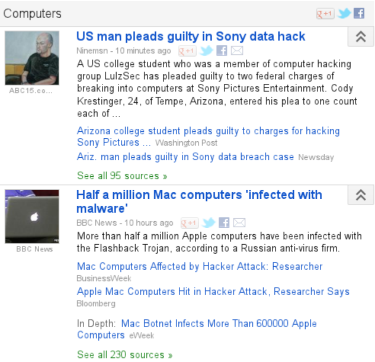
Finding topical collocations with adaptor grammars

Project report on Wray's and my project

Conclusions and future work

Topic models for document processing

- Topic models *cluster documents into one or more topics*
  - usually *unsupervised* (i.e., topics aren't given in training data)
- Important for document analysis and information extraction
  - Example: clustering news stories for information retrieval
  - Example: tracking evolution of a research topic over time



MACQUARIE UNIVERSITY

Mixture versus admixture topic models

- In a *mixture model*, each document has a *single topic*
  - all words in the document come from this topic
- In *admixture models*, each document has a *distribution over topics*
  - a single document can have multiple topics (number of topics in a document controlled by prior)
  - $\Rightarrow$ can capture more complex relationships between documents than a mixture model
- Both mixture and admixture topic models typically use a *"bag of words"* representation of a document

MACQUARIE UNIVERSITY

Example: documents from NIPS corpus

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

Example (cont): ignore function words

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services . . .

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, . . .

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for . . .

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some . . .

Example (cont): mixture topic model

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services . . .

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, . . .

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for . . .

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some . . .

Example (cont): admixture topic model

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

MACQUARIE UNIVERSITY

Wray's and my project: collocation topic models

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

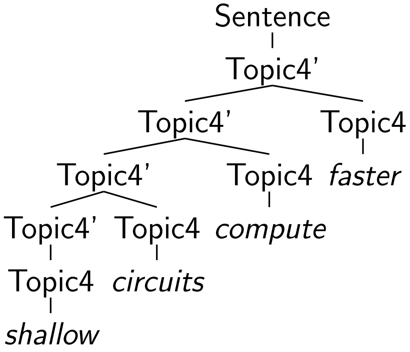
Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

MACQUARIE UNIVERSITY

Mixture topic models as PCFGs (1)

- Idea: Design PCFG so that:
  - non-deterministic rules implement generative steps in topic model
  - deterministic rules propagate information to appropriate place

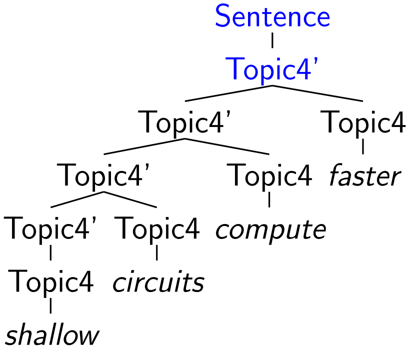
$$
\begin{array}{lll}
\text{Sentence} \rightarrow & \text{Topic}'_i & i \in 1,\ldots,\ell \\
\text{Topic}'_i \rightarrow & \text{Topic}'_i\ \text{Topic}_i & i \in 1,\ldots,\ell \\
\text{Topic}'_i \rightarrow & \text{Topic}_i & i \in 1,\ldots,\ell \\
\text{Topic}_i \rightarrow & w & i \in 1,\ldots,\ell \\
 & & w \in \mathcal{W}
\end{array}
$$



Mixture topic models as PCFGs (2)

- Choose a topic for sentence (non-deterministically)

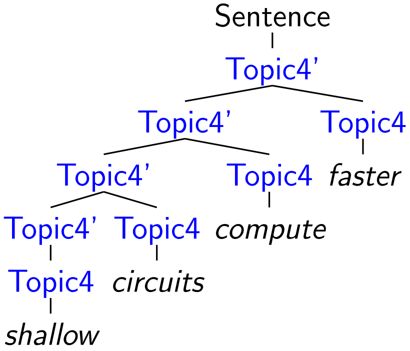
$$\begin{array}{lll}
\text{Sentence} \to & \text{Topic}'_i & i \in 1, \ldots, \ell \\
\text{Topic}'_i \to & \text{Topic}'_i \ \text{Topic}_i & i \in 1, \ldots, \ell \\
\text{Topic}'_i \to & \text{Topic}_i & i \in 1, \ldots, \ell \\
\text{Topic}_i \to & w & i \in 1, \ldots, \ell \\
 & & w \in \mathcal{W}
\end{array}$$



Mixture topic models as PCFGs (3)

- Copy sentence topic to each word (deterministically)

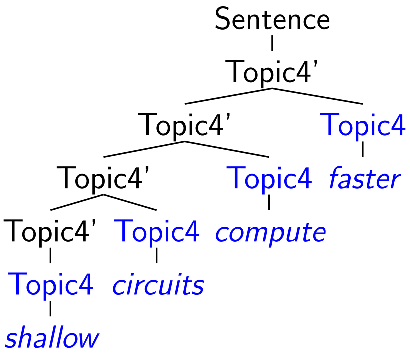
$$
\begin{array}{lll}
\text{Sentence} \rightarrow & \text{Topic}'_i & i \in 1, \ldots, \ell \\
\text{Topic}'_i \rightarrow & \text{Topic}'_i \ \text{Topic}_i & i \in 1, \ldots, \ell \\
\text{Topic}'_i \rightarrow & \text{Topic}_i & i \in 1, \ldots, \ell \\
\text{Topic}_i \rightarrow & w & i \in 1, \ldots, \ell \\
 & & w \in \mathcal{W}
\end{array}
$$



Mixture topic models as PCFGs (4)

- Generate each word from sentence topic (non-deterministically)

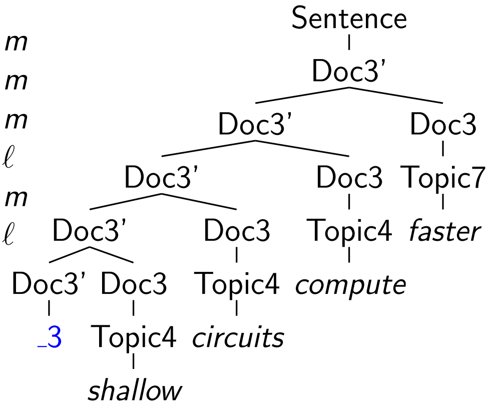
$\text{Sentence} \rightarrow \text{Topic}'_i$	$i \in 1,\ldots,\ell$
$\text{Topic}'_i \rightarrow \text{Topic}'_i\ \text{Topic}_i$	$i \in 1,\ldots,\ell$
$\text{Topic}'_i \rightarrow \text{Topic}_i$	$i \in 1,\ldots,\ell$
$\text{Topic}_i \rightarrow w$	$i \in 1,\ldots,\ell$ $w \in \mathcal{W}$



Admixture topic models as PCFGs (1)

- Prefix strings from document j with a *document identifier* "$\_j$"

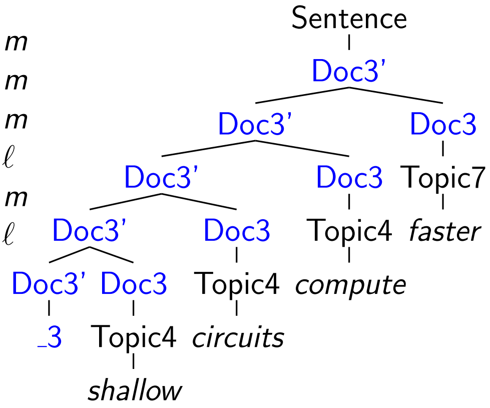
$$
\begin{array}{lll}
\text{Sentence} \to \text{Doc}'_j & & j \in 1, \ldots, m \\
\text{Doc}'_j \to \_j & & j \in 1, \ldots, m \\
\text{Doc}'_j \to \text{Doc}'_j\ \text{Doc}_j & & j \in 1, \ldots, m \\
\text{Doc}_j \to \text{Topic}_i & & i \in 1, \ldots, \ell \\
 & & j \in 1, \ldots, m \\
\text{Topic}_i \to w & & i \in 1, \ldots, \ell \\
 & & w \in \mathcal{W}
\end{array}
$$



Admixture topic models as PCFGs (2)

- Spine deterministically *propagates document id up through tree*

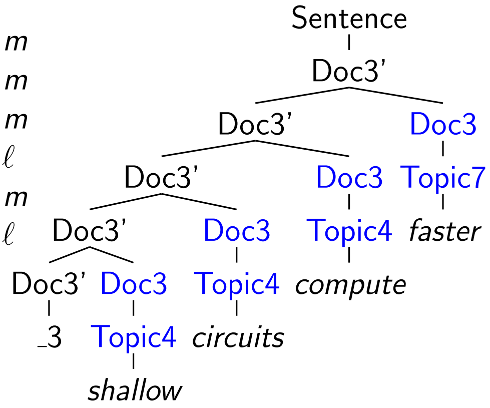
$$
\begin{array}{lll}
\text{Sentence} \rightarrow & \text{Doc}'_j & j \in 1, \ldots, m \\
\text{Doc}'_j \rightarrow & \_j & j \in 1, \ldots, m \\
\text{Doc}'_j \rightarrow & \text{Doc}'_j \ \text{Doc}_j & j \in 1, \ldots, m \\
\text{Doc}_j \rightarrow & \text{Topic}_i & i \in 1, \ldots, \ell \\
 & & j \in 1, \ldots, m \\
\text{Topic}_i \rightarrow & w & i \in 1, \ldots, \ell \\
 & & w \in \mathcal{W}
\end{array}
$$



Admixture topic models as PCFGs (3)

- $\mathrm{Doc}_j \rightarrow \mathrm{Topic}_i$ rules nondeterministically map *documents to topics*

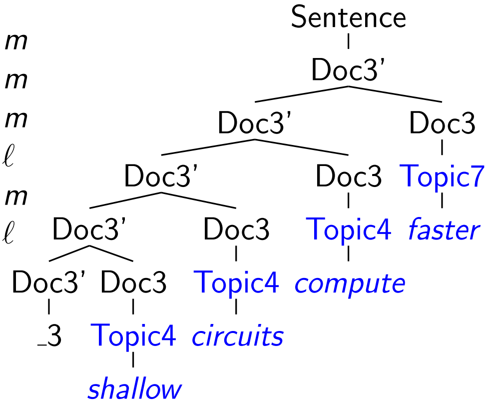
$$
\begin{array}{lll}
\mathrm{Sentence} \rightarrow & \mathrm{Doc}'_j & j \in 1, \ldots, m \\
\mathrm{Doc}'_j \rightarrow & \_j & j \in 1, \ldots, m \\
\mathrm{Doc}'_j \rightarrow & \mathrm{Doc}'_j \ \mathrm{Doc}_j & j \in 1, \ldots, m \\
\mathrm{Doc}_j \rightarrow & \mathrm{Topic}_i & i \in 1, \ldots, \ell \\
 & & j \in 1, \ldots, m \\
\mathrm{Topic}_i \rightarrow & w & i \in 1, \ldots, \ell \\
 & & w \in \mathcal{W}
\end{array}
$$



Admixture topic models as PCFGs (4)

- $\text{Topic}_i \rightarrow\ w$ rules nondeterministically map *topics to words*

$$
\begin{array}{lll}
\text{Sentence} \rightarrow \text{Doc}'_j & & j \in 1, \ldots, m \\
\text{Doc}'_j \rightarrow \_j & & j \in 1, \ldots, m \\
\text{Doc}'_j \rightarrow \text{Doc}'_j\ \text{Doc}_j & & j \in 1, \ldots, m \\
\text{Doc}_j \rightarrow \text{Topic}_i & & i \in 1, \ldots, \ell \\
 & & j \in 1, \ldots, m \\
\text{Topic}_i \rightarrow w & & i \in 1, \ldots, \ell \\
 & & w \in \mathcal{W}
\end{array}
$$



Why are these reductions interesting?

- *Not* claiming that topic modelling should be done using PCFGs
  - PCFG parsing takes time proportional to *cube* of document length
  - standard topic model algorithms take time *linear* in document length
- The PCFG reductions suggest *new kinds of models that merge grammars and topic models*
  - easily implemented and evaluated (on small corpora at least)
- Grammars are good at:
  - grouping words into hierarchically-structured larger units
  - tracking relative ordering of these units

Outline

Bayesian nonparametrics for learning rules

- PCFGs are products of multinomials
  - each rule expansion is a draw from a multinomial (roll of a die)
- Dirichlet Processes extend multinomials to *an unbounded number of outcomes*
  - Chinese Restaurant Processes (CRP) are the predictive distributions associated with Dirichlet Processes (needed to implement MCMC algorithms)
- Provides a framework for *learning the rules* as well as their probabilities
  - specify a generative process for possible rules
  - CRP sampler *learns the useful rules* and their probabilities
- In an adaptor grammar, the possible rules are *subtrees generated by a base PCFG*

MACQUARIE UNIVERSITY

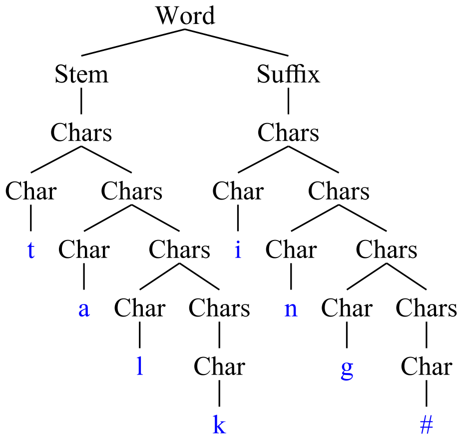
Adaptor grammars: informal description

- The trees generated by an adaptor grammar are defined by CFG rules as in a CFG
- A subset of the nonterminals are *adapted*
- *Unadapted nonterminals* expand by picking a rule and recursively expanding its children, as in a PCFG
- *Adapted nonterminals* can expand in two ways:
  - by picking a rule and recursively expanding its children, or
  - by generating a previously generated tree (with probability proportional to the number of times previously generated)
- Implemented by having a CRP for each adapted nonterminal
- The CFG rules of the adapted nonterminals determine the *base distributions* of these CRPs

MACQUARIE UNIVERSITY

A CFG for stem-suffix morphology

Word	→	Stem Suffix	Chars	→	Char
Stem	→	Chars	Chars	→	Char Chars
Suffix	→	Chars	Char	→	a | b | c | . . .



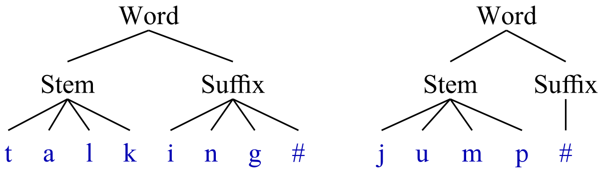
- Grammar's trees can represent any segmentation of words into stems and suffixes

⇒ Can *represent* true segmentation

- But grammar's *units of generalization (PCFG rules) are "too small"* to learn morphemes

A “CFG” with one rule per possible morpheme

Word	$\rightarrow$	Stem Suffix
Stem	$\rightarrow$	*all possible stems*
Suffix	$\rightarrow$	*all possible suffixes*



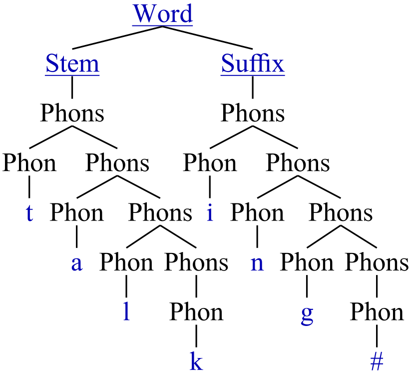
- A rule for each morpheme
  $\Rightarrow$ “PCFG” can represent probability of each morpheme
- *Unbounded number of possible rules, so this is not a PCFG*
  - not a practical problem, as only a finite set of rules could possibly be used in any particular data set

From PCFGs to Adaptor grammars

- An adaptor grammar is a PCFG where a subset of the nonterminals are *adapted*
- **Adaptor grammar generative process:**
  - to expand an *unadapted nonterminal* B: (just as in PCFG)
    - select a *rule* $B \to \beta \in R$ with prob. $\theta_{B \to \beta}$, and recursively expand nonterminals in β
  - to expand an *adapted nonterminal* B:
    - select a *previously generated subtree* T_B with prob. $\propto$ number of times T_B was generated, or
    - select a *rule* $B \to \beta \in R$ with prob. $\propto \alpha_B \, \theta_{B \to \beta}$, and recursively expand nonterminals in β

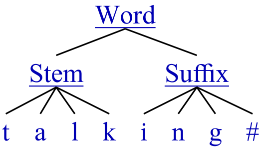
Adaptor grammar for stem-suffix morphology

Word → Stem Suffix
Stem → Phons
Suffix → Phons
Phons → Phon
Phons → Phon Phons



or in *abbreviated form* with
non-adapted nonterminals suppressed

Word → Stem Suffix
Stem → Phon$^+$
Suffix → Phon$^+$



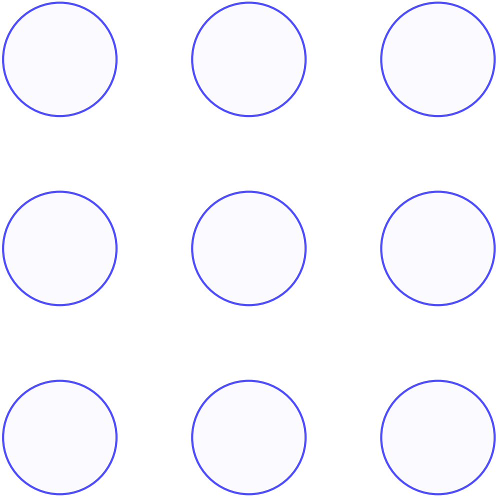
MACQUARIE UNIVERSITY

Adaptor grammar for stem-suffix morphology (0)

Word $\rightarrow$ Stem Suffix

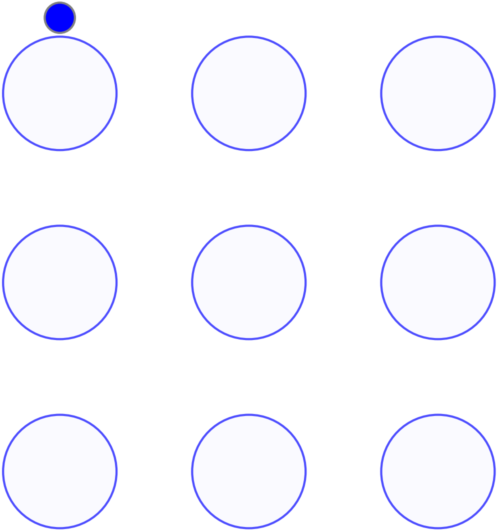
Stem $\rightarrow$ Phoneme^{+}

Suffix $\rightarrow$ Phoneme*



Generated words:

Adaptor grammar for stem-suffix morphology (1a)



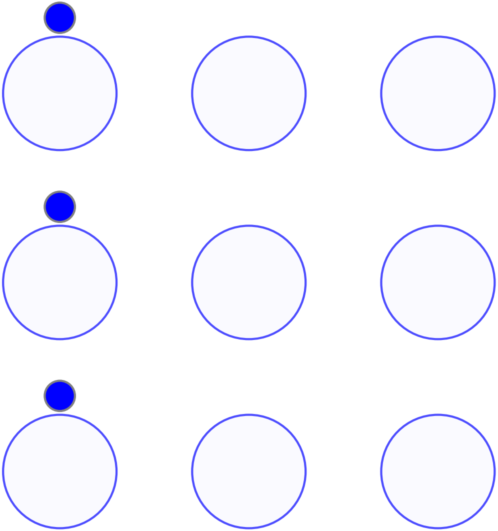
Word $\rightarrow$ Stem Suffix

Stem $\rightarrow$ Phoneme^{+}

Suffix $\rightarrow$ Phoneme*

Generated words:

Adaptor grammar for stem-suffix morphology (1b)



Word $\rightarrow$ Stem Suffix

Stem $\rightarrow$ Phoneme^{+}

Suffix $\rightarrow$ Phoneme*

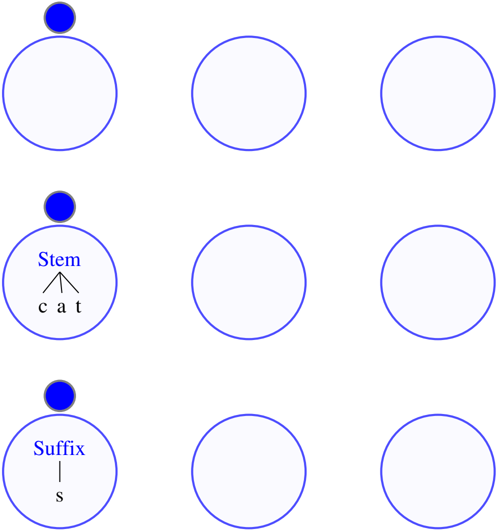
Generated words:

Adaptor grammar for stem-suffix morphology (1c)

Word $\rightarrow$ Stem Suffix

Stem $\rightarrow$ Phoneme^{+}

Suffix $\rightarrow$ Phoneme*



Generated words:

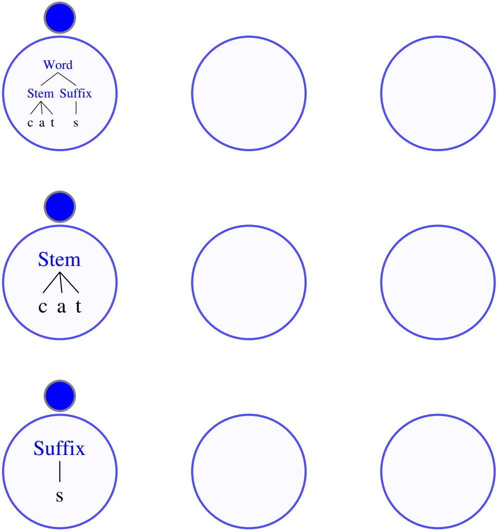
MACQUARIE UNIVERSITY

Adaptor grammar for stem-suffix morphology (1d)

Word $\rightarrow$ Stem Suffix

Stem $\rightarrow$ Phoneme^{+}

Suffix $\rightarrow$ Phoneme*



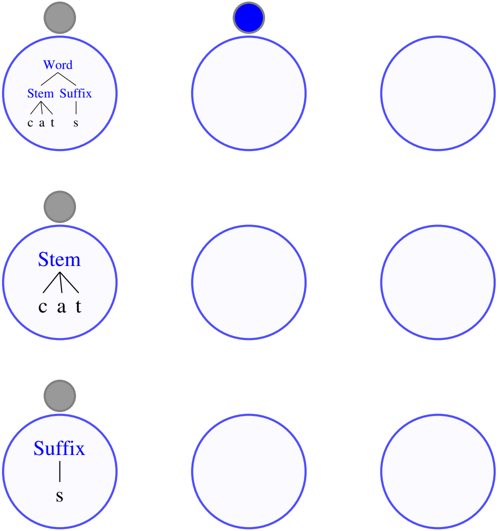
Generated words: cats

Adaptor grammar for stem-suffix morphology (2a)

Word → Stem Suffix

Stem → Phoneme$^+$

Suffix → Phoneme*



Generated words: cats

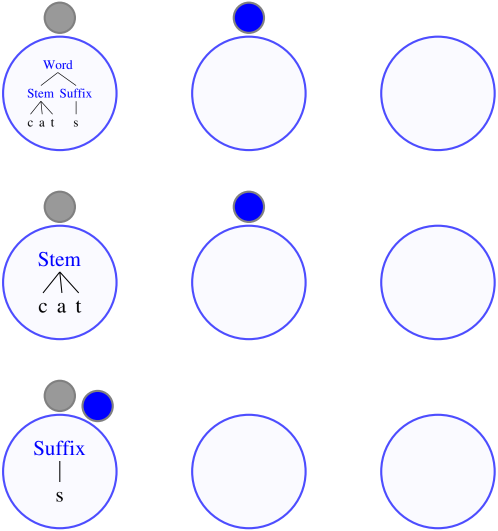
MACQUARIE UNIVERSITY

Adaptor grammar for stem-suffix morphology (2b)

Word $\rightarrow$ Stem Suffix

Stem $\rightarrow$ Phoneme^{+}

Suffix $\rightarrow$ Phoneme*



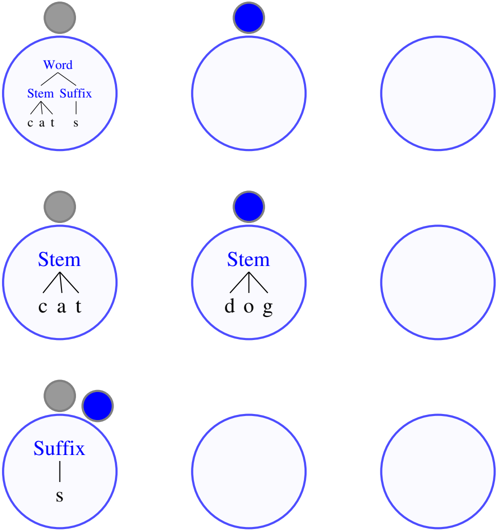
Generated words: cats

Adaptor grammar for stem-suffix morphology (2c)

Word → Stem Suffix

Stem → Phoneme^{+}

Suffix → Phoneme*



Generated words: cats

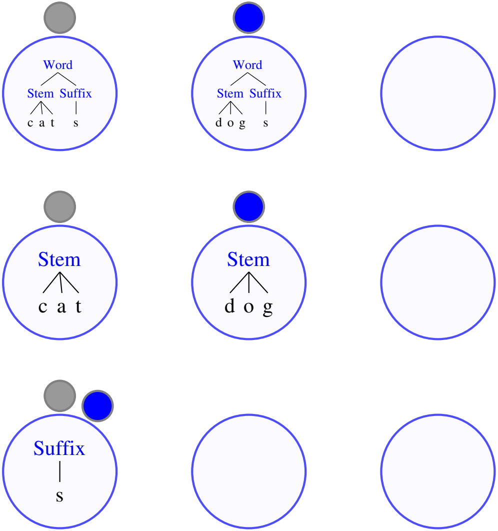
MACQUARIE UNIVERSITY

Adaptor grammar for stem-suffix morphology (2d)

Word $\rightarrow$ Stem Suffix

Stem $\rightarrow$ Phoneme^{+}

Suffix $\rightarrow$ Phoneme*



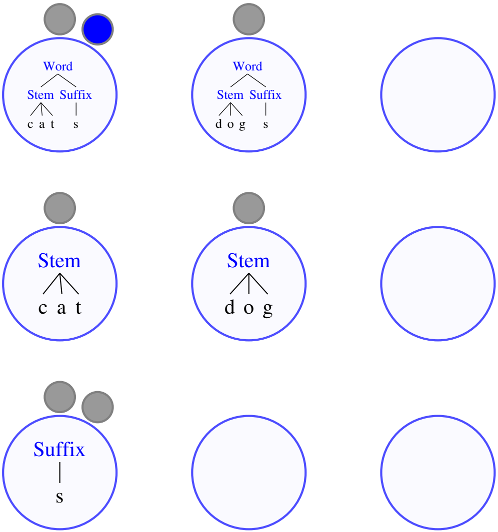
Generated words: cats, dogs

Adaptor grammar for stem-suffix morphology (3)

Word $\rightarrow$ Stem Suffix

Stem $\rightarrow$ Phoneme^{+}

Suffix $\rightarrow$ Phoneme*



Generated words: cats, dogs, cats

MACQUARIE UNIVERSITY

Posterior samples from adaptor grammar

$\alpha = 0.1$		$\alpha = 10^{-5}$		$\alpha = 10^{-10}$		$\alpha = 10^{-15}$	
expect		expect		expect		exp	ect
expects		expect	s	expect	s	exp	ects
expected		expect	ed	expect	ed	exp	ected
expect	ing	expect	ing	expect	ing	exp	ecting
include		includ	e	includ	e	includ	e
include	s	includ	es	includ	es	includ	es
included		includ	ed	includ	ed	includ	ed
including		includ	ing	includ	ing	includ	ing
add		add		add		add	
adds		add	s	add	s	add	s
add	ed	add	ed	add	ed	add	ed
adding		add	ing	add	ing	add	ing
continue		continu	e	continu	e	continu	e
continue	s	continu	es	continu	es	continu	es
continu	ed	continu	ed	continu	ed	continu	ed
continuing		continu	ing	continu	ing	continu	ing
report		report		repo	rt	rep	ort

Adaptor grammars as generative processes

- The sequence of trees generated by an adaptor grammar are *not* independent
  - it *learns* from the trees it generates
  - if an adapted subtree has been used frequently in the past, it's more likely to be used again
- but the sequence of trees is *exchangable* (important for sampling)
- An *unadapted nonterminal* A expands using $A \to \beta$ with probability $\theta_{A\to\beta}$
- Each adapted nonterminal A is associated with a CRP (or PYP) that caches previously generated subtrees rooted in A
- An *adapted nonterminal* A expands:
  - to a subtree T_A rooted in A with probability proportional to the number of times T_A was previously generated
  - using $A \to \beta$ with probability proportional to $\alpha_A \theta_{A\to\beta}$

Adaptor grammars as non-parametric PCFGs

- An adaptor grammar *reuses whole previously-generated subtrees* T_A of adapted nonterminals A
- This is equivalent to *adding a rule* $A \rightarrow w$ to the grammar, where w is the yield of T_A
- If the base CFG generates an *infinite number of trees* T_A for A, then the adaptor grammar is *non-parametric*
- But any set of sample parses for a *finite training corpus* only contains a *finite number of number of adapted subtrees*

  $\Rightarrow$ *sampling methods* (e.g., MCMC) are a natural approach to learning and parsing adaptor grammars
  - in implementation terms, an adaptor grammar is like a PCFG with a *constantly changing set of rules*

MACQUARIE UNIVERSITY

Outline

Probabilistic Context-Free Grammars

Topic models as PCFGs

Adaptor grammars

Learning word pronunciations

Finding topical collocations with adaptor grammars

Project report on Wray's and my project

Conclusions and future work

MACQUARIE UNIVERSITY

Unsupervised word segmentation

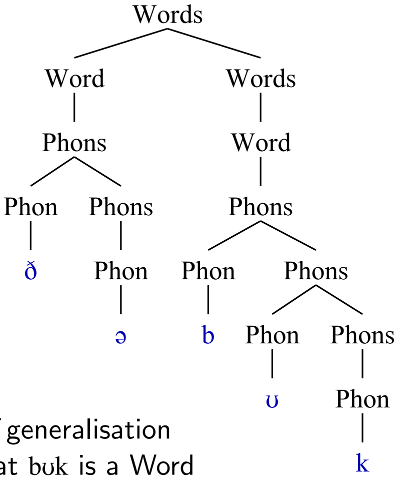
- Input: phoneme sequences with *sentence boundaries* (Brent)
- Task: identify *word boundaries*, and hence words

j △ u ▲ w △ ɑ △ n △ t ▲ t △ u ▲ s △ i ▲ ð △ ə ▲ b △ ʊ △ k
"you want to see the book"

- Ignoring phonology and morphology, this involves learning the pronunciations of the lexical items in the language

CFG models of word segmentation

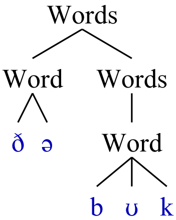
Words $\rightarrow$ Word
Words $\rightarrow$ Word Words
Word $\rightarrow$ Phons
Phons $\rightarrow$ Phon
Phons $\rightarrow$ Phon Phons
Phon $\rightarrow$ $a \mid b \mid \ldots$



- CFG trees can *describe* segmentation, but
- PCFGs *can't distinguish* good segmentations from bad ones
  - PCFG rules are *too small* a unit of generalisation
  - need to learn e.g., probability that bʊk is a Word

Towards non-parametric grammars

Words → Word
Words → Word Words
Word → *all possible phoneme sequences*

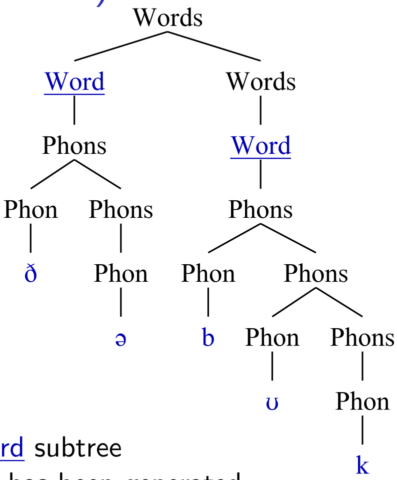


- Learn probability Word → b ʊ k
- But *infinitely many possible Word expansions*
  ⇒ this grammar is *not a PCFG*
- Given *fixed training data*, only finitely many useful rules
  ⇒ use data to choose Word rules as well as their probabilities
- An adaptor grammar can do precisely this!

MACQUARIE UNIVERSITY

Unigram adaptor grammar (Brent)

Words → Word
Words → Word Words
Word → Phons
Phons → Phon
Phons → Phon Phons



- Word nonterminal is adapted

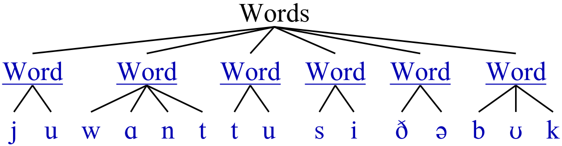
⇒ To generate a Word:
  - select a previously generated Word subtree with prob. $\propto$ number of times it has been generated
  - expand using Word → Phons rule with prob. $\propto \alpha_{\text{Word}}$ and recursively expand Phons

MACQUARIE UNIVERSITY

Unigram model of word segmentation

- Unigram “bag of words” model (Brent):
    - generate a *dictionary*, i.e., a set of words, where each word is a random sequence of phonemes
        - Bayesian prior prefers smaller dictionaries
    - generate each utterance by choosing each word at random from dictionary
- Brent’s unigram model as an adaptor grammar:

$$\text{Words} \rightarrow \text{Word}^{+}$$
$$\underline{\text{Word}} \rightarrow \text{Phoneme}^{+}$$



- Accuracy of word segmentation learnt: *56% token f-score* (same as Brent model)
- But we can construct many more word segmentation models using AGs

Adaptor grammar learnt from Brent corpus

- **Initial grammar**

1	Words → <u>Word</u> Words	1	Words → <u>Word</u>
1	<u>Word</u> → Phon		
1	Phons → Phon Phons	1	Phons → Phon
1	Phon → *D*	1	Phon → *G*
1	Phon → *A*	1	Phon → *E*

- **A grammar learnt from Brent corpus**

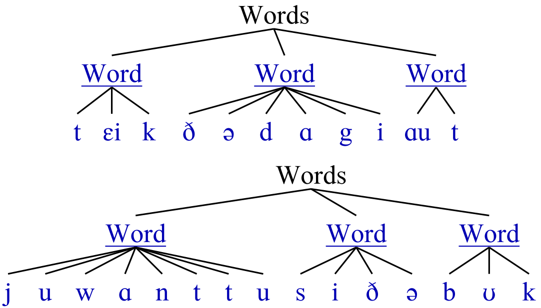
16625	Words → <u>Word</u> Words	9791	Words → <u>Word</u>
1575	<u>Word</u> → Phons		
4962	Phons → Phon Phons	1575	Phons → Phon
134	Phon → *D*	41	Phon → *G*
180	Phon → *A*	152	Phon → *E*
460	<u>Word</u> → (Phons (Phon *y*) (Phons (Phon *u*)))		
446	<u>Word</u> → (Phons (Phon *w*) (Phons (Phon *A*) (Phons (Phon *t*)))		
374	<u>Word</u> → (Phons (Phon *D*) (Phons (Phon *6*)))		
372	<u>Word</u> → (Phons (Phon *&*) (Phons (Phon *n*) (Phons (Phon *d*)))		

MACQUARIE UNIVERSITY

Undersegmentation errors with Unigram model

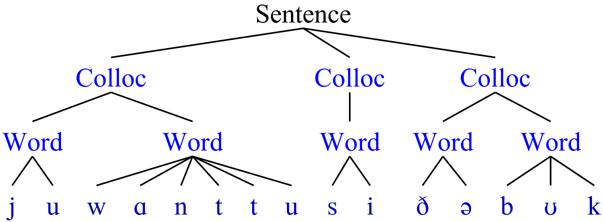
Words → <u>Word</u>$^+$ <u>Word</u> → Phon$^+$

- Unigram word segmentation model assumes each word is generated independently
- But there are strong inter-word dependencies (collocations)
- Unigram model can only capture such dependencies by analyzing collocations as words (Goldwater 2006)



Collocations ⇒ Words

$$\text{Sentence} \rightarrow \text{Colloc}^{+}$$
$$\underline{\text{Colloc}} \rightarrow \text{Word}^{+}$$
$$\underline{\text{Word}} \rightarrow \text{Phon}^{+}$$



- A Colloc(ation) consists of one or more words
- Both Words and Collocs are adapted (learnt)
- Significantly improves word segmentation accuracy over unigram model (76% f-score; ≈ Goldwater's bigram model)

MACQUARIE UNIVERSITY

More complex adaptor grammar models of word segmentation

- Because adaptor grammar models generalise PCFGs, we can combine the topic model grammars and word segmentation grammars
  - topical information does improve word segmentation
  - social cues do not improve word segmentation (as far as we can tell)
- We can learn the internal structure of words too
  - words are a sequence of syllables
  - learn syllable structure jointly with word segmentation
  - we can learn different structures for word-peripheral and word-internal syllables

  $\Rightarrow$ the best reported accuracy for unsupervised word segmentation (87% f-score)

Outline

Probabilistic Context-Free Grammars

Topic models as PCFGs

Adaptor grammars

Learning word pronunciations

Finding topical collocations with adaptor grammars

Project report on Wray's and my project

Conclusions and future work

Topical collocation models

Annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. Getting a dataset labeled by experts can be expensive and time consuming. With the advent of crowdsourcing services ...

The task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading, shadows, specular highlights, ...

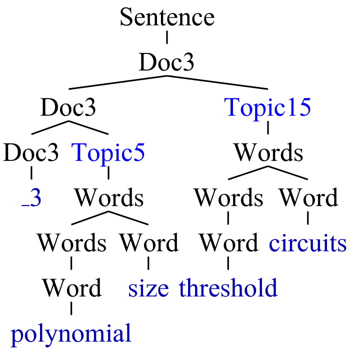
In each trial of a standard visual short-term memory experiment, subjects are first presented with a display containing multiple items with simple features (e.g. colored squares) for a brief duration and then, after a delay interval, their memory for ...

Many studies have uncovered evidence that visual cortex contains specialized regions involved in processing faces but not other object classes. Recent electrophysiology studies of cells in several of these specialized regions revealed that at least some ...

Topic model with collocations

- Combines *PCFG for admixture topic model* and *segmentation adaptor grammar*

$\text{Sentence} \rightarrow \text{Doc}_j$	$j \in 1, \ldots, m$
$\text{Doc}_j \rightarrow \_j$	$j \in 1, \ldots, m$
$\text{Doc}_j \rightarrow \text{Doc}_j\ \text{Topic}_i$	$i \in 1, \ldots, \ell;$
	$j \in 1, \ldots, m$
$\underline{\text{Topic}_i} \rightarrow \text{Words}$	$i \in 1, \ldots, \ell$
$\text{Words} \rightarrow \text{Word}$	
$\text{Words} \rightarrow \text{Words Word}$	
$\text{Word} \rightarrow w$	$w \in \mathcal{W}$



Data preparation in Griffiths et al (2007)

- Documents are papers from NIPS proceedings ($\sim$ 3 million words)
- Case normalised
- Segmented at *punctuation* and *function words*

annotating an unlabeled dataset is one of the bottlenecks in using supervised learning to build good predictive models. getting a dataset labeled by experts can be expensive and time consuming. with the advent of crowdsourcing services ...

---

the task of recovering intrinsic images is to separate a given input image into its material-dependent properties, known as reflectance or albedo, and its light-dependent properties, such as shading , shadows , specular highlights , ...

MACQUARIE UNIVERSITY

Finding topical collocations in NIPS abstracts

- Run topical collocation adaptor grammar on NIPS corpus
- Run with $\ell = 20$ topics (i.e., 20 distinct Topic_i nonterminals)
- Corpus is segmented by punctuation
  - terminal strings are fairly short
  - $\Rightarrow$ inference is fairly efficient
- Used Pitman-Yor adaptors
  - sampled Pitman-Yor a and b parameters
  - flat and “vague Gamma” priors on Pitman-Yor a and b parameters
- See Griffiths et al (2007) for an alternative topical collocation model, and Johnson and Goldwater (2009) for details on inference

Sample output on NIPS corpus, 20 topics

- Multiword subtrees learned by adaptor grammar:

  T_0 → gradient descent
  T_0 → cost function
  T_0 → fixed point
  T_0 → learning rates
  T_3 → membrane potential
  T_3 → action potentials
  T_3 → visual system
  T_3 → primary visual cortex
  T_1 → associative memory
  T_1 → standard deviation
  T_1 → randomly chosen
  T_1 → hamming distance
  T_10 → ocular dominance
  T_10 → visual field
  T_10 → nervous system
  T_10 → action potential

- Sample skeletal parses:

  _3 (T_5 polynomial size) (T_15 threshold circuits)
  _4 (T_11 studied) (T_19 pattern recognition algorithms)
  _4 (T_2 feedforward neural network) (T_1 implements)
  _5 (T_11 single) (T_10 ocular dominance stripe) (T_12 low)
  (T_3 ocularity) (T_12 drift rate)

Some collocations found in NIPS corpus

Count	Topic	Collocation
2	T0	unites states israeli binational science foundation bsf
2	T5	batch k-means empty circles online gradient
12	T1	partially observable markov decision processes
12	T2	defense advanced research projects agency
7	T5	radial basis function rbf network
5	T6	analog vlsi neural network chip
4	T12	national science foundation graduate fellowship
3	T10	globally optimal on-line learning rules
3	T12	radial basis function rbf units
3	T13	non-parametric multi-scale statistical image model
3	T15	weight vector estimate requires knowledge
3	T17	orientation bands intersect ocular dominance
3	T18	optimal brain damage le cun
3	T6	normalized mean squared prediction error
47	T5	markov chain monte carlo
43	T12	radial basis function rbf
41	T12	radial basis function networks
39	T7	independent component analysis ica
35	T11	principal component analysis pca

MACQUARIE UNIVERSITY

Some collocations found in NIPS corpus (cont.)

Count	Topic	Collocation
17	T11	principal components analysis pca
16	T11	hidden markov models hmm
14	T18	artificial neural network ann
13	T15	optimal brain damage obd
12	T4	kanerva sparse distributed memory
11	T14	hybrid monte carlo method
11	T19	artificial neural networks ann
10	T0	mean square error mse
10	T12	radial basis functions rbfs
10	T16	markov decision process pomdp
10	T11	hidden markov model hmm
10	T3	atr human information processing
10	T18	artificial neural networks anns
10	T9	spin spin correlation function
9	T2	naive mean field approximation
9	T0	mean squared error mse
9	T7	support vector machines svms
9	T8	owl sound localization system
8	T1	compatible lateral bipolar transistors

Application: “perspective” and sentiment analysis

- Hardisty et al (2010) use a topical collocation model in a “perspective” sentiment analysis
- Data: the *Bitter Lemons* corpus
  essays on mid-East issues from Israeli and Palestinian perspectives
- Supervised training: training sentences belong to one of two “super documents”
  - learns distributions over topics associated with each perspective
  - can be viewed as a “semi-supervised” approach
- Label test documents by finding “super document” most likely to generate them
- Compared a number of other supervised and semi-supervised methods (including SVMs, other collocation-based approaches)
- Found that *adaptor grammar topical collocations (with a hierarchical topic structure) performed best of all*

Outline

MACQUARIE UNIVERSITY

Project aims

- Are the topical collocations found by our model:
  - better than those found by other topical collocation procedures?
  - better than finding collocations separately and retokenising?
- There are several different adaptor grammars for topical collocations: which one works best?
- The adaptor grammar inference procedure relies on a general-purpose PCFG parsing procedure: can we find a faster inference procedure for topical collocations?

Evaluating topical collocation models

- Standard evaluation procedures for topic models:
  - *Perplexity:* how well does the model predict held-out data
  - *Information retrieval:* evaluate models by how well they score the similarity between a query and documents in an information-retrieval task
  - *Human evaluation:* can humans spot the "intruder" in a list of topical words and collocations?
- Subtask: find a proxy measure that approximates the human evaluation results (useful for selecting between and tuning models)
- We are about to begin human evaluation using Mechanical Turk

Speeding topical collocation model inference

- Current adaptor grammar models require repeatedly reparsing the input
  $\Rightarrow$ slow on multi-million word collections
- Take advantage of recent work on speeding (single-word) topic model inference
  - parallel point-wise sampling algorithms
  - variational Bayesian approximations
- We have generalised these algorithms to apply to topical collocation models, hopefully yielding a significant speed-up

Accomplishments so far

- NAACL 2013 paper accepted "Topic Segmentation with a Structured Topic Model"
  - segments documents (e.g., meeting transcripts) into topically-coherent units
  - generalises the word segmentation problem (replace "words" with "document subsection")
  - sampling algorithm for finding topically-coherent unit boundaries generalises Goldwater et al word boundary sampling algorithm
  - key technical challenge is finding methods for "splitting" and "merging" topic models as sampler introduces and removes unit boundaries

Outline

Conclusions

- Although PCFGs are generally thought of as methods for syntactic analysis, they can be used to model a variety of other phenomena as well
  - both mixture and admixture topic models can be expressed as PCFGs
- Adaptor grammars can express a variety of useful models
  - unsupervised models of word learning
  - finding topical collocations
  - generic AG inference code makes it easy to compare and explore a variety of models
- These models and associated inference techniques can be generalised to new kinds of models

Future work: modelling "life stories"

- A person's *life story* is the sequence of events that occur to them
  - Life stories are a mixture of one or more *careers*
  - A career consists of a sequence of *events*
- This can be regarded as *generalised topic model*:

Topic model	**Life story model**
words	events and properties
documents	life stories
topics	*careers*

MACQUARIE UNIVERSITY

Life story models for entity linking

- Query: "What did Jim Jones do before his recent hit song?"
- Wikipedia lists eight different Jim Jones:
  - two are politicians
  - two are sportsmen
  - one is a judge
  - one is a cult leader
  - one is a rapper
  - one is a guitarist
  - three of them are dead (including the guitarist)
- Which entry would you look at?

Hierarchical Bayesian models for careers

- Everyone's life story is different, but there are important commonalities:
  - everyone dies at most 110 years after they are born
  - not everyone goes to university, but if they do, they go *after* they've been to high school
  - politicians run an election campaign *before* they win an election
  - releasing a music CD is often associated with a release party, a tour, reviews, etc.
- A career is a temporally-ordered cluster of events intended to capture the shared structure of life stories
- Aim: *learn a "grammar" of careers*
- Use hierarchical Bayesian models to share information across careers

MACQUARIE UNIVERSITY

Life stories as admixtures of careers

- Bill Clinton's life story is primarily that of a successful politician, but it contains events from a musician career
  - $\Rightarrow$ a life story is an *admixture model* of one or more careers
- We want to capture correlations between careers:
  - a lawyer is much more likely than a carpenter to become a politician
  - an academic is more likely than a plumber to become an author
  - a singer is more likely than a mechanic to become a movie star

Learning and using life story models

- Freebase is a structured database built from Wikipedia
- We intend to mine Freebase for life stories to train our Bayesian models
- We will apply the life story models to improve entity linking in free text documents (e.g., newswire)

- We submitted a proposal to develop life story models to Google Research in April