Dendrophilia squared

Alexander Clark

CLASP,
Department of Philosophy, Linguistics and Theory of Science,
University of Gothenburg
alexsclark@gmail.com

CLASP 2022

Dendrophilia

Fitch [2014]

> *Experiments to date strongly suggest that there is an important difference between humans and most other species, best characterized cognitively as a propensity by our species to infer tree structures from sequential data.*

Dendrophilia

Fitch [2014]

> *Experiments to date strongly suggest that there is an important difference between humans and most other species, best characterized cognitively as a propensity by our species to infer tree structures from sequential data.*

- How does this work exactly?
- Tree structures are inadequate for natural language syntax.

Strings and trees and something else

Naively we are dealing with three sorts of objects:

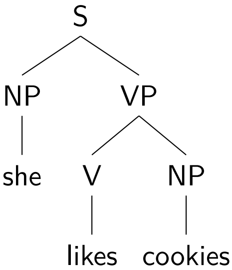
1. Strings of words
2. Constituent structures
3. and something else to handle movement?

This is theoretically bad.

Sequential data: string

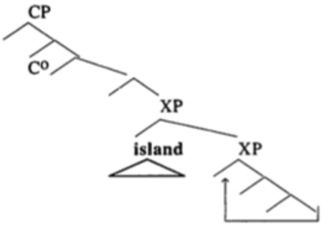
She likes cookies.

Hierarchically structured data: Tree

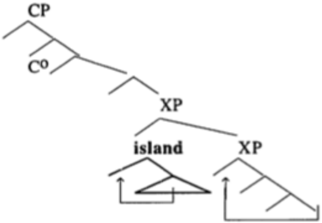


Movement [Richards, 1997]

(51) a.

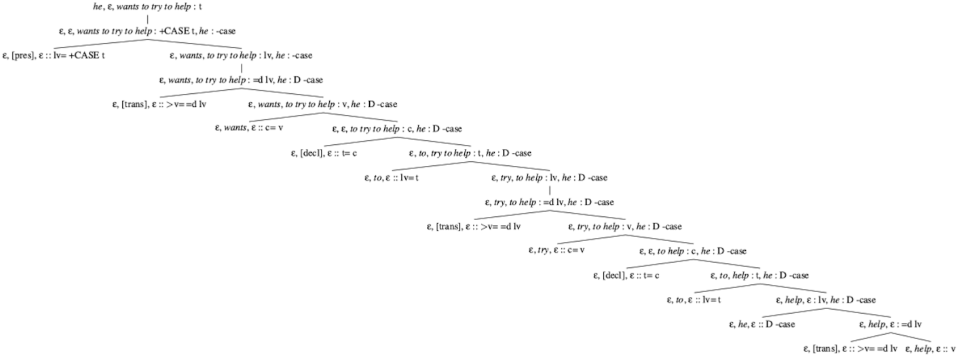


b.

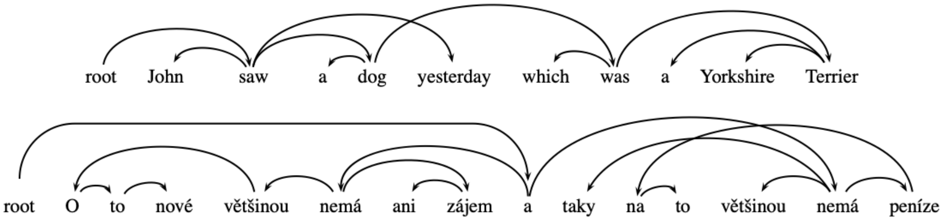


c.

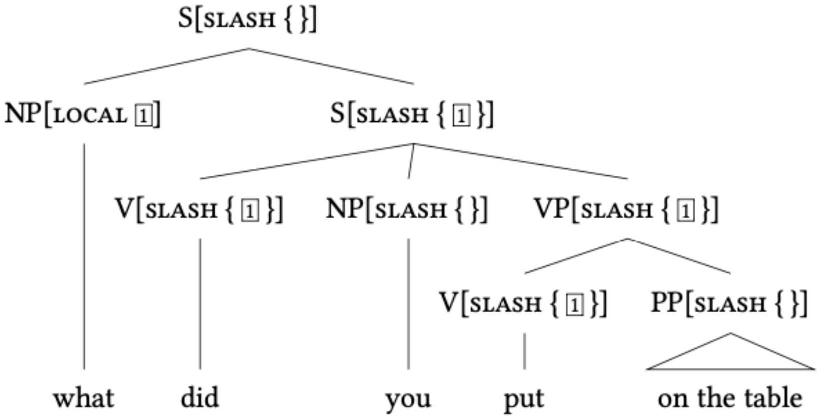
Derivation tree of MG [Torr, 2019]



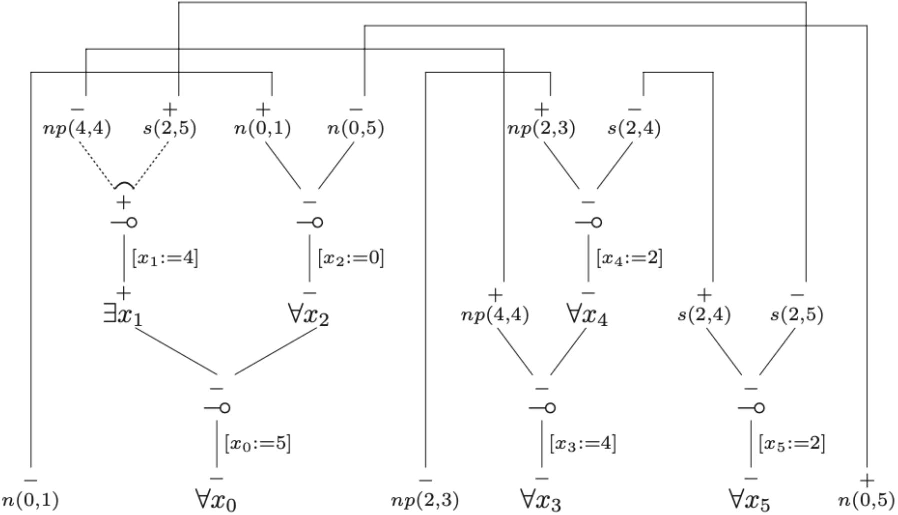
Non-projective dependency structures [McDonald et al., 2005]



HPSG feature structures [Borsley and Crysmann, 2021]



Proof Nets [Moot, 2002]



Desiderata

- Descriptively adequate
- Easy for humans to reason about
  - Natural diagrams on a 2d page
  - Have clean mathematical properties

Desiderata

- Descriptively adequate
- Easy for humans to reason about
  - Natural diagrams on a 2d page
  - Have clean mathematical properties
- Where do these structures come from?
  1. Processing: efficiently parseable
  2. Acquisition: learnable from evidence available to the child
  3. Cultural Evolution: why do languages have these structures?
  4. Biological Evolution: why do we have the ability to learn these structures?

5 minute introduction to strings and trees and 3d trees

Rogers [2003]

How to construct a string of length 2 *ab*?

- Take *a* and *b* and concatenate them to make *ab*.

Construction of a string

How to make a string *abc*?

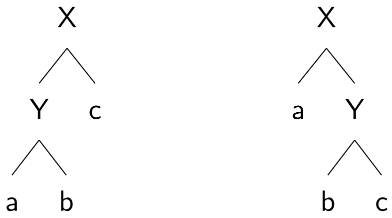
1. a and b $\rightarrow$ ab.
2. ab and c $\rightarrow$ abc

OR

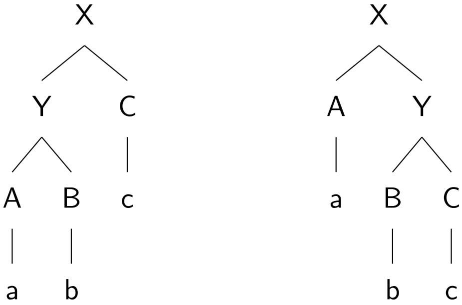
1. b and c $\rightarrow$ bc.
2. a and bc $\rightarrow$ abc

Construction of a string

We can represent these as trees:



Construction of a string

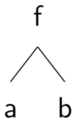


Finite amount of state + Markov assumption gives (probabilistic/weighted) context-free grammars

$X \to YC$, $A \to a$, $B \to b$, ...

Construction of a trivial tree

How to make the tree

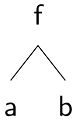


?

Construction of a trivial tree

We assemble three parts:

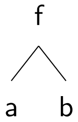
and a and b →



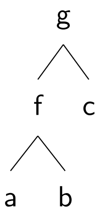
Two sorts of objects

- Rank 2:

- Rank 0: *a* *b*



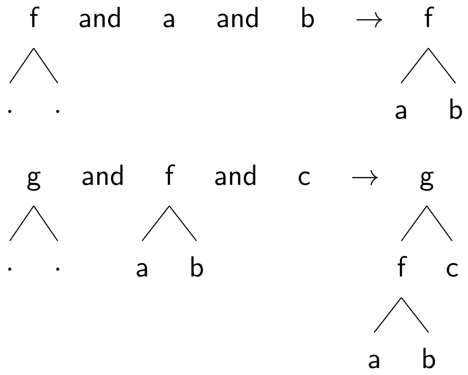
Construction of a bigger tree

How to make the tree 

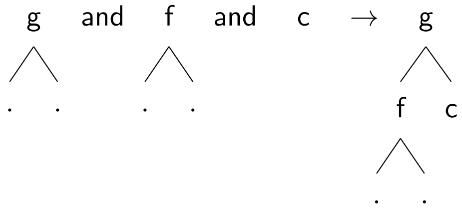
?

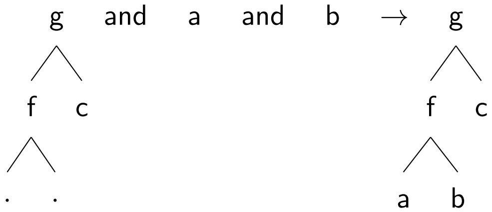
Construction 1

1.
2.

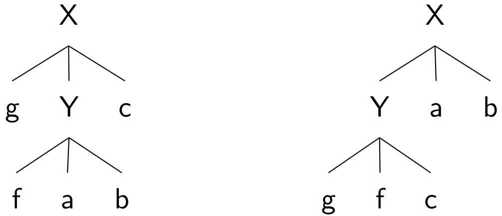


Construction 2

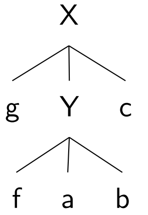
1. 

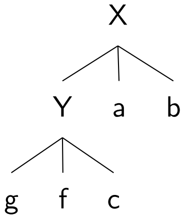
2. 

Represent these construction methods as trees



Represent these construction methods as trees

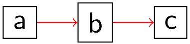




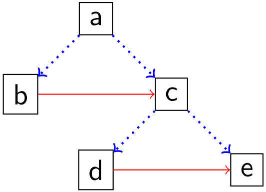
These aren't trees! They are 3d trees

3d trees

string: one relation

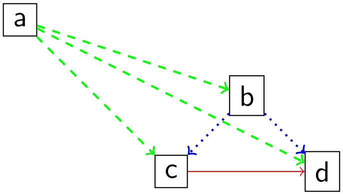


tree: two relations



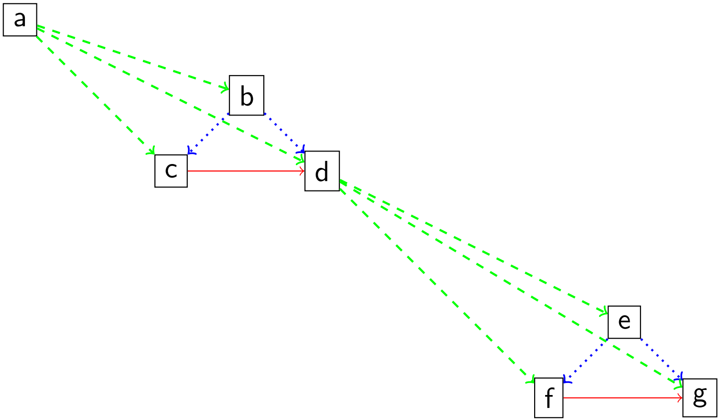
3d trees

Horrible diagram



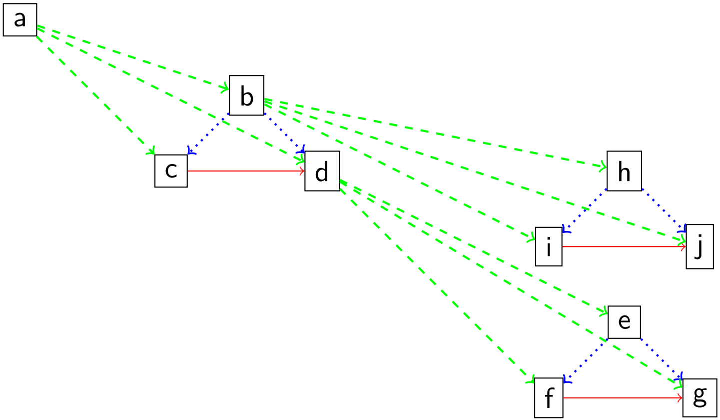
3d trees

Horrible diagram



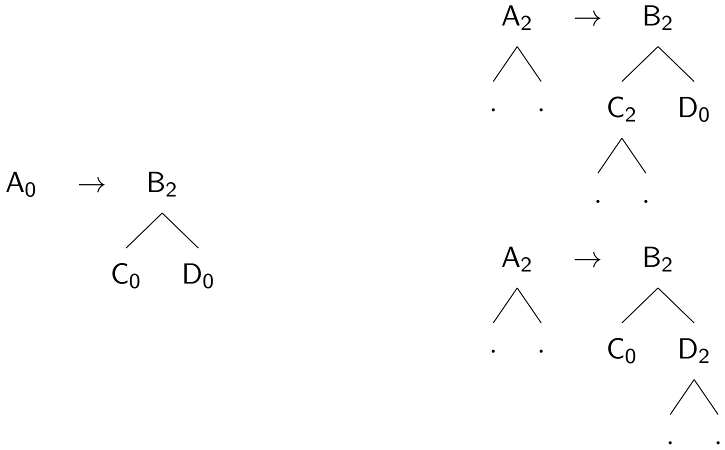
3d trees

Horrible diagram

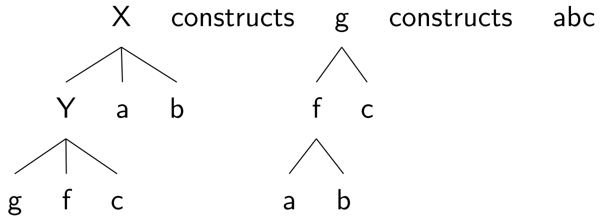


Footed linear context-free tree grammars

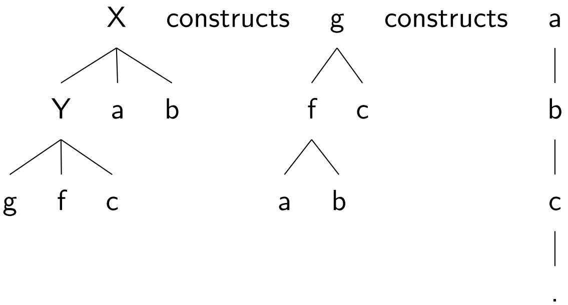
Finite amount of state + Markov assumption gives you a standard mildly context-sensitive formalism, equivalent to TAG, CCG, LIG, well-nested MCFGs of dimensions 2, etc.



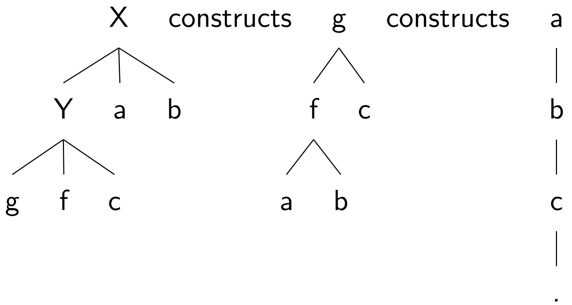
Two step derivation



Same operation twice



Same operation twice



Main point of this talk

We can use the same learning operation twice.

1. Learn CFGs from strings [Clark and Fijalkow, 2020]
2. Learn these context-free tree grammars from trees. [Clark, 2021]

Strong learning

Horning [1969]

Ignore the (unobserved) semantics, and try to generate the right set/distribution of **trees (t) and strings (s)**:

Input forms according to G_{PARENT}

$s_1, \ldots, s_k, \ldots\ldots$

Output Require that $\mathbb{P}(t; G_{\text{CHILD}}) \approx \mathbb{P}(t; G_{\text{PARENT}})$ and $\mathbb{P}(s \mid t; G_{\text{CHILD}}) \approx \mathbb{P}(s \mid t; G_{\text{PARENT}})$

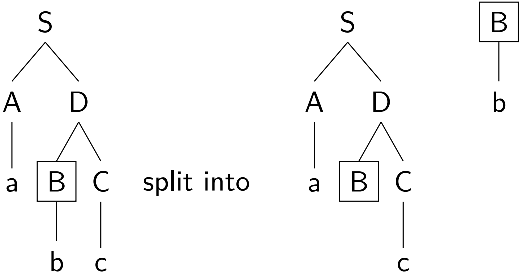
- Realizability assumption: the samples are drawn i.i.d. according to a grammar in the class we are learning.
- Consistent estimator: should converge to the true grammar and parameters.

Context Free Grammars

CFG in Chomsky Normal Form:

Set of productions P of the form $A \to BC$ or $A \to a$

S only occurs on the left hand side of productions.



Tree

Context

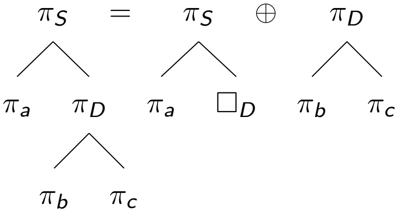
$a\Box c$

Yield

b

Context Free Grammars

- Label derivation tree with productions



Notation

$\Omega(A)$ is the set of all trees with A at the root.
$\Xi(A)$ is the set of all contexts of A, with S at the root.

Weighted Context Free Grammars

Smith and Johnson [2007]

Parameter θ for each production in $\mathbb{R}^+$, defines the weight of a tree as

$$w(\tau) = \prod_{\pi} \theta(\pi)^{n(\pi;\tau)}$$

For each nonterminal A define:

$$I(A) = w(\Omega(A)) \text{ (sum over yields)}$$

$$O(A) = w(\Xi(A)) \text{ (sum over contexts)}$$

Stipulate that $I(S) = 1$ and define $\mathbb{P}(\tau) = w(\tau)$

$$\mathbb{P}(s \mid \tau) = \begin{cases} 1 \text{ if } s \text{ is the yield of } \tau \\ 0 \text{ otherwise} \end{cases}$$

Probabilistic Context Free Grammars

Trivial Identity for WCFGs

$$I(A)O(A) = \mathbb{E}(A)$$

Stipulate that $I(A) = 1$, and so $O(A) = \mathbb{E}(A)$. Each nonterminal defines a probability distribution over its yields.
Parameters are in $[0, 1]$ and satisfy:

$$\theta(A \to BC) = \frac{\mathbb{E}(A \to BC)}{\mathbb{E}(A)}$$

$$\theta(A \to a)) = \frac{\mathbb{E}(A \to a)}{\mathbb{E}(A)}$$

Parameters have interpretation as conditional probabilities in a top down generative process starting with S.

Bottom up parameterization of Weighted CFGs

Trivial Identity for WCFGs

$$I(A)O(A) = \mathbb{E}(A)$$

Stipulate that $O(A) = 1$, and $I(A) = \mathbb{E}(A)$: each nonterminal defines a probability distribution over its contexts.
Parameters are no longer in $[0, 1]$ but satisfy:

$$\theta(A \leftarrow BC) = \frac{\mathbb{E}(A \leftarrow BC)}{\mathbb{E}(B)\mathbb{E}(C)}$$

$$\theta(A \leftarrow a) = \mathbb{E}(A \leftarrow a)$$

Identifiability from trees

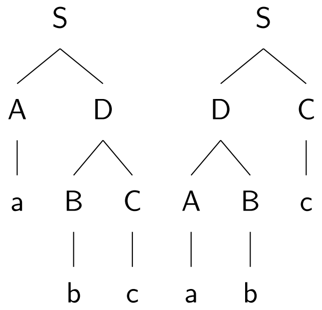
$$\underbrace{\theta(A \to BC)}_{\substack{\text{parameter} \\ \text{of the grammar}}} = \underbrace{\frac{\mathbb{E}(A \to BC)}{\mathbb{E}(A)}}_{\substack{\text{depends only on} \\ \text{distribution over trees}}}$$

- We can't have two PCFGs that generate the same distribution over trees: $\mathbb{P}(t; G_1) = \mathbb{P}(t; G_2)$ implies $G_1 = G_2$
- This gives us a recipe to learn the parameters: count and normalise.

The major problem:

Non identifiability of PCFGs and CFGs from strings [Hsu et al., 2013]

We *can* have two PCFGs that generate the same distribution over *strings*; for example $\mathbb{P}(abc) = 1$



Distributional learning

- The kitten is over there.
- I want a kitten for Christmas.
- What a cute kitten!

Distributional learning

- The kitten is over there.
- I want a kitten for Christmas.
- What a cute kitten!

The work "kitten" occurs in these contexts:

- The □ is over there.
- I want a □ for Christmas.
- What a cute □!

So does "dog". But the word "the" does not.

Probabilistic language

Given a probability distribution, $\mathbb{P}$, over strings of symbols (Σ^*).

Distributional distribution

A string u defines a probability distribution $\mathbf{u}$ over its contexts:

$$l \Box r \text{ has probability } \frac{\mathbb{P}(lur)}{\mathbb{E}(u)}$$

Anchored Context Free Grammars

Stratos et al. [2016]

Assume that for every nonterminal A there is a terminal a which occurs only in the production $A \to a$.
Reasonable assumption if number of words is much greater than number of nonterminals.

Example in English

- she (NP)
- the (Det)
- kitten (N)

The strings

she and *the kitten*

The production

$NP \rightarrow Det\ N$

The strings

she and *the kitten*

The production

$NP \rightarrow Det\ N$

Two old ideas [Harris, 1955]:

1. There should be high MI between *the* and *kitten*
2. *she* and *the kitten* should occur in the same contexts
   she and **the kitten** should be similar.

Divergence between context distributions

Rényi divergence, $\alpha = \infty$, between discrete distributions P and Q:

$$\mathcal{R}_\infty (P \| Q) = \log \sup_x \frac{P(x)}{Q(x)}$$

- Asymmetric
- Satisfies triangle inequality
- In $[0, \infty]$

Define for strings u and v

$$\mathcal{R}_\infty (\mathbf{u} \| \mathbf{v}) = \log \sup_{l,r} \frac{\mathbb{P}(lur)/\mathbb{E}(u)}{\mathbb{P}(lvr)/\mathbb{E}(v)}$$

Two further conditions

Strict Upward Monotonicity

Adding any new production will increase the set of strings generated by the grammar.

Local Unambiguity

A weak condition limiting how ambiguous the grammar is:
For every production $A \to \alpha$, there is a string which always uses that production "in the same place".
For $\pi = A \leftarrow BC$, there is a string $w = luvr$ such that

$$\Omega(S, w) = \Xi(A, l\Box r) \oplus \pi(\Omega(B, u), \Omega(C, v))$$

Under these three conditions:

Given nonterminals A, B, C anchored by a, b, c resp.:

$$\underbrace{\log \theta(A \leftarrow BC)}_{\text{bottom-up parameter}}$$

Under these three conditions:

Given nonterminals A, B, C anchored by a, b, c resp.:

$$\underbrace{\log \theta(A \leftarrow BC)}_{\text{bottom-up parameter}} = \underbrace{\log \frac{\mathbb{E}(bc)}{\mathbb{E}(b)\mathbb{E}(c)}}_{\text{PMI of rhs}}$$

Under these three conditions:

Given nonterminals A, B, C anchored by a, b, c resp.:

$$\underbrace{\log \theta(A \leftarrow BC)}_{\text{bottom-up parameter}} = \underbrace{\log \frac{\mathbb{E}(bc)}{\mathbb{E}(b)\mathbb{E}(c)}}_{\text{PMI of rhs}} - \underbrace{\mathcal{R}_\infty(\mathbf{a}\|\mathbf{bc})}_{\text{divergence of lhs from rhs}}$$

Right hand side depends only on the distribution over strings.

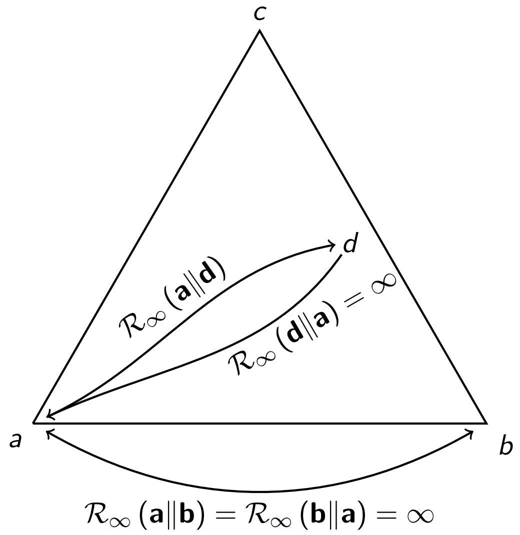
Lexical rule

Given nonterminal A anchored by a, and a terminal d:

$$\underbrace{\log \theta(A \leftarrow d)}_{\text{bottom-up parameter}} = \underbrace{\log \mathbb{E}(d)}_{\text{lexical frequency}} - \underbrace{\mathcal{R}_\infty(\mathbf{a} \| \mathbf{d})}_{\text{divergence of lhs from rhs}}$$

Identifying terminals as anchors

Context distributions of all terminals will lie in the convex hull of the anchors:



Theorem [Clark and Fijalkow, 2020]

There is a computationally efficient (trivial) consistent estimator from strings, for all PCFGs whose underlying CFG is

1. In Chomsky Normal Form
2. Anchored
3. Strictly Upward Monotonic
4. Locally Unambiguous

Using naive plug-in estimators that are slow to converge.

Identifiability

For this class of grammars $\mathbb{P}(s \mid G_1) = \mathbb{P}(s \mid G_1)$ implies G_1 is isomorphic to G_2.

Positive Discussion

The first[1] strong probabilistic result for learning PCFGs for strings. (in 2020!)

- Hyper-parameter free; Input is just a sequence of strings.
- Learns
  - The nonterminals, and how many there are
  - The lexicon
  - The syntactic rules for combining these categories
  - The correct probabilities for each production.

[1] with some caveats.

The grammar is the parser [Phillips, 1996]

This algorithm contains no parsing:

- Which comes first the parser or the grammar?

Formalised as a grammar learning algorithm:

- One can equally well parse on the fly using just exemplars, and get the same $\mathrm{argmax}_t \mathbb{P}(t \mid s)$.
- Then the best parse is the shortest path from the yield to S: [Klein and Manning, 2005], using $\mathcal{R}_\infty (\cdot \| \cdot)$ as a distance

Critical Discussion

Any learning theorem

If conditions are satisfied then we learn under some model.

Two questions:

- Is the antecedent too strong?
- Is the consequent too weak?

- Do natural languages satisfy these conditions? Putting aside the intrinsic limitations of CFGs.
- The theorem says nothing about the speed of convergence: are the algorithms too slow to converge?

Do natural languages fit in this class?

Obviously not since CFGs are inadequate but are the assumptions reasonable?
A nonterminal A can have no anchors if:

- It doesn't generate any strings of length 1.
- Or all of them are ambiguous.

We can look at a syntatically annotated corpus of English Child directed speech [Pearl and Sprouse, 2012].

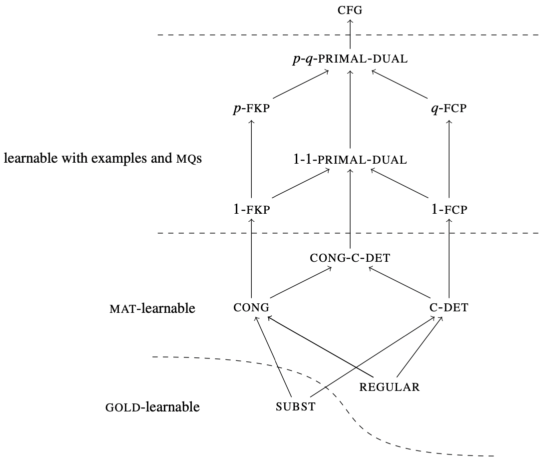
Corpus study

t	$P(I = 1)$	w_{max}	$P(\text{t}\|w_{\text{max}})$
ADJP	0.67	careful	0.85
ADVP	0.84	already	1.0
FRAG	0.3	seal	0.2
INTJ	0.87	hmm	1.0
NP	0.7	he	1.0
PP	0.078	for	0.13
PRT	0.99	off	0.72
S	0.017	-	-
SBAR	0.0046	if	0.0024
SBARQ	0.0	-	-
SQ	0.021	-	-
VP	0.11	crying	0.82
WHADVP	0.98	when	1.0
WHNP	0.8	who	0.95

Weak learning hierarchy

[Clark and Yoshinaka, 2016]

Instead of having a single anchor a, have a small set of strings $w_1, \ldots, w_k$, of arbitrary length, such that the shared distribution of these strings correctly defines the nonterminal.



Computational experiments

- Language acquisition happens not asymptotically but with fairly small amounts of data: in the worst cases the divergences are very hard to estimate.
- What happens if the conditions don't hold, or hold only approximately?

Replace a simple algorithm that is easy to analyse with something more data efficient.

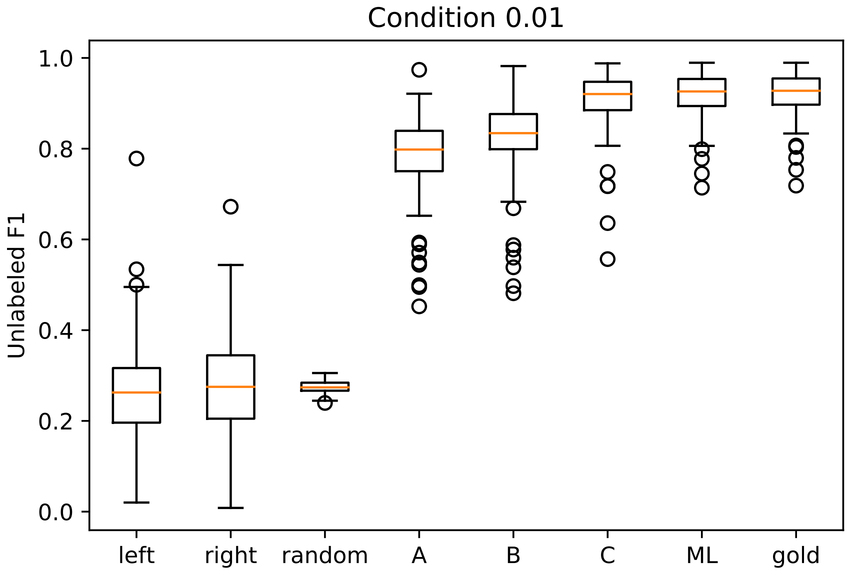
Protocol for simulations

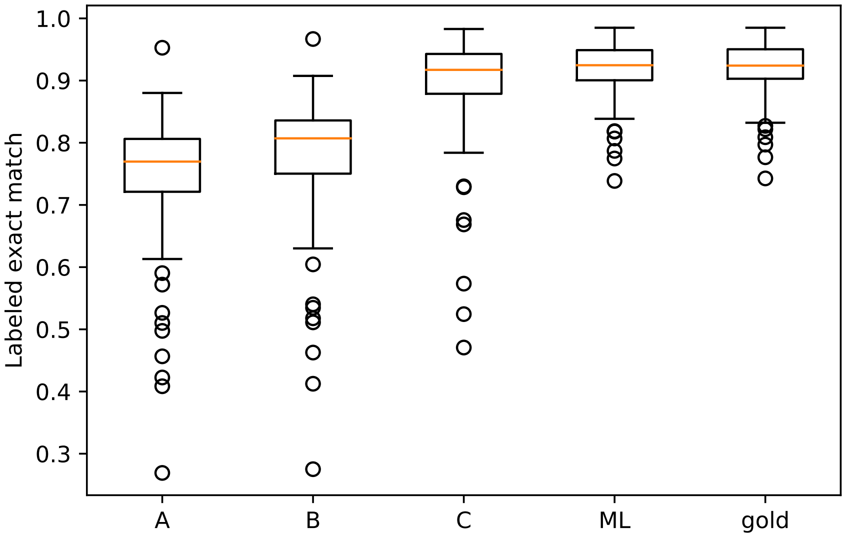
- Generate synthetic grammars that match the observed statistical properties of Child-Directed Speech:
  - distribution of sentence lengths (zero truncated Poisson, mean 5)
  - Zipfian unigram distribution
- $|V| = 10$, $|\Sigma| = 1000$, with *all CNF productions allowed*.
- Control ambiguity with a Dirichlet hyperparameter α for the binary rules.
- Sample 10^6 strings for a training set.
- Give true number of nonterminals (10) to the algorithm.
- Evaluate using supervised parsing metrics on 10^3 trees, with a maximum length of 20.

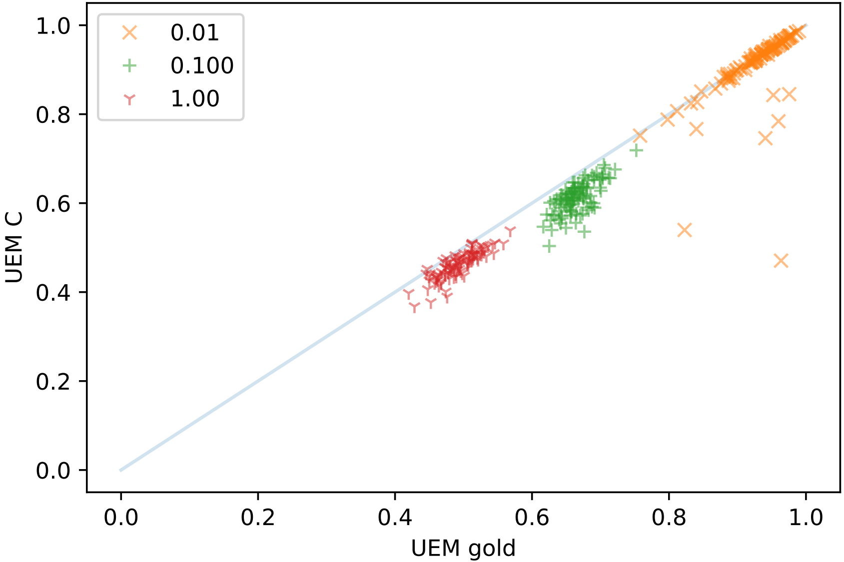
Implementation

- Standard NLP techniques: cluster words based on local distributional context to get a low dimensional approximation.
- Approximate $\mathcal{R}_\infty(\cdot\|\cdot)$ with $\mathcal{R}_5$
- Three outputs:

  A Bottom up WCFG
  B plus 1 iteration of EM
  C plus 10 iterations of EM
- Set hyperparameters (a few thresholds etc), debug etc. on some example grammars and then test on fresh grammars without any further tuning.


Condition 0.01
Unlabeled F1
1.0
0.8
0.6
0.4
0.2
0.0
left
right
random
A
B
C
ML
gold


Labeled exact match
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
A
B
C
ML
gold



0.01
0.100
1.00
UEM C
UEM gold
0.0
0.2
0.4
0.6
0.8
1.0

Discussion

- None of the grammars here satisfy the conditions since they contain all productions.
- The hypothesis class of the learner now is effectively all grammars in CNF.
- Cheap algorithms ($1 per language).
- Learning the number of nonterminals is straightforward but a bit more expensive, and complicates the evaluation.
- These are an order of magnitude smaller than natural language grammars; but we can learn nearly all of them effectively.

Take-home point

In the average case, PCFGs are strongly learnable if they are not too ambiguous.

Weak and strong inadequacy of CFGs

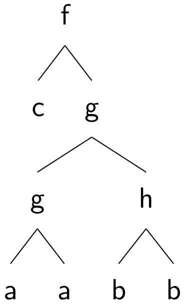
Shieber [1985] showed that CFGs are weakly inadequate but we already knew that they were strongly inadequate (e.g. Gazdar et al. [1985]).

One step up: Vijay-Shanker and Weir [1994]

Four equivalent formalisms: here we use the first one:

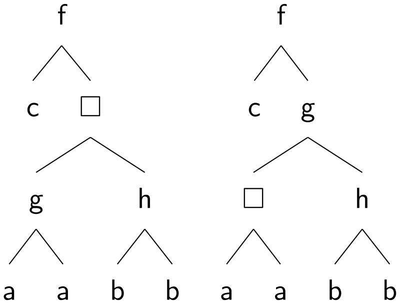
- Tree-adjoining grammar via footed simple context-free tree grammars [Kepser and Rogers, 2011]
- Head grammars: well-nested multiple CFGs of dimension 2 [Seki et al., 1991]
- Linear Indexed Grammars
- Combinatory Categorial Grammar

A stochastic language of binary trees

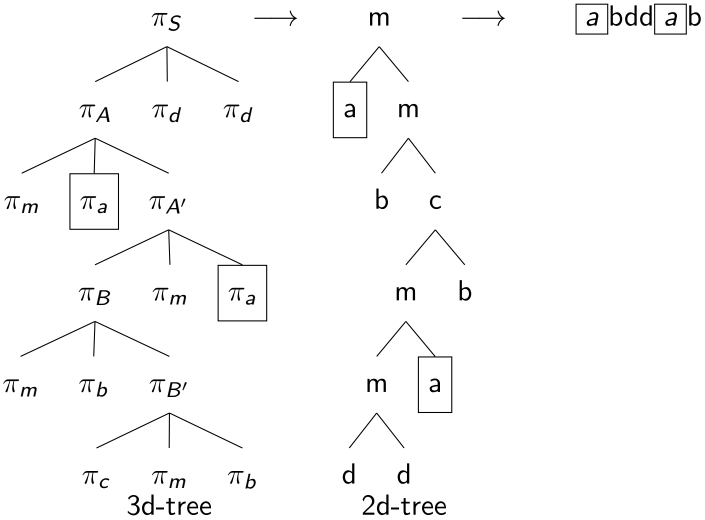


Generalise the notion of context

g occurs in the contexts

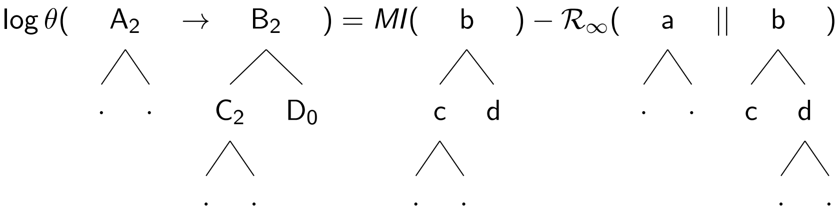


[Rogers, 2003]



$\pi_S \longrightarrow$ m $\longrightarrow$ $\boxed{a}$bdd$\boxed{a}$b
π_A π_d π_d
π_m $\boxed{\pi_a}$ $\pi_{A'}$
π_B π_m $\boxed{\pi_a}$
π_m π_b $\pi_{B'}$
π_c π_m π_b
3d-tree
$\boxed{\text{a}}$ m
b c
m b
m $\boxed{\text{a}}$
d d
2d-tree

Same parameter identities



Result [Clark, 2021]

Exactly the same algorithm except we need to handle nonterminals of rank 0 and rank 2 and perhaps more.

- Anchored
- Locally unambiguous
- Strictly upward monotonic

Dendrophilia-squared

Apply the same algorithm twice:

1. Apply to strings (1d trees) to get some surface structure trees (2d trees)
2. ...
3. Apply to 2d trees to get a "deep structure" derivation tree.

Result [Clark, 2021]

Exactly the same algorithm except we need to handle nonterminals of rank 0 and rank 2 and perhaps more.

- Anchored
- Locally unambiguous
- Strictly upward monotonic

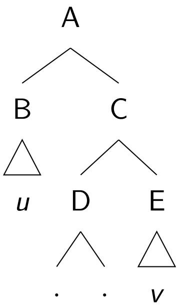
Dendrophilia-squared

Apply the same algorithm twice:

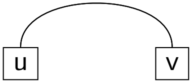
1. Apply to strings (1d trees) to get some surface structure trees (2d trees)
2. ...
3. Apply to 2d trees to get a "deep structure" derivation tree.

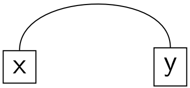
Can we go directly from strings to a suitable structure?

Wellnestedness

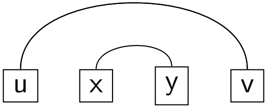


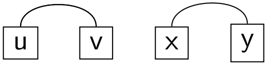
When we consider it as a string is $u\square v$



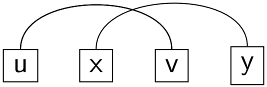


Well-nested





Not well-nested



Well-nestedness

Kanazawa and Salvati [2012]

- Productions can be binarised, which implies more efficient parsing [Gómez-Rodríguez et al., 2010]
- Excludes excessively free word order (MIX language) [Kanazawa and Salvati, 2012]
- Corpus studies suggest it generally holds [Kuhlmann and Nivre, 2006].

Corpus: Kuhlmann and Nivre [2006]

	Danish (DDT)	Czech (PDT)
projective	84.95%	76.86%
well-nested	99.89%	99.89%

Simplistic model

The more you learn the easier it is to learn: but how does the whole process start? There are regularities that learners can and do exploit, but they need to know them first.

- Ignores other information sources:
  - Phonology: Morgan and Demuth [1996]
  - Semantics: Pinker [1996],Abend et al. [2017]
- Can't expect a single model to account for all of language acquisition.
- The two steps do not fit together:
  - Even if we have a perfect grammar we still need semantics to recover some structure needed as input to the second phase.
  - What are the symbols in the trees?

Desiderata

- Descriptively adequate
- Easy for humans to reason about
  - Natural diagrams on a 2d page
  - Have clean mathematical properties

Desiderata

- Descriptively adequate
- Easy for humans to reason about
  - Natural diagrams on a 2d page
  - Have clean mathematical properties
- Where do these structures come from?
  1. Processing: efficiently parseable
  2. Acquisition: learnable from evidence available to the child
  3. Cultural Evolution: why do languages have these structures?
  4. Biological Evolution: why do we have the ability to learn these structures?

Pure Speculation

Questions

- How can we account for the origin of "movement"?
- Why are there syntactic islands as constraints?

Pure Speculation

Questions

- How can we account for the origin of "movement"?
- Why are there syntactic islands as constraints?

A sketch of an argument:

- Dendrophilia will apply to trees as well as strings unless stipulated otherwise.
- Strict Upward Monotonicity implies that all "legal" rules will be learned: so the learner *must* hypothesize "movement" rules when the situation permits.
- Local Unambiguity implies that we must have restrictions on movement or the tree grammar component will be too ambiguous.

Take home points

Technical claim

Learning large classes of phrase structure grammars, including mildly context-sensitive grammars, defined by explicit structural constraints is possible just from strings; in a computationally efficient, strong, probabilistic model.

Theoretical claims

- Early acquisition of syntax is driven by distributional learning.
- We can unify various levels of syntactic structure using multidimensional trees [Rogers, 2003].
- Movement is acquired by the same mechanism as phrase structure acquisition.
- Well-nestedness seems to be an important restriction.

Bibliography I

Omri Abend, Tom Kwiatkowski, Nathaniel J Smith, Sharon Goldwater, and Mark Steedman. Bootstrapping language acquisition. *Cognition*, 164: 116–143, 2017.

Robert D Borsley and Berthold Crysmann. Unbounded dependencies. *Head-Driven Phrase Structure Grammar: The handbook*, pages 537–594, 2021.

Alexander Clark. Strong learning of probabilistic tree adjoining grammars. *Proceedings of the Society for Computation in Linguistics*, 4(48), 2021. URL `https://scholarworks.umass.edu/scil/vol4/iss1/48`.

Alexander Clark and Nathanaël Fijalkow. Consistent unsupervised estimators for anchored PCFGs. *Transactions of the Association for Computational Linguistics*, 8:409–422, 2020. doi: 10.1162/tacl_a_00323. URL `https://doi.org/10.1162/tacl_a_00323`.

Bibliography II

Alexander Clark and Ryo Yoshinaka. Distributional learning of context-free and multiple context-free grammars. In Jeffrey Heinz and M. José Sempere, editors, *Topics in Grammatical Inference*, pages 143–172. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016. ISBN 978-3-662-48395-4. doi: 10.1007/978-3-662-48395-4_6. URL `http://dx.doi.org/10.1007/978-3-662-48395-4_6`.

W Tecumseh Fitch. Toward a computational framework for cognitive biology: unifying approaches from cognitive neuroscience and comparative cognition. *Physics of Life Reviews*, 11(3):329–364, 2014.

G. Gazdar, E. Klein, G. Pullum, and I. Sag. *Generalised Phrase Structure Grammar*. Basil Blackwell, 1985.

Carlos Gómez-Rodríguez, Marco Kuhlmann, and Giorgio Satta. Efficient parsing of well-nested linear context-free rewriting systems. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 276–284, 2010.

Bibliography III

Zellig Harris. From phonemes to morphemes. *Language*, 31:190–222, 1955.

James Jay Horning. *A study of grammatical inference*. PhD thesis, Computer Science Department, Stanford University, 1969.

D. Hsu, S. M. Kakade, and P. Liang. Identifiability and unmixing of latent parse trees. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1520–1528, 2013.

Makoto Kanazawa and Sylvain Salvati. MIX is not a tree-adjoining language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 666–674. Association for Computational Linguistics, 2012.

Stephan Kepser and Jim Rogers. The equivalence of tree adjoining grammars and monadic linear context-free tree grammars. *Journal of Logic, Language and Information*, 20(3):361–384, 2011.

Bibliography IV

D. Klein and C.D. Manning. Parsing and hypergraphs. *New developments in parsing technology*, pages 351–372, 2005.

Marco Kuhlmann and Joakim Nivre. Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL 2006 main conference poster sessions*, pages 507–514, 2006.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 523–530, 2005.

Richard Cornelis Antonius Moot. *Proof nets for linguistic analysis*. PhD thesis, University of Utrecht, 2002.

J.L. Morgan and K. Demuth. *Signal to Syntax: Bootstrapping from Speech to Grammar in Early Acquisition*. Lawrence Erlbaum Associates, 1996.

Bibliography V

L. Pearl and J. Sprouse. Computational models of acquisition for islands. In J. Sprouse and N. Hornstein, editors, *Experimental syntax and island effects*. Cambridge University Press, Cambridge, UK, 2012.

Colin Phillips. *Order and Structure*. PhD thesis, MIT, 1996.

Steven Pinker. *Language Learnability and Language Development*. Harvard University Press, second edition, 1996.

Norvin Waldemar Richards. *What moves where when in which languages?* PhD thesis, Massachusetts Institute of Technology, 1997.

James Rogers. Syntactic structures as multi-dimensional trees. *Research on Language and Computation*, 1(3-4):265–305, 2003.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):229, 1991.

Bibliography VI

Stuart M. Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343, 1985.

Noah A Smith and Mark Johnson. Weighted and probabilistic context-free grammars are equally expressive. *Computational Linguistics*, 33(4):477–491, 2007.

Karl Stratos, Michael Collins, and Daniel Hsu. Unsupervised part-of-speech tagging with anchor hidden markov models. *Transactions of the Association for Computational Linguistics*, 4:245–257, 2016.

John Philip Torr. *Wide-coverage statistical parsing with Minimalist Grammars*. PhD thesis, The University of Edinburgh, 2019.

K. Vijay-Shanker and David J. Weir. The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory*, 27(6):511–546, 1994.