Unsupervised machine learning as acquisition modeling

Cory Shain, Dept of Linguistics, The Ohio State University

Collaborators: William Bryce, Finale Doshi-Velez, Micha Elsner, Lifeng Jin, Victoria Krakovna, Timoth Miller, William Schuler, Lane Schwartz

16 Aug. 2017, MIT
Introduction
Unsupervised machine learning as acquisition modeling

- Both humans and computers can learn (aspects of) language
- Human language acquisition is not well understood
- Computer “language acquisition” is well understood
- Can we use machine learning to shed light on human learning?
Unsupervised machine learning as acquisition modeling

- Both humans and computers can learn (aspects of) language
- Human language acquisition is not well understood
- Computer “language acquisition” is well understood
- Can we use machine learning to shed light on human learning?
Unsupervised machine learning as acquisition modeling

- Both humans and computers can learn (aspects of) language
- Human language acquisition is not well understood
- Computer “language acquisition” is well understood
- Can we use machine learning to shed light on human learning?
Unsupervised machine learning as acquisition modeling

+ Both humans and computers can learn (aspects of) language
+ Human language acquisition is not well understood
+ Computer “language acquisition” is well understood
+ Can we use machine learning to shed light on human learning?
Roadmap for this talk

+ Motivation
  + Modeling lexical acquisition with unsupervised speech segmentation
  + Modeling grammar acquisition with unsupervised PCFG induction
  + Discussion and future directions
Roadmap for this talk

+ Motivation
+ Modeling lexical acquisition with unsupervised speech segmentation
+ Modeling grammar acquisition with unsupervised PCFG induction
+ Discussion and future directions
Roadmap for this talk

+ Motivation
+ Modeling lexical acquisition with unsupervised speech segmentation
+ Modeling grammar acquisition with unsupervised PCFG induction
+ Discussion and future directions
Roadmap for this talk

+ Motivation
+ Modeling lexical acquisition with unsupervised speech segmentation
+ Modeling grammar acquisition with unsupervised PCFG induction
+ Discussion and future directions
Motivation
Why model language acquisition computationally?

Of interest to both science and engineering
Why model language acquisition computationally?

Science:

- Test predictions of hypotheses about language acquisition
- Dissect the language learning problem
- Explore learnability of linguistic phenomena
Why model language acquisition computationally?

Science:
- Test predictions of hypotheses about language acquisition
- Dissect the language learning problem
- Explore learnability of linguistic phenomena
Why model language acquisition computationally?

+ Science:
  + Test predictions of hypotheses about language acquisition
  + Dissect the language learning problem
  + Explore learnability of linguistic phenomena
Why model language acquisition computationally?

Engineering:
+ Humans are better than computers at learning and using language
+ We learn from cheap and abundant sources of data
+ Low-resource NLP
+ Study and preservation of endangered languages
Why model language acquisition computationally?

**Engineering:**
- Humans are better than computers at learning and using language
- We learn from cheap and abundant sources of data
- Low-resource NLP
- Study and preservation of endangered languages
Why model language acquisition computationally?

+ **Engineering:**
  + Humans are better than computers at learning and using language
  + We learn from cheap and abundant sources of data
  + Low-resource NLP
  + Study and preservation of endangered languages
Why model language acquisition computationally?

Engineering:
- Humans are better than computers at learning and using language
- We learn from cheap and abundant sources of data
- Low-resource NLP
- Study and preservation of endangered languages
Why model language acquisition computationally?

Engineering:
- Humans are better than computers at learning and using language
- We learn from cheap and abundant sources of data
- Low-resource NLP
- Study and preservation of endangered languages
But...
NLP is (usually) cognitively implausible

- Normally requires lots of annotated data
  - Microsoft's Bing used 2100 hours of transcribed speech to train its speech recognizer (Dahl et al. 2011)
  - Humans don’t have direct access to the right answers
- Unrealistically large memory capacity
  - Human working memory constraints are severe compared to those of computers (Miller 1956; Cowan 2001; McElree 2001)
- Non-incremental processing
  - Humans process language incrementally (Marslen-Wilson 1979; Tanenhaus et al. 1985)
NLP is (usually) cognitively implausible

+ Normally requires lots of annotated data
  + Microsoft’s Bing used 2100 hours of transcribed speech to train its speech recognizer (Dahl et al. 2011)
  + Humans don’t have direct access to the right answers

+ Unrealistically large memory capacity
  + Human working memory constraints are severe compared to those of computers (Miller 1956; Cowan 2001; McElree 2001)

+ Non-incremental processing
  + Humans process language incrementally (Marslen-Wilson 1979; Tanenhaus et al. 1995)
NLP is (usually) cognitively implausible

+ Normally requires lots of annotated data
  + Microsoft’s Bing used 2100 hours of transcribed speech to train its speech recognizer (Dahl et al. 2011)
  + Humans don’t have direct access to the right answers

+ Unrealistically large memory capacity
  + Human working memory constraints are severe compared to those of computers (Miller 1956; Cowan 2001; McElree 2001)

+ Non-incremental processing
  + Humans process language incrementally (Marslen-Wilson 1973; Tanenhaus et al. 1995)
NLP is (usually) cognitively implausible

- Normally requires lots of annotated data
  - Microsoft’s Bing used 2100 hours of transcribed speech to train its speech recognizer (Dahl et al. 2011)
  - Humans don’t have direct access to the right answers
- Unrealistically large memory capacity
  - Human working memory constraints are severe compared to those of computers (Miller 1956; Cowan 2001; McElree 2001)
- Non-incremental processing
  - Humans process language incrementally (Marslen-Wilson 1979; Tanenhaus et al. 1988)
NLP is (usually) cognitively implausible

+ Normally requires lots of annotated data
  + Microsoft’s Bing used 2100 hours of transcribed speech to train its speech recognizer (Dahl et al. 2011)
  + Humans don’t have direct access to the right answers
+ Unrealistically large memory capacity
  + Human working memory constraints are severe compared to those of computers (Miller 1956; Cowan 2001; McElree 2001)
+ Non-incremental processing
  + Humans process language incrementally (Marslen-Wilson 1979; Tanenhaus et al. 1995)
NLP is (usually) cognitively implausible

- Normally requires lots of annotated data
  - Microsoft’s Bing used 2100 hours of transcribed speech to train its speech recognizer (Dahl et al. 2011)
  - Humans don’t have direct access to the right answers
- Unrealistically large memory capacity
  - Human working memory constraints are severe compared to those of computers (Miller 1956; Cowan 2001; McElree 2001)
- Non-incremental processing
  - Humans process language incrementally (Marslen-Wilson 1975; Tanenhaus et al. 1995)
NLP is (usually) cognitively implausible

- Normally requires lots of annotated data
  - Microsoft’s Bing used 2100 hours of transcribed speech to train its speech recognizer (Dahl et al. 2011)
  - Humans don’t have direct access to the right answers

- Unrealistically large memory capacity
  - Human working memory constraints are severe compared to those of computers (Miller 1956; Cowan 2001; McElree 2001)

- Non-incremental processing
  - Humans process language incrementally (Marslen-Wilson 1975; Tanenhaus et al. 1995)
Unsupervised NLP uses a more plausible training objective

+ Unsupervised = no direct access to the right answers
+ More similar training feedback to that received by humans
+ Can still be cognitively implausible in other ways
+ Cognitively-constrained unsupervised NLP can shed light on language acquisition
Unsupervised NLP uses a more plausible training objective

- Unsupervised = no direct access to the right answers
- More similar training feedback to that received by humans
- Can still be cognitively implausible in other ways
- Cognitively-constrained unsupervised NLP can shed light on language acquisition
Unsupervised NLP uses a more plausible training objective

- Unsupervised = no direct access to the right answers
- More similar training feedback to that received by humans
- Can still be cognitively implausible in other ways
- Cognitively-constrained unsupervised NLP can shed light on language acquisition
Unsupervised NLP uses a more plausible training objective

- Unsupervised = no direct access to the right answers
- More similar training feedback to that received by humans
- Can still be cognitively implausible in other ways
- Cognitively-constrained unsupervised NLP can shed light on language acquisition
This work: Modeling lexical and grammar acquisition

- Models of two related acquisition tasks:
  - **Lexical acquisition**: Learning to segment the speech signal
  - **Grammar acquisition**: Learning to parse
This work: Modeling lexical and grammar acquisition

Models of two related acquisition tasks:

- **Lexical acquisition**: Learning to segment the speech signal
- **Grammar acquisition**: Learning to parse
This work: Modeling lexical and grammar acquisition

- Models of two related acquisition tasks:
  - **Lexical acquisition**: Learning to segment the speech signal
  - **Grammar acquisition**: Learning to parse
Both of these models accept arbitrary naturally-occurring training data in any language.

C.f. computational models that use “toy” input (e.g. Elman 1991; Briscoe 2000; Fodor and Sakas 2004)

Our approach more accurately represents input to human learners.
This work: Modeling lexical and grammar acquisition

- Both of these models accept arbitrary naturally-occurring training data in any language.
- C.f. computational models that use “toy” input (e.g. Elman 1991; Briscoe 2000; Fodor and Sakas 2004)
- Our approach more accurately represents input to human learners.
This work: Modeling lexical and grammar acquisition

- Both of these models accept arbitrary naturally-occurring training data in any language
- C.f. computational models that use “toy” input (e.g. Elman 1991; Briscoe 2000; Fodor and Sakas 2004)
- Our approach more accurately represents input to human learners
Modeling lexical acquisition with unsupervised speech segmentation
Speech segmentation: Cognitive background

- Phonological memory limits may encourage sparse encodings (Baddeley and Hitch 1974)
- Thought to affect learning as well as processing (Baddeley, Gathercole, and Papagno 1998)
- We model this learning pressure by seeking compressible segmentations

Speech segmentation: Cognitive background

- Phonological memory limits may encourage sparse encodings (Baddeley and Hitch 1974)
- Thought to affect learning as well as processing (Baddeley, Gathercole, and Papagno 1998)
- We model this learning pressure by seeking compressible segmentations

Speech segmentation: Cognitive background

- Phonological memory limits may encourage sparse encodings (Baddeley and Hitch 1974)
- Thought to affect learning as well as processing (Baddeley, Gathercole, and Papagno 1998)
- We model this learning pressure by seeking compressible segmentations

Speech segmentation: Model overview

+ Two RNN’s:
  + Auto-encoder (AE) network: Reconstructs its input
  + Proposal network: Predicts segmentation points

Speech segmentation: Model overview

+ Two RNN’s:
  + Auto-encoder (AE) network: Reconstructs its input
  + Proposal network: Predicts segmentation points

Speech segmentation: Model overview

+ Two RNN's:
  + Auto-encoder (AE) network: Reconstructs its input
  + Proposal network: Predicts segmentation points

Speech segmentation: Auto-encoder network architecture

Speech segmentation: Segmenter network architecture

+ LSTM trained to predict segmentation probability at each time step

Speech segmentation: Model overview

- Overall segmentation loss is non-differentiable (segmentation decisions are hard)
- Estimated via importance sampling (e.g. Mnih et al. 2014; Xu et al. 2015), using reconstruction loss for scoring

Speech segmentation: Model overview

- Overall segmentation loss is non-differentiable (segmentation decisions are hard)
- Estimated via importance sampling (e.g. Mnih et al. 2014; Xu et al. 2015), using reconstruction loss for scoring

Speech segmentation: Model overview

- Memory limits simulated using:
  - Dropout
    - Simulates forgetting
    - Phonemes drop at rate $D_p$, words drop at rate $D_u$
  - LSTM hidden state size
    - Simulates capacity limitations
    - Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    - Too little memory might cause total reconstruction failure
    - Too much memory might not encourage efficient segmentations

Speech segmentation: Model overview

- Memory limits simulated using:
  - Dropout
    - Simulates forgetting
    - Phonemes drop at rate $D_p$, words drop at rate $D_u$
  - LSTM hidden state size
    - Simulates capacity limitations
    - Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    - Too little memory might cause total reconstruction failure
    - Too much memory might not encourage efficient segmentations

Micha Elsner and Cory Shain (to appear). “Speech segmentation with a neural encoder model of working memory”. In: *EMNLP 2017*
Speech segmentation: Model overview

+ Memory limits simulated using:
  + Dropout
    + Simulates forgetting
    + Phonemes drop at rate $D_p$, words drop at rate $D_u$
  + LSTM hidden state size
    + Simulates capacity limitations
    Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    Too little memory might cause total reconstruction failure
    Too much memory might not encourage efficient segmentations

Speech segmentation: Model overview

+ Memory limits simulated using:
  + Dropout
    + Simulates forgetting
    + Phonemes drop at rate $D_p$, words drop at rate $D_u$
  + LSTM hidden state size
    + Simulates capacity limitations
    Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    Too little memory might cause total reconstruction failure
    Too much memory might not encourage efficient segmentations

Speech segmentation: Model overview

+ Memory limits simulated using:
  + Dropout
    + Simulates forgetting
    + Phonemes drop at rate $D_p$, words drop at rate $D_u$
  + LSTM hidden state size
    + Simulates capacity limitations
    + Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    + Too little memory might cause total reconstruction failure
    + Too much memory might not encourage efficient segmentations

Speech segmentation: Model overview

- Memory limits simulated using:
  - Dropout
    - Simulates forgetting
    - Phonemes drop at rate $D_p$, words drop at rate $D_u$
  - LSTM hidden state size
    - Simulates capacity limitations
    - Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    - Too little memory might cause total reconstruction failure
    - Too much memory might not encourage efficient segmentations

Speech segmentation: Model overview

+ Memory limits simulated using:
  + Dropout
    + Simulates forgetting
    + Phonemes drop at rate $D_p$, words drop at rate $D_u$
  + LSTM hidden state size
    + Simulates capacity limitations
    + Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    + Too little memory might cause total reconstruction failure
    + Too much memory might not encourage efficient segmentations

Speech segmentation: Model overview

- Memory limits simulated using:
  - Dropout
    - Simulates forgetting
    - Phonemes drop at rate $D_p$, words drop at rate $D_u$
  - LSTM hidden state size
    - Simulates capacity limitations
    - Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    - Too little memory might cause total reconstruction failure
    - Too much memory might not encourage efficient segmentations

Speech segmentation: Model overview

- Memory limits simulated using:
  - Dropout
    - Simulates forgetting
    - Phonemes drop at rate $D_p$, words drop at rate $D_u$
  - LSTM hidden state size
    - Simulates capacity limitations
    - Number of hidden units in phonological ($H_p$) and utterance ($H_u$) auto-encoders
    - Too little memory might cause total reconstruction failure
    - Too much memory might not encourage efficient segmentations

Speech segmentation: Input

- Architecture is very flexible
- Can accept any vectorial representation of the input sequence
- Characters (1-hot)
- Acoustic features (MFCC)
- First system to perform unsupervised segmentation of either text or acoustics using same code base

Speech segmentation: Input

+ Architecture is very flexible
+ Can accept any vectorial representation of the input sequence
  + Characters (1-hot)
  + Acoustic features (MFCC)
+ First system to perform unsupervised segmentation of either text or acoustics using same code base

Speech segmentation: Input

- Architecture is very flexible
- Can accept any vectorial representation of the input sequence
- Characters (1-hot)
- Acoustic features (MFCC)
- First system to perform unsupervised segmentation of either text or acoustics using same code base

Speech segmentation: Input

+ Architecture is very flexible
+ Can accept any vectorial representation of the input sequence
+ Characters (1-hot)
+ Acoustic features (MFCC)
+ First system to perform unsupervised segmentation of either text or acoustics using same code base

Speech segmentation: Input

- Architecture is very flexible
- Can accept any vectorial representation of the input sequence
- Characters (1-hot)
- Acoustic features (MFCC)
- First system to perform unsupervised segmentation of either text or acoustics using same code base

Speech segmentation: Experiments

- Text: Brent corpus (Brent 1999)
- Acoustics: Zerospeech ’15 English (Versteegh et al. 2015)

Speech segmentation: Results (Brent)

<table>
<thead>
<tr>
<th></th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our system</td>
<td>81</td>
<td>85</td>
<td>83</td>
<td>72</td>
</tr>
</tbody>
</table>

Examples:

- `yu wanttu si D6bUk`
  
  `You wantto see thebook?`

- `oke yusIt D* &nd 9l pUty) Suz b&kan`
  
  `Okay, yousit there and I’ll putyour shoes backon`

- `&nd IUK&t WAt D6kltiz pleIN wiT`
  
  `And lookat what thekitty’s playing with`

- `dld yu kQnt Ol6v DEm`
  
  `Did you count allof them?`

- `wan6 IUK&t 6nADR bUk`
  
  `Wanna lookat another book?`

Speech segmentation: Results (Brent)

<table>
<thead>
<tr>
<th></th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our system</td>
<td>81</td>
<td>85</td>
<td>83</td>
<td>72</td>
</tr>
</tbody>
</table>

Examples:

+ you want to see the book?
+ okay, you sit there and I'll put your shoes back on
+ and look at what the kitty's playing with
+ did you count all of them?
+ wanna look at another book?
+ if puppy bites, puppy gets panked.

Speech segmentation: Results (Brent)

<table>
<thead>
<tr>
<th></th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our system</td>
<td>81</td>
<td>85</td>
<td>83</td>
<td>72</td>
</tr>
</tbody>
</table>

Examples:

+ yu wanttu si D6bUk
  *You wantto see thebook?*
+ oke YusIt D* nnd 9I pUty) Suz b&kan
  *Okay, yousit there and I'll putyour shoes backon*
+ &nd IUk&t WA* D6kltiz pleIN wiT
  *And lookat what thekitty's playing with*
+ dId yu kQnt Ol6v DEm
  *Did you count allof them?*
+ wan6 IUk&t 6nADR bUk
  *Wanna lookat another book?*
+ If pApi b9ts pApi gEtss p&Nkt
  *If puppy bites, puppy getss panked.*

Micha Elsner and Cory Shain (to appear). “Speech segmentation with a neural encoder model of working memory”. In: *EMNLP 2017*
Speech segmentation: Results (Brent)

<table>
<thead>
<tr>
<th></th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our system</td>
<td>81</td>
<td>85</td>
<td>83</td>
<td>72</td>
</tr>
</tbody>
</table>

Examples:

- yu wanttu si D6bUk
  *You wantto see the book?*
- oke yusIt D* &nd 9l pUty) Suz b&kan
  *Okay, yousit there and I'll put your shoes backon*
- &nd IUk&t WAAt D6kItiz pleIN wIT
  *And lookat what the kitty’s playing with*
- dld yu kQnt Ol6v DEm
  *Did you count allof them?*
- wan6 IUk&t 6nADR bUk
  *Wanna lookat another book?*
- If pApi b9ts pApi gEtss p&Nkt
  *If puppy bites, puppy getss panked.*

Micha Elsner and Cory Shain (to appear). “Speech segmentation with a neural encoder model of working memory”. In: *EMNLP 2017*
Speech segmentation: Results (Brent)

<table>
<thead>
<tr>
<th></th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our system</td>
<td>81</td>
<td>85</td>
<td>83</td>
<td>72</td>
</tr>
</tbody>
</table>

+ Examples:
  + yu wanttu si D6bUk
    *You want to see the book?*
  + oke yusIt D* &nd 9l pUty) Suz b&kan
    *Okay, you sit there and I'll put your shoes back on*
  + &nd lUk&t WAi D6kItiz pleIN wIT
    *And look at what the kitty's playing with*
  + dId yu kQnt Ol6v DEm
    *Did you count all of them?*
  + wan6 lUk&t 6nADR bUk
    *Wanna look at another book?*
  + If pApi b9ts pApi gEtss p&Nkt
    *If puppy bites, puppy getss panked.*

Speech segmentation: Results (Brent)

<table>
<thead>
<tr>
<th></th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our system</td>
<td>81</td>
<td>85</td>
<td>83</td>
<td>72</td>
</tr>
</tbody>
</table>

Examples:

- yu wanttu si D6bUk  
  *You wantto see thebook?*
- oke yusIt D* &nd 9l pUty) Suz b&kan  
  *Okay, yousit there and I'll putyour shoes backon*
- &nd IUk&t WAAt D6kItiz pleIIN wIIT  
  *And lookat what thekitty’s playing with*
- dld yu kQnt Ol6v DEm  
  *Did you count allof them?*
- wan6 IUk&t 6nADR bUk  
  *Wanna lookat another book?*
- If pApi b9ts pApi gEtss p&Nkt  
  *If puppy bites, puppy getss panked.*

Speech segmentation: Results (Brent)

<table>
<thead>
<tr>
<th></th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our system</td>
<td>81</td>
<td>85</td>
<td>83</td>
<td>72</td>
</tr>
</tbody>
</table>

Examples:

- you want tu si D6bUk
  You want to see the book?
- oke yuslt D* &nd 9I pUty) Suz b&kan
  Okay, you sit there and I'll put your shoes back on
- &nd IUk&t WA t D6kltiz pleIN wIT
  And look at what the kitty's playing with
- dld yu kQnt Ol6v DEm
  Did you count all of them?
- wan6 IUk&t 6nADR bUk
  Wanna look at another book?
- If pApi b9ts pApi gEtss p&Nkt
  If puppy bites, puppy getss panked.

### Speech segmentation: Results (Brent)

<table>
<thead>
<tr>
<th>System</th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goldwater 09</td>
<td>90</td>
<td>74</td>
<td>87</td>
<td>74</td>
</tr>
<tr>
<td>Johnson 09</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>88</td>
</tr>
<tr>
<td>Berg-Kirkpatrick 10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>88</td>
</tr>
<tr>
<td>Fleck 08</td>
<td>95</td>
<td>74</td>
<td>83</td>
<td>71</td>
</tr>
<tr>
<td>Our system</td>
<td>81</td>
<td>85</td>
<td>83</td>
<td>72</td>
</tr>
</tbody>
</table>

## Speech segmentation: Results (Zerospeech ’15)

<table>
<thead>
<tr>
<th>System</th>
<th>Bd P</th>
<th>Bd R</th>
<th>Bd F</th>
<th>Wd F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lyzinski 15</td>
<td>18.8</td>
<td>64.0</td>
<td>29.0</td>
<td>2.4</td>
</tr>
<tr>
<td>Räsänen 15</td>
<td>75.7</td>
<td>33.7</td>
<td>46.7</td>
<td>9.6</td>
</tr>
<tr>
<td>Räsänen new</td>
<td>61.1</td>
<td>50.1</td>
<td>55.2</td>
<td>12.4</td>
</tr>
<tr>
<td>Kamper 16</td>
<td>66.5</td>
<td>58.8</td>
<td>62.4</td>
<td>20.6</td>
</tr>
<tr>
<td>Ours</td>
<td>62.4</td>
<td>43.2</td>
<td>51.1</td>
<td>9.3</td>
</tr>
</tbody>
</table>

Micha Elsner and Cory Shain (to appear). “Speech segmentation with a neural encoder model of working memory”. In: *EMNLP 2017*
Speech segmentation: Dropout

Dropout and memory limits encourage better segmentations

Speech segmentation: Conclusion

Our results support the hypothesis that limited phonological memory facilitates lexical acquisition by encouraging efficient segmentation.

Modeling grammar acquisition with unsupervised PCFG induction
Grammar induction: Cognitive background

Humans have been shown to use distributional statistics in language acquisition (Saffran et al. 1999)

Cognitively-constrained grammar induction allows us to study:

- Utility of word distributions to syntax acquisition
- Advantages/disadvantages of cognitive constraints

Humans have been shown to use distributional statistics in language acquisition (Saffran et al. 1999)

Cognitively-constrained grammar induction allows us to study:
- Utility of word distributions to syntax acquisition
- Advantages/disadvantages of cognitive constraints

Grammar induction: Cognitive background

Humans have been shown to use distributional statistics in language acquisition (Saffran et al. 1999)

Cognitively-constrained grammar induction allows us to study:
  - Utility of word distributions to syntax acquisition
  - Advantages/disadvantages of cognitive constraints

Grammar induction: Cognitive background

- Humans have been shown to use distributional statistics in language acquisition (Saffran et al. 1999)
- Cognitively-constrained grammar induction allows us to study:
  - Utility of word distributions to syntax acquisition
  - Advantages/disadvantages of cognitive constraints

Grammar induction: Previous work

+ Several raw-text constituency parsers exist (e.g. Seginer 2007; Ponvert, Baldridge, and Erik 2011)
+ No system besides ours is
  + Depth-bounded (memory-limited)
  + Incremental
+ Typically constituents are not labeled

Grammar induction: Previous work

+ Several raw-text constituency parsers exist (e.g. Seginer 2007; Ponvert, Baldridge, and Erik 2011)
+ No system besides ours is
  + Depth-bounded (memory-limited)
  + Incremental
+ Typically constituents are not labeled

Grammar induction: Previous work

+ Several raw-text constituency parsers exist (e.g. Seginer 2007; Ponvert, Baldridge, and Erik 2011)
+ No system besides ours is
  + Depth-bounded (memory-limited)
  + Incremental
+ Typically constituents are not labeled

Grammar induction: Previous work

- Several raw-text constituency parsers exist (e.g. Seginer 2007; Ponvert, Baldridge, and Erik 2011)
- No system besides ours is
  - Depth-bounded (memory-limited)
  - Incremental
- Typically constituents are not labeled

Grammar induction: Previous work

- Several raw-text constituency parsers exist (e.g. Seginer 2007; Ponvert, Baldridge, and Erik 2011)
- No system besides ours is
  - Depth-bounded (memory-limited)
  - Incremental
- Typically constituents are not labeled

Grammar induction: Model overview

- Bayesian depth-bounded incremental left-corner PCFG induction system
- Parses with depth-bounded hierarchical hidden Markov model (Schuler et al. 2010)
- Trained using block Gibbs sampling
- Produces a full labeled tree structure and PCFG model

Grammar induction: Model overview

- Bayesian depth-bounded incremental left-corner PCFG induction system
- Parses with depth-bounded hierarchical hidden Markov model (Schuler et al. 2010)
- Trained using block Gibbs sampling
- Produces a full labeled tree structure and PCFG model

Grammar induction: Model overview

+ Bayesian depth-bounded incremental left-corner PCFG induction system
+ Parses with depth-bounded hierarchical hidden Markov model (Schuler et al. 2010)
+ Trained using block Gibbs sampling
+ Produces a full labeled tree structure and PCFG model

Grammar induction: Model overview

+ Bayesian depth-bounded incremental left-corner PCFG induction system
+ Parses with depth-bounded hierarchical hidden Markov model (Schuler et al. 2010)
+ Trained using block Gibbs sampling
+ Produces a full labeled tree structure and PCFG model
Grammar induction: Experiment

- Experimental conditions designed to mimic conditions of early language learning:
  - **Child-directed input**: Child-directed utterances from the Eve corpus of Brown (1973), distributed with CHILDES (MacWhinney 2000)
  - **Limited depth**: Depth was limited to 2
    - Children have more severe working memory limits than adults (Gathercole 1998)
    - Greater depths rarely needed for child-directed utterances
  - **Small hypothesis space** (Newport 1990): 4 left child categories, 4 right child categories, 8 parts of speech

Grammar induction: Experiment

+ Experimental conditions designed to mimic conditions of early language learning:
  + **Child-directed input**: Child-directed utterances from the Eve corpus of Brown (1973), distributed with CHILDES (MacWhinney 2000)
  + **Limited depth**: Depth was limited to 2
    - Children have more severe working memory limits than adults (Gathercole 1998)
    - Greater depths rarely needed for child-directed utterances
  + **Small hypothesis space** (Newport 1990): 4 left child categories, 4 right child categories, 8 parts of speech

Grammar induction: Experiment

+ Experimental conditions designed to mimic conditions of early language learning:
  + **Child-directed input**: Child-directed utterances from the Eve corpus of Brown (1973), distributed with CHILDES (MacWhinney 2000)
  + **Limited depth**: Depth was limited to 2
    + Children have more severe working memory limits than adults (Gathercole 1998)
    + Greater depths rarely needed for child-directed utterances
  + Small hypothesis space (Newport 1990): 4 left child categories, 4 right child categories, 8 parts of speech

Grammar induction: Experiment

+ Experimental conditions designed to mimic conditions of early language learning:
  + **Child-directed input**: Child-directed utterances from the Eve corpus of Brown (1973), distributed with CHILDES (MacWhinney 2000)
  + **Limited depth**: Depth was limited to 2
    + Children have more severe working memory limits than adults (Gathercole 1998)
    + Greater depths rarely needed for child-directed utterances
  + **Small hypothesis space (Newport 1990)**: 4 left child categories, 4 right child categories, 8 parts of speech

Grammar induction: Experiment

- Experimental conditions designed to mimic conditions of early language learning:
  - **Child-directed input**: Child-directed utterances from the Eve corpus of Brown (1973), distributed with CHILDES (MacWhinney 2000)
  - **Limited depth**: Depth was limited to 2
    - Children have more severe working memory limits than adults (Gathercole 1998)
    - Greater depths rarely needed for child-directed utterances
  - **Small hypothesis space** (Newport 1990): 4 left child categories, 4 right child categories, 8 parts of speech

Grammar induction: Experiment

+ Experimental conditions designed to mimic conditions of early language learning:
  + **Child-directed input**: Child-directed utterances from the Eve corpus of Brown (1973), distributed with CHILDES (MacWhinney 2000)
  + **Limited depth**: Depth was limited to 2
    - Children have more severe working memory limits than adults (Gathercole 1998)
    - Greater depths rarely needed for child-directed utterances
  + **Small hypothesis space (Newport 1990)**: 4 left child categories, 4 right child categories, 8 parts of speech

Grammar induction: Evaluation

+ **Gold standard:** Hand-corrected PTB-style trees for Eve (Pearl and Sprouse 2013)
+ **Competitors:**
  - CCL (Seginer 2007)
  - UPPARSE (Ponvert, Baldridge, and Eck 2011)
  - BMNN-OMV (Christodoulopoulos, Goldwater, and Steedman 2012)

Grammar induction: Evaluation

- **Gold standard:** Hand-corrected PTB-style trees for Eve (Pearl and Sprouse 2013)
- **Competitors:**
  - CCL (Seginer 2007)
  - UPPARSE (Ponvert, Baldridge, and Erik 2011)
  - BMMM+DMV (Christodoulopoulos, Goldwater, and Steedman 2012)

Grammar induction: Evaluation

- **Gold standard**: Hand-corrected PTB-style trees for Eve (Pearl and Sprouse 2013)
- **Competitors**:
  - CCL (Seginer 2007)
  - UPPARSE (Ponvert, Baldridge, and Erik 2011)
  - BMMM+DMV (Christodoulopoulos, Goldwater, and Steedman 2012)

*Cory Shain et al. (2016)*. “Memory-bounded left-corner unsupervised grammar induction on child-directed input”. In: *Proceedings of The 26th International Conference on Computational Linguistics*. Osaka, pp. 964–975
Grammar induction: Evaluation

+ **Gold standard:** Hand-corrected PTB-style trees for Eve (Pearl and Sprouse 2013)

+ **Competitors:**
  + CCL (Seginer 2007)
  + UPPARSE (Ponvert, Baldridge, and Erik 2011)
  + BMMM+DMV (Christodoulopoulos, Goldwater, and Steedman 2012)

*Cory Shain et al. (2016)*. “Memory-bounded left-corner unsupervised grammar induction on child-directed input”. In: *Proceedings of The 26th International Conference on Computational Linguistics*. Osaka, pp. 964–975
Grammar induction: Evaluation

- **Gold standard:** Hand-corrected PTB-style trees for Eve (Pearl and Sprouse 2013)
- **Competitors:**
  - CCL (Seginer 2007)
  - UPPARSE (Ponvert, Baldridge, and Erik 2011)
  - BMMM+DMV (Christodoulopoulos, Goldwater, and Steedman 2012)

Grammar induction: COLING results

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our system</td>
<td>68.83</td>
<td>57.18</td>
<td>62.47</td>
</tr>
<tr>
<td>Random baseline (Ours 1st iter)</td>
<td>51.69</td>
<td>38.75</td>
<td>44.30</td>
</tr>
</tbody>
</table>

Unlabeled bracketing accuracy on Eve

### Grammar induction: COLING results

<table>
<thead>
<tr>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPPARSE</td>
<td>60.50</td>
<td>51.96</td>
<td>55.90</td>
</tr>
<tr>
<td>CCL</td>
<td>64.70</td>
<td>53.47</td>
<td>58.55</td>
</tr>
<tr>
<td>BMMM+DMV</td>
<td>63.63</td>
<td>64.02</td>
<td>63.82</td>
</tr>
<tr>
<td>Our system</td>
<td>68.83</td>
<td>57.18</td>
<td>62.47</td>
</tr>
<tr>
<td>Random baseline (Ours 1st iter)</td>
<td>51.69</td>
<td>38.75</td>
<td>44.30</td>
</tr>
</tbody>
</table>

Unlabeled bracketing accuracy on Eve

---

*Cory Shain et al. (2016)*. “Memory-bounded left-corner unsupervised grammar induction on child-directed input”. In: *Proceedings of The 26th International Conference on Computational Linguistics*. Osaka, pp. 964–975
Grammar induction: Error analysis

Percent gold noun phrases (NPs) discovered
Grammar induction: Error analysis

Percent gold verb phrases (VPs) discovered
Grammar induction: Error analysis

Part-of-speech tagging (V-Measure)
Grammar induction: Constructions of interest

Subject-auxiliary inversion: (c.f. Chomsky 1968)

Grammar induction: Constructions of interest

Ditransitive:

Grammar induction: Hot off the press

- Since COLING:
  - Merged left, right, and PoS category spaces
  - Depth=1 run on Eve got $F_1 = 71$
- Additional constraints on search space facilitate learning
Grammar induction: Hot off the press

Since COLING:
- Merged left, right, and PoS category spaces
- Depth=1 run on Eve got $F_1 = 71$

- Additional constraints on search space facilitate learning
Grammar induction: Hot off the press

Since COLING:
  - Merged left, right, and PoS category spaces
  - Depth=1 run on Eve got $F_1 = 71$

- Additional constraints on search space facilitate learning
Grammar induction: Hot off the press

- Since COLING:
  - Merged left, right, and PoS category spaces
  - Depth=1 run on Eve got $F_1 = 71$
- Additional constraints on search space facilitate learning
Grammar induction: Conclusions

- Word distributions contain a substantial amount of information about English syntax
- This information is detectible by a cognitively-constrained learner
- There is still much room for improvement
  - Some residue may be unlearnable without additional cues (e.g., vision) or innate bias
  - Some residue may be captured by improved induction techniques
Word distributions contain a substantial amount of information about English syntax. This information is detectible by a cognitively-constrained learner. There is still much room for improvement. Some residue may be unlearnable without additional cues (e.g., vision) or innate bias. Some residue may be captured by improved induction techniques.
Grammar induction: Conclusions

- Word distributions contain a substantial amount of information about English syntax
- This information is detectible by a cognitively-constrained learner
- There is still much room for improvement
  - Some residue may be unlearnable without additional cues (e.g. vision) or innate bias
  - Some residue may be captured by improved induction techniques
Grammar induction: Conclusions

- Word distributions contain a substantial amount of information about English syntax.
- This information is detectible by a cognitively-constrained learner.
- There is still much room for improvement.
  - Some residue may be unlearnable without additional cues (e.g. vision) or innate bias.
  - Some residue may be captured by improved induction techniques.
Grammar induction: Conclusions

- Word distributions contain a substantial amount of information about English syntax.
- This information is detectible by a cognitively-constrained learner.
- There is still much room for improvement.
  - Some residue may be unlearnable without additional cues (e.g. vision) or innate bias.
  - Some residue may be captured by improved induction techniques.
Conclusion
Unsupervised NLP approaches to speech segmentation and parsing can shed light on language acquisition.

Speech segmenter results show that memory pressures encourage learning efficient representations.

Grammar induction results show that much syntax can be acquired from word distributions alone.
Conclusion

+ Unsupervised NLP approaches to speech segmentation and parsing can shed light on language acquisition
+ Speech segmenter results show that memory pressures encourage learning efficient representations
+ Grammar induction results show that much syntax can be acquired from word distributions alone
Conclusion

+ Unsupervised NLP approaches to speech segmentation and parsing can shed light on language acquisition
+ Speech segmenter results show that memory pressures encourage learning efficient representations
+ Grammar induction results show that much syntax can be acquired from word distributions alone
Conclusion

+ Together, these systems might take us closer to a full computational model of language acquisition
+ Dense word embeddings can be obtained from raw speech
+ (Soon:) PCFG can be trained from dense word representations
+ If pipelined, these approaches could go from acoustics to syntax trees, completely unsupervised
Together, these systems might take us closer to a full computational model of language acquisition. Dense word embeddings can be obtained from raw speech. (Soon:) PCFG can be trained from dense word representations. If pipelined, these approaches could go from acoustics to syntax trees, completely unsupervised.
Conclusion

- Together, these systems might take us closer to a full computational model of language acquisition
- Dense word embeddings can be obtained from raw speech
- (Soon:) PCFG can be trained from dense word representations
- If pipelined, these approaches could go from acoustics to syntax trees, completely unsupervised
Together, these systems might take us closer to a full computational model of language acquisition

- Dense word embeddings can be obtained from raw speech
- (Soon:) PCFG can be trained from dense word representations
- If pipelined, these approaches could go from acoustics to syntax trees, completely unsupervised
Thank you!

To you, co-authors, anonymous reviewers of submitted papers, and members of various discussion groups who gave feedback.

Computations for this project were run on a Titan-X GPU donated by the NVIDIA Hardware Grant program and on the Ohio Supercomputer (1987). Funding was provided by NSF #1422987.

This project was sponsored by the Defense Advanced Research Projects Agency award #HR0011-15-2-0022. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

**Segmenter Github:**
https://github.com/melsner/neural-segmentation

**Parser Github:**
https://github.com/tmills/uhhmm/


References II


References III


Pearl, Lisa and Jon Sprouse (2013). “Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem”. In: *Language Acquisition* 20, pp. 23–68.


Appendix
Speech segmentation: Algorithm

1. For each training epoch:
   1.1 For each batch of \( n \) utterances in the training data
      1.1.1 Generate a proposal distribution (segmenter network output)
      1.1.2 Sample \( m \) segmentations from proposal distribution
      1.1.3 Compute new proposal distribution as normalized sum of segmentations weighted by
            reconstruction loss
      1.1.4 Train segmenter network on new proposal distribution

Speech segmentation: Algorithm

1. For each training epoch:
   1.1 For each batch of $n$ utterances in the training data
      1.1.1 Generate a proposal distribution (segmenter network output)
      1.1.2 Sample $m$ segmentations from proposal distribution
      1.1.3 Compute new proposal distribution as normalized sum of segmentations weighted by reconstruction loss
      1.1.4 Train segmenter network on new proposal distribution

Speech segmentation: Algorithm

1. For each training epoch:
   1.1 For each batch of $n$ utterances in the training data
      1.1.1 Generate a proposal distribution (segmenter network output)
      1.1.2 Sample $m$ segmentations from proposal distribution
      1.1.3 Compute new proposal distribution as normalized sum of segmentations weighted by reconstruction loss
      1.1.4 Train segmenter network on new proposal distribution

Speech segmentation: Algorithm

1. For each training epoch:
   1.1 For each batch of $n$ utterances in the training data
      1.1.1 Generate a proposal distribution (segmenter network output)
      1.1.2 Sample $m$ segmentations from proposal distribution
      1.1.3 Compute new proposal distribution as normalized sum of segmentations weighted by reconstruction loss
      1.1.4 Train segmenter network on new proposal distribution

Speech segmentation: Algorithm

1. For each training epoch:
   1.1 For each batch of $n$ utterances in the training data
      1.1.1 Generate a proposal distribution (segmenter network output)
      1.1.2 Sample $m$ segmentations from proposal distribution
      1.1.3 Compute new proposal distribution as normalized sum of segmentations weighted by reconstruction loss
      1.1.4 Train segmenter network on new proposal distribution

Speech segmentation: Algorithm

1. For each training epoch:
   1.1 For each batch of $n$ utterances in the training data
      1.1.1 Generate a proposal distribution (segmenter network output)
      1.1.2 Sample $m$ segmentations from proposal distribution
      1.1.3 Compute new proposal distribution as normalized sum of segmentations weighted by reconstruction loss
      1.1.4 Train segmenter network on new proposal distribution

Speech segmentation: Sampling procedure

Given a set of \( m \) sampled boundary sequences \( B_1..B_m \) with associated reconstruction losses \( L_1...L_m \):

\[
P(x|B_i) = \frac{P(B_i|x)P(B_i)}{P(x)} \approx \frac{\exp(L_i)}{\sum_j \exp(L_j)} \tag{1}
\]

\[
w_i^t = \frac{P(x|B_i)}{P_{seg}(B_i^t)} \tag{2}
\]

\[
\mathbb{E}[B(t)] \approx \frac{1}{\sum_i w_i^t} \sum_i w_i^t B_i^t \tag{3}
\]
Importance sampling caused oversegmentation

We suspect that this is due to non-independence between samples, exaggerated by longer sequences

Acoustic results were obtained via 1-best sampling
Speech segmentation: Experiment parameters

Brent:

- Max characters per utterance: 30
- Max words per utterance: 10
- Max characters per word: 7
- Phonological AE hidden units: 80
- Utterance AE hidden units: 400
- Segmenter hidden units: 100
- Phonological AE dropout probability: 0.5
- Utterance AE dropout probability: 0.25
Speech segmentation: Experiment parameters

**Zerospeech:**
- Max frames per utterance: 400
- Max words per utterance: 16
- Max frames per word: 100
- Phonological AE hidden units: 20
- Utterance AE hidden units: 400
- Segmenter hidden units: 1500
- Phonological AE dropout probability: 0
- Utterance AE dropout probability: 0.25
1. **Initialization**: Randomly sample HHMM parameters

2. For each training iteration:
   
   2.1 **Parsing**: For each sentence in input:
      
      2.1.1 **Forward pass**: Compute posterior over HHMM states left to right
      
      2.1.2 **Backward pass**: Sample states right to left

   2.2 Update HHMM parameters from sampled counts

---

Grammar induction: Algorithm

1. **Initialization**: Randomly sample HHMM parameters

2. For each training iteration:
   2.1 **Parsing**: For each sentence in input:
      2.1.1 Forward pass: Compute posterior over HHMM states left to right
      2.1.2 Backward pass: Sample states right to left
   2.2 Update HHMM parameters from sampled counts

Grammar induction: Algorithm

1. **Initialization**: Randomly sample HHMM parameters
2. For each training iteration:
   2.1 **Parsing**: For each sentence in input:
      2.1.1 **Forward pass**: Compute posterior over HHMM states left to right
      2.1.2 **Backward pass**: Sample states right to left
   2.2 Update HHMM parameters from sampled counts

Grammar induction: Algorithm

1. **Initialization**: Randomly sample HHMM parameters
2. For each training iteration:
   2.1 **Parsing**: For each sentence in input:
      2.1.1 **Forward pass**: Compute posterior over HHMM states left to right
      2.1.2 **Backward pass**: Sample states right to left
   2.2 Update HHMM parameters from sampled counts

*Cory Shain et al. (2016)*. “Memory-bounded left-corner unsupervised grammar induction on child-directed input”. In: *Proceedings of The 26th International Conference on Computational Linguistics*. Osaka, pp. 964–975
Grammar induction: Algorithm

1. **Initialization**: Randomly sample HHMM parameters
2. For each training iteration:
   2.1 **Parsing**: For each sentence in input:
      2.1.1 **Forward pass**: Compute posterior over HHMM states left to right
      2.1.2 **Backward pass**: Sample states right to left
   2.2 Update HHMM parameters from sampled counts

Grammar induction: Algorithm

1. **Initialization**: Randomly sample HHMM parameters

2. For each training iteration:
   2.1 **Parsing**: For each sentence in input:
      2.1.1 **Forward pass**: Compute posterior over HHMM states left to right
      2.1.2 **Backward pass**: Sample states right to left

2.2 Update HHMM parameters from sampled counts

---

Grammar induction: HHMM Graphical model
Grammar induction: Punctuation

- Punctuation poses a problem — keep or remove?
  - **Remove:** Doesn’t exist in input to human learners.
  - **Keep:** Might be proxy for intonational phrasal cues.

- Punctuation was kept in training data in main result presented above.

- We did an additional UHHMM run trained on data with punctuation removed (2000 iterations).
Grammar induction: Full COLING Results

<table>
<thead>
<tr>
<th></th>
<th>With punct</th>
<th></th>
<th></th>
<th>No punct</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
<td>R</td>
<td>F1</td>
</tr>
<tr>
<td>UPPARSE</td>
<td>60.50</td>
<td>51.96</td>
<td>55.90</td>
<td>38.17</td>
<td>48.38</td>
<td>42.67</td>
</tr>
<tr>
<td>CCL</td>
<td>64.70</td>
<td>53.47</td>
<td>58.55</td>
<td>56.87</td>
<td>47.69</td>
<td>51.88</td>
</tr>
<tr>
<td>BMMM+DMV (directed)</td>
<td>62.08</td>
<td>62.51</td>
<td>62.30</td>
<td>61.01</td>
<td>59.24</td>
<td>60.14</td>
</tr>
<tr>
<td>BMMM+DMV (undirected)</td>
<td>63.63</td>
<td><strong>64.02</strong></td>
<td><strong>63.82</strong></td>
<td>61.34</td>
<td><strong>59.33</strong></td>
<td><strong>60.32</strong></td>
</tr>
<tr>
<td>UHHMM-4000, binary</td>
<td>46.68</td>
<td>58.28</td>
<td>51.84</td>
<td>37.62</td>
<td>46.97</td>
<td>41.78</td>
</tr>
<tr>
<td>UHHMM-4000, flattened</td>
<td><strong>68.83</strong></td>
<td>57.18</td>
<td>62.47</td>
<td><strong>61.78</strong></td>
<td>45.52</td>
<td>52.42</td>
</tr>
<tr>
<td>Right-branching</td>
<td>68.73</td>
<td><strong>85.81</strong></td>
<td><strong>76.33</strong></td>
<td>68.73</td>
<td><strong>85.81</strong></td>
<td><strong>76.33</strong></td>
</tr>
</tbody>
</table>

Table 1: Parsing accuracy on Eve with and without punctuation (phrasal cues) in the input. The UHHMM systems were given 8 PoS categories while the BMMM+DMV systems were given 45. UPPARSE and CCL do not learn PoS tags. Only the UHHMM systems model limited working memory capacity or incremental left-corner parsing.

Grammar induction: Newer results

Learning curves on Eve
Grammar induction: Newer results

Category learning on Eve