Probabilistic Learning of Labeled Grammars

William Schuler
Dept. of Linguistics, The Ohio State University

September 6, 2017

Overview

Organization of this Talk:

Overview

Organization of this Talk:

1. What does it mean to learn a language?

Overview

Organization of this Talk:

1. What does it mean to learn a language?
   Do we learn possible rules or probabilistically weighted rules?

Overview

Organization of this Talk:

1. What does it mean to learn a language?
   Do we learn possible rules or probabilistically weighted rules?
2. Problems with learning possible rules.

Overview

Organization of this Talk:

1. What does it mean to learn a language?
   Do we learn possible rules or probabilistically weighted rules?
2. Problems with learning possible rules.
3. A successful experiment in probabilistic learning.

Grammar as rules to accept/reject sentences

Can formalize knowledge about sentence structure as ‘context-free’ rules:

Grammar as rules to accept/reject sentences

Can formalize knowledge about sentence structure as 'context-free' rules:

Sentence $\rightarrow$ Noun Phrase (*you*), Verb Phrase (*have a cookie*)

Grammar as rules to accept/reject sentences

Can formalize knowledge about sentence structure as 'context-free' rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)

Grammar as rules to accept/reject sentences

Can formalize knowledge about sentence structure as ‘context-free’ rules:

Sentence → Noun Phrase (*you*), Verb Phrase (*have a cookie*)
Verb Phrase → Verb (*have*), Noun Phrase (*a cookie*)
Noun Phrase → *you*

Grammar as rules to accept/reject sentences

Can formalize knowledge about sentence structure as 'context-free' rules:

Sentence $\rightarrow$ Noun Phrase (*you*), Verb Phrase (*have a cookie*)
Verb Phrase $\rightarrow$ Verb (*have*), Noun Phrase (*a cookie*)
Noun Phrase $\rightarrow$ *you*
Noun Phrase $\rightarrow$ Determiner (*a*), Noun (*cookie*)

Grammar as rules to accept/reject sentences

Can formalize knowledge about sentence structure as ‘context-free’ rules:

Sentence $\rightarrow$ Noun Phrase (*you*), Verb Phrase (*have a cookie*)
Verb Phrase $\rightarrow$ Verb (*have*), Noun Phrase (*a cookie*)
Noun Phrase $\rightarrow$ *you*
Noun Phrase $\rightarrow$ Determiner (*a*), Noun (*cookie*)

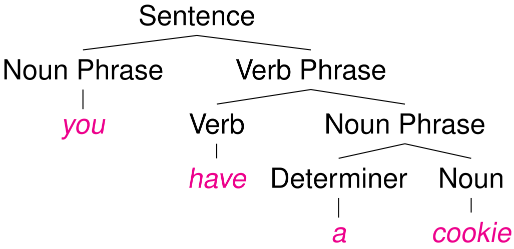
Strings that obey the rules have a derivation or ‘parse:’

Grammar as rules to accept/reject sentences

Can formalize knowledge about sentence structure as ‘context-free’ rules:

Sentence → Noun Phrase (*you*), Verb Phrase (*have a cookie*)
Verb Phrase → Verb (*have*), Noun Phrase (*a cookie*)
Noun Phrase → *you*
Noun Phrase → Determiner (*a*), Noun (*cookie*)

Strings that obey the rules have a derivation or ‘parse:’

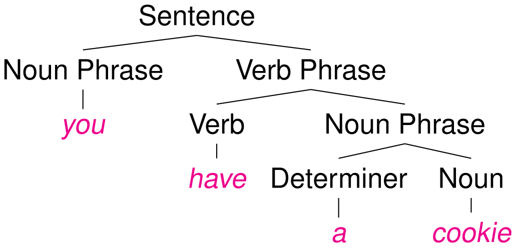


Grammar as rules to accept/reject sentences

Can formalize knowledge about sentence structure as ‘context-free’ rules:

Sentence → Noun Phrase (*you*), Verb Phrase (*have a cookie*)
Verb Phrase → Verb (*have*), Noun Phrase (*a cookie*)
Noun Phrase → *you*
Noun Phrase → Determiner (*a*), Noun (*cookie*)

Strings that obey the rules have a derivation or ‘parse:’



Strings that don’t obey (*have a cookie you*) are considered ‘ungrammatical.’

Claims about acquisition: ‘poverty of stimulus’

Formulated in this way, grammars are very hard to learn.

Claims about acquisition: ‘poverty of stimulus’

Formulated in this way, grammars are very hard to learn.

For example, learner can’t rule out unused rules, as they may just be rare:

Claims about acquisition: ‘poverty of stimulus’

Formulated in this way, grammars are very hard to learn.

For example, learner can’t rule out unused rules, as they may just be rare:

Sentence → Verb Phrase (*have a cookie*),   Noun Phrase (*you*)

Claims about acquisition: ‘poverty of stimulus’

Formulated in this way, grammars are very hard to learn.

For example, learner can’t rule out unused rules, as they may just be rare:

Sentence → Verb Phrase (*have a cookie*), Noun Phrase (*you*)

Caregivers don’t and can’t give negative examples of all unused rules.

Claims about acquisition: ‘poverty of stimulus’

Formulated in this way, grammars are very hard to learn.

For example, learner can’t rule out unused rules, as they may just be rare:

Sentence $\rightarrow$ Verb Phrase (*have a cookie*), Noun Phrase (*you*)

Caregivers don’t and can’t give negative examples of all unused rules.
(And even if they did, children don’t seem to pay attention to this.)

Claims about acquisition: ‘poverty of stimulus’

Formulated in this way, grammars are very hard to learn.

For example, learner can’t rule out unused rules, as they may just be rare:

Sentence $\rightarrow$ Verb Phrase (*have a cookie*), Noun Phrase (*you*)

Caregivers don’t and can’t give negative examples of all unused rules.
(And even if they did, children don’t seem to pay attention to this.)

This ‘poverty of stimulus’ argument used to justify ‘universal grammar’ (UG):
(Chomsky, 1965)

Claims about acquisition: 'poverty of stimulus'

Formulated in this way, grammars are very hard to learn.

For example, learner can't rule out unused rules, as they may just be rare:

Sentence $\rightarrow$ Verb Phrase (*have a cookie*), Noun Phrase (*you*)

Caregivers don't and can't give negative examples of all unused rules. (And even if they did, children don't seem to pay attention to this.)

This 'poverty of stimulus' argument used to justify 'universal grammar' (UG): (Chomsky, 1965)

- In UG, structural rules are innate, learners just set true/false parameters (e.g.: allow pronominal subject to be dropped = true/false).

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence $\rightarrow$ Noun Phrase (*you*), Verb Phrase (*have a cookie*) = 0.999

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence → Noun Phrase (*you*), Verb Phrase (*have a cookie*) = 0.999
Sentence → Verb Phrase (*have a cookie*), Noun Phrase (*you*) = 0.001

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)	= 1.0

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*),	Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*),	Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*),	Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→ *you*		= 0.5

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→	Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→	Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→	Verb (*have*), Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→	*you*	= 0.5
Noun Phrase	→	Determiner (*a*), Noun (*cookie*)	= 0.5

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→ *you*	= 0.5
Noun Phrase	→ Determiner (*a*), Noun (*cookie*)	= 0.5

The grammar is now a probabilistic process for generating a string.

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→ *you*	= 0.5
Noun Phrase	→ Determiner (*a*), Noun (*cookie*)	= 0.5

The grammar is now a probabilistic process for generating a string.
Strings with high probability sound more fluent: *you have a cookie*

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→ *you*	= 0.5
Noun Phrase	→ Determiner (*a*), Noun (*cookie*)	= 0.5

The grammar is now a probabilistic process for generating a string.
Strings with high probability sound more fluent: *you have a cookie*
Strings with low probability sound less fluent: *have a cookie you*

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→ *you*	= 0.5
Noun Phrase	→ Determiner (*a*), Noun (*cookie*)	= 0.5

The grammar is now a probabilistic process for generating a string.
Strings with high probability sound more fluent: *you have a cookie*
Strings with low probability sound less fluent: *have a cookie you*

Defined this way, grammars can be learned probabilistically with no UG.

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→ *you*	= 0.5
Noun Phrase	→ Determiner (*a*), Noun (*cookie*)	= 0.5

The grammar is now a probabilistic process for generating a string.
Strings with high probability sound more fluent: *you have a cookie*
Strings with low probability sound less fluent: *have a cookie you*

Defined this way, grammars can be learned probabilistically with no UG.
They are not learned *exactly*, but to some probabilistic distance

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→ *you*	= 0.5
Noun Phrase	→ Determiner (*a*), Noun (*cookie*)	= 0.5

The grammar is now a probabilistic process for generating a string.
Strings with high probability sound more fluent: *you have a cookie*
Strings with low probability sound less fluent: *have a cookie you*

Defined this way, grammars can be learned probabilistically with no UG.
They are not learned *exactly*, but to some probabilistic distance
(ranked by the probability the grammar assigns to training sentences).

Grammar as preferences over speech decisions

But we could also formulate grammar as probabilistically weighted rules:

Sentence	→ Noun Phrase (*you*), Verb Phrase (*have a cookie*)	= 0.999
Sentence	→ Verb Phrase (*have a cookie*), Noun Phrase (*you*)	= 0.001
Verb Phrase	→ Verb (*have*), Noun Phrase (*a cookie*)	= 1.0
Noun Phrase	→ *you*	= 0.5
Noun Phrase	→ Determiner (*a*), Noun (*cookie*)	= 0.5

The grammar is now a probabilistic process for generating a string.
Strings with high probability sound more fluent: *you have a cookie*
Strings with low probability sound less fluent: *have a cookie you*

Defined this way, grammars can be learned probabilistically with no UG.
They are not learned *exactly*, but to some probabilistic distance
(ranked by the probability the grammar assigns to training sentences).
Incentive to assign high weights to common rules, low weights to rare rules.

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities.

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts;

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts; e.g. given 2 heads, 10 tails, coin is more likely biased than fair.)

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts; e.g. given 2 heads, 10 tails, coin is more likely biased than fair.)
- Generate (sample) many possible sets of trees given these weights.

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts; e.g. given 2 heads, 10 tails, coin is more likely biased than fair.)
- Generate (sample) many possible sets of trees given these weights. (Generate random number and select outcome from rule distrubution.)

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts; e.g. given 2 heads, 10 tails, coin is more likely biased than fair.)
- Generate (sample) many possible sets of trees given these weights. (Generate random number and select outcome from rule distrubution.)
- Remove all trees whose terminals (words) are not in the sentences.

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts; e.g. given 2 heads, 10 tails, coin is more likely biased than fair.)
- Generate (sample) many possible sets of trees given these weights. (Generate random number and select outcome from rule distrubution.)
- Remove all trees whose terminals (words) are not in the sentences. (Can't just write in words; must sample proportionally to grammar!)

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts; e.g. given 2 heads, 10 tails, coin is more likely biased than fair.)
- Generate (sample) many possible sets of trees given these weights. (Generate random number and select outcome from rule distrubution.)
- Remove all trees whose terminals (words) are not in the sentences. (Can't just write in words; must sample proportionally to grammar!)

Trees that remain incorporate constraints of observations (common co-occurrences are chunked together).

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts; e.g. given 2 heads, 10 tails, coin is more likely biased than fair.)
- Generate (sample) many possible sets of trees given these weights. (Generate random number and select outcome from rule distrubution.)
- Remove all trees whose terminals (words) are not in the sentences. (Can't just write in words; must sample proportionally to grammar!)

Trees that remain incorporate constraints of observations (common co-occurrences are chunked together).

This is called rejection sampling.

Probabilistically learning grammars from sentences

How can grammars be learned probabilistically?

Consider a space of possible probabilistic grammars with 15 labels:

- Generate (sample) many possible distributions of rule probabilities. (Distributions are generated randomly from a Dirichlet prior model, which is a model of distributions consistent with observed counts; e.g. given 2 heads, 10 tails, coin is more likely biased than fair.)
- Generate (sample) many possible sets of trees given these weights. (Generate random number and select outcome from rule distrubution.)
- Remove all trees whose terminals (words) are not in the sentences. (Can't just write in words; must sample proportionally to grammar!)

Trees that remain incorporate constraints of observations (common co-occurrences are chunked together).

This is called rejection sampling.
It is very inefficient: odds of generating actual corpus sentence are very low.

Probabilistically learning grammars from sentences

Alternate model:

Probabilistically learning grammars from sentences

Alternate model:

Consider space of possible CFGs with 15 labels

Probabilistically learning grammars from sentences

Alternate model:

Consider space of possible CFGs with 15 labels

- Start with random set of values for rule distributions and trees.

Probabilistically learning grammars from sentences

Alternate model:

Consider space of possible CFGs with 15 labels

- Start with random set of values for rule distributions and trees.
- Iterate through rule distributions and tree decisions:

Probabilistically learning grammars from sentences

Alternate model:

Consider space of possible CFGs with 15 labels

- Start with random set of values for rule distributions and trees.
- Iterate through rule distributions and tree decisions:
  - Resample distributions/decision given surrounding context (posterior).

Probabilistically learning grammars from sentences

Alternate model:

Consider space of possible CFGs with 15 labels

- Start with random set of values for rule distributions and trees.
- Iterate through rule distributions and tree decisions:
    - Resample distributions/decision given surrounding context (posterior).

The model gradually comes to accommodate observations.

Probabilistically learning grammars from sentences

Alternate model:

Consider space of possible CFGs with 15 labels

- Start with random set of values for rule distributions and trees.
- Iterate through rule distributions and tree decisions:
    - Resample distributions/decision given surrounding context (posterior).

The model gradually comes to accommodate observations.

This is called Gibbs sampling.

Probabilistically learning grammars from sentences

Alternate model:

Consider space of possible CFGs with 15 labels

- Start with random set of values for rule distributions and trees.
- Iterate through rule distributions and tree decisions:
    - Resample distributions/decision given surrounding context (posterior).

The model gradually comes to accommodate observations.

This is called Gibbs sampling.
It is way more efficient. We do this.

Acquisition experiments

We run this probabilistic learning process on child-directed speech data.

Acquisition experiments

We run this probabilistic learning process on child-directed speech data.

Training data: CHILDES corpus of child-directed speech, Eve section.
(MacWhinney, 2000)

Acquisition experiments

We run this probabilistic learning process on child-directed speech data.

Training data: CHILDES corpus of child-directed speech, Eve section. (MacWhinney, 2000)

14,251 sentences of varying lengths.

Acquisition experiments

We run this probabilistic learning process on child-directed speech data.

Training data: CHILDES corpus of child-directed speech, Eve section.
(MacWhinney, 2000)

14,251 sentences of varying lengths.
Recorded during interaction between child and caregiver, then transcribed.

Acquisition experiments

We run this probabilistic learning process on child-directed speech data.

Training data: CHILDES corpus of child-directed speech, Eve section.
(MacWhinney, 2000)

14,251 sentences of varying lengths.
Recorded during interaction between child and caregiver, then transcribed.

E.g. *You have another cookie right on the table.*

Acquisition experiments

We run this probabilistic learning process on child-directed speech data.

Training data: CHILDES corpus of child-directed speech, Eve section.
(MacWhinney, 2000)

14,251 sentences of varying lengths.
Recorded during interaction between child and caregiver, then transcribed.

E.g. *You have another cookie right on the table.*

Experiments run for a week on 10 GPUs in Ohio Supercomputer Center.

Evaluation Parameters

We evaluate several configurations of the learner:

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),
- CCL (Seginer, 2007),

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),
- CCL (Seginer, 2007),
- BMMM+DMV (Christodoulopoulos et al., 2012),

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),
- CCL (Seginer, 2007),
- BMMM+DMV (Christodoulopoulos et al., 2012),
- UHHMM (Shain et al., 2016),

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),
- CCL (Seginer, 2007),
- BMMM+DMV (Christodoulopoulos et al., 2012),
- UHHMM (Shain et al., 2016),
- right-branching baseline: left children are always terminals (words).

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),
- CCL (Seginer, 2007),
- BMMM+DMV (Christodoulopoulos et al., 2012),
- UHHMM (Shain et al., 2016),
- right-branching baseline: left children are always terminals (words).

Evaluate vs. unlabeled versions of human-annotated ‘gold standard’ trees:

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),
- CCL (Seginer, 2007),
- BMMM+DMV (Christodoulopoulos et al., 2012),
- UHHMM (Shain et al., 2016),
- right-branching baseline: left children are always terminals (words).

Evaluate vs. unlabeled versions of human-annotated 'gold standard' trees:

- recall: % of actual constituents that model predicts.

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),
- CCL (Seginer, 2007),
- BMMM+DMV (Christodoulopoulos et al., 2012),
- UHHMM (Shain et al., 2016),
- right-branching baseline: left children are always terminals (words).

Evaluate vs. unlabeled versions of human-annotated 'gold standard' trees:

- recall: % of actual constituents that model predicts.
- precision: % of model's predictions that are actual constituents.

Evaluation Parameters

We evaluate several configurations of the learner:

- Manipulate number of categories: $K \in \{15, 30, 45\}$.
- Manipulate maximum center-embedding depth: $D \in \{1, 2\}$.

We also compare against other recent learners & right-branching baseline:

- UPPARSE (Ponvert et al., 2011),
- CCL (Seginer, 2007),
- BMMM+DMV (Christodoulopoulos et al., 2012),
- UHHMM (Shain et al., 2016),
- right-branching baseline: left children are always terminals (words).

Evaluate vs. unlabeled versions of human-annotated 'gold standard' trees:

- recall: % of actual constituents that model predicts.
- precision: % of model's predictions that are actual constituents.
- F1 score: product of recall & precision / average of recall & precision.

Results

Results on constituent trees with punctuation removed after training:

System	Precision	Recall	F1
(rival) CCL	60.1	48.7	53.8
(rival) UPPARSE	60.5	51.9	55.9
(rival) UHHMM	55.5	69.3	61.7
(rival) BMMM+DMV	**63.5**	63.3	63.4
(rival) UHHMM(flattened)	62.9	68.4	65.6
This model w. D=1,K=15	55.5	69.3	61.6
This model w. D=1,K=30	61.6	**76.7**	**68.4**
This model w. D=1,K=45	53.9	66.9	59.5
This model w. D=2,K=15	50.6	63.2	56.2
(baseline) Right-branching	**68.7**	**85.8**	**76.3**

This model is competitive with rivals, but not better than right-branching.

Evaluation Parameters

Model also learns category labels — do these correspond to NP, PP, etc?

Evaluation Parameters

Model also learns category labels — do these correspond to NP, PP, etc?

Problem: different theories make different predictions about category labels.

Evaluation Parameters

Model also learns category labels — do these correspond to NP, PP, etc?

Problem: different theories make different predictions about category labels.

Solution: most theories make same predictions about NPs; just test these.

Evaluation Parameters

Model also learns category labels — do these correspond to NP, PP, etc?

Problem: different theories make different predictions about category labels.

Solution: most theories make same predictions about NPs; just test these.

- NP recall: % of actual NPs hypothesized with any label,

Evaluation Parameters

Model also learns category labels — do these correspond to NP, PP, etc?

Problem: different theories make different predictions about category labels.

Solution: most theories make same predictions about NPs; just test these.

- NP recall: % of actual NPs hypothesized with any label,
- NP identification: % of actual NPs hypothesized w. label mapped to NP.

Evaluation Parameters

Model also learns category labels — do these correspond to NP, PP, etc?

Problem: different theories make different predictions about category labels.

Solution: most theories make same predictions about NPs; just test these.

- NP recall: % of actual NPs hypothesized with any label,
- NP identification: % of actual NPs hypothesized w. label mapped to NP. (Mapping function trained on separate data w. human NP annotation.)

Results

Results for noun phrase recall and noun phrase identification:

System	NP recall	NP ident
(rival) CCL	32.4	-
(rival) UPPARSE	69.1	-
(rival) UHHMM (flattened)	61.4	34.7
(rival) BMMM+DMV	71.3	60.8
This model w. D=1,K=15	81.9	57.4
This model w. D=1,K=30	80.1	**63.1**
This model w. D=1,K=45	77.1	60.8
This model w. D=2,K=15	**86.3**	**63.1**
Right-branching baseline	64.2	-

Category labels appear to be quite coherent!

Results

Results for noun phrase recall and noun phrase identification:

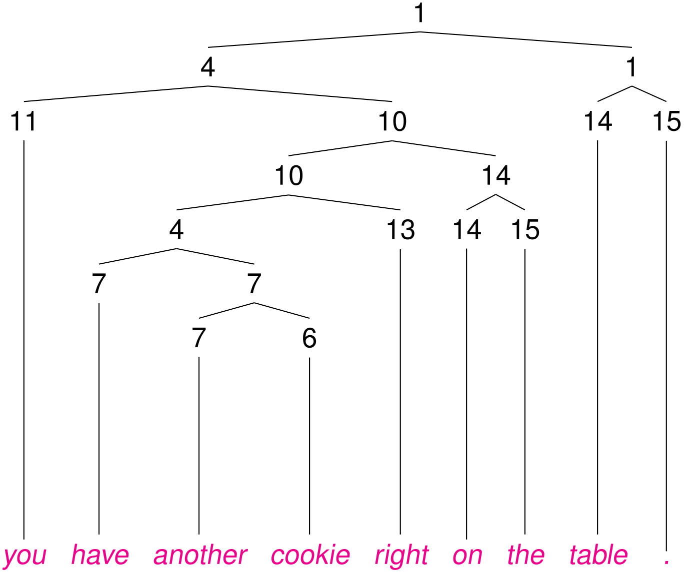
System	NP recall	NP ident
(rival) CCL	32.4	-
(rival) UPPARSE	69.1	-
(rival) UHHMM (flattened)	61.4	34.7
(rival) BMMM+DMV	71.3	60.8
This model w. D=1,K=15	81.9	57.4
This model w. D=1,K=30	80.1	**63.1**
This model w. D=1,K=45	77.1	60.8
This model w. D=2,K=15	**86.3**	**63.1**
Right-branching baseline	64.2	-

Category labels appear to be quite coherent!

(Similar results obtain for PP and, to a lesser extent, VP.)

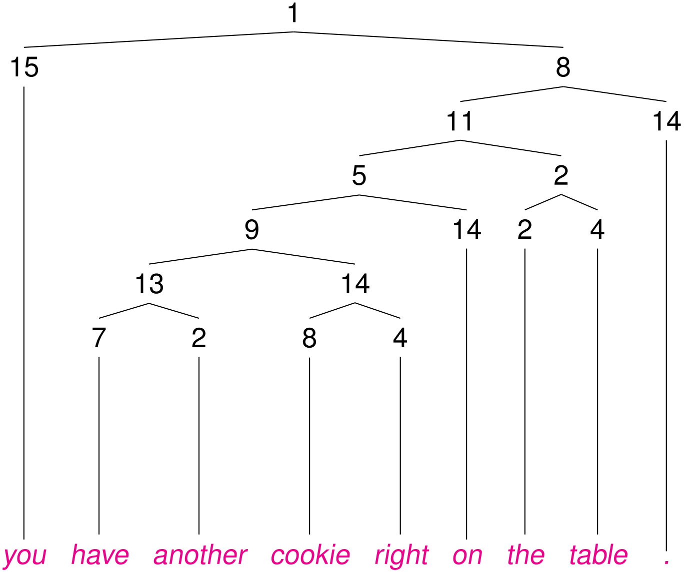
Structures hypothesized during training

Iteration 5 (first iteration after re-initialization trials) — not much familiar:



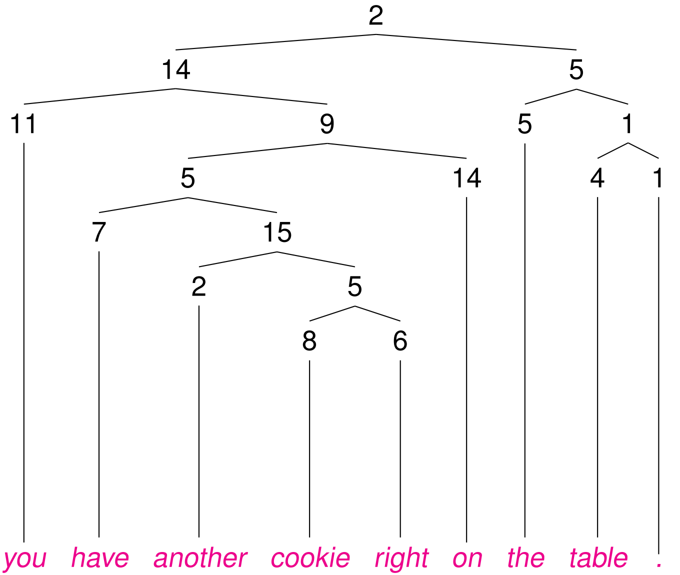
Structures hypothesized during training

Iteration 6:



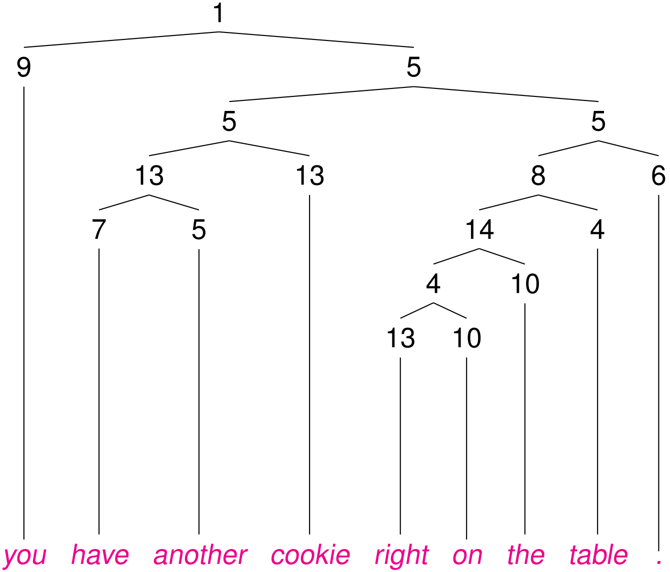
Structures hypothesized during training

Iteration 7:



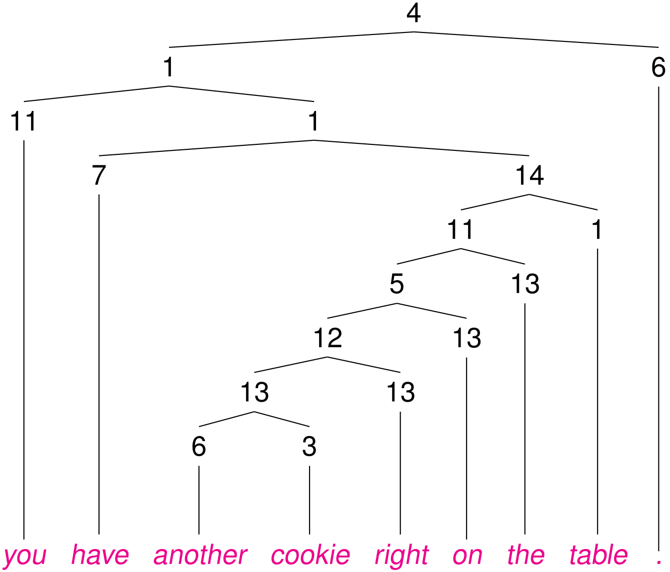
Structures hypothesized during training

Iteration 8:



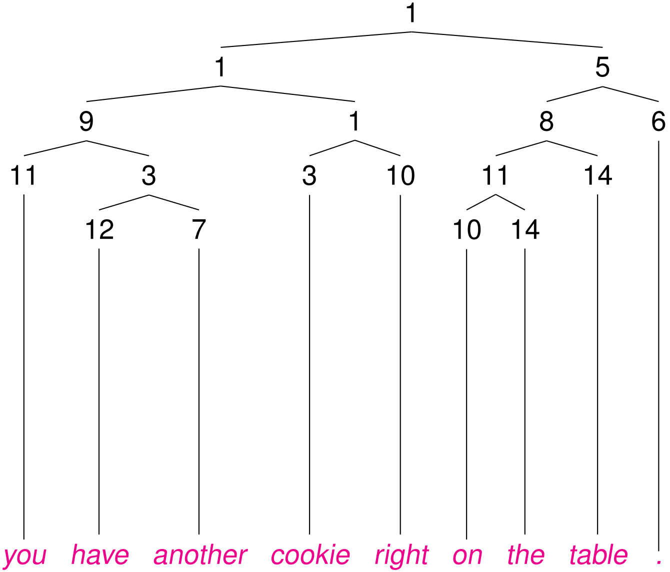
Structures hypothesized during training

Iteration 9:



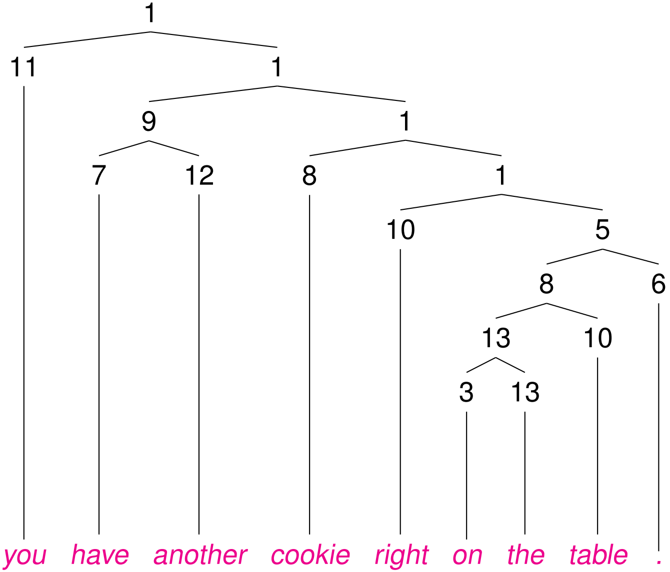
Structures hypothesized during training

Iteration 10 — the model discovers *on* and *the* co-occur a lot, clumps them:



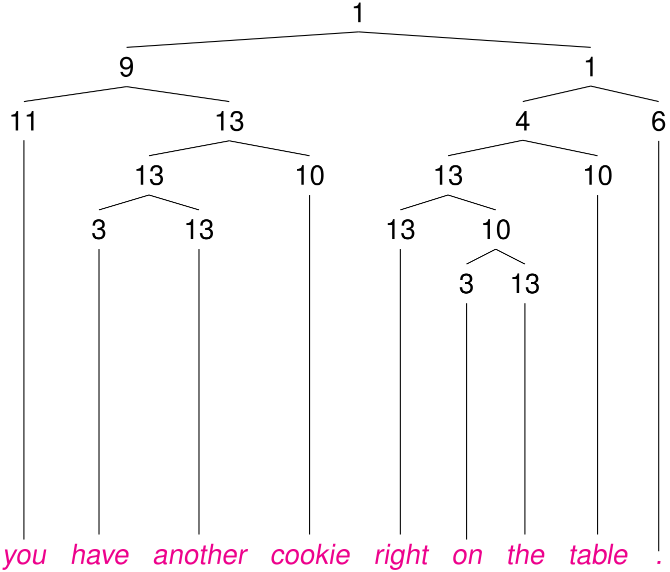
Structures hypothesized during training

Iteration 25 (now showing every 25th iteration):



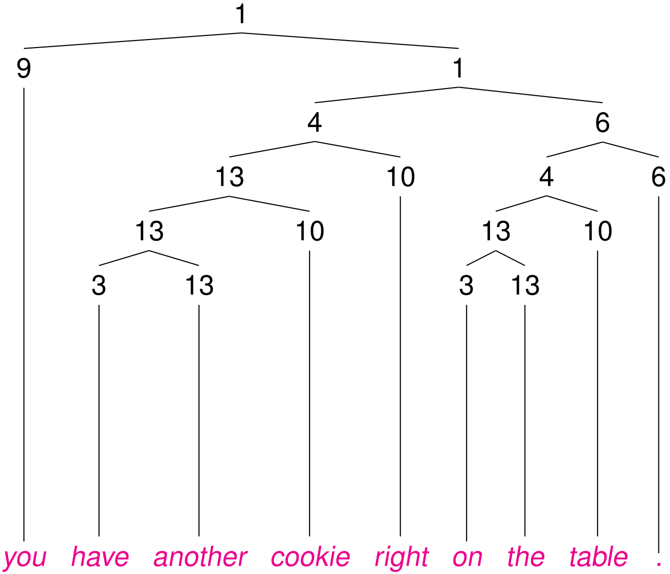
Structures hypothesized during training

Iteration 50:



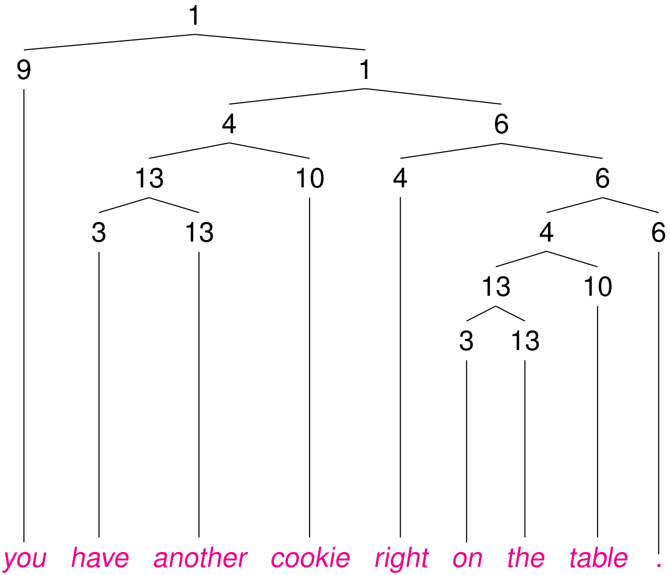
Structures hypothesized during training

Iteration 75:



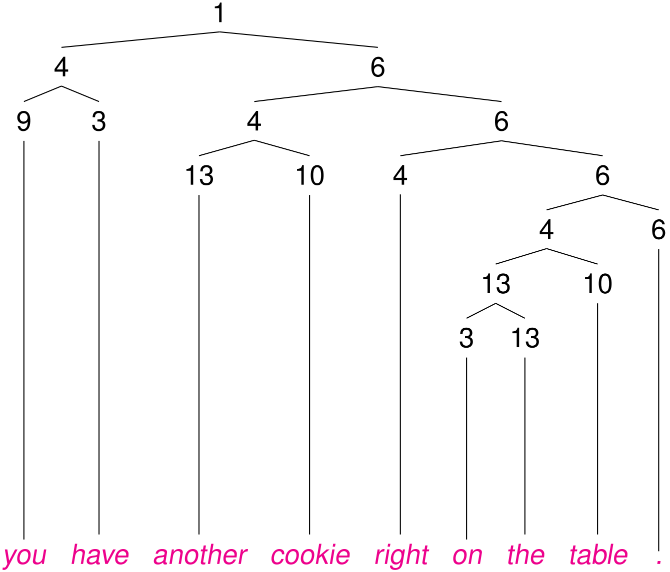
Structures hypothesized during training

Iteration 100:



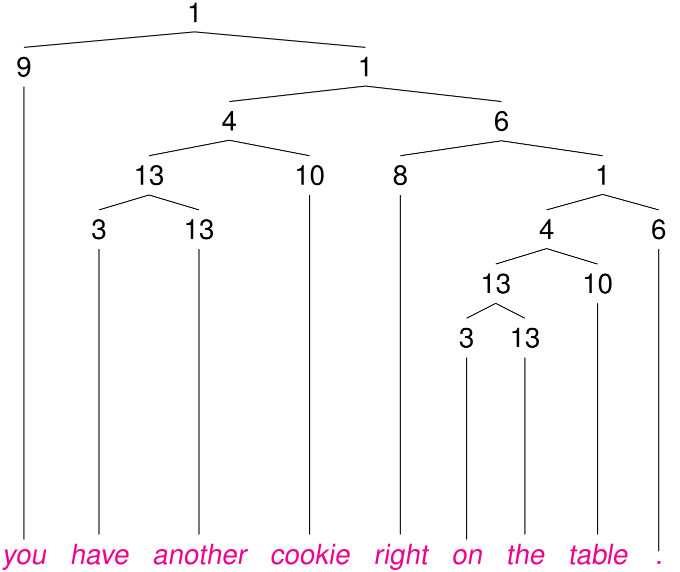
Structures hypothesized during training

Iteration 125:



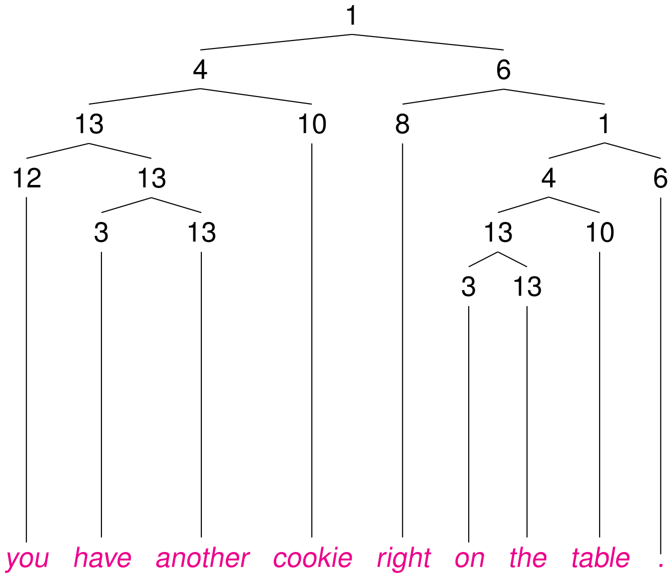
Structures hypothesized during training

Iteration 150:



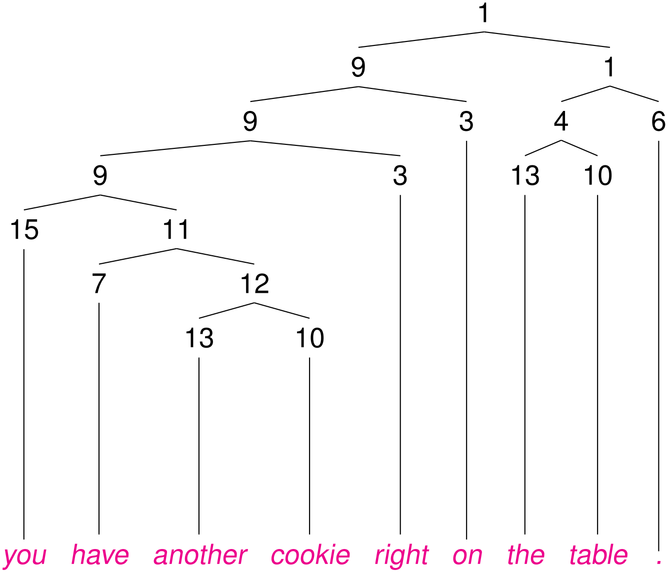
Structures hypothesized during training

Iteration 200 (now showing every 50th iteration):



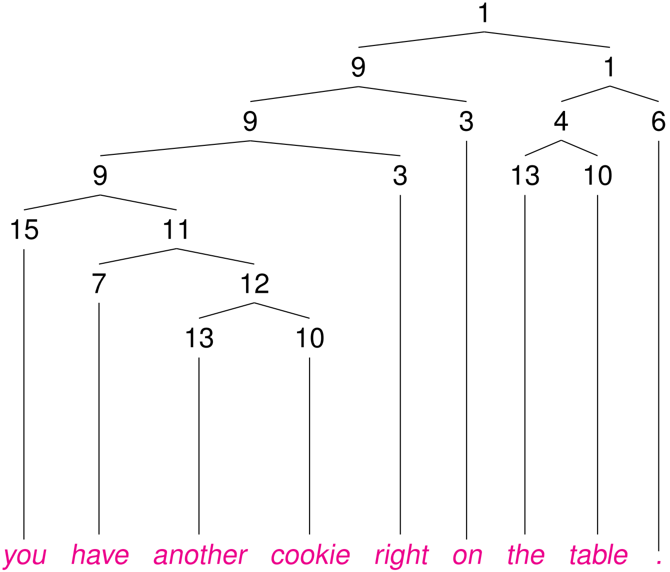
Structures hypothesized during training

Iteration 250 – determiners (*the/another*), nouns (*table/cookie*) clumped:



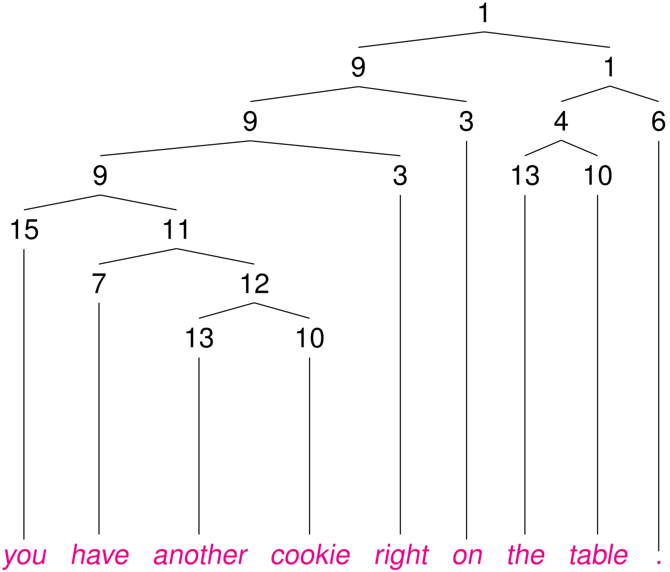
Structures hypothesized during training

Iteration 250 – learner can re-use Det+Noun rule more than Prep+Det:



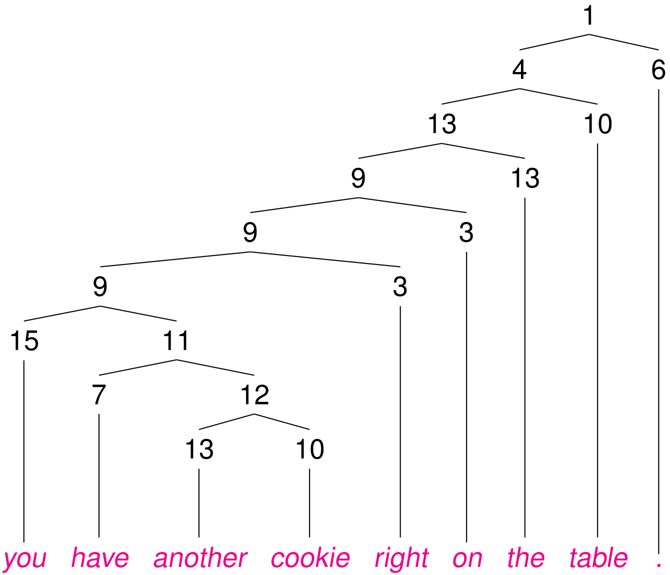
Structures hypothesized during training

Iteration 250 – also, verb *have* clumped with noun phrase *another cookie*:



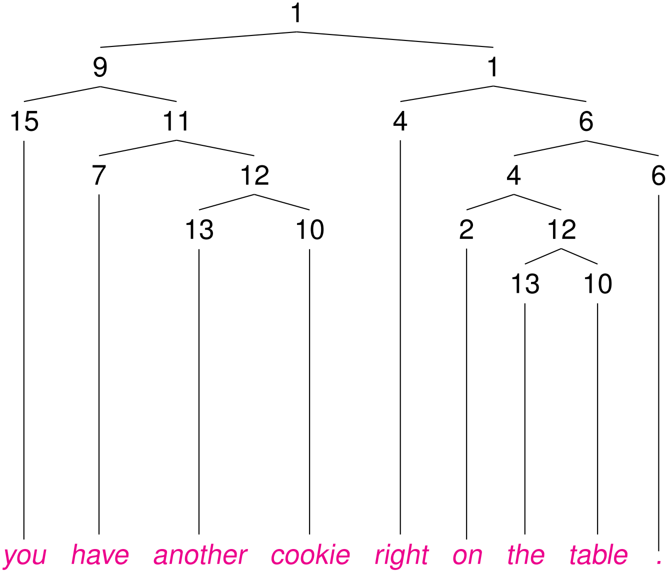
Structures hypothesized during training

Iteration 300:



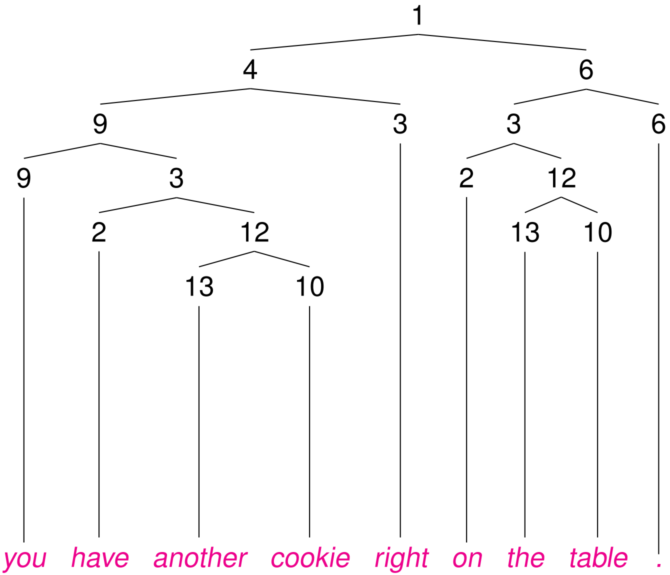
Structures hypothesized during training

Iteration 350 – preposition *on* and noun phrase *the table* now clumped:



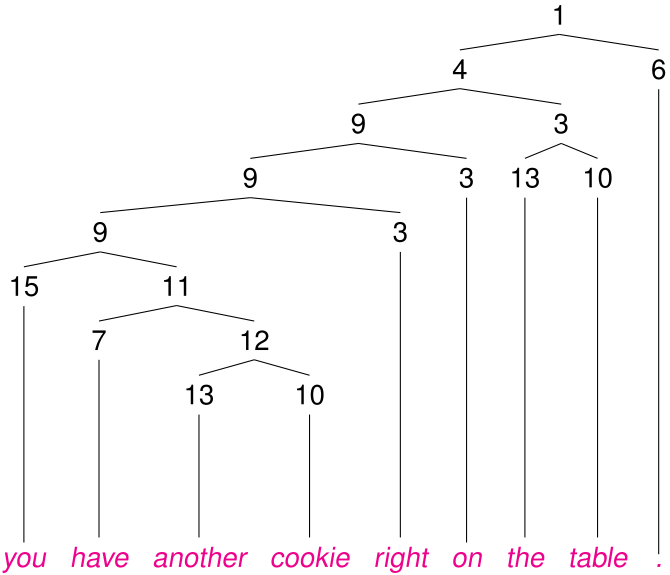
Structures hypothesized during training

Iteration 400:



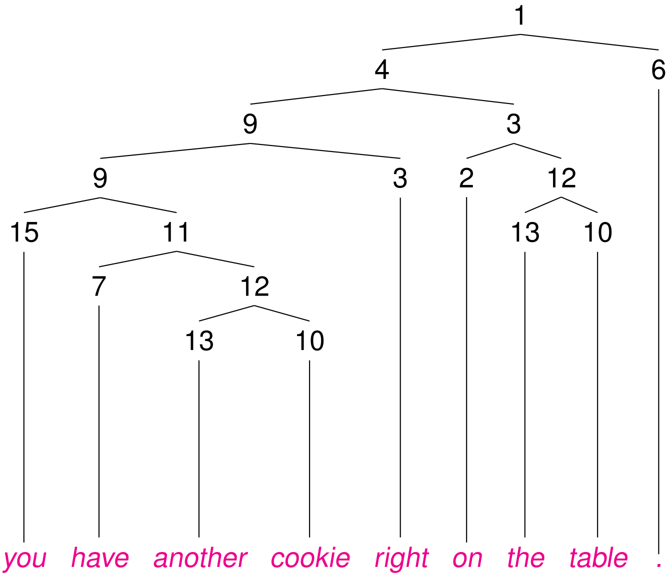
Structures hypothesized during training

Iteration 450:



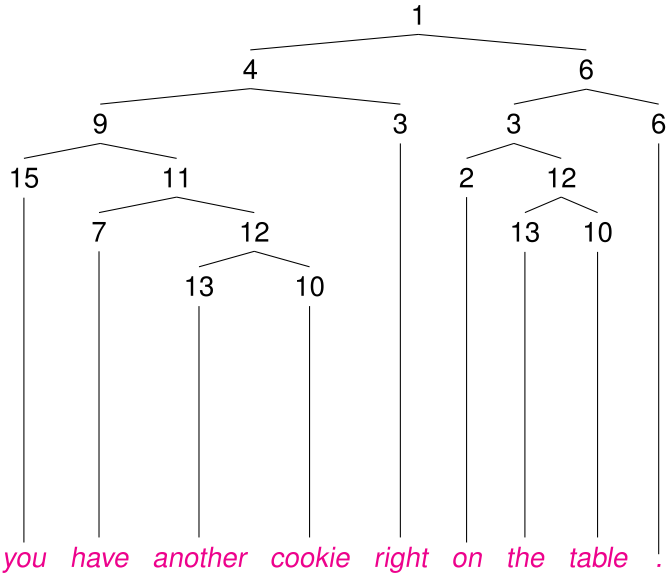
Structures hypothesized during training

Iteration 500 — category labels for Prep/Det/Noun/NP/PP mostly stable:



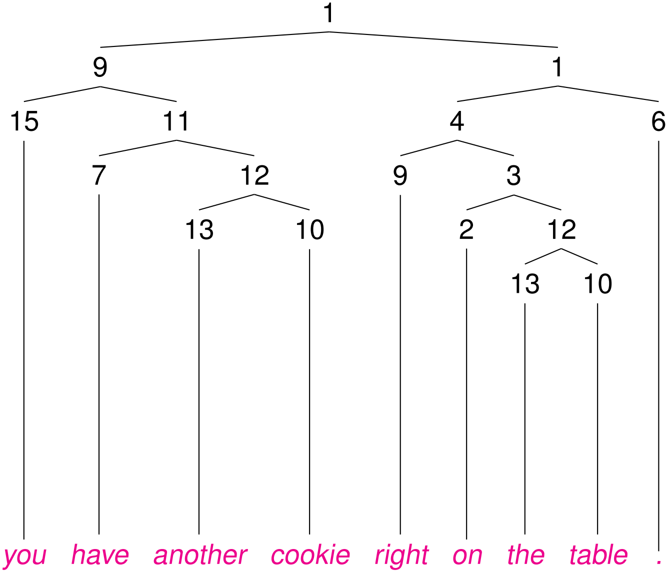
Structures hypothesized during training

Iteration 550:



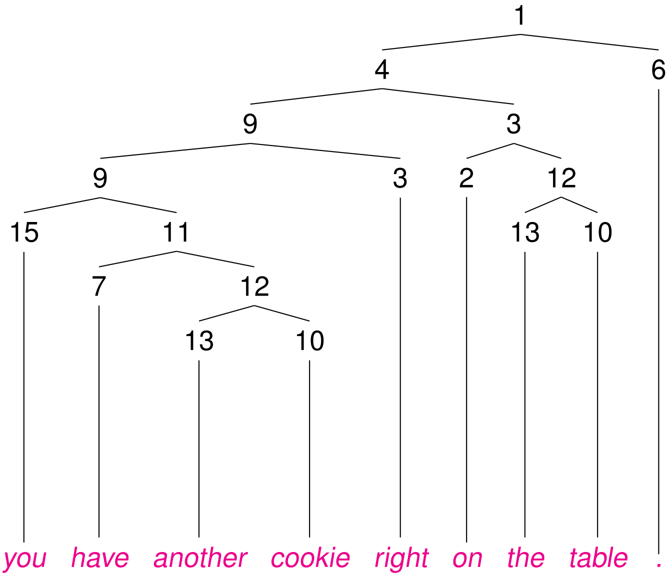
Structures hypothesized during training

Iteration 600 — adverb *right* clumped with prepositional phrase *on the table*:



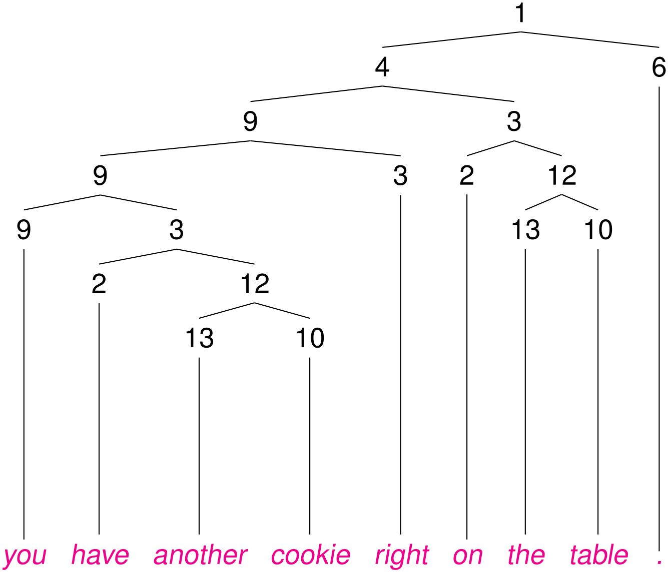
Structures hypothesized during training

Iteration 650 — adverb *right* now clumped with sentence *you . . . cookie*:



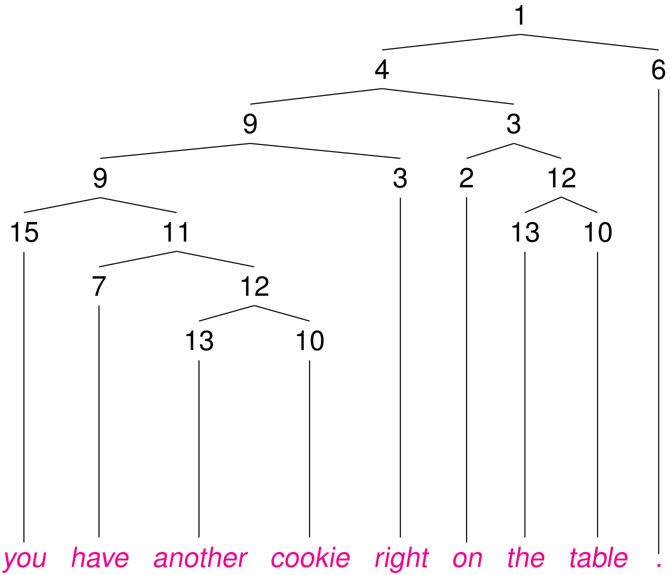
Structures hypothesized during training

Iteration 700 — re-type noun phrase *you* and verb phrase *have . . . cookie*:



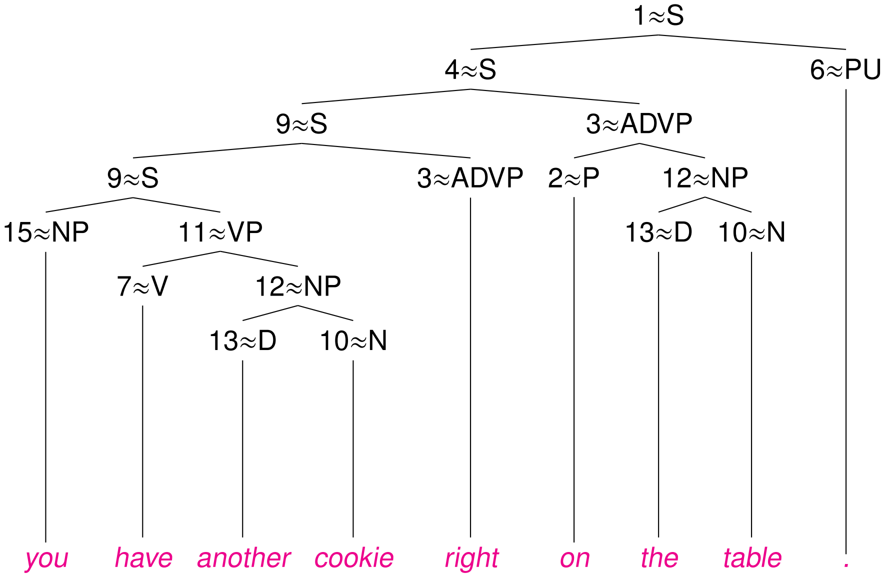
Structures hypothesized during training

Iteration 750 – change back noun phrase and verb phrase:



Structures hypothesized during training

Final constituent types consistent with linguistic theory:



Conclusion

In this talk:

Conclusion

In this talk:

1. Learning possible rules from just words is hard: anything's possible!

Conclusion

In this talk:

1. Learning possible rules from just words is hard: anything's possible!
2. But defined probabilistically, grammar learning is feasible.

Conclusion

In this talk:

1. Learning possible rules from just words is hard: anything's possible!
2. But defined probabilistically, grammar learning is feasible.
3. This makes justification of Universal Grammar more tenuous.

Conclusion

In this talk:

1. Learning possible rules from just words is hard: anything's possible!
2. But defined probabilistically, grammar learning is feasible.
3. This makes justification of Universal Grammar more tenuous.

Thanks!

Bibliography I

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, Mass.: MIT Press.

Christodoulopoulos, C., Goldwater, S., & Steedman, M. (2012, 6). Turning the pipeline into a loop: Iterated unsupervised dependency parsing and PoS induction. In *NAACL-HLT Workshop on the Induction of Linguistic Structure* (p. 96-99). Montreal, Canada.

MacWhinney, B. (2000). *The childes project: Tools for analyzing talk* (Third ed.). Mahwah, NJ: Lawrence Elrbaum Associates.

Ponvert, E., Baldridge, J., & Erik, K. (2011, 6). Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the 49th annual meeting of the association for computational linguistics* (p. 1077-1086). Portland, Oregon.

Seginer, Y. (2007). Fast unsupervised incremental parsing. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 384–391).

Bibliography II

Shain, C., van Schijndel, M., Futrell, R., Gibson, E., & Schuler, W. (2016). Memory access during incremental sentence processing causes reading time latency. In *Proceedings of the computational linguistics for linguistic complexity workshop.* Association for Computational Linguistics.