

# How Ideal Are We?

## Incorporating Human Limitations Into Bayesian Models of Word Segmentation

Lisa Pearl★

Sharon Goldwater◆

Mark Steyvers★

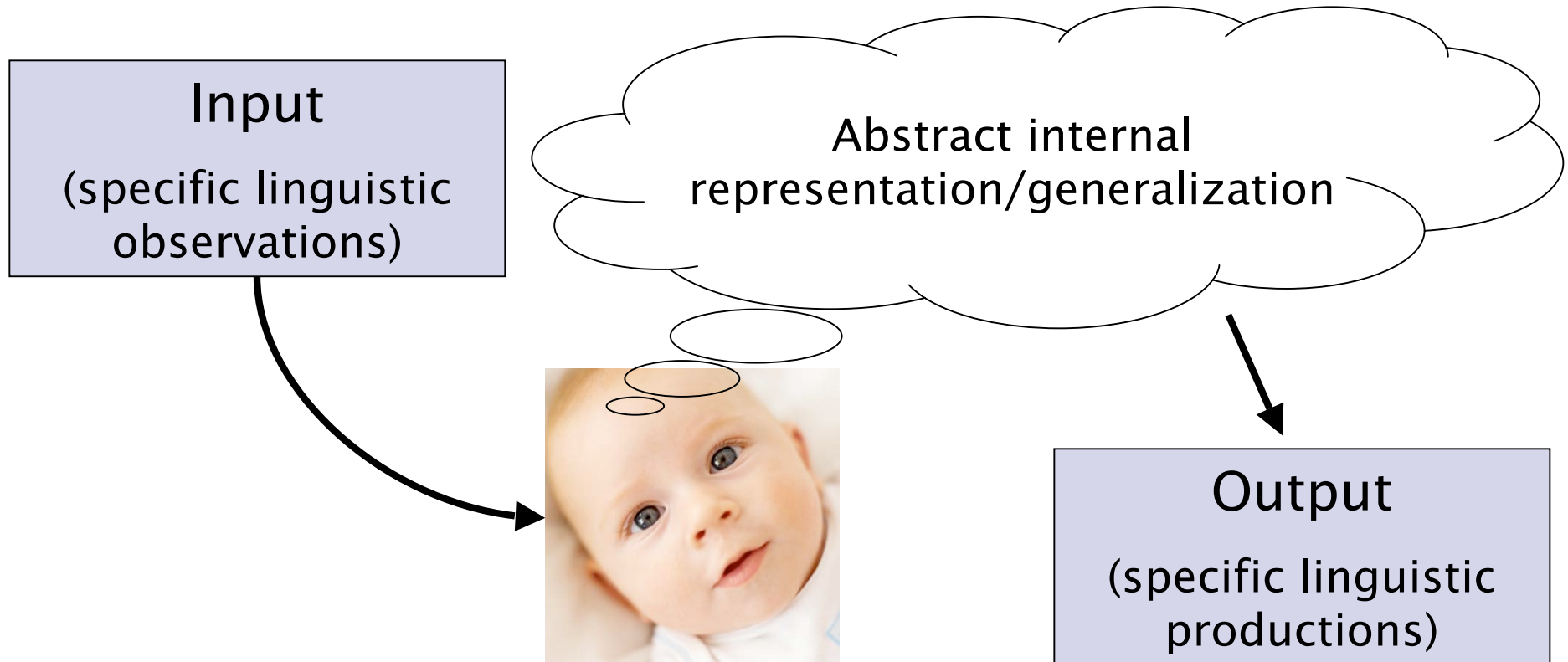
★Department of Cognitive Sciences  
University of California, Irvine

◆School of Informatics  
University of Edinburgh

Boston University Conference on Language Development

November 6, 2009

# Language acquisition as induction



# Bayesian modeling: ideal vs. constrained

---

- Typically an **ideal observer** approach asks what the optimal solution to the induction problem is, given particular assumptions about representation and available information.
- Here we investigate **constrained** learners that implement ideal learners in cognitively plausible ways.
  - How might **limitations on memory and processing** affect learning?

# Word segmentation



- Given a corpus of fluent speech or text (no utterance-internal word boundaries), we want to identify the words.

whatsthat  
thedoggie  
yeah  
wheresthedoggie



whats that  
the doggie  
yeah  
wheres the doggie

# Word segmentation

---

- One of the first problems infants must solve when learning language.
- Infants make use of many different cues.
  - Phonotactics, allophonic variation, metrical (stress) patterns, effects of coarticulation, and statistical regularities in syllable sequences.
- Statistics may provide initial bootstrapping.
  - Used very early (Thiessen & Saffran, 2003)
  - Language-independent, so doesn't require children to know some words already

# Bayesian learning

- The Bayesian learner seeks to identify an explanatory linguistic hypothesis that
  - accounts for the observed data.
  - conforms to prior expectations.

$$\underbrace{P(h|d)}_{\text{posterior}} \propto \underbrace{P(d|h)}_{\text{likelihood}} \underbrace{P(h)}_{\text{prior}}$$

- **Ideal learner**: Focus is on the goal of computation, not the procedure (algorithm) used to achieve the goal.
- **Constrained learner**: Use same probabilistic model, but algorithm reflects how humans might implement the computation.

# Bayesian segmentation

- In the domain of segmentation, we have:
  - Data: unsegmented corpus (transcriptions)
  - Hypotheses: sequences of word tokens

$$\underbrace{P(h|d)}_{\text{posterior}} \propto \underbrace{P(d|h)}_{\text{likelihood}} \underbrace{P(h)}_{\text{prior}}$$

= 1 if concatenating words forms corpus,  
= 0 otherwise.

Encodes assumptions or  
biases in the learner.

- Optimal solution is the segmentation with highest prior probability.

# An ideal Bayesian learner for word segmentation

- Model considers hypothesis space of segmentations, preferring those where
  - The lexicon is relatively small.
  - Words are relatively short.
- The learner has a perfect memory for the data
  - Order of data presentation doesn't matter.
  - The entire corpus (or equivalent) is available in memory.
- Note: only counts of lexicon items are required to compute highest probability segmentation.



# Investigating learner assumptions

---

- If a learner assumes that words are **independent units**, what is learned from realistic data? [**unigram model**]
- What if the learner assumes that words are units that **help predict** other units? [**bigram model**]

Approach of Goldwater, Griffiths, & Johnson (2007, 2009): use a Bayesian **ideal observer** to examine the consequences of making these different assumptions.

# Corpus: child-directed speech samples

- Bernstein-Ratner corpus:

- 9790 utterances of phonemically transcribed child-directed speech (19-23 months), 33399 tokens and 1321 unique types.
- Average utterance length: 3.4 words
- Average word length: 2.9 phonemes

- Example input:

```
yuwanttusid6bUk  
lUkD*z6b7wIThIzh&t  
&nd6dOgi  
yuwanttulUk&tDI  
...
```

≈

```
youwanttoseethebook  
looktheresaboywithhishat  
andadoggie  
youwanttolookatthis  
...
```

# Results: Ideal learner (Standard MCMC)

**Precision:**  $\#correct / \#found$ , “How many of what I found are right?”

**Recall:**  $\#found / \#true$ , “How many did I find that I should have found?”

|                        | Word Tokens |             | Boundaries  |             | Lexicon     |             |
|------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                        | Prec        | Rec         | Prec        | Rec         | Prec        | Rec         |
| <b>Ideal (unigram)</b> | 61.7        | 47.1        | <b>92.7</b> | 61.6        | 55.1        | <b>66.0</b> |
| <b>Ideal (bigram)</b>  | <b>74.6</b> | <b>68.4</b> | 90.4        | <b>79.8</b> | <b>63.3</b> | 62.6        |

- The assumption that words predict other words is good: bigram model generally has superior performance
- Both models tend to undersegment, though the bigram model does so less (boundary precision > boundary recall)
- Note: Training set was used as test set

# Results: Ideal learner sample segmentations

## Unigram model

```
youwant to see thebook  
look theres aboy with his hat  
and adoggie  
you wantto lookatthis  
lookatthis  
havea drink  
okay now  
whatsthis  
whatsthat  
whatisit  
look canyou take itout  
...
```

## Bigram model

```
you want to see the book  
look theres a boy with his hat  
and a doggie  
you want to lookat this  
lookat this  
have a drink  
okay now  
whats this  
whats that  
whatis it  
look canyou take it out  
...
```

# How about constrained learners?

---

- Our constrained learners use the same probabilistic model, but process the data incrementally (one utterance at a time), rather than all at once.
  - Dynamic Programming with Maximization (DPM)
  - Dynamic Programming with Sampling (DPS)
  - Decayed Markov Chain Monte Carlo (DMCMC)

# Considering human limitations

---

What is the most direct translation of the ideal learner to an online learner that must process utterances one at a time?

# Dynamic Programming: Maximization

For each utterance:

- Use dynamic programming to compute probabilities of all segmentations, given the current lexicon.
- Choose the best segmentation.
- Add counts of segmented words to **lexicon**.

|   |      |                                 |
|---|------|---------------------------------|
|   |      | <i>you want to see the book</i> |
| → | 0.33 | yu want tusi D6bUk              |
|   | 0.21 | yu wanttusi D6bUk               |
|   | 0.15 | yuwant tusi D6 bUk              |
|   | ...  | ...                             |

- Algorithm used by Brent (1999), with different model.

# Considering human limitations

---


What if humans don't always choose the most probable hypothesis, but instead sample among the different hypotheses available?



# Dynamic Programming: Sampling

For each utterance:

- Use dynamic programming to compute probabilities of all segmentations, given the current lexicon.
- Sample a segmentation.
- Add counts of segmented words to **lexicon**.

|  |                                 |
|--|---------------------------------|
|  | <i>you want to see the book</i> |
| 0.33   | yu want tusi D6bUk              |
| 0.21   | yu wanttusi D6bUk               |
|  0.15 | <b>yuwant tusi D6 bUk</b>       |
| ...  | ...                             |

# Considering human limitations

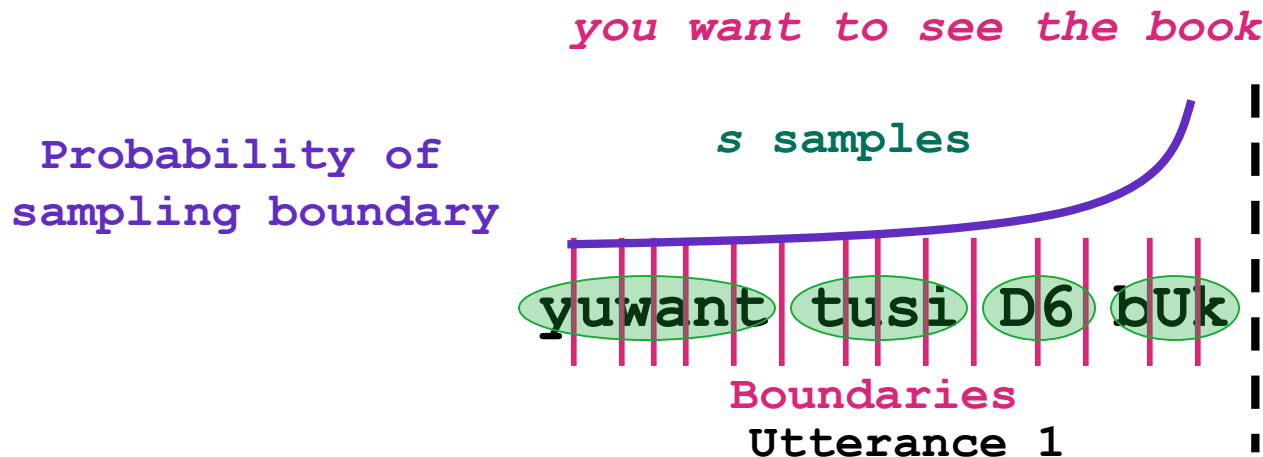
---

What if humans are more likely to sample potential word boundaries that they have heard more recently (decaying memory = recency effect)?

# Decayed Markov Chain Monte Carlo

For each utterance:

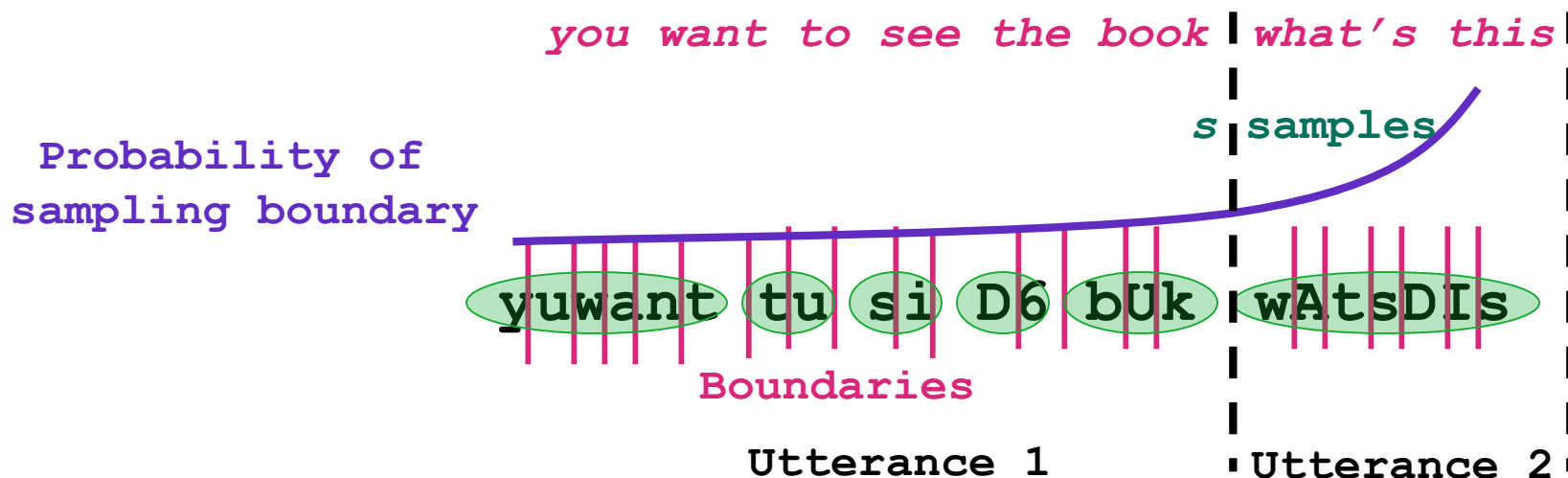
- Probabilistically **sample  $s$  boundaries** from all utterances encountered so far.
- $\text{Prob}(\text{sample } b) \propto b_a^{-d}$  where  $b_a$  is the number of potential boundary locations between  $b$  and the end of the current utterance and  $d$  is the decay rate (Marthi et al. 2002).
- Update **lexicon** after the  $s$  samples are completed.



# Decayed Markov Chain Monte Carlo

For each utterance:

- Probabilistically **sample  $s$  boundaries** from all utterances encountered so far.
- $\text{Prob}(\text{sample } b) \propto b_a^{-d}$  where  $b_a$  is the number of potential boundary locations between  $b$  and the end of the current utterance and  $d$  is the decay rate (Marthi et al. 2002).
- Update **lexicon** after the  $s$  samples are completed.

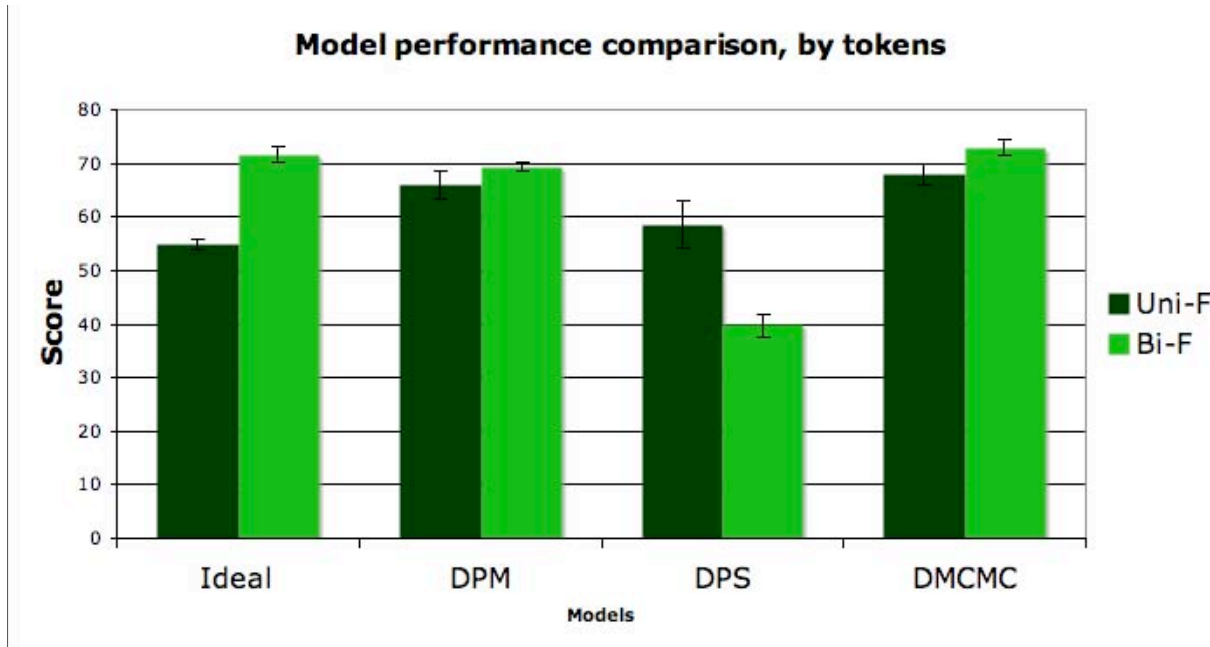


# Decayed Markov Chain Monte Carlo

Decay rates tested: 2, 1.5, 1, 0.75, 0.5, 0.25, 0.125

|             | <b>Probability of<br/>sampling within<br/>current utterance</b> |
|-------------|---|
| $d = 2$     | .942  |
| $d = 1.5$   | .772  |
| $d = 1$     | .323  |
| $d = 0.75$  | .125  |
| $d = 0.5$   | .036  |
| $d = 0.25$  | .009  |
| $d = 0.125$ | .004  |

# Results: unigrams vs. bigrams



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

#correct / #found

Recall:

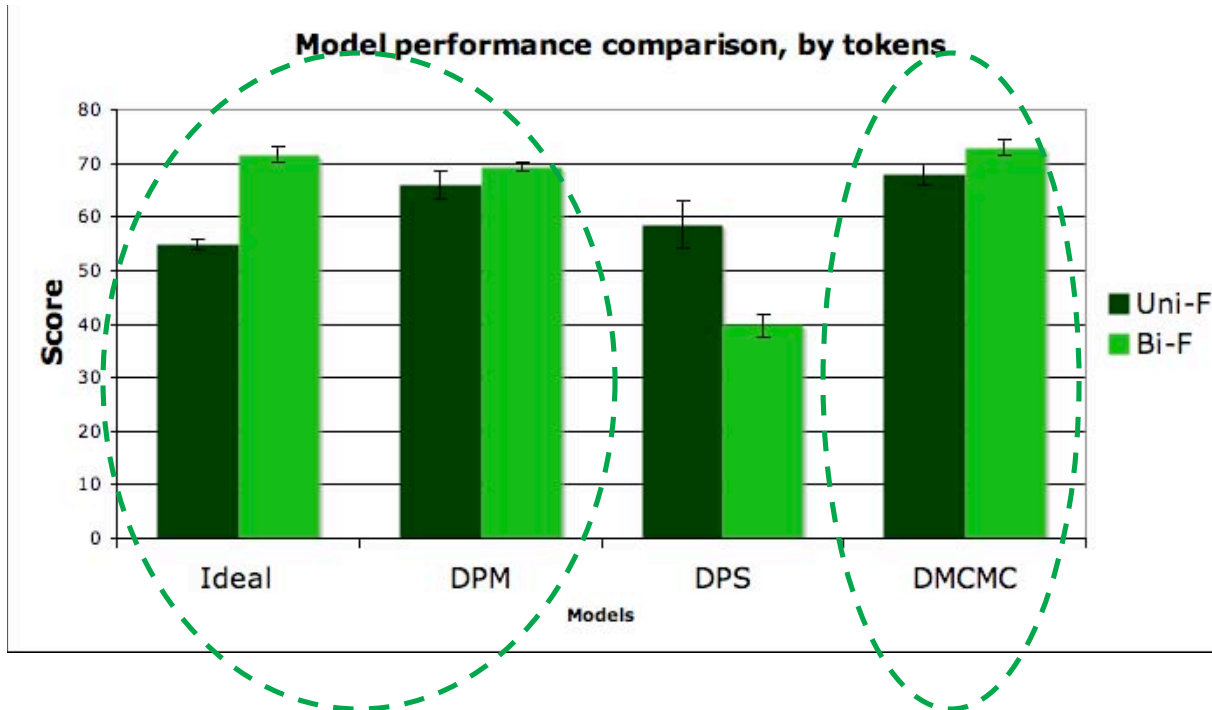
#found / #true

Results averaged over 5 randomly generated test sets (~900 utterances) that were separate from the training sets (~8800 utterances), all generated from the Bernstein Ratner corpus

DMCMC Unigram:  $d=1, s=20000$   
DMCMC Bigram:  $d=0.25, s=20000$

Note:  $s=20000$  means DMCMC learner samples 89% less often than the Ideal learner.

# Results: unigrams vs. bigrams



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

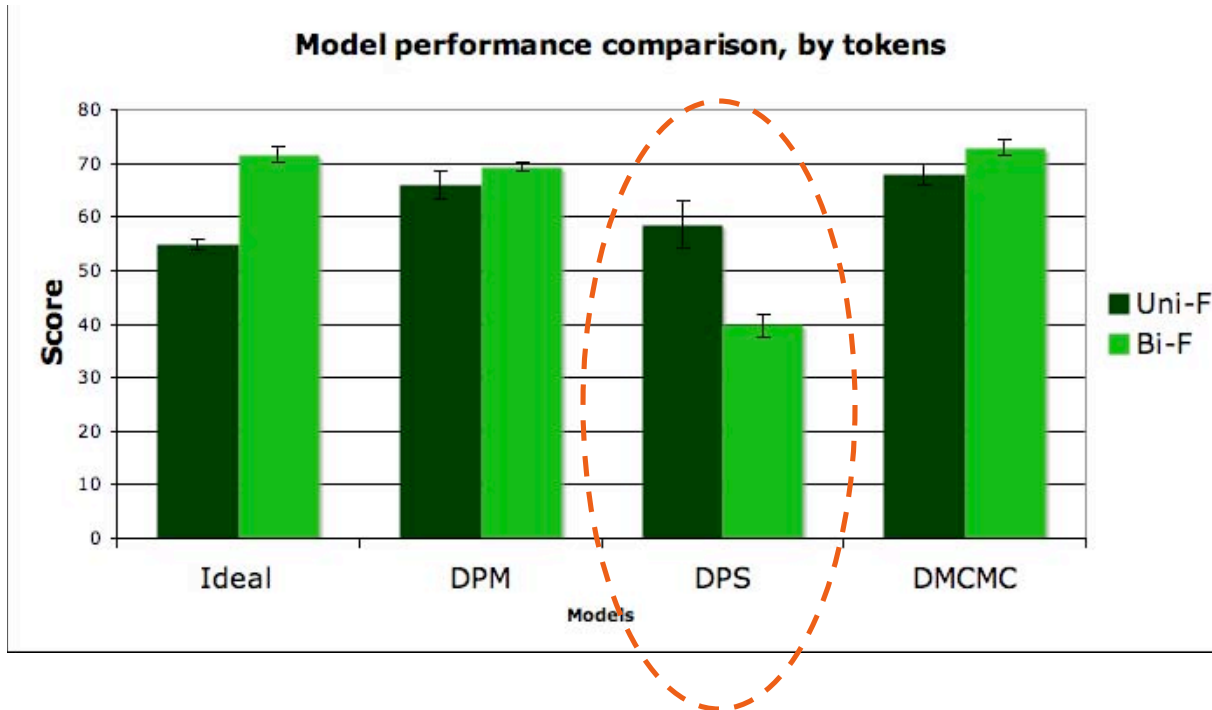
#correct / #found

Recall:

#found / #true

Like the Ideal learner, the DPM & DMCMC bigram learners perform better than the unigram learner, though improvement is not as great as in the Ideal learner. The bigram assumption is helpful.

# Results: unigrams vs. bigrams



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

#correct / #found

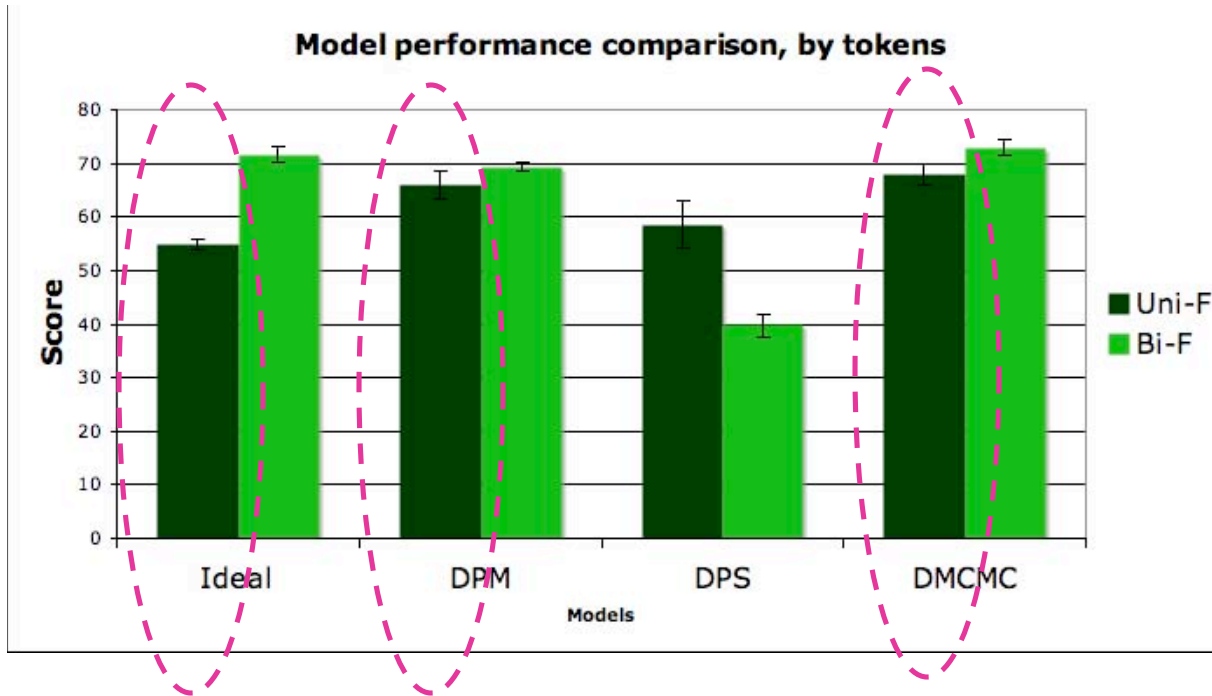
Recall:

#found / #true

However, the DPS bigram learner performs worse than the unigram learner. The bigram assumption is not helpful.



# Results: unigrams vs. bigrams



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

#correct / #found

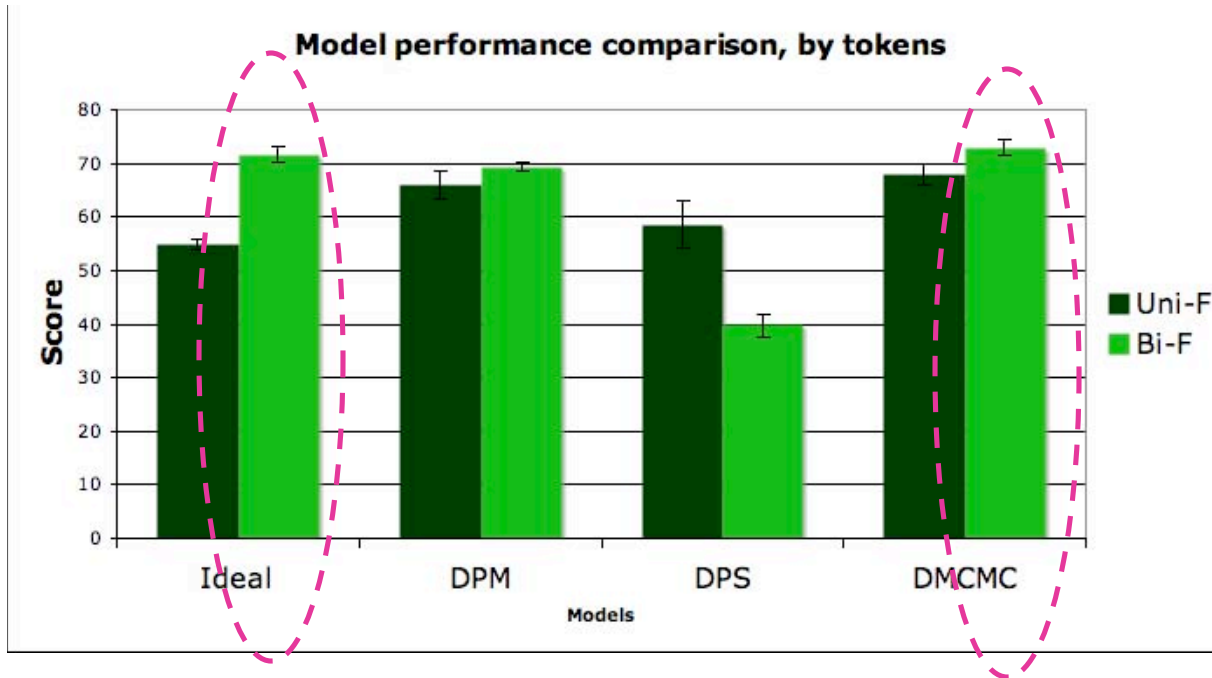
Recall:

#found / #true

Unigram comparison: DPM, DMCMC > Ideal, DPS performance

Interesting: Constrained learners outperforming unconstrained learner when context is not considered.

# Results: unigrams vs. bigrams



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

$\frac{\# \text{correct}}{\# \text{found}}$

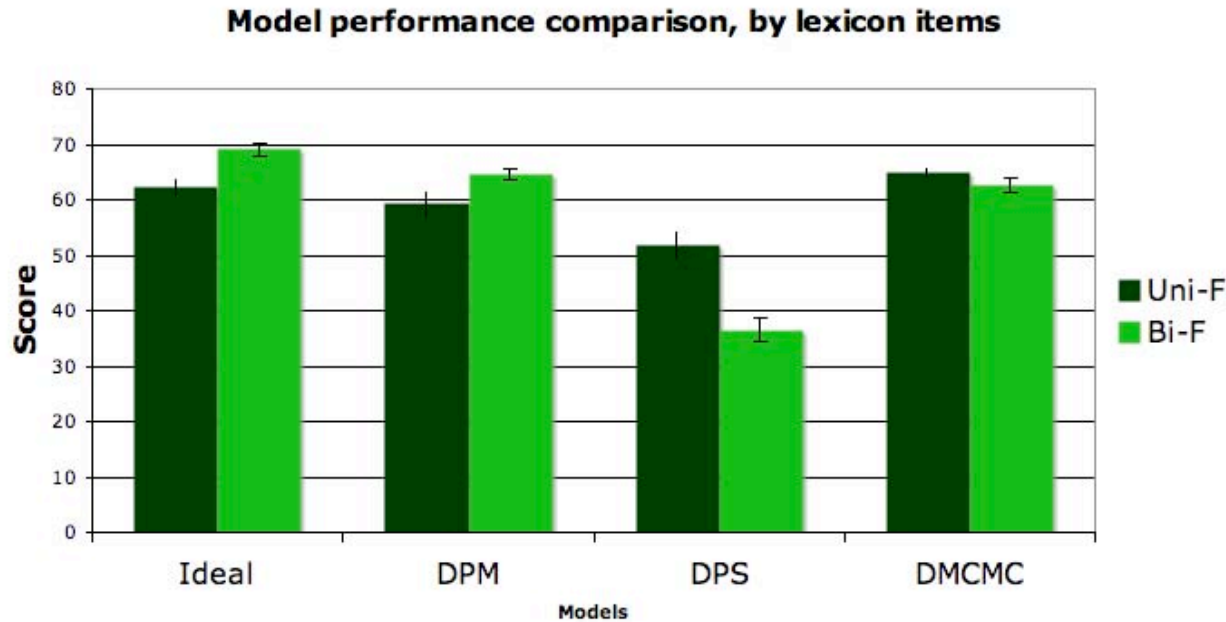
Recall:

$\frac{\# \text{found}}{\# \text{true}}$

Bigram comparison: Ideal, DMCMC > DPM > DPS performance

Interesting: Constrained learner performing equivalently to unconstrained learner when context is considered.

# Results: unigrams vs. bigrams for the lexicon



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

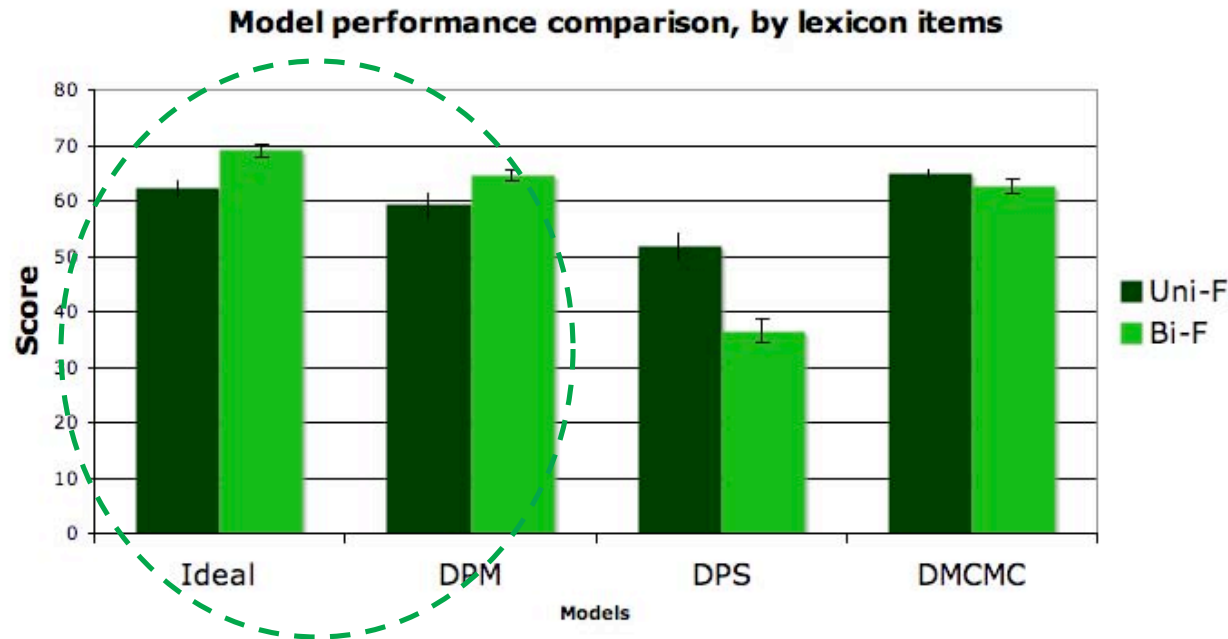
#correct / #found

Recall:

#found / #true

Lexicon = a seed pool of words for children to use to figure out language-dependent word segmentation strategies.

# Results: unigrams vs. bigrams for the lexicon



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

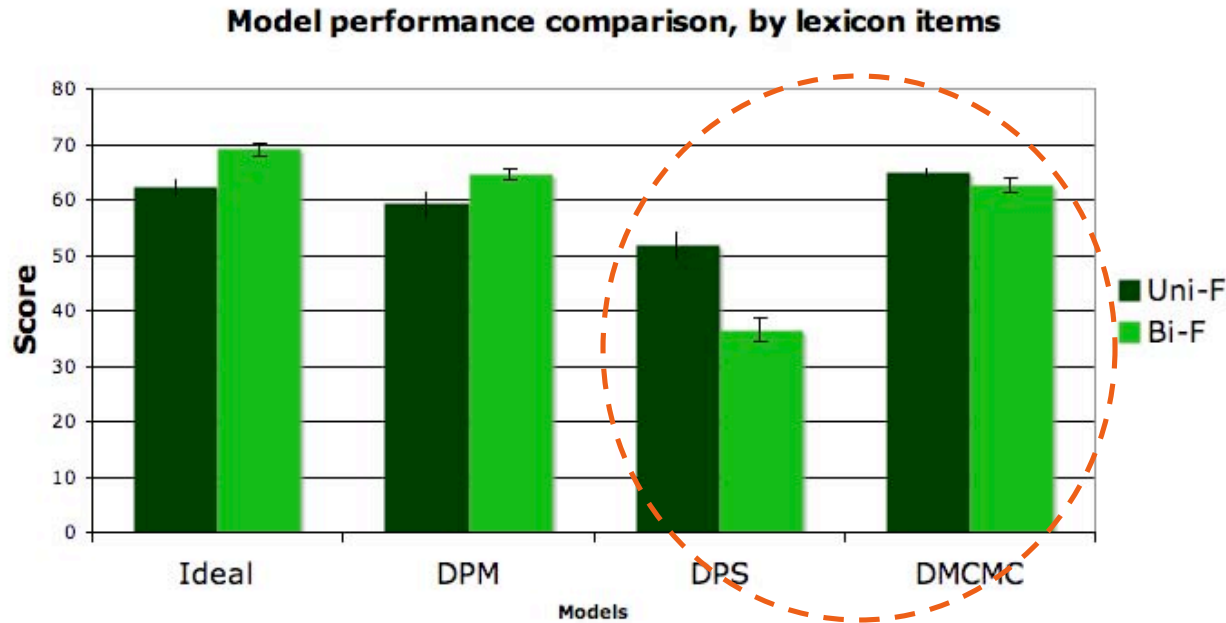
#correct / #found

Recall:

#found / #true

Like the Ideal learner, the DPM bigram learner yields a more reliable lexicon than the unigram learner.

# Results: unigrams vs. bigrams for the lexicon



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

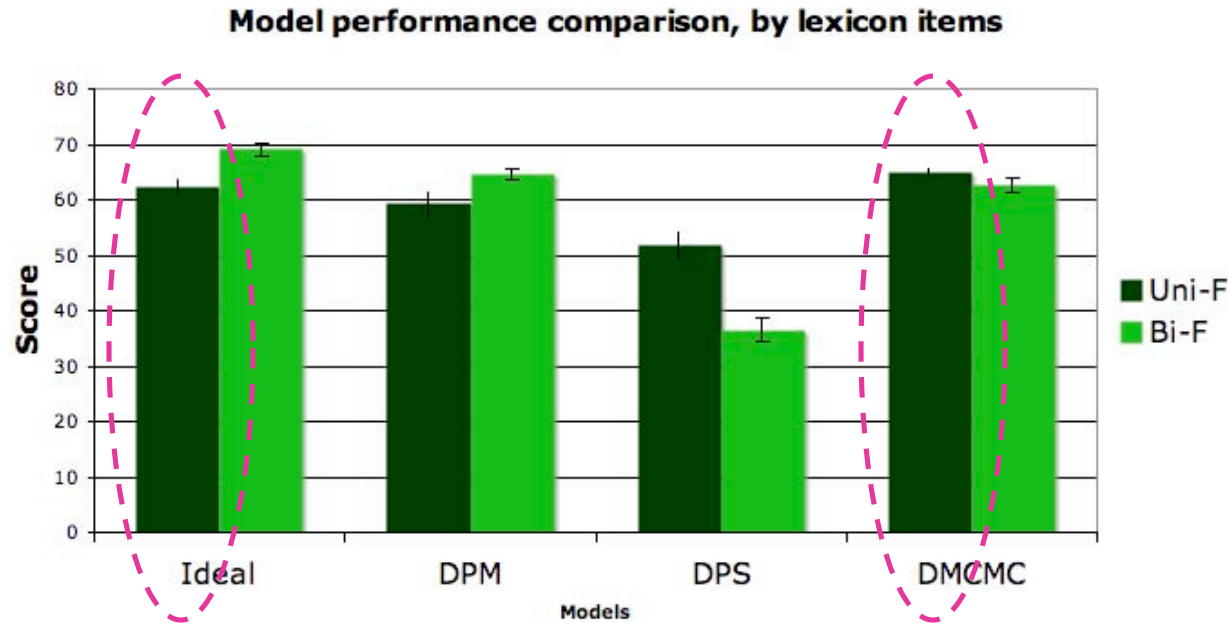
$\frac{\# \text{correct}}{\# \text{found}}$

Recall:

$\frac{\# \text{found}}{\# \text{true}}$

However, the DPS and DMCMC bigram learners yield less reliable lexicons than the unigram learners.

# Results: unigrams vs. bigrams for the lexicon



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

#correct / #found

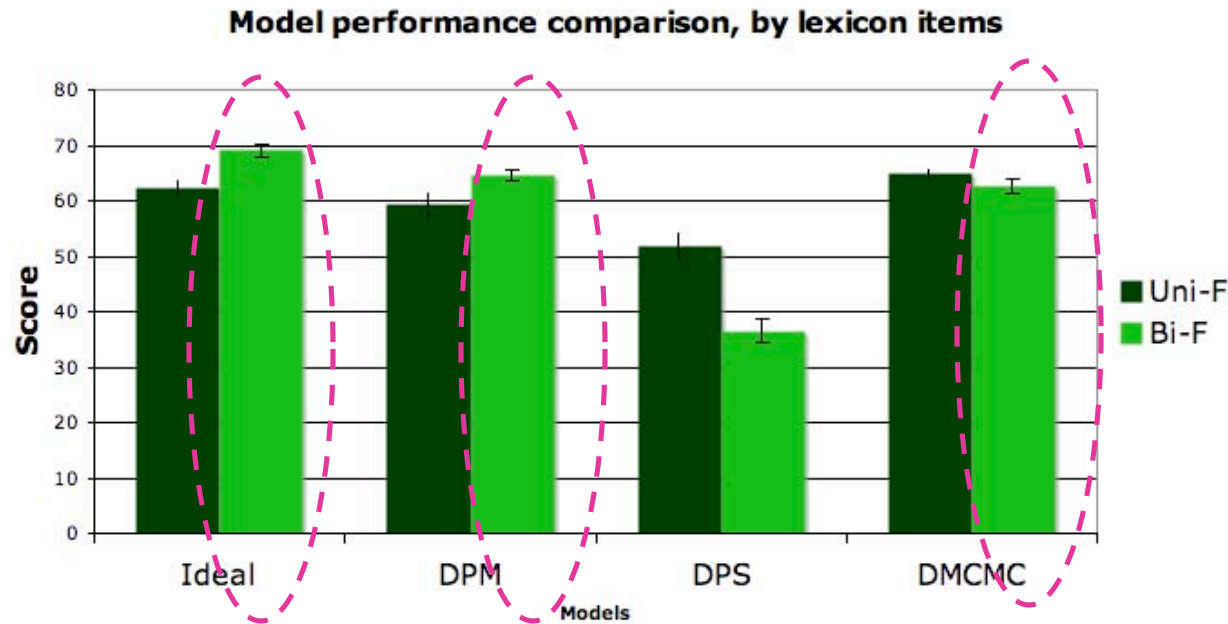
Recall:

#found / #true

Unigram comparison: DMCMC > Ideal > DPM > DPS performance

Interesting: Constrained learner outperforming unconstrained learner when context is not considered.

# Results: unigrams vs. bigrams for the lexicon



$$F = \frac{2 * \text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}}$$

Precision:

$\frac{\# \text{correct}}{\# \text{found}}$

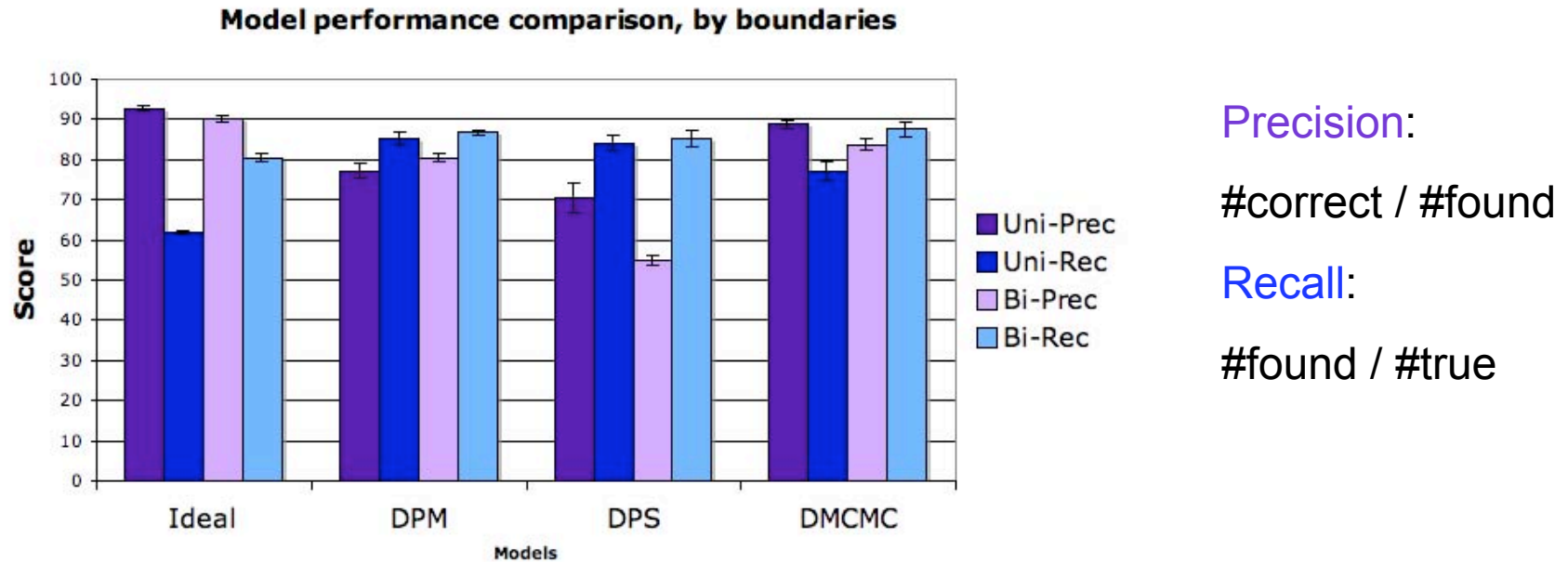
Recall:

$\frac{\# \text{found}}{\# \text{true}}$

Bigram comparison: Ideal > DPM > DMCMC > DPS performance

More expected: Unconstrained learner outperforming constrained learners when context is considered (though not by a lot).

# Results: under vs. oversegmentation

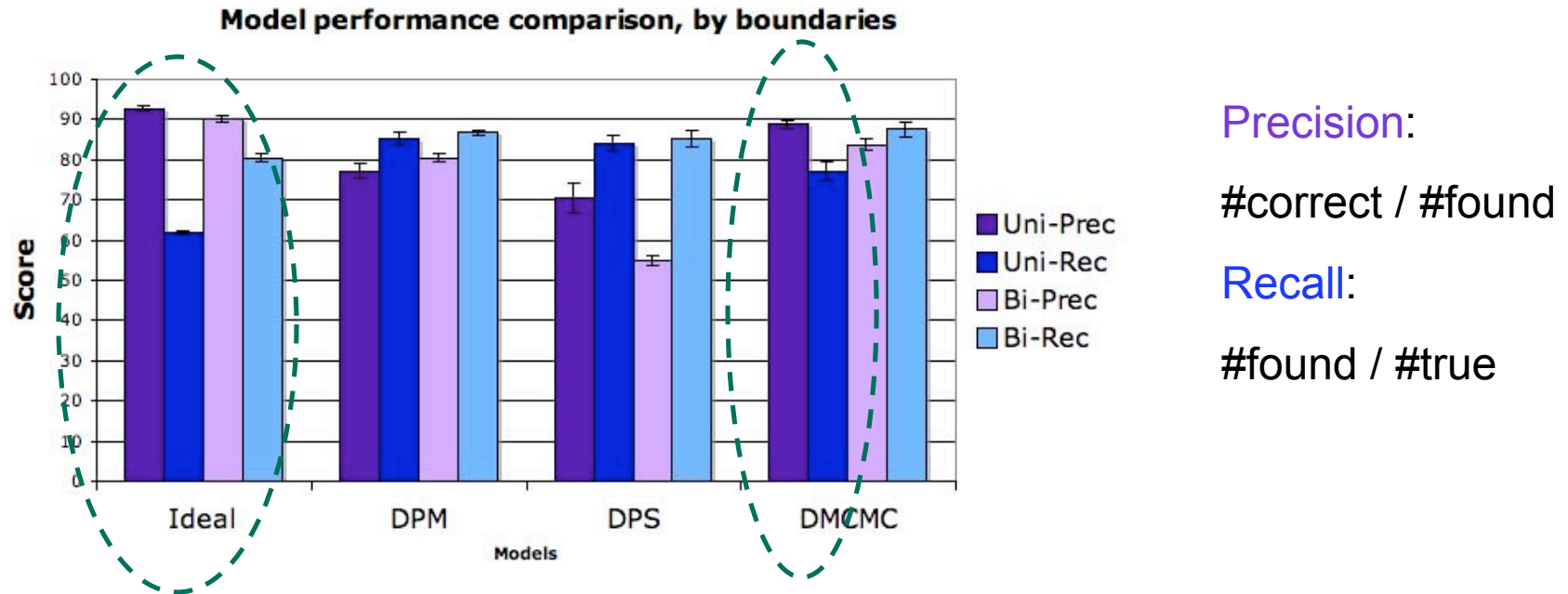


Undersegmentation: boundary precision > boundary recall

Oversegmentation: boundary precision < boundary recall

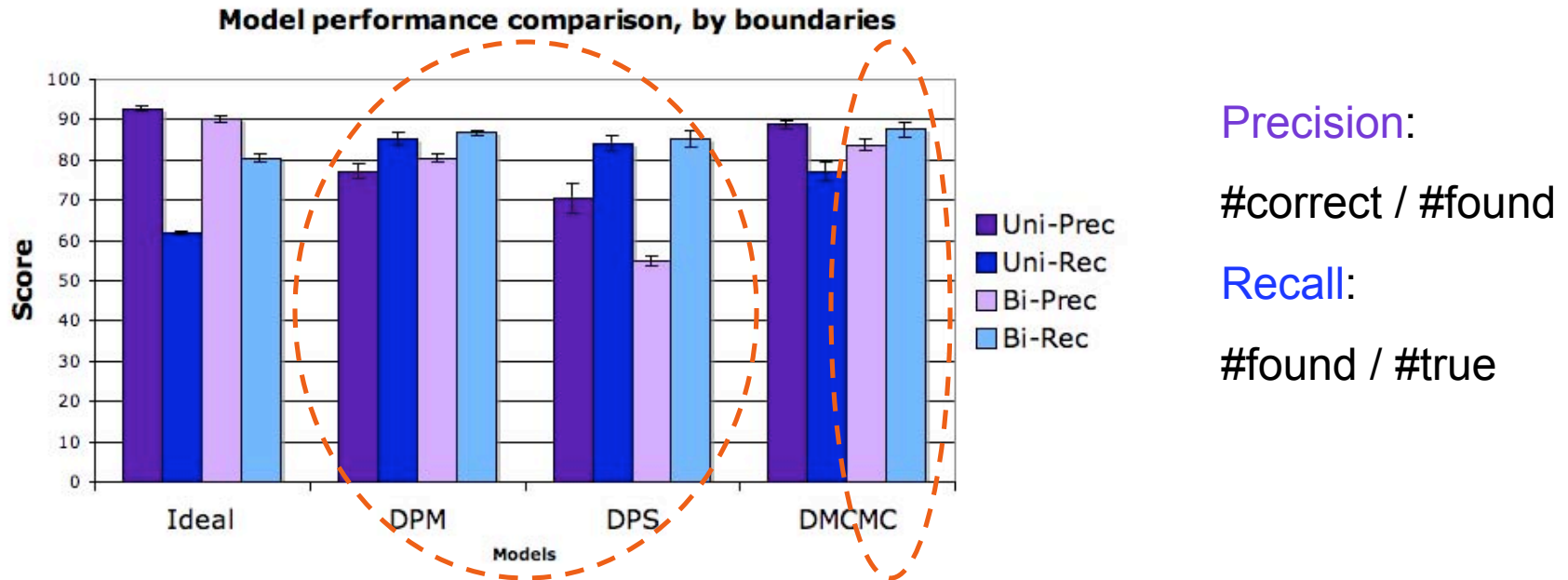


# Results: under vs. oversegmentation



The DMCMC unigram learner, like the Ideal learner, tends to undersegment.

# Results: under vs. oversegmentation



All other learners, however, tend to oversegment.

# Results: interim summary

---

- While no constrained learners outperform the best ideal learner on all measures, **all perform better on realistic child-directed speech data than a syllable transitional probability learner**, which achieves a token F score of 29.9 (Gambell & Yang 2006).
- While assuming words are predictive units (**bigram model**) significantly helped **the ideal learner**, this assumption may not be as useful to a **constrained learner** (depending on how memory limitations are implemented). Moreover, **constrained unigram learners** can sometimes outperform the ideal (standard MCMC) unigram learner (“Less is More” Hypothesis: Newport 1990).

# Results: interim summary

---

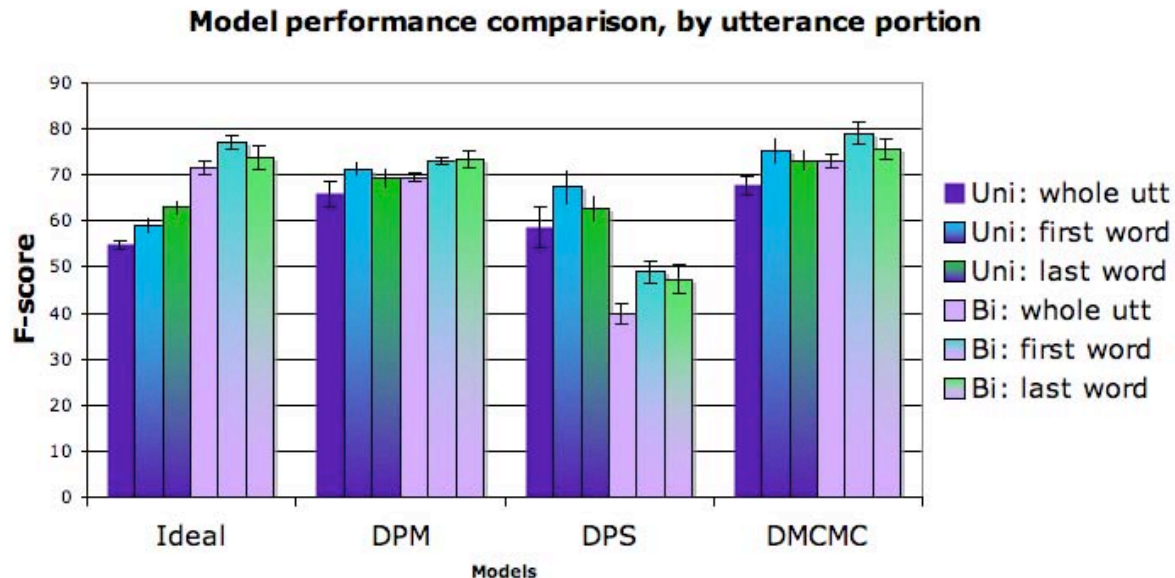
- The tendency to undersegment the corpus also depends on how memory limitations are implemented. Undersegmentation may match children's performance better than oversegmentation (Peters 1983).
- The lower the decay rate in the DMCMC learner, the more the learner tends to undersegment. (Ask for details!)
- DMCMC learners can actually perform fairly well, even with significantly fewer samples per utterance. (Ask for details!)

# Results: Exploring different performance measures

- Some positions in the utterance are more easily segmented by infants, such as the **first** and **last** word of the utterance (Seidl & Johnson 2006).
  - The first and last word are less ambiguous (one boundary known)  
(**first**, **last** > **whole utterance**)
  - Memory effects & prosodic prominence make the last word easier  
(**last** > **first**, **whole utterance**)
  - The first/last word are more regular, due to syntactic properties  
(**first**, **last** > **whole utterance**)

```
look theres a boy with his hat  
and a doggie  
you want to look at this  
Look at this
```

# Results: Exploring different performance measures



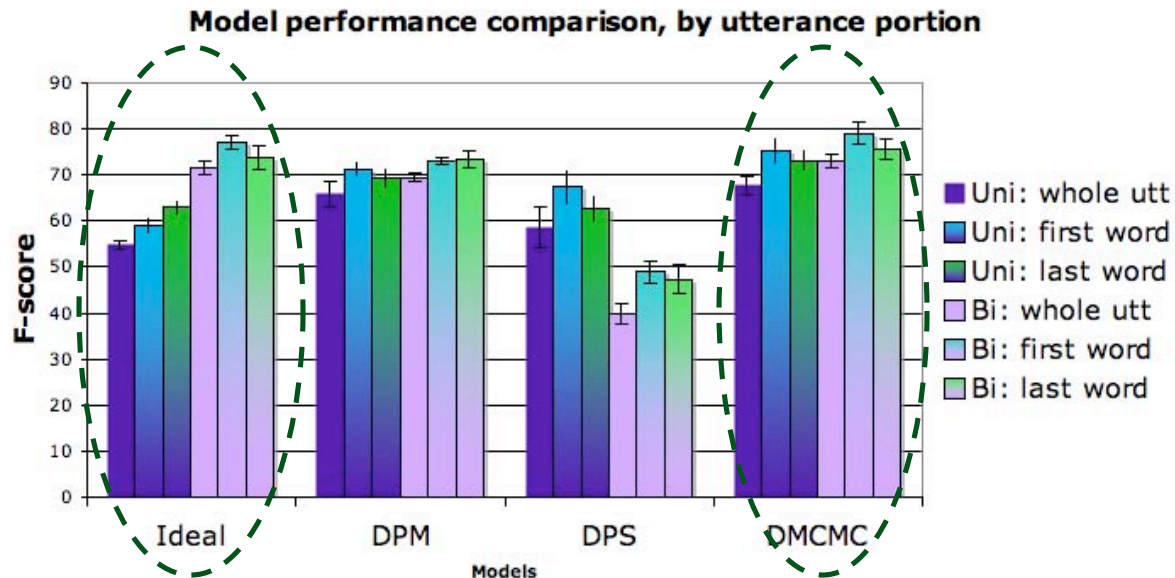
Unigrams vs. Bigrams,  
Token F-scores

whole utterance  
first word  
last word

*Results averaged over 5 randomly generated test sets (~900 utterances) that were separate from the training sets (~8800 utterances), all generated from the Bernstein Ratner corpus*

*DMCMC Unigram:  $d=1, s=20000$   
DMCMC Bigram:  $d=0.25, s=20000$*

# Results: Exploring different performance measures

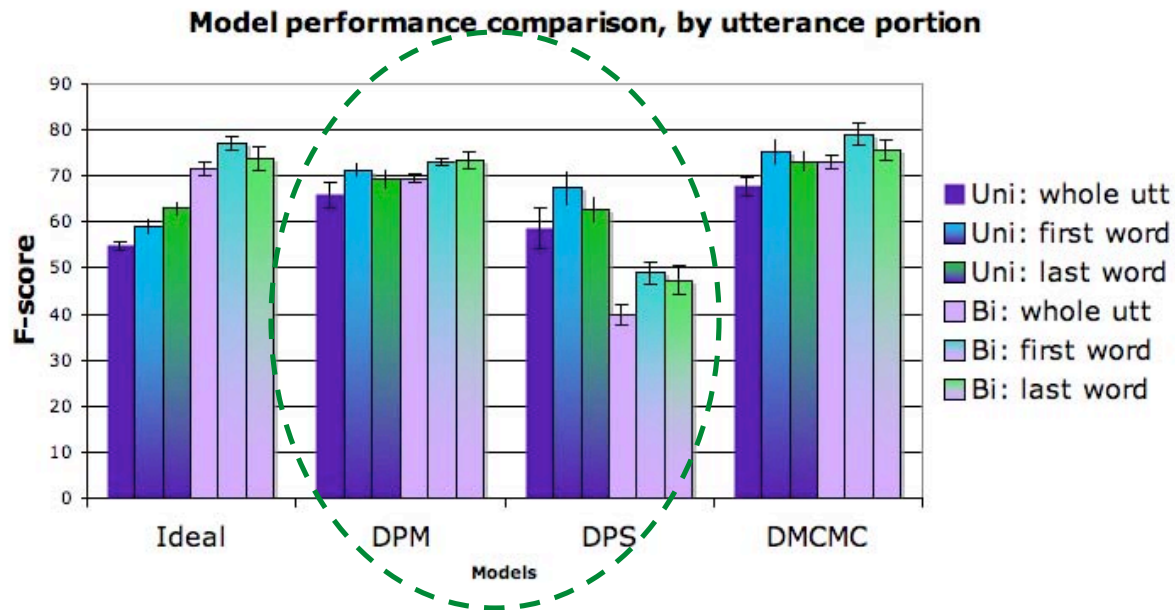


Unigrams vs. Bigrams,  
Token F-scores

whole utterance  
first word  
last word

The Ideal unigram learner performs better on the first and last words in the utterance, while the bigram learner only improves for the first words. The DMCMC follows this trend.

# Results: Exploring different performance measures



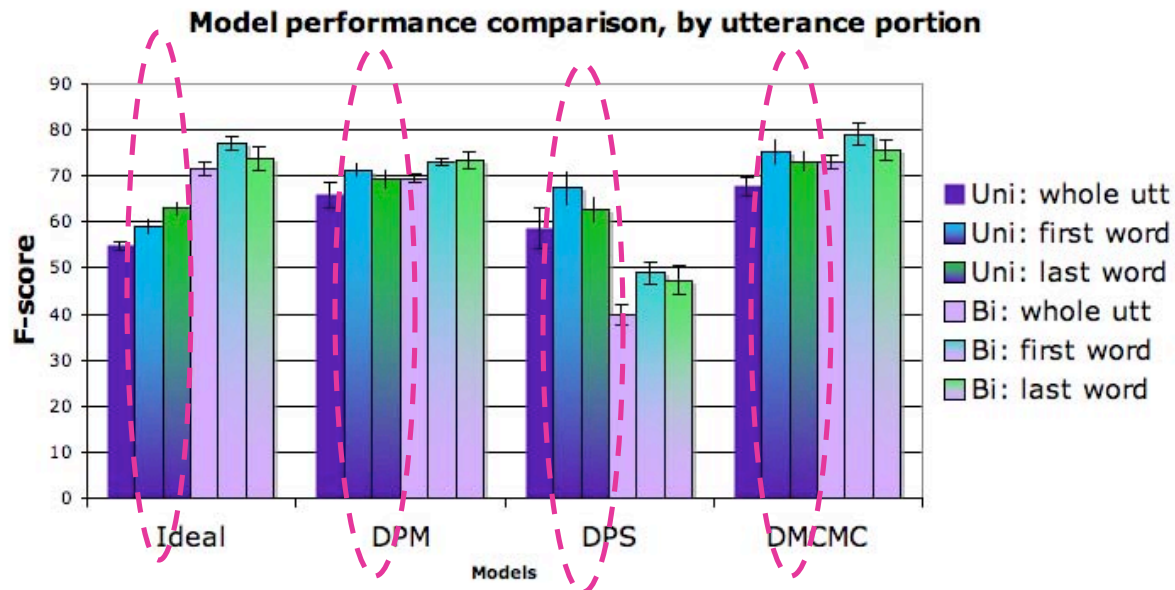
Unigrams vs. Bigrams,  
Token F-scores

whole utterance  
first word  
last word

The DPM and DPS learners usually improve on the first and last words, irrespective of n-gram model. The first word tends to improve as much as the last word.



# Results: Exploring different performance measures



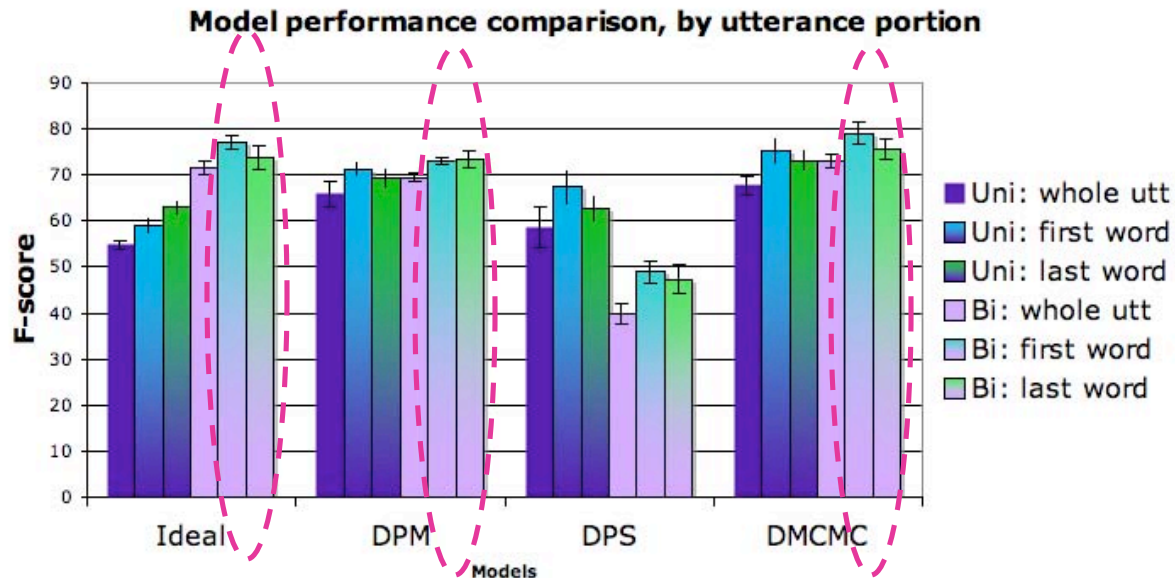
Unigrams vs. Bigrams,  
Token F-scores

whole utterance  
first word  
last word

Interesting:

Constrained unigram learners outperform the Ideal learner for first and last words.

# Results: Exploring different performance measures



Unigrams vs. Bigrams,  
Token F-scores

whole utterance  
first word  
last word

Interesting:

Constrained unigram learners outperform the Ideal learner for first and last words.

Some constrained bigram learners are equivalent to the unconstrained learner for first and last words.

# Summary: Constrained Learners

- Simple intuitions about human cognition (e.g. memory limitations) can be translated in multiple ways
  - processing utterances incrementally
  - keeping a single lexicon hypothesis in memory
  - implementing recency effects
- Learning biases/assumptions that are helpful in an ideal learner may hinder a learner with processing constraints.
- Constrained learners can still use statistical regularity available in the data. Sometimes learners with processing constraints may even outperform unconstrained learners.
- Statistical learning doesn't have to be perfect to reflect acquisition: online statistical learning may provide a lexicon reliable enough for children to learn language-dependent strategies from.

# The End & Thank You!

---

Special thanks to...

Tom Griffiths

Michael Frank

the Computational Models of Language Learning Seminar at UCI 2008

the Psychocomputational Models of Human Language Acquisition

Workshop 2009

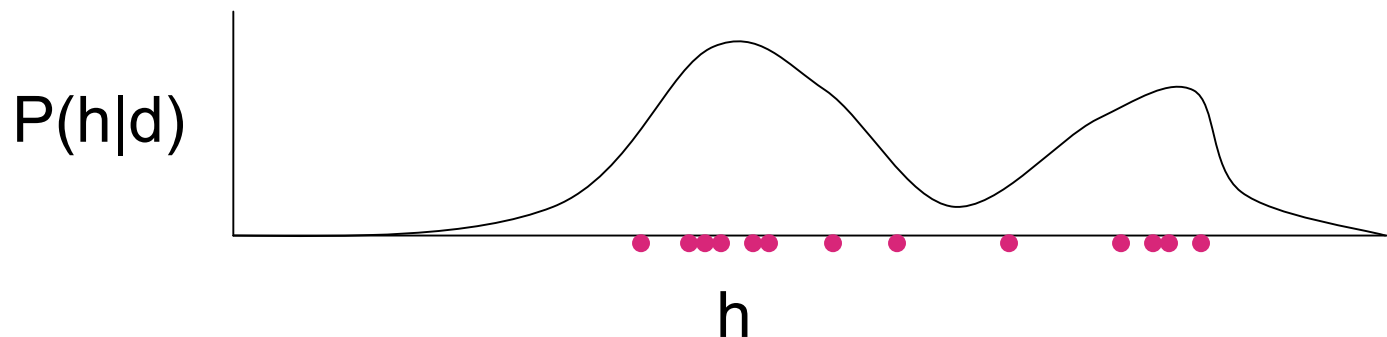
*This work was supported by NSF grant BCS-0843896 to LP.*



# Search algorithm comparison

Model defines a distribution over hypotheses. We use **Gibbs sampling** to find a good hypothesis.

- Iterative procedure produces samples from the posterior distribution of hypotheses.



- **Ideal (Standard)**: A batch algorithm  
vs. **DMCMC**: incremental algorithm that uses the same sampling equation

# Gibbs sampler

- Compares pairs of hypotheses differing by a single word boundary:

```
whats . that  
the .doggie  
yeah  
wheres . the .doggie  
...
```

```
whats . that  
the .dog.gie  
yeah  
wheres . the .doggie  
...
```

- Calculate the probabilities of the words that differ, given current analysis of all other words.
- Sample a hypothesis according to the ratio of probabilities.





# The unigram model

Assumes word  $w_i$  is generated as follows:

1. Is  $w_i$  a novel lexical item?

$$P(\text{yes}) = \frac{\alpha}{n + \alpha}$$

$$P(\text{no}) = \frac{n}{n + \alpha}$$

Fewer word types =  
Higher probability

# The unigram model

Assume word  $w_i$  is generated as follows:

2. **If novel**, generate phonemic form  $x_1 \dots x_m$  :

$$P(w_i = x_1 \dots x_m) = \prod_{i=1}^m P(x_i)$$

Shorter words =  
Higher probability

**If not**, choose lexical identity of  $w_i$  from previously occurring words:

$$P(w_i = w) = \frac{n_w}{n}$$

Power law =  
Higher probability

# Notes

- Distribution over words is a **Dirichlet Process** (DP) with concentration parameter  $\alpha$  and base distribution  $P_0$ :

$$P(w_i = w \mid w_1 \dots w_{i-1}) = \frac{n_w + \alpha P_0(w)}{i - 1 + \alpha}$$

- Also (nearly) equivalent to Anderson's (1990) Rational Model of Categorization.

# Bigram model

Assume word  $w_i$  is generated as follows:

1. Is  $(w_{i-1}, w_i)$  a novel bigram?

$$P(\text{yes}) = \frac{\beta}{n_{w_{i-1}} + \beta} \quad P(\text{no}) = \frac{n_{w_{i-1}}}{n_{w_{i-1}} + \beta}$$

2. **If novel**, generate  $w_i$  using unigram model (almost).

**If not**, choose lexical identity of  $w_i$  from words previously occurring after  $w_{i-1}$ .

$$P(w_i = w \mid w_{i-1} = w') = \frac{n_{(w', w)}}{n_{w'}}$$

# Notes

- Bigram model is a **hierarchical Dirichlet process** (Teh et al., 2005):

$$P(w_i = w \mid w_{i-1} = w', w_1 \dots w_{i-2}) = \frac{n_{(w',w)} + \beta P_1(w)}{i - 1 + \beta}$$

$$P_1(w_i = w \mid w_1 \dots w_{i-1}) = \frac{b_w + \alpha P_0(w)}{b + \alpha}$$



# Results: Exploring decay rates in DMCMC

Unigram learners,  $s = 10000$ , on training and test set 1

|           | Word Tokens |             | Boundaries  |             | Lexicon     |             |
|-----------|-------------|-------------|-------------|-------------|-------------|-------------|
|           | Prec        | Rec         | Prec        | Rec         | Prec        | Rec         |
| $d=2$     | 23.2        | 33.7        | 46.8        | <b>77.2</b> | 24.6        | 15.1        |
| $d=1.5$   | 59.8        | 55.1        | 79.3        | 70.4        | 44.2        | 53.7        |
| $d=1$     | <b>71.5</b> | <b>65.7</b> | 88.1        | 73.0        | 60.9        | 68.7        |
| $d=0.75$  | 69.5        | 62.4        | 88.5        | 75.7        | 60.8        | 69.6        |
| $d=0.5$   | 69.3        | 60.9        | 89.5        | 74.1        | <b>62.2</b> | 71.5        |
| $d=0.25$  | 65.9        | 54.9        | <b>90.0</b> | 68.6        | 56.9        | <b>72.0</b> |
| $d=0.125$ | 64.0        | 52.8        | 89.4        | 67.0        | 54.9        | 71.1        |

- Decay rate 1 has best performance by tokens.
- Undersegmentation occurs more as decay rate decreases.
- Lexicon recall increases as decay rate decreases, and is generally higher than lexicon precision.

# Results: Exploring decay rates in DMCMC

Bigram learners,  $s = 10000$ , on training and test set 1

|           | Word Tokens |             | Boundaries  |             | Lexicon     |             |
|-----------|-------------|-------------|-------------|-------------|-------------|-------------|
|           | Prec        | Rec         | Prec        | Rec         | Prec        | Rec         |
| $d=2$     | 38.9        | 55.4        | 57.4        | 92.2        | 42.1        | 33.6        |
| $d=1.5$   | 51.5        | 67.3        | 66.0        | <b>94.9</b> | 51.8        | 42.9        |
| $d=1$     | 61.9        | 73.8        | 73.1        | 93.0        | 55.6        | 49.1        |
| $d=0.75$  | 59.9        | 71.0        | 72.6        | 91.9        | 55.1        | 51.5        |
| $d=0.5$   | 63.1        | 71.2        | 75.5        | 89.5        | 57.1        | 54.1        |
| $d=0.25$  | <b>70.7</b> | <b>72.9</b> | 82.4        | 86.1        | <b>58.8</b> | 61.8        |
| $d=0.125$ | 69.6        | 69.4        | <b>83.5</b> | 83.1        | 58.5        | <b>63.4</b> |

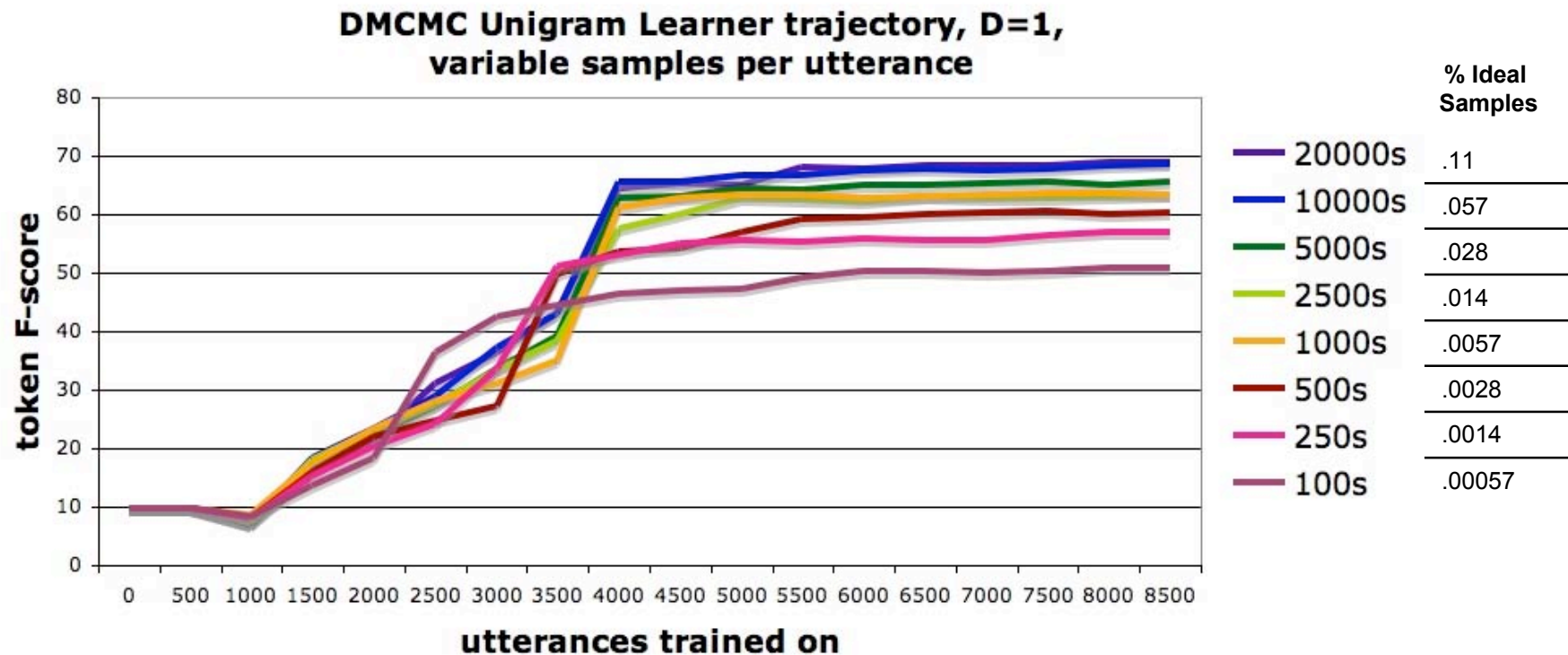
- Decay rate 0.25 has the best performance by tokens.
- Oversegmentation occurs less as decay rate decreases.
- Lexicon scores behind unigram lexicon scores, though increase as decay rate decreases.





# Results: The effect of number of samples

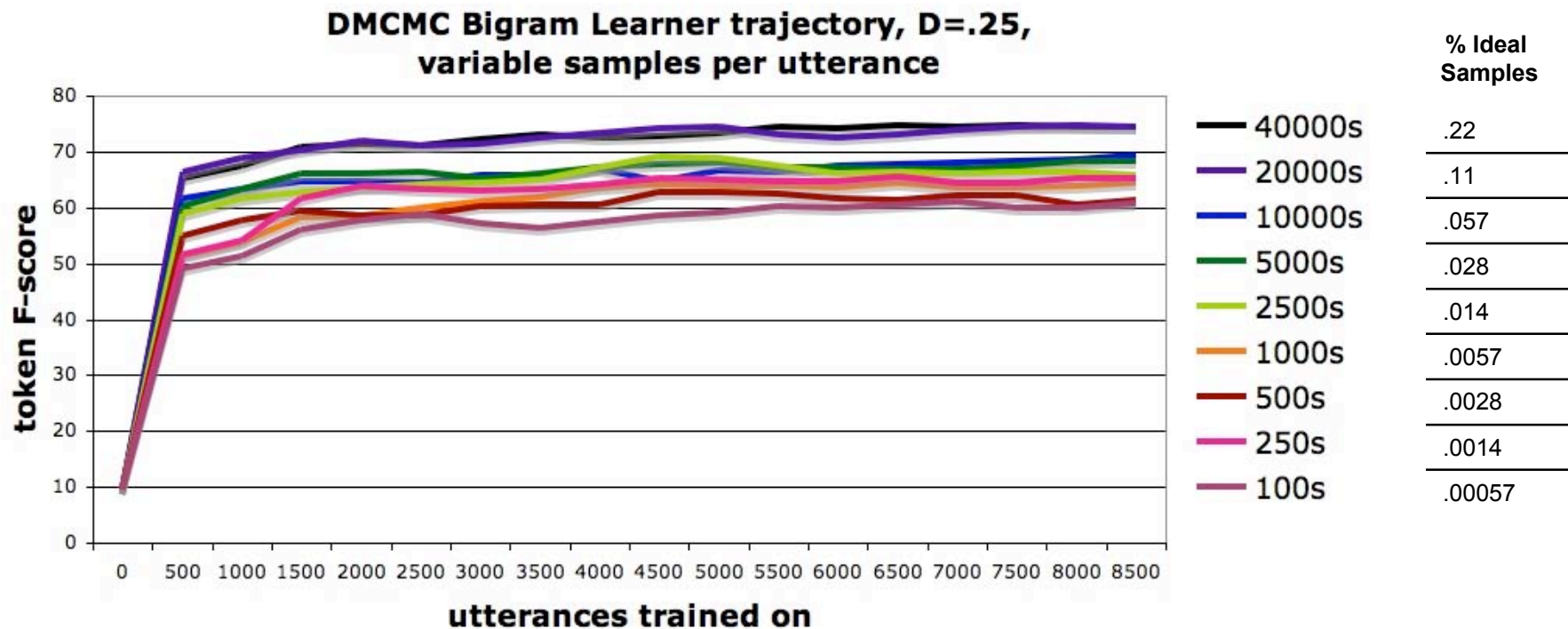
Unigram learners, on training and test set 1



- Even down to 500 samples per utterance, token F score is still above 60. Can still get reasonably high score with fairly few samples.
- Scores somewhat stable after about 4000 utterances trained on.

# Results: The effect of number of samples

Bigram learners, on training and test set 1

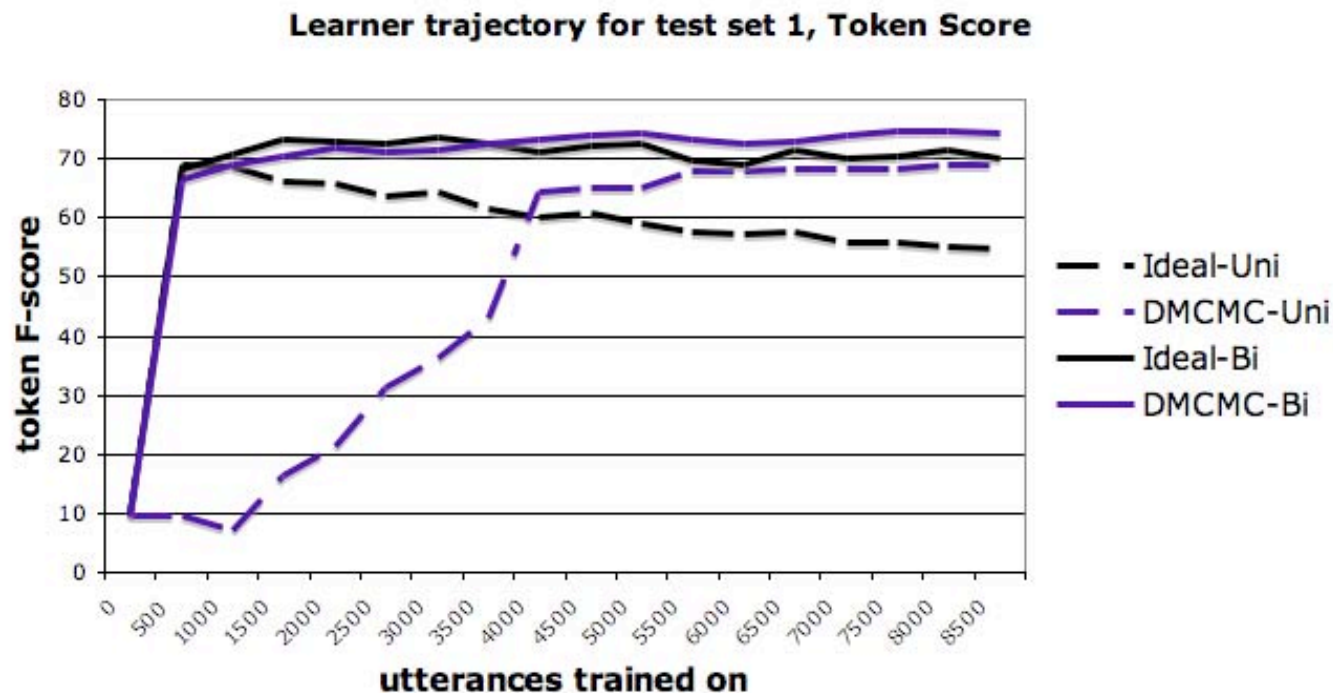


- Even down to 100 samples per utterance, token F score is still above 60. (Less samples required to get high score.)
- Jump in score occurs quickly, after only 500 utterances trained on.



# Results: Standard vs. Decayed MCMC

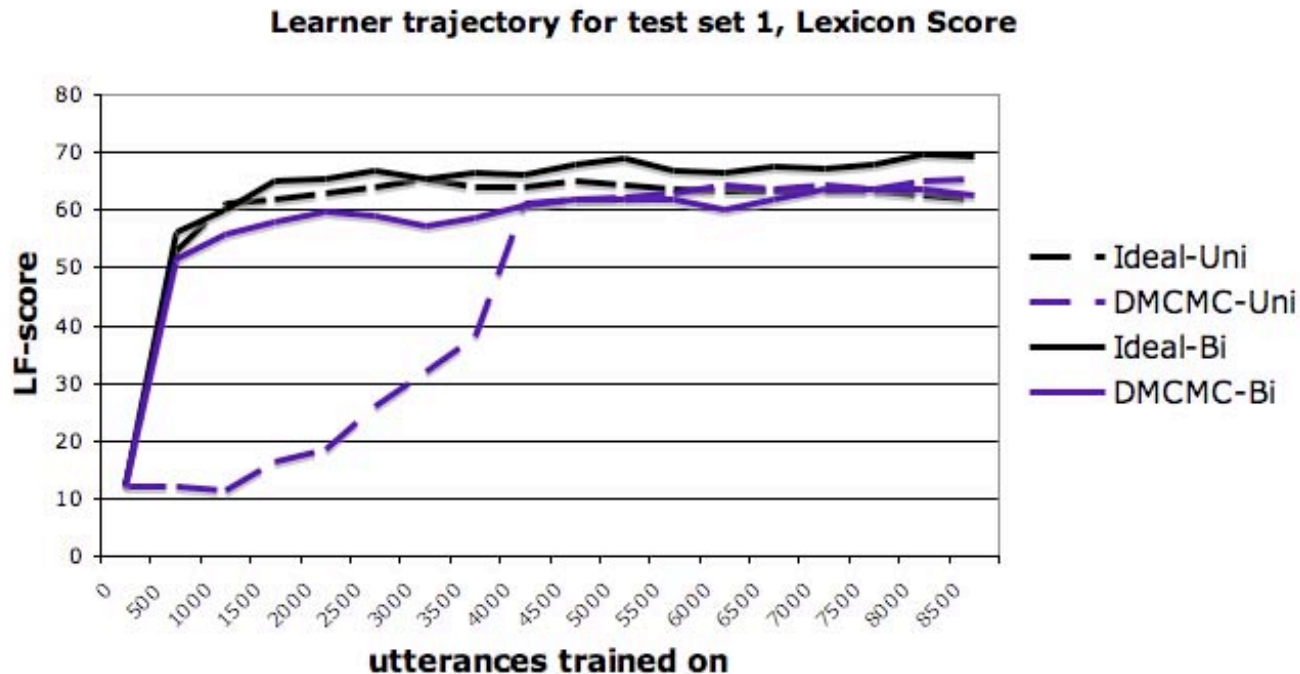
DMCMC vaues: Unigram  $d=1$ ; Bigram  $d = .25$ ;  $s = 20000$



- DMCMC actually performs better over time with respect to tokens when trained on successively larger quantities of data.

# Results: Standard vs. Decayed MCMC

DMCMC vaues: Unigram  $d=1$ ; Bigram  $d = .25$ ;  $s = 20000$



- Ideal (Standard MCMC) learner continually outperforms DMCMC for lexicon.
- Unigram DMCMC only does well after about 4000 utterances have been trained on.