Bonobos, children, and fear of trees

Rob Truswell
rob.truswell@ed.ac.uk

DLLD lecture, Ghent, 8/5/17
1. The dendrophilia hypothesis (Fitch 2014): Humans pair strings with hierarchically structured representations to a much greater extent than other animals.

2. Results from, and limitations of, previous studies.


4. Comparison with an acquisition study (Gertner & Fisher 2012) suggests that children are not particularly dendrophilic (they don’t automatically see trees, even when it should be easy). Rather, Kanzi is dendrophobic.
Section 1

Background
Big questions

The Faculty of Language: What Is It, Who Has It, and How Did It Evolve?

Marc D. Hauser,¹* Noam Chomsky,² W. Tecumseh Fitch¹

Available online at www.sciencedirect.com

Cognition 95 (2005) 201–236

www.elsevier.com/locate/COGNIT

The faculty of language: what’s special about it?*

Steven Pinkerᵃ,*, Ray Jackendoffᵇ
Linguistic cognition is special in lots of way
Claims of specialization from Pinker & Jackendoff (2005)

- Distinctive neuroanatomy of speech processing (Hickok & Poeppel 2007);
- Distinctive speech production anatomy (Lieberman 2003);
- Duality of patterning (Hockett 1960);
- Words;
- Syntax, including
  - Phrase structure
  - Syntactic categories
  - Long-distance dependencies
  - Recursion
Phrase structure is very, very useful

- One design feature of language is **productivity** (Chomsky 1957, Hockett 1960), ‘the capacity to say things that have never been said or heard before and yet to be understood by other speakers of the language’ (Hockett 1960: 6)

- Productivity is grounded in a combination of:
  1. Phrase structure
  2. Compositional semantics

- If you know 1,000 uncombinable words, you can express 1,000 things.

- If you know 500 nouns, 495 verbs, 2 determiners, and 3 rules:
  1. $S \to NP \ VP$
  2. $VP \to V \ NP$
  3. $NP \to D \ N$

  Then you know $500 + 495 + 2 + 3 = 1000$ things.

- But you can express $2 \times 500 \times 495 \times 2 \times 500 = 495,000,000$ things.
Phrase structure and productivity

▶ If you live for 80 years, you’d have to say one of these sentences every 5 seconds to get through them all.

▶ And never sleep.

▶ And if you did that, no-one would listen to you anyway.
Phrase structure and recursion

- Rewrite rules like $S \rightarrow NP \ VP$ reflect the intuitions that:
  - Constituency is an important part of sentence structure;
  - Constituents nest:
    - Take some little units (e.g. words) and combine them to make bigger units.
    - Combine the bigger units with other units to make even bigger units.
    - Etc.
  - The labels on the right of ‘$\rightarrow$’ tell us what we can combine.
  - The label on the left of ‘$\rightarrow$’ tells us what we get when we combine those units.
  - Given this format, recursive embedding of like within like is inevitable.
Phrase structure and recursion

1. $S \rightarrow NP \ VP$
2. $VP \rightarrow V \ NP$
3. $NP \rightarrow D \ N$
4. $VP \rightarrow V \ that \ S$

The monkey caressed the bishop.
   - The bishop thinks that the monkey caressed the bishop.
   - The monkey thinks that the bishop thinks that the monkey caressed the bishop.
   - The bishop thinks that the monkey thinks that the bishop thinks that the monkey caressed the bishop.
   - ...

Suddenly, we have infinitely many sentences to play with.

(This can be even simpler: $N \rightarrow N \ N$).

In sum,
   - Phrase structure leads us to expect recursion.
   - Recursion captures the ‘infinite use of finite media’.
   - But phrase structure (recursive or not) already explains productivity.
The recursion-only hypothesis

- Hauser, Chomsky & Fitch (2002) divided ‘What’s special about the Faculty of Language?’ into three subparts.
  1. Linguistic behaviour draws on a wide range of cognitive capacities.
  2. Which of those capacities are domain-specific?
  3. Which of those capacities are species-specific?
- They then hypothesize that the only domain-specific and the only species-specific component of linguistic cognition (‘Faculty of Language in the Narrow sense’, or FLN) is ‘recursion’ (p.1569), or ‘the core computational mechanisms of recursion as they appear in narrow syntax and the mappings to the interfaces’ (p.1573).
- This is the recursion-only hypothesis.
Criticism of the recursion-only hypothesis

- The recursion-only hypothesis is hard to pin down, because it’s worded in contradictory ways in different parts of Hauser et al. (2002).
- It generated a huge amount of discussion, partly because of these contradictions.
- But two clear points emerged.
  1. There are other domain-specific and species-specific aspects of linguistic cognition (see list above).
  2. Recursive structure (as generated by syntactic mechanisms) is not unique to language.
     - But it may be unique to humans.
     - And the nature of recursive linguistic structure may be distinctive in other ways (‘... and the mapping to the interfaces’).
Recursive structure in visual processing
(Pinker & Jackendoff 2005)

Fig. 1. Recursion in visual grouping.
Recursive structure in planning

(Jackendoff 2007)
The Dendrophilia hypothesis

▶ In part because of these criticisms, Fitch (2014) proposed a weaker hypothesis:

‘Humans have a multi-domain capacity and proclivity to infer tree structures from strings, to a degree that is difficult or impossible for most non-human animal species.’ (Fitch 2014: 352)

▶ Three differences:

1. Tree structures (e.g. phrase structure), not narrowly recursion.
2. Domain-general, not domain-specific.

▶ But this retains the really useful property of Hauser et al. (2002): It encourages targeted comparisons (between species, between domains).
Section 2

Artificial Grammar Learning studies
Fitch & Hauser’s tamarins

- Fitch & Hauser (2004) tested the ability of humans and cotton-top tamarins to learn to recognize two patterns:

1. \((ab)^n\)
   - \(ab\)
   - \(abab\)
   - \(ababab\)
   - \(abababab\)
   - \(\ldots\)

2. \(a^n b^n\)
   - \(ab\)
   - \(aabb\)
   - \(aaabbb\)
   - \(\ldots\)

- Humans could learn both; cotton-top tamarins only learned the first.
Why these strings?

▶ You can use phrase structure rules to analyse either of these sets of strings.

1. \( S \rightarrow a \ T \)
2. \( T \rightarrow b \ S \)
3. \( T \rightarrow b \)

But you can also generate them in other, less complicated ways.
Memory and phrase structure

To produce strings generated by a phrase structure grammar, you can follow these steps:

1. Read a rewrite rule for the start symbol (S), left to right.
2. If the next character is the name of a terminal node, output that terminal node.
3. If the next character is the name for a nonterminal node, start a subroutine.
   3.1 Look at a rewrite rule for that nonterminal node.
   3.2 If the next character is the name of a terminal node, output that terminal node.
   3.3 If the next character is the name for a nonterminal node, start a subroutine.

...  

...  

When you start a subroutine within a subroutine within, ..., you need to keep a record of what subroutines you’re in the middle of, in which order.

This is called a stack.
Phrase structure with stacks

- $S \rightarrow NP \ VP$
- $NP \rightarrow D \ N$
- $VP \rightarrow V$
- $D \rightarrow a$
- $N \rightarrow dog$
- $V \rightarrow barked$

1. Stack: $S$
Phrase structure with stacks

- $S \rightarrow NP \ VP$
- $NP \rightarrow D \ N$
- $VP \rightarrow V$
- $D \rightarrow a$
- $N \rightarrow dog$
- $V \rightarrow barked$

1. Stack: $S$
2. Stack: $S, NP$
Phrase structure with stacks

- $S \rightarrow NP \ VP$
- $NP \rightarrow D \ N$
- $VP \rightarrow V$
- $D \rightarrow a$
- $N \rightarrow \text{dog}$
- $V \rightarrow \text{barked}$

1. Stack: S
2. Stack: S, NP
3. Stack: S, NP, D
Phrase structure with stacks

- $S \rightarrow NP \ VP$
- $NP \rightarrow D \ N$
- $VP \rightarrow V$
- $D \rightarrow a$
- $N \rightarrow dog$
- $V \rightarrow barked$

1. Stack: $S$
2. Stack: $S$, $NP$
3. Stack: $S$, $NP$, $D$
4. Stack: $S$, $NP$, $N$

Output: $a$
Phrase structure with stacks

- $S \rightarrow NP \ VP$
- $NP \rightarrow D \ N$
- $VP \rightarrow V$
- $D \rightarrow a$
- $N \rightarrow dog$
- $V \rightarrow barked$

1. Stack: $S$
2. Stack: $S$, $NP$
3. Stack: $S$, $NP$, $D$
4. Stack: $S$, $NP$, $N$
5. Stack: $S$, $VP$

Output: a
dog

Output: a dog
Phrase structure with stacks

- $S \rightarrow NP \ VP$
- $NP \rightarrow D \ N$
- $VP \rightarrow V$
- $D \rightarrow a$
- $N \rightarrow dog$
- $V \rightarrow barked$

1. Stack: $S$
2. Stack: $S, NP$
3. Stack: $S, NP, D$
4. Stack: $S, NP, N$  
   Output: a
5. Stack: $S, VP$  
   Output: a dog
6. Stack: $S, VP, V$  
   Output: a dog
Phrase structure with stacks

▶ $S \rightarrow NP\ VP$
▶ $NP \rightarrow D\ N$
▶ $VP \rightarrow V$
▶ $D \rightarrow a$
▶ $N \rightarrow dog$
▶ $V \rightarrow barked$

1. Stack: $S$
2. Stack: $S, NP$
3. Stack: $S, NP, D$
4. Stack: $S, NP, N$
5. Stack: $S, VP$
6. Stack: $S, VP, V$
7. Stack: $\emptyset$

Output: a
Output: a dog
Output: a dog barked
The power of a stack

- The point of the stack is to allow us to say:
  - I just produced a determiner.
    - But I’m still in the middle of producing an NP.
      - And once I’m done with the NP, I’ll get back to producing an S.
  - This allows us to represent arbitrarily many dependencies between units, regardless of the complexity of intervening material.
    - $S \rightarrow \text{if } S \text{ then } S$
    - $S \rightarrow \text{either } S \text{ or } S$
    - $S \rightarrow \text{both } S \text{ and } S$
    - if [either [she leaves] or [both [he arrives] and [he starts drinking]]], then [either [we’re doomed] or [we need a miracle]].
- If you produce *either*, later you must produce *or*.
- But the intervening sentence can be as complicated as you like (can be as deeply embedded as you like).
Our phrase structure grammar for $a^n b^n$ uses arbitrarily many long distance dependencies:

- $S \rightarrow a \ S \ b$: ‘I’ll produce an $a$, then start another $S$, which can be as complicated as you like, but eventually I’ll produce a $b$ to go with that $a$.

1. Stack: $S$ 
   Output: $a$


$a^n b^n$ in a phrase structure grammar

- Our phrase structure grammar for $a^n b^n$ uses arbitrarily many long distance dependencies:
  - $S \to a \ S \ b$: ‘I’ll produce an $a$, then start another $S$, which can be as complicated as you like, but eventually I’ll produce a $b$ to go with that $a$.

1. Stack: $S$  
   Output: $a$
2. Stack: $S \ S$  
   Output: $a \ a$
Our phrase structure grammar for $a^n b^n$ uses arbitrarily many long distance dependencies:

- $S \rightarrow a S b$: ‘I’ll produce an $a$, then start another $S$, which can be as complicated as you like, but eventually I’ll produce a $b$ to go with that $a$.

1. Stack: $S$  
   Output: $a$
2. Stack: $S S$  
   Output: $a a$
3. Stack: $S S S$  
   Output: $a a a$
$a^n b^n$ in a phrase structure grammar

Our phrase structure grammar for $a^n b^n$ uses arbitrarily many long distance dependencies:

- $S \rightarrow a \ S \ b$: ‘I’ll produce an $a$, then start another $S$, which can be as complicated as you like, but eventually I’ll produce a $b$ to go with that $a$.

1. Stack: $S$ 
   Output: $a$
2. Stack: $S \ S$ 
   Output: $a \ a$
3. Stack: $S \ S \ S$ 
   Output: $a \ a \ a$
4. Stack: $S \ S$ 
   Output: $a \ a \ a \ b$
Our phrase structure grammar for $a^n b^n$ uses arbitrarily many long distance dependencies:

- $S \rightarrow a S b$: ‘I’ll produce an $a$, then start another $S$, which can be as complicated as you like, but eventually I’ll produce a $b$ to go with that $a$.

1. Stack: $S$  
   Output: $a$
2. Stack: $S S$  
   Output: $a a$
3. Stack: $S S S$  
   Output: $a a a$
4. Stack: $S S$  
   Output: $a a a b$
5. Stack: $S$  
   Output: $a a a b b$
$a^n b^n$ in a phrase structure grammar

Our phrase structure grammar for $a^n b^n$ uses arbitrarily many long distance dependencies:

- $S \rightarrow a \ S \ b$: ‘I’ll produce an $a$, then start another $S$, which can be as complicated as you like, but eventually I’ll produce a $b$ to go with that $a$.

1. Stack: $S$  Output: $a$
2. Stack: $S \ S$  Output: $a \ a$
3. Stack: $S \ S \ S$  Output: $a \ a \ a$
4. Stack: $S \ S$  Output: $a \ a \ a \ b$
5. Stack: $S$  Output: $a \ a \ a \ b \ b$
6. Stack: $\emptyset$  Output: $a \ a \ a \ b \ b \ b$
(ab)^n without phrase structure

- You don’t need anything so complicated to make (ab)^n.
- What you can do next is entirely determined by what you’ve just done.
  - Start: make an a
  - Just made an a: make a b.
  - Just made a b: make an a, or stop.

No subroutines, no remembering which subroutines you’re in the middle of. Machines for recognizing patterns like this are called finite state machines.

- You can’t do this with a^n b^n: you need to remember how many a’s you’ve produced, so you can pair each a with a b.
Implications and criticism

- When Fitch & Hauser’s tamarins learned \((ab)^n\) but not \(a^n b^n\), this was interpreted as telling us something about the kind of things they could (and couldn’t) remember.

- And the phrase-structure grammar for \(a^n b^n\) made it very tempting to relate the difference to grammatical cognition (e.g. Dendrophilia).

- But several criticisms emerged.
  - Starlings can learn \(a^n b^n\) (with huge amounts of training; Gentner et al. 2005).
  - Humans (undergraduates) often learned \(a^n b^m\) rather than \(a^n b^n\) (they didn’t notice that it was the same number of a’s and b’s; Hochmann et al. 2008).
  - There are lots of grammars compatible with these two stringsets; neither stringset is representative of the class of grammars that it’s meant to represent (Rogers & Pullum 2011).
  - The phrase-structure grammar capable of generating \(a^n b^n\) involves centre-embedding, which humans struggle hugely with.
$a^n b^n$ and centre-embedding

- The last points are deadly.
- Rogers & Pullum (2011) on $(ab)^n$: easy.
  - Ding dong
  - Ding dong ding dong
  - Ding dong ding dong ding dong
- Rogers & Pullum (2011) on $a^n b^n$: quickly becomes incomprehensible.
  - $[S [NP \text{ People}] [VP \text{ left}]]$
  - $[S [NP \text{ People} [RC [NP \text{ people}] [VP \text{ left}]]] [VP \text{ left}]]$
  - $[S[NP \text{ People} [RC[NP \text{ people} [RC[NP \text{ people} [VP \text{ left}]]]]] [VP \text{ left}]] [VP \text{ left}]]$
- However we recognize $aaabbb$, it surely isn’t like the way we fail to understand People people people left left left.
- Natural intuition: we recognize $a^n b^n$ by counting.
- Counter grammars (Chomsky 1959) are weaker than phrase structure grammars.
- Rather than putting ‘I’ve started an S/NP/etc.’ on the stack, you just put a ‘1’ on the stack (no differentiation of categories).
A counter grammar implementation of $a^nb^n$

- Start by producing $a$ and adding a 1 to the stack.
- Do that as often as you like.
- Then start producing $b$ and removing a 1 from the stack.
- Do that until there are no 1s left.
- Stop.
Artificial Grammar Learning studies: Summary

- Real results have emerged from studies like Fitch & Hauser (2004).
- But those results are more nuanced than originally claimed.
- And as they become more nuanced, they become harder to relate to natural language.
  - Counter grammars have no known linguistic use.
  - Linguistically relevant formalisms are not plausible ways of capturing $a^n b^n$.

‘So what can one say about a mechanism that can learn a properly context-free pattern [like $a^n b^n$]? For one thing, it is not finite-state... Beyond that, there is very little if anything that we can determine about the nature of that information and how it is used simply from the evidence that an organism can learn the pattern.’ (Jäger & Rogers 2012: 1962)
Section 3

The Kanzi corpus
Interpreted strings

- A problem with $a^n b^n$ is that it is uninterpreted.
- Because all participants had to do was recognize it, it’s impossible to tell how they represented it.
- We will look instead at a source of evidence about a nonhuman primate’s behaviour in response to a language with hierarchical phrase structure, namely spoken English.
- We can then infer aspects of the subject’s interpretation of an utterance from his behaviour, and aspects of the grammatical representation of the utterance from that interpretation.
- The evidence we will find broadly supports Fitch’s Dendrophilia hypothesis, from a complementary perspective.
Kanzi

- 36-year-old bonobo, 8 at time of tests discussed here.
- Spontaneously acquired language-like communication systems during training sessions targeted at his mother, Matata.
- Housed in the Ape Cognition and Conservation Initiative (formerly Great Ape Trust), Des Moines, Iowa.
- See Savage-Rumbaugh & Lewin (1994), Savage-Rumbaugh et al. (1998) for accessible (though sometimes overinterpreted) introductions to research on Kanzi and other great apes at the Great Ape Trust.
The Kanzi corpus

- Savage-Rumbaugh et al. (1993): parallel corpora of 660 instructions directed to Kanzi (a bonobo) and Alia (a human infant), and description of their responses.
- Each has this format:

  287. (C) Kanzi, take the tomato to the colony room. (Kanzi makes a sound like “orange”; he then takes both the tomato and the orange to the colony room.) [C is scored because it is assumed that Kanzi is announcing that he wants to take an orange and have it to eat.]

- 287: item number
- (C): code (C, C1–C5: correct; others: incorrect in various ways)
- Kanzi, take the tomato . . . : utterance
- (Kanzi makes a sound like “orange” . . .): description of response
- [C is scored because . . .]: justification of code
What Kanzi gets right

- Savage-Rumbaugh et al.: Kanzi responds correctly 71.5% of the time (Alia: 66.6%).
- For 420/660 trials, a ‘semantic soup’ strategy would give a correct response.
  1. Take word meanings;
  2. Stir them together in a noncrazy way;
  3. Interpret.
- So most of the trials are informative about Kanzi’s vocabulary, but not grammar.
Reversible ditransitives

- Kanzi is also fine on 43 ‘reversible ditransitive’ pairs (76.7% correct, one example repeated).
  
  525. (C) *Put the tomato in the oil.* (Kanzi does so.)
  528. (C) *Put some oil in the tomato.* (Kanzi picks up the liquid Baby Magic oil and pours it in a bowl with the tomato.)

This requires sensitivity to linear order, a step beyond semantic soup.

- This may indicate ability to learn some ‘linear’ grammatical rules, but could also be interpreted as iconic.
Where we can’t tell

- From an adult-anglophone perspective, the instructions given to Kanzi are full of phrases.
- The noun phrases he correctly interprets are of the form
  \[D–(A)–N–(RC)\]
  where relative clauses are of the form
  \[that’s–[\text{PP}P–[\text{NP}D–N]]\]
- Unfortunately, we don’t know how he interprets any of this.
  - No evidence that he understands determiners or prepositions; slight evidence that he doesn’t.
  - No evidence that A or RC is taken to modify specifically the head noun.
    - 564. (C) Can you pour the ice water in the potty? Pour the ice water in the potty.
      Untested: Can you pour the ice water in the hot water?
    - 500. (C) Get the lighter that’s in the bedroom.
      Untested: Put the lighter that’s in the bedroom in the kitchen.
What Kanzi gets wrong

- A subset of these utterances (involving NP-coordination) require hierarchical phrase structure for correct interpretation.
- Kanzi fails to interpret those utterances correctly, while Alia has no problem.
- Because these sentences are no longer or otherwise more complex than the sentences that he responds appropriately to, we can take this as evidence that Kanzi does not represent hierarchical phrase structure.
Coordinate structures

- In simple cases, there’s a 1–1 mapping between nouns and NPs.
- Kanzi arguably interprets the noun rather than the full NP.
- That leads to trouble with coordinate NP objects.

  Fetch the ball and the rock.

  Which noun describes the patient of *fetch*? And what’s the other noun doing there?

  - Same problem arises in principle with *Get the lighter that’s in the bedroom*, only lighters are much easier to get than bedrooms.

- ‘Standard English’ solution: *the ball and the rock* is a complex constituent, part of a hierarchical representation of phrase structure.
- Kanzi hasn’t found that solution.
Coordinate structures

▶ We don’t need all this to get the bare bones of coordination:

```
VP
  / \   /  \\
 V   NP   \
   /     /
 fetch   \
 NP   CONJ   NP
   /    /
 D    N   D    N
 the   ball and the rock
```

▶ But we minimally need this:

```
fetch X
  / \   /
 X   X
 \   /
 ball rock
```

(plus a rule that fetching X involves fetching everything named in X).
Kanzi’s responses

- Ignore first noun: 9/18 trials.
  428. (PC) *Give the water and the doggie to Rose.* (Kanzi picks up the dog and hands it to Rose.)

- Ignore second noun: 5/18 trials.
  526. (PC) *Give the lighter and the shoe to Rose.* (Kanzi hands Rose the lighter, then points to some food in a bowl in the array that he would like to have to eat.)

- Respond correctly: 4/18 trials (22.2%; Alia 68.4%)
  281. (C) *Give me the milk and the lighter.* (Kanzi does so.)

- This represents a species-specific deficit (Fisher exact test, $p = 0.008$), and a construction-specific deficit (binomial test, $p = 1.1 \times 10^{-5}$).
Section 4

Coda: Acquisition studies
Gertner & Fisher’s gorping

- Gertner & Fisher (2012) show 21-month-olds parallel videos of coordinated actions (boy and girl act independently) and transitive actions (boy acts on girl), and play one of:
  1. The boy is gorping the girl!
  2. The boy and the girl are gorping!
  3. The girl and the boy are gorping!

*Transitive: The boy is gorping the girl!*
*Agent-first: The boy and the girl are gorping!*
*Patient-first: The girl and the boy are gorping!*
Gertner & Fisher’s gorping

- Subjects look more to the transitive action for both (1) and (2).
- Conclusion: 21-month-olds use linear order of nouns to determine who the agent is.
- They don’t automatically represent *the boy and the girl* as a structurally complex subject.
- But they can be encouraged to (*Oh yes, they are gorping*, Arunachalam, Escovar, Hansen & Waxman 2011), and they slowly learn to over coming months (e.g. Hirsh-Pasek & Golinkoff 1996).
- This is not dendrophile behaviour.
  - English is not fond of SOV order.
  - That should be a clear cue that another analysis is needed.
  - But children need months to find that other analysis.
Conclusion

▶ There are serious obstacles to interpreting results like Fitch & Hauser (2004) as evidence supporting the Dendrophilia hypothesis.
▶ But the evidence from the Kanzi corpus broadly supports that conclusion.
▶ Alia has no trouble with hierarchical phrase structure; Kanzi performs roughly at chance.
▶ We can see this because of evidence about interpretation, not just string recognition.
▶ We can infer aspects of interpretation from behaviour, and aspects of grammatical representation from interpretation.
▶ Nevertheless, acquisition studies suggest that human infants are not quick to represent coordinate NPs as hierarchically structured.
▶ In other words, we are not dendrophiles, but Kanzi is a dendrophobe.


