Unsupervised word segmentation for Sesotho using Adaptor Grammars

Mark Johnson
Brown University

SIGMORPHON workshop
June 2008

based on joint work with
Sharon Goldwater and Tom Griffiths

Outline

Motivations for this work

- Can non-parametric hierarchical Bayesian models help us understand language acquisition?
  - *Adaptor grammars* are a framework for easily constructing these models
- How useful are various potential information sources for language acquisition? (here, word segmentation)
  - Bayesian prior can express Universal Grammar and markedness preferences
  - Different adaptor grammars learn different kinds of generalizations
- *Are the information sources most useful for English also useful for other languages?*

Word segmentation in English

- Task: segment utterances in *broad phonemic representation*
  Example: $y_{\vartriangle}u_{\blacktriangle}w_{\vartriangle}a_{\vartriangle}n_{\vartriangle}t_{\blacktriangle}t_{\vartriangle}u_{\blacktriangle}s_{\vartriangle}i_{\blacktriangle}D_{\vartriangle}6_{\blacktriangle}b_{\vartriangle}U_{\vartriangle}k$
- Previous work has mainly focused on English
  - Brent corpus constructed by looking up transcribed child-directed speech in pronouncing dictionary
  - Goldwater et al (2006) demonstrate importance of interword dependencies
  - Johnson (2008) used *adaptor grammars* to explore a variety of word segmentation models
    - found no improvement simultaneously learning stem-suffix morphology
    - but did find a significant improvement simultaneously learning syllable structure
- *Do these results hold in other languages as well?*
- *Are different kinds of models useful for other languages?*

The Sesotho corpus

- Sesotho is a Bantu language spoken in southern Africa
- Orthography is (roughly) phonemic
  ⇒ use orthographic forms as broad phonemic representations
- Rich agglutinative morphology (especially in verbs)

u- e- nk- il- e	kae
SM-OM-take-PERF-IN	where

  "You took it from where?"
- The Demuth Sesotho corpus (1992) contains transcripts of child and child-directed speech
- Here used a subset of size roughly comparable to Brent corpus of infant-directed speech

	Brent	Demuth
utterances	9,790	8,503
word tokens	33,399	30,200
phonemes	95,809	100,113

Adaptor grammars

- Adaptor grammars are a *non-parametric* Bayesian extension of PCFGs
  - the set of possible trees defined using rules as in a PCFG
  - the units of generalization are the subtrees associated with *adapted nonterminals*
- Each adapted nonterminal A has a *concentration parameter* α_A
- An *unadapted nonterminal* U expands just as in a PCFG
  - to children $V_1 \ldots V_m$ with probability $\theta_{U \to V_1 \ldots V_m}$
- An *adapted nonterminal* A expands:
  - to a previously generated subtree t rooted in A with probability $\propto$ number of times t was previously selected
  - to children $B_1 \ldots B_m$ with probability $\propto \alpha_A \, \theta_{A \to B_1 \ldots B_m}$

Outline

Introduction

Adaptor grammars for English word segmentation

Adaptor grammars for Sesotho word segmentation

Summary and conclusions

Unigram adaptor grammar for English

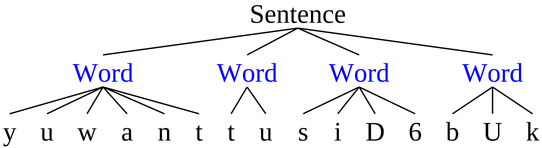
- Adaptor grammar (adapted nonterminals highlighted):

Sentence $\rightarrow$ Words
Words $\rightarrow$ Word
Words $\rightarrow$ Word Words
Word $\rightarrow$ Phonemes
Phonemes $\rightarrow$ Phoneme
Phonemes $\rightarrow$ Phoneme Phonemes

or in abbreviated format:

Sentence $\rightarrow$ Word$^+$
Word $\rightarrow$ Phoneme$^+$

- Sample parse (only showing root and adapted nonterminals):



- Word segmentation f-score = 0.55 (same as Goldwater et al)
- Can't capture dependencies between words
  $\Rightarrow$ tends to undersegment

Unigram word grammar as a Dirichlet Process

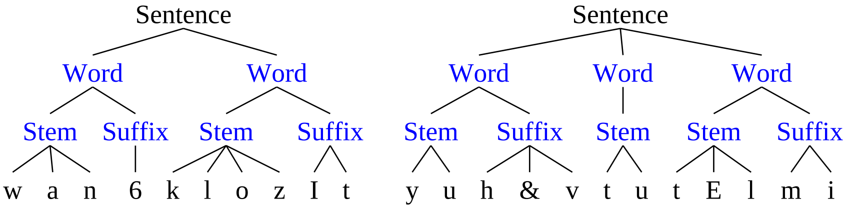
- Unigram word grammar implements unigram word segmentation model of Goldwater et al (2006)
- Generative process:
  - expand Sentence into a sequence of Words using PCFG rules
  - expand each Word into:
    - a sequence of Phonemes with prob. $\propto$ number of times Word expanded to this sequence before
    - a sequence of phonemes generated by PCFG rules with prob. $\propto \alpha_{\text{Word}}$
- This is a *Dirichlet Process* where the PCFG rules expanding Word define the *base distribution*

Unigram morphology adaptor grammar

- Adaptor grammar memorizes Word, Stem and Suffix:

$$
\begin{aligned}
&\text{Sentence} \rightarrow \text{Word}^{+}\\
&\text{Word} \rightarrow \text{Stem (Suffix)}\\
&\text{Stem} \rightarrow \text{Phoneme}^{+}\\
&\text{Suffix} \rightarrow \text{Phoneme}^{+}
\end{aligned}
$$

- Sample parse:



- Combines Goldwater's morphology and unigram model
- Word segmentation f-score = 0.46 (worse than unigram)
- Tends to misanalyse words as Stems or Suffixes

Morphology grammar as a Hierarchical Dirichlet Process

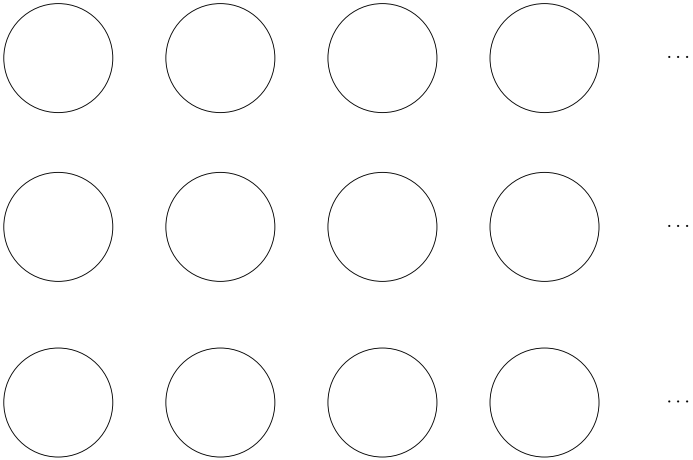
- Expand Sentence into a sequence of Word
- Expand each Word into:
  - a sequence of Phonemes with prob. $\propto$ number of times sequence was generated before
  - a Stem and optional Suffix with prob. $\propto \alpha_{\text{Word}}$
- Expand Stem into:
  - a sequence of Phoneme with prob. $\propto$ number of times Stem expanded to this sequence before
  - a sequence of Phoneme generated by PCFG rules with prob. $\propto \alpha_{\text{Stem}}$
- Suffix expands in same way as Stem
- This is a *Hierarchical Dirichlet Process* where Stem and Suffix distributions define the base distribution for Word DP

Morphology adaptor grammar (0)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme$^+$

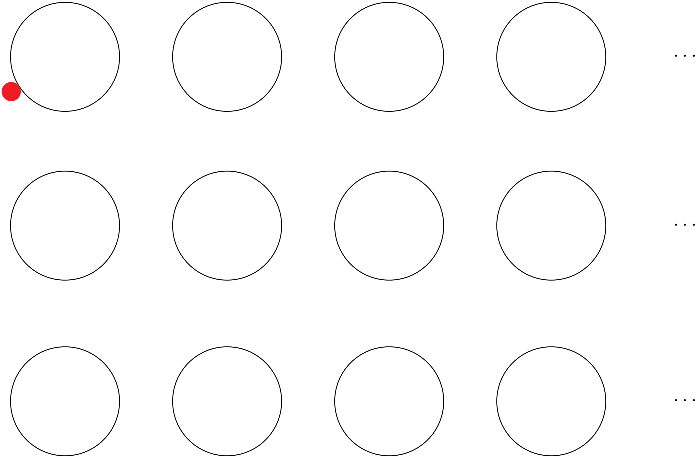


Morphology adaptor grammar (1a)

Word restaurant
Word $\rightarrow$ Stem Suffix

Stem restaurant
Stem $\rightarrow$ Phoneme*

Suffix restaurant
Suffix $\rightarrow$ Phoneme$^+$

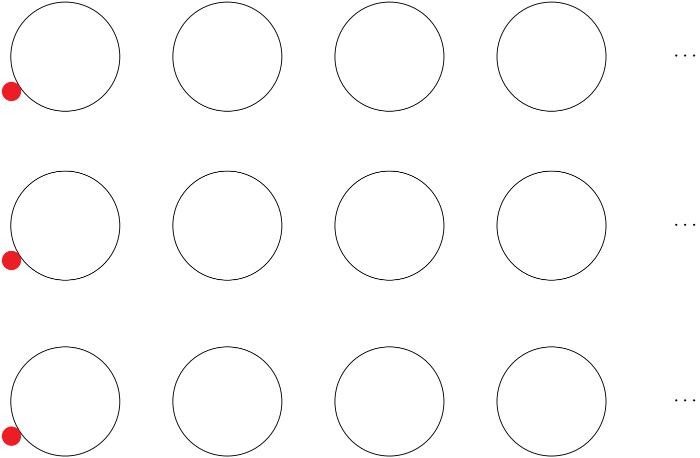


Morphology adaptor grammar (1b)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme$^+$

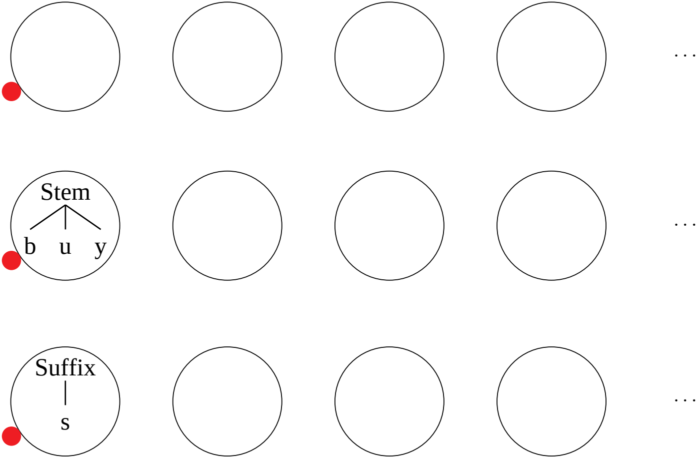


Morphology adaptor grammar (1c)

Word restaurant
Word $\rightarrow$ Stem Suffix

Stem restaurant
Stem $\rightarrow$ Phoneme*

Suffix restaurant
Suffix $\rightarrow$ Phoneme^{+}

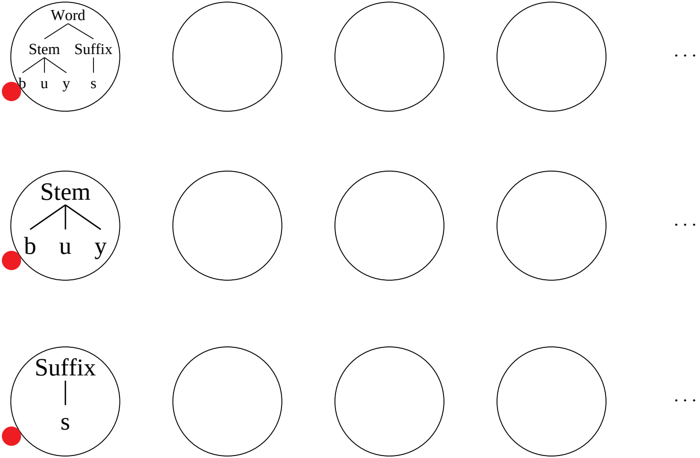


Morphology adaptor grammar (1d)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme$^+$

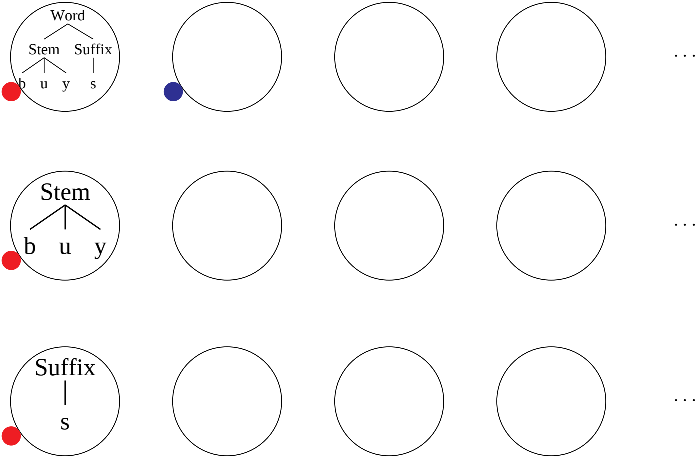


Morphology adaptor grammar (2a)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme^{+}

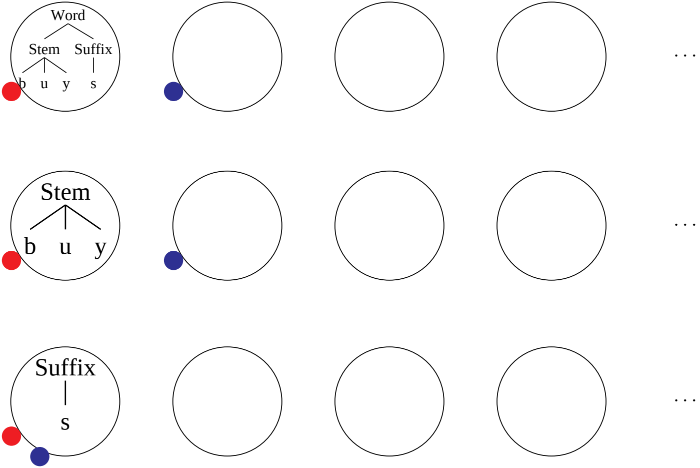


Morphology adaptor grammar (2b)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme$^+$

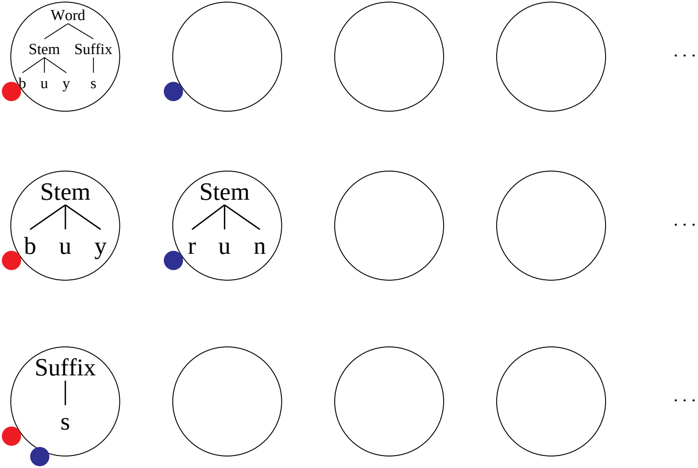


Morphology adaptor grammar (2c)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme^{+}

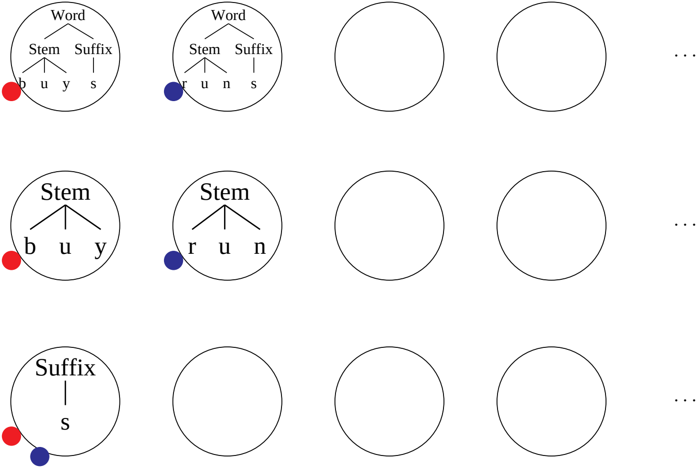


Morphology adaptor grammar (2d)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme$^+$

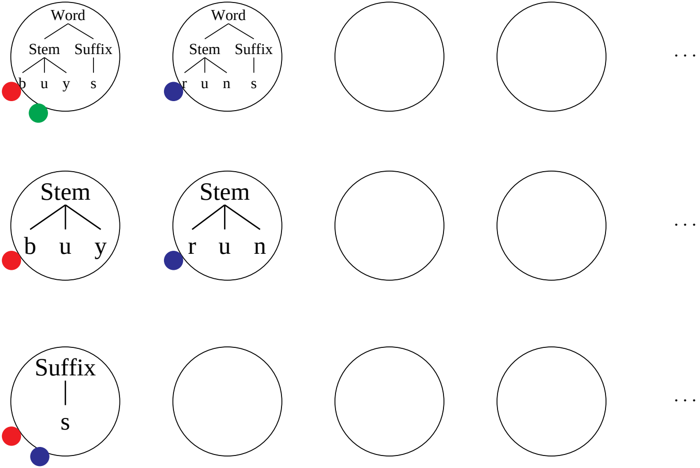


Morphology adaptor grammar (3)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme^{+}

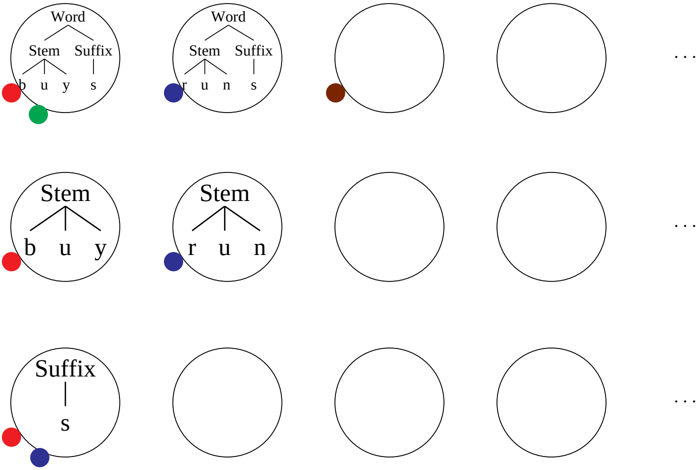


Morphology adaptor grammar (4a)

Word restaurant
Word $\rightarrow$ Stem Suffix

Stem restaurant
Stem $\rightarrow$ Phoneme*

Suffix restaurant
Suffix $\rightarrow$ Phoneme^{+}

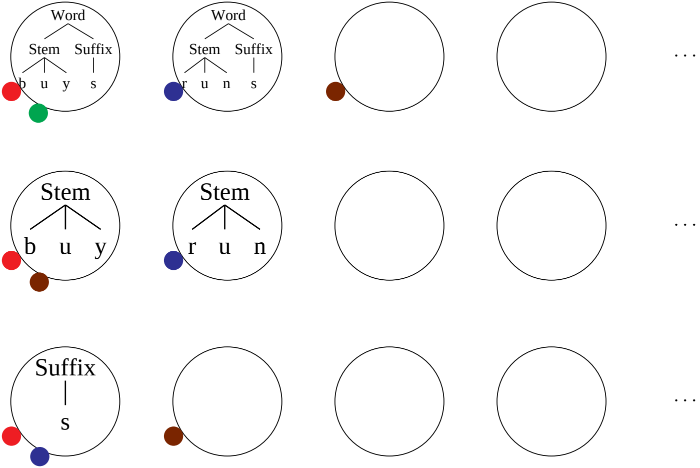


Morphology adaptor grammar (4b)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme$^+$

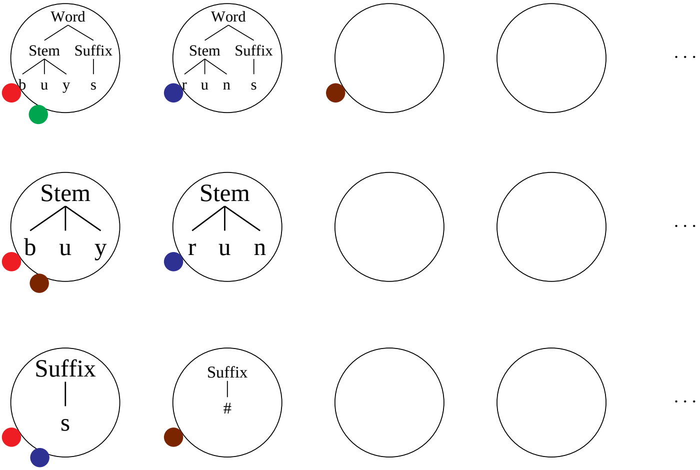


Morphology adaptor grammar (4c)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme^{+}

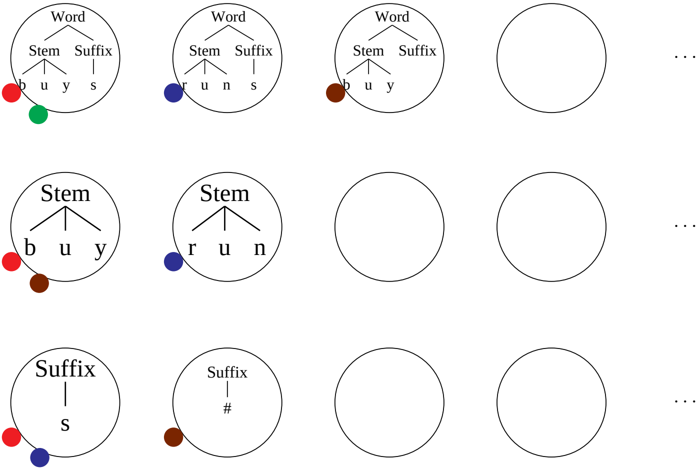


Morphology adaptor grammar (4d)

Word restaurant
Word → Stem Suffix

Stem restaurant
Stem → Phoneme*

Suffix restaurant
Suffix → Phoneme$^+$



Properties of adaptor grammars

- Possible trees generated by CFG rules
  but the probability of each adapted tree is estimated separately
- Probability of a subtree τ is proportional to:
  - the number of times τ was seen before
    $\Rightarrow$ "rich get richer" dynamics (Zipf distributions)
  - plus α_A times prob. of generating it via PCFG expansion

$\Rightarrow$ Useful compound structures can be *more probable than their parts*

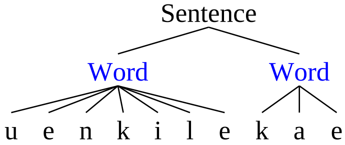
- PCFG rule probabilities estimated *from table labels*
  $\Rightarrow$ learns from types, not tokens
  $\Rightarrow$ dampens frequency variation

Outline

Unigram segmentation grammar – word

u- e- nk- il- e kae
SM-OM-take-PERF-IN where
"You took it from where?"

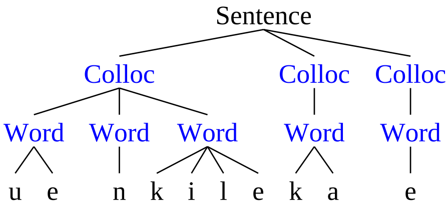
Sentence → Word$^+$
Word → Phoneme$^+$



- The word grammar has a word segmentation f-score of 43%
- Lower than 56% f-score on the Brent corpus.
- Sesotho words are longer and more complex.

Collocation grammar – colloc

$$\text{Sentence} \rightarrow \text{Colloc}^{+}$$
$$\text{Colloc} \rightarrow \text{Word}^{+}$$
$$\text{Word} \rightarrow \text{Phoneme}^{+}$$

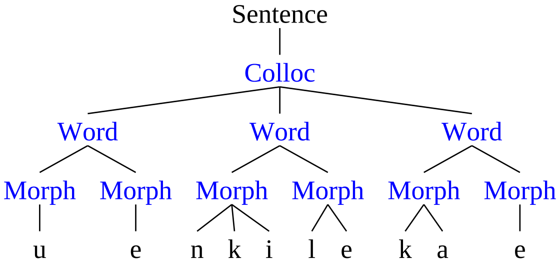


- Goldwater et al (2006) found that modelling bigram dependencies greatly improved English segmentation accuracy
- Johnson (2008) showed similar improvements by learning English collocations
- If we treat lower-level units as Words, f-score = 27%
- If we treat upper-level units as Words, f-score = 48%
- English improves by learning dependencies above words, but Sesotho improves by learning generalizations below words

Adding more levels – colloc2

u- e- nk- il- e kae
SM-OM-take-PERF-IN where
"You took it from where?"

Sentence → Colloc^+
Colloc → Word^+
Word → Morph^+
Morph → Phoneme^+



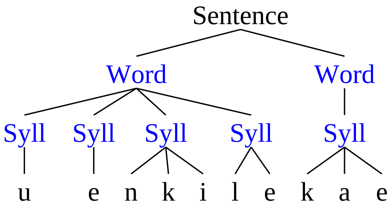
- If two levels are good, maybe three would be better?
- Word segmentation f-score drops to 47%
- Doesn't seem to be much value in adding dependencies above Word level in Sesotho

Using syllable structure – word – syll

u-	e-	nk-	il-	e	kae
SM-OM-take-PERF-IN					where

"You took it from where?"

Sentence → Word^+
Word → Syll^+
Syll → (Onset) Nuc (Coda)
Syll → SC
Onset → C^+
Nuc → V^+
Coda → C^+

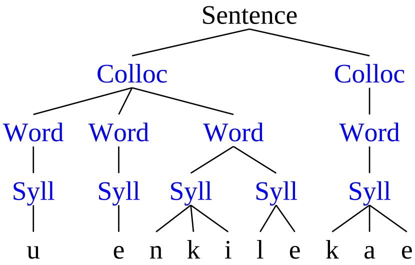


- SC (syllabic consonants) are '*l*', '*m*' '*n*' and '*r*'
- Word segmentation f-score = 50%
- Assuming that words are composed of syllables does improve Sesotho word segmentation

Using syllable structure – colloc – syll

u- e- nk- il- e kae
SM-OM-take-PERF-IN where
"You took it from where?"

$$\text{Sentence} \rightarrow \text{Colloc}^{+}$$
$$\text{Colloc} \rightarrow \text{Word}^{+}$$
$$\text{Syll} \rightarrow (\text{Onset})\,\text{Nuc}\,(\text{Coda})$$
$$\text{Syll} \rightarrow \text{SC}$$
$$\text{Onset} \rightarrow \text{C}^{+}$$
$$\text{Nuc} \rightarrow \text{V}^{+}$$
$$\text{Coda} \rightarrow \text{C}^{+}$$

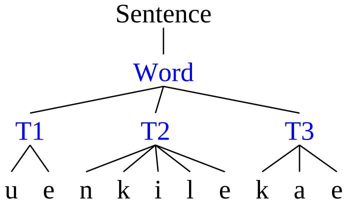


- Word segmentation f-score = 48%
- Additional collocation level doesn't help

Morpheme positions – word – morph

u- e- nk- il- e kae
SM-OM-take-PERF-IN where
"You took it from where?"

Sentence → Word$^+$
Word → T1 (T2 (T3 (T4 (T5))))
T1 → Phoneme$^+$
T2 → Phoneme$^+$
T3 → Phoneme$^+$
T4 → Phoneme$^+$
T5 → Phoneme$^+$



- Each word consists of 1–5 morphemes
- Learn separate morphemes for each position
- Improves word segmentation f-score to 53%

Building in language-specific information – word – smorph

u- e- nk- il- e kae

SM-OM-take-PERF-IN where

"You took it from where?"

Sentence → Word$^+$

Word → (P1 (P2 (P3))) T (S)

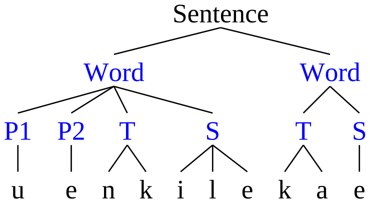
P1 → Phoneme$^+$

P2 → Phoneme$^+$

P3 → Phoneme$^+$

T → Phoneme$^+$

S → Phoneme$^+$



- In Sesotho many words consist of a stem T, an optional suffix S and up to 3 prefixes P1, P2 and P3
- Achieves highest f-score = 56%

Outline

Summary of results

Model	word f-score	morpheme f-score
word	0.431	0.352
colloc	0.478	0.387
colloc2	0.467	0.389
word – syll	0.502	0.349
colloc – syll	0.476	0.372
colloc2 – syll	0.490	0.393
word – morph	0.529	0.321
word – smorph	0.556	0.378
colloc – smorph	0.537	0.352

Conclusions

- The same kinds of models that Goldwater et al (2006) developed for English can be applied to other languages
- Adaptor grammars permit us to easily develop and apply such models
- Learning dependencies above the Word (which are important for English) doesn't seem so important for Sesotho
- Learning dependencies below the Word is much more important for Sesotho
- Building in language-specific information improves word-segmentation f-score
  - is there a suitable "universal grammar" of morphology?

PCFGs as recursive mixtures

For simplicity assume all rules are of the form $A \to B\,C$ or $A \to w$, where $A, B, C \in N$ (nonterminals) and $w \in T$ (terminals).
Each nonterminal $A \in N$ generates a distribution G_A over the trees rooted in A.

$$G_A \;=\; \sum_{A \to B\,C \in R_A} \theta_{A \to B\,C} \textsc{Tree}_A(G_B, G_C) + \sum_{A \to w \in R_A} \theta_{A \to w} \textsc{Tree}_A(w)$$

where $\textsc{Tree}_A(w)$ puts all of its mass on the tree with child w and $\textsc{Tree}_A(P, Q)$ is the distribution over trees rooted in A with children distributed according to P and Q respectively.

$$\textsc{Tree}_A(P, Q)\left(\begin{array}{c} A \\ \diagup \quad \diagdown \\ t_1 \qquad t_2 \end{array}\right) \;=\; P(t_1)\, Q(t_2)$$

The tree language generated by the PCFG is G_S.

Adaptor grammars as recursive mixtures

An adaptor grammar $(G, \boldsymbol{\theta}, \boldsymbol{\alpha})$ is a PCFG $(G, \boldsymbol{\theta})$ together with a parameter vector $\boldsymbol{\alpha}$ where for each $A \in N$, α_A is the parameter of the Dirichlet process associated with A.

$$
\begin{aligned}
G_A & \sim \mathrm{DP}(\alpha_A, H_A) \text{ if } \alpha_A > 0 \\
& = H_A \qquad\qquad\quad\;\; \text{if } \alpha_A = 0
\end{aligned}
$$

$$
H_A = \sum_{A \to B\,C \in R_A} \theta_{A \to B\,C} \textsc{Tree}_A(G_B, G_C) + \sum_{A \to w \in R_A} \theta_{A \to w} \textsc{Tree}_A(w)
$$

The grammar generates the distribution G_S.
There is one Dirichlet Process for each non-terminal A where $\alpha_A > 0$. Its base distribution H_A is a mixture of the language generated by the Dirichlet processes associated with other non-terminals.

Bayesian priors on adaptor grammar parameters

- Parameters of adaptor grammars:
  - probabilities $\theta_{A \to \beta}$ of base grammar rules $A \to \beta$
  - concentration parameters α_A of adapted nonterminals A
- Put Bayesian priors on these parameters
  - (Uniform) Dirichlet prior on base grammar rule probabilities $\boldsymbol{\theta}$
  - Vague Gamma prior on concentration parameter on α_A
- We also use a generalization of CRPs called "Pitman-Yor processes", and put a uniform Dirichlet prior on its a parameter

Estimating adaptor grammars

- Need to estimate:
  - cached subtrees τ for adapted nonterminals
  - (optional) DP parameters $\boldsymbol{\alpha}$ for adapted nonterminals
  - (optional) probabilities $\boldsymbol{\theta}$ of base grammar rules
- Component-wise Metropolis-within-Gibbs sampler
  - components are parse tree T_i for each string W_i
  - sample T_i from $\mathrm{P}(T|W_i, \vec{T}_{-i}, \boldsymbol{\alpha}, \boldsymbol{\theta})$ for each sentence W_i in turn
- Sampling directly from conditional distribution of parses seems intractable
  - construct PCFG proposal grammar $G'(\vec{T}_{-i})$ on the fly
  - each table label τ corresponds to a production in PCFG approximation
  - Use accept/reject to convert samples from PCFG approx to samples from adaptor grammar

PCFG proposal grammar

- Recall that in a CRP,
  - pick old table τ with prob. $\propto n_\tau$ (number of customers seated at τ)
  - pick new table with prob. $\propto \alpha$ (DP concentration parameter)
- Rules of PCFG proposal grammar G' consist of:
  - rules $A \rightarrow \beta$ from base PCFG: $\theta'_{A\rightarrow\beta} \propto \alpha_A \theta_{A\rightarrow\beta}$
  - A rule $A \rightarrow \textsc{Yield}(\tau)$ for each table τ in A's restaurant: $\theta'_{A\rightarrow\textsc{Yield}(\tau)} \propto n_\tau$, the number of customers at table τ
- Parses of G' can be mapped back to adaptor grammar parses