Sharpening the empirical claims of generative syntax through formalization

Tim Hunter

University of Minnesota, Twin Cities

NASSLLI, June 2014
Part 1: Grammars and cognitive hypotheses
  What is a grammar?
  What can grammars do?
  Concrete illustration of a target: Surprisal

Parts 2–4: Assembling the pieces
  Minimalist Grammars (MGs)
  MGs and MCFGs
  Probabilities on MGs

Part 5: Learning and wrap-up
  Something slightly different: Learning model
  Recap and open questions
Sharpening the empirical claims of generative syntax through formalization

Tim Hunter — NASSLLI, June 2014

Part 1

Grammars and Cognitive Hypotheses
Outline

1. What we want to do with grammars
2. How to get grammars to do it
3. Derivations and representations
4. Information-theoretic complexity metrics
Outline

1. What we want to do with grammars
2. How to get grammars to do it
3. Derivations and representations
4. Information-theoretic complexity metrics
What are grammars used for?

- “Mostly” for accounting for acceptability judgements
- But there are other ways a grammar can figure in claims about cognition
Claims made by grammars

What are grammars used for?

- “Mostly” for accounting for acceptability judgements
- But there are other ways a grammar can figure in claims about cognition

Often tempting to draw a distinction between “linguistic evidence” (where grammar lives) and “experimental evidence” (where cognition lives)

- One need not make this distinction
- We will proceed without it, i.e. it's all linguistic (and/or all experimental)
Claims made by grammars

There’s a “boring” sense in which every syntax paper makes a cognitive claim, i.e. a claim testable via acceptability facts.
There’s a “boring” sense in which every syntax paper makes a cognitive claim, i.e. a claim testable via acceptability facts.

- For one thing, this is not a cop-out, even if it might seem like it is!
  - Why does it seem like a cop-out?
  - Lingering externalism/Platonism?
  - Perhaps partly because it’s just relatively rare to see anything being tested by other measures
Claims made by grammars

There’s a “boring” sense in which every syntax paper makes a cognitive claim, i.e. a claim testable via acceptability facts.

- For one thing, this is not a cop-out, even if it might seem like it is!
  - Why does it seem like a cop-out?
  - Lingering externalism/Platonism?
  - Perhaps partly because it's just relatively rare to see anything being tested by other measures

- For another, we can incorporate grammars into claims that are testable by other measures.
  - This is the main point of the course!
  - The claims/predictions will depend on internal properties of grammars, not just what they say is good and what they say is bad
  - And we'll do it without seeing grammatical derivations as real-time operations
Claims made by grammars

If we accept — as I do — . . . that the rules of grammar enter into the processing mechanisms, then evidence concerning production, recognition, recall, and language use in general can be expected (in principle) to have bearing on the investigation of rules of grammar, on what is sometimes called “grammatical competence” or “knowledge of language”.

(Chomsky 1980: pp.200-201)

[S]ince a competence theory must be incorporated in a performance model, evidence about the actual organization of behavior may prove crucial to advancing the theory of underlying competence.

(Chomsky 1980: p.226)
If we accept — as I do — ... that the rules of grammar enter into the processing mechanisms, then evidence concerning production, recognition, recall, and language use in general can be expected (in principle) to have bearing on the investigation of rules of grammar, on what is sometimes called “grammatical competence” or “knowledge of language”.

(Chomsky 1980: pp.200-201)

Since a competence theory must be incorporated in a performance model, evidence about the actual organization of behavior may prove crucial to advancing the theory of underlying competence.

(Chomsky 1980: p.226)

Evidence about X can only advance Y if Y makes claims about X!
What we will do:

- Put together a chain of linking hypotheses that bring “experimental evidence” to bear on “grammar questions”
  - e.g. reading times, acquisition patterns
  - e.g. move as distinct operation from merge vs. unified with merge
- Illustrate with some toy examples

What we will not do:

- Engage with state-of-the-art findings in the sentence processing literature
- End up with claims that one particular set of derivational operations is empirically better than another
We’ll take pairs of equivalent grammars that differ only in the move/re-merge dimension.

- They will make different predictions about sentence comprehension difficulty.
- They will make different predictions about what a learner will conclude from a common input corpus.
We’ll take pairs of equivalent grammars that differ only in the move/re-merge dimension.

- They will make different predictions about sentence comprehension difficulty.
- They will make different predictions about what a learner will conclude from a common input corpus.

The issues become “distant but empirical questions”. That’s all we’re aiming for, for now.
Outline

1. What we want to do with grammars
2. How to get grammars to do it
3. Derivations and representations
4. Information-theoretic complexity metrics
Interpretation functions

\[
\{ \\
S \\
NP \quad VP \\
John \quad V \quad NP \\
loves \quad Mary \\
\} \\
S \\
NP \quad VP \\
John \quad V \quad NP \\
loves \quad everyone \\
, \\
, \\
, \\
\} \\
\]

Caveats:
- Maybe we're interested in the finite specification of the set
- Maybe there's no clear line between observable and not
- Maybe some evidence is based on relativities among interpretations
Interpretation functions

\[
\begin{align*}
S & \quad \text{NP} \quad \text{VP} \\
\text{John} & \quad \text{loves} \quad \text{Mary} \\
L(m)(j) & \quad \text{John loves Mary} \\
\forall x \ L(x)(j) & \quad \text{John loves everyone} \\
\exists y \forall x \ L(x)(y) & \quad \text{someone loves everyone}
\end{align*}
\]
Interpretation functions

Caveats:

- Maybe we’re interested in the finite specification of the set
- Maybe there’s no clear line between observable and not
- Maybe some evidence is based on relativities among interpretations
Telling grammars apart

So, what if we have two different grammars — systems that define different sets of objects — that we can’t tell apart via the sound and meaning interpretations?

(Perhaps because they’re provably equivalent, or perhaps because the evidence just happens to be unavailable.)
So, what if we have two different grammars — systems that define different sets of objects — that we can't tell apart via the sound and meaning interpretations?

(Perhaps because they’re provably equivalent, or perhaps because the evidence just happens to be unavailable.)

- Option 1: Conclude that the differences are irrelevant to us (or “they’re not actually different”).
- Option 2: Make the differences matter ...somehow ...
Morrill (1994) in favour of Option 1:

The construal of a language as a collection of signs [sound-meaning pairs] presents as an investigative task the characterisation of this collection. This is usually taken to mean the specification of a set of “structural descriptions” (or: “syntactic structures”). Observe however that on our understanding a sign is an association of prosodic [phonological] and semantic properties. It is these properties that can be observed and that are to be modelled. There appears to be no observation which bears directly on syntactic as opposed to prosodic and/or semantic properties, and this implies an asymmetry in the status of these levels. A structural description is only significant insofar as it is understood as predicting prosodic and semantic properties (e.g. in interpreting the yield of a tree as word order). Attribution of syntactic (or prosodic or semantic) structure does not of itself predict anything.
What are syntactic representations for?

Morrill (1994) in favour of Option 1:

*The construal of a language as a collection of signs [sound-meaning pairs] presents as an investigative task the characterisation of this collection. This is usually taken to mean the specification of a set of “structural descriptions” (or: “syntactic structures”). Observe however that on our understanding a sign is an association of prosodic [phonological] and semantic properties. It is these properties that can be observed and that are to be modelled. There appears to be no observation which bears directly on syntactic as opposed to prosodic and/or semantic properties, and this implies an asymmetry in the status of these levels. A structural description is only significant insofar as it is understood as predicting prosodic and semantic properties (e.g. in interpreting the yield of a tree as word order). Attribution of syntactic (or prosodic or semantic) structure does not of itself predict anything.*

Where might we depart from this (to pursue Option 2)?

- Object that syntactic structure **does** matter “of itself”
- Object that prosodic and semantic properties are **not** the only ones we can observe
Interpretation functions

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow John \\
VP & \rightarrow V \ NP \\
V & \rightarrow \text{lives} \\
NP & \rightarrow Mary \\
\end{align*}
\]

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow John \\
VP & \rightarrow V \ NP \\
V & \rightarrow \text{lives} \\
NP & \rightarrow \text{everyone} \\
\end{align*}
\]

\[
\begin{align*}
S & \rightarrow \ldots \\
\ldots & \rightarrow \ldots \\
\ldots & \rightarrow \ldots \\
\ldots & \rightarrow \ldots \\
\end{align*}
\]

\[L(m)(j) \rightarrow John \text{ loves Mary}\]

\[\forall x L(x)(j) \rightarrow John \text{ loves everyone}\]

\[\exists y \forall x L(x)(y) \rightarrow \text{someone loves everyone}\]

Caveats:

- Maybe we’re interested in the finite specification of the set
- Maybe there’s no clear line between observable and not
- Maybe some evidence is based on relativities among interpretations
What we want to do with grammars
How to get grammars to do it
Derivations and representations
Information-theoretic complexity metrics

Interpretation functions

John loves Mary

John loves everyone

someone loves everyone

Caveats:
- Maybe we’re interested in the finite specification of the set
- Maybe there’s no clear line between observable and not
- Maybe some evidence is based on relativities among interpretations
Interpretation functions for “complexity”

What are some other interpretation functions?

- number of nodes
Interpretation functions for “complexity”

What are some other interpretation functions?

- number of nodes
- ratio of total nodes to terminal nodes (Miller and Chomsky 1963)
Ratio of total nodes to terminal nodes

Fig. 8. Illustrating a measure of structural complexity. $N(Q)$ for the $P$-marker (a) is $7/4$; for (b), $N(Q) = 5/4$.  

(Miller and Chomsky 1963)
Ratio of total nodes to terminal nodes

![Diagram of tree structures]

Fig. 8. Illustrating a measure of structural complexity. $N(Q)$ for the P-marker (a) is 7/4; for (b), $N(Q) = 5/4$.

Won’t distinguish center-embedding from left- and right-embedding

1. The mouse [the cat [the dog bit] chased] died. (center)
2. The dog bit the cat [which chased the mouse [which died]]. (right)
3. [[the dog] ’s owner] ’s friend (left)

(Miller and Chomsky 1963)
Interpretation functions for “complexity”

What are some other interpretation functions?

- number of nodes
- ratio of total nodes to terminal nodes (Miller and Chomsky 1963)
- degree of self-embedding (Miller and Chomsky 1963)
Degree of (centre-)self-embedding

A tree’s degree of self-embedding is \( m \) iff:
“there is . . . a continuous path passing through \( m + 1 \) nodes \( N_0, \ldots, N_m \), each with the same label, where each \( N_i \) (\( i \geq 1 \)) is fully self-embedded (with something to the left and something to the right) in the subtree dominated by \( N_{i-1} \)”

(Miller and Chomsky 1963)
A tree’s degree of self-embedding is $m$ iff:

“there is ... a continuous path passing through $m + 1$ nodes $N_0, \ldots, N_m$, each with the same label, where each $N_i$ ($i \geq 1$) is fully self-embedded (with something to the left and something to the right) in the subtree dominated by $N_{i-1}$”

(Miller and Chomsky 1963)
Interpretation functions for “complexity”

What are some other interpretation functions?

- number of nodes
- ratio of total nodes to terminal nodes (Miller and Chomsky 1963)
- degree of self-embedding (Miller and Chomsky 1963)
- “depth” of memory required by a top-down parser (Yngve 1960)
Yngve’s depth

Number of constituents expected but not yet started:

```
S
  |   
NP  VP
  |   
N  S  V
died
  |   
NP  VP
  |   
N  S  V
chased
  |   
NP  VP
  |   
N  V
bit
  |   
NP
  |   
Det the
  |   
Det the
  |   
Det the
  |   
Det the
  |   
Det the
  |   
Det the
  |   
N  N
mouse  cat  dog
  |   
```

Unlike (center-)self-embedding, right-embedding doesn’t create such large lists of expected constituents (because the expected stuff is all part of one constituent). But left-embedding does.

Yngve’s theory was set within — perhaps justified by — a procedural story, but we can arguably detach it from that and treat depth as just another property of trees. (Yngve 1960)
Yngve’s depth

Number of constituents expected but not yet started:

```
S
  /\    
 NP  VP
   /\    
 Det  N mouse
   /\    
 Det  N cat
   /\    
 Det  N dog
```

Unlike (center-)self-embedding, right-embedding doesn’t create such large lists of expected constituents (because the expected stuff is all part of one constituent). But left-embedding does.

Yngve’s theory was set within — perhaps justified by — a procedural story, but we can arguably detach it from that and treat depth as just another property of trees. (Yngve 1960)
Yngve's depth

Number of constituents expected but not yet started:

Unlike (center-)self-embedding, right-embedding doesn’t create such large lists of expected constituents (because the expected stuff is all part of one constituent).

But left-embedding does.
Yngve’s depth

Number of constituents expected but not yet started:

Unlike (center-)self-embedding, right-embedding doesn’t create such large lists of expected constituents (because the expected stuff is all part of one constituent).

But left-embedding does.

Yngve’s theory was set within — perhaps justified by — a procedural story, but we can arguably detach it from that and treat depth as just another property of trees.

(Yngve 1960)
What are some other interpretation functions?

- number of nodes
- ratio of total nodes to terminal nodes \(^{(\text{Miller and Chomsky 1963})}\)
- degree of self-embedding \(^{(\text{Miller and Chomsky 1963})}\)
- “depth” of memory required by a top-down parser \(^{(\text{Yngve 1960})}\)
- minimal attachment, late closure, etc.?
Reaching conclusions about grammars

complexity metric + grammar → prediction

Typically, arguments hold the grammar fixed and present evidence in favour of a metric.
Reaching conclusions about grammars

complexity metric + grammar → prediction

Typically, arguments hold the grammar fixed and present evidence in favour of a metric.

We can flip this around: hold the metric fixed and present evidence in favour of a grammar.

If we accept — as I do — . . . that the rules of grammar enter into the processing mechanisms, then evidence concerning production, recognition, recall, and language use in general can be expected (in principle) to have bearing on the investigation of rules of grammar, on what is sometimes called “grammatical competence” or “knowledge of language”.

(Chomsky 1980: pp.200-201)
Reaching conclusions about grammars

complexity metric $+$ grammar $\rightarrow$ prediction

Example: hold self-embedding fixed as the complexity metric.

(4) That [the food that [John ordered] tasted good] pleased him.

(5) That [that [the food was good] pleased John] surprised Mary.

Grammar question: Does a relative clause have a node labeled S?
Reaching conclusions about grammars

complexity metric  +  grammar  ⟷  prediction

Example: hold self-embedding fixed as the complexity metric.

(4) That [the food that [John ordered] tasted good] pleased him.
(5) That [that [the food was good] pleased John] surprised Mary.

Grammar question: Does a relative clause have a node labeled S?

<table>
<thead>
<tr>
<th>Proposed answer</th>
<th>(4) structure</th>
<th>(5) structure</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>\ldots [s \ldots [s \ldots ] ]</td>
<td>\ldots [s \ldots [s \ldots ] ]</td>
<td>(4) &amp; (5) same</td>
</tr>
<tr>
<td>No</td>
<td>\ldots [s \ldots [RC \ldots ] ]</td>
<td>\ldots [s \ldots [s \ldots ] ]</td>
<td>(5) harder</td>
</tr>
</tbody>
</table>
Reaching conclusions about grammars

complexity metric  +  grammar  →  prediction

Example: hold self-embedding fixed as the complexity metric.

(4) That [the food that [John ordered] tasted good] pleased him.

(5) That [that [the food was good] pleased John] surprised Mary.

Grammar question: Does a relative clause have a node labeled S?

<table>
<thead>
<tr>
<th>Proposed answer</th>
<th>(4) structure</th>
<th>(5) structure</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>…[S …[S …]]</td>
<td>…[S …[S …]]</td>
<td>(4) &amp; (5) same</td>
</tr>
<tr>
<td>No</td>
<td>…[S …[RC …]]</td>
<td>…[S …[S …]]</td>
<td>(5) harder</td>
</tr>
</tbody>
</table>

Conclusion: The fact that (5) is harder supports the “No” answer.
Outline

1. What we want to do with grammars
2. How to get grammars to do it
3. Derivations and representations
4. Information-theoretic complexity metrics
Derivations and representations

Question

But these metrics are all properties of a final, fully-constructed tree. How can anything like this be sensitive to differences in the derivational operations that build these trees? (e.g. TAG vs. MG, whether move is re-merge)
Interpretation functions for “complexity”

What are some other interpretation functions?

- number of nodes
- ratio of total nodes to terminal nodes (Miller and Chomsky 1963)
- degree of self-embedding (Miller and Chomsky 1963)
- “depth” of memory required by a top-down parser (Yngve 1960)
- minimal attachment, late closure, etc.?
- “nature, number and complexity of” transformations (Miller and Chomsky 1963)
The psychological plausibility of a transformational model of the language user would be strengthened, of course, if it could be shown that our performance on tasks requiring an appreciation of the structure of transformed sentences is some function of the nature, number and complexity of the grammatical transformations involved.

(Miller and Chomsky 1963: p.481)
But these metrics are all properties of a final, **fully-constructed tree**. How can anything like this be sensitive to differences in the derivational operations that build these trees? (e.g. TAG vs. MG, whether move is re-merge)
Derivations and representations

Question
But these metrics are all properties of a final, fully-constructed tree. How can anything like this be sensitive to differences in the derivational operations that build these trees? (e.g. TAG vs. MG, whether move is re-merge)

Answer
The relevant objects on which the interpretation functions are defined encode a complete derivational history.
Derivations and representations

**Question**

But these metrics are all properties of a final, **fully-constructed tree**. How can anything like this be sensitive to differences in the derivational operations that build these trees? (e.g. TAG vs. MG, whether move is re-merge)

**Answer**

The relevant objects on which the interpretation functions are defined encode a complete **derivational history**.

e.g. The function which, given a complete “recipe” for carrying out a derivation, returns the number of movement steps called for by the recipe.

(Maybe only useful when we’re holding a grammar fixed)
Full derivation recipes?

Are the inputs to these functions **really** full derivation recipes?

For minimalist syntax it’s hard to tell, because the final derived object very often uniquely identifies a derivational history/recipe.
Full derivation recipes?

Are the inputs to these functions **really** full derivation recipes?

For minimalist syntax it’s hard to tell, because the final derived object very often uniquely identifies a derivational history/recipe.

- merge Y with RP
- merge the result with ZP
- merge the result with WP
- merge X with the result
- move ZP
Full derivation recipes?

A few cases reveal that (we must all be already assuming that) it’s full derivations/recipes that count.
Full derivation recipes?

A few cases reveal that (we must all be already assuming that) it’s full derivations/recipes that count.

(6) *Which claim [that Mary\textsubscript{i} was a thief] did she\textsubscript{i} deny?

(7) Which claim [that Mary\textsubscript{i} made] did she\textsubscript{i} deny?
Full derivation recipes?

A few cases reveal that (we must all be already assuming that) it’s full derivations/recipes that count.

(6) *Which claim [that Mary$_i$ was a thief] did she$_i$ deny?

(7) Which claim [that Mary$_i$ made] did she$_i$ deny?
Full derivation recipes?

A few cases reveal that (we must all be already assuming that) it’s full derivations/recipes that count.

(6) *Which claim [that Mary\textsubscript{i} was a thief] did she\textsubscript{i} deny?

(7) Which claim [that Mary\textsubscript{i} made] did she\textsubscript{i} deny?
Full derivation recipes?

Also:

- subjacency effects without traces
- compare categorial grammar
Full derivation recipes?

Also:

- subjacency effects without traces
- compare categorial grammar

And this is not a new idea!

[The perceptual model] will utilize the full resources of the transformational grammar to provide a structural description, consisting of a set of P-markers and a transformational history

Miller and Chomsky (1963: p.480)
Full derivation recipes?

[The perceptual model] will utilize the full resources of the transformational grammar to provide a structural description, consisting of a set of P-markers and a transformational history

_Miller and Chomsky (1963: p.480)_

```
S
  NP  VP
    John is eager to please
```

```
S
  NP  VP
     it is Adj COMP
```

```
S
  NP  VP
    someone V NP
```

\[ T_5: \text{front embedded object, replacing ‘it’} \]

\[ T_4: \text{delete ‘for someone’} \]

\[ T_1: \text{replace COMP} \]
Full derivation recipes?

[The perceptual model] will utilize the full resources of the transformational grammar to provide a structural description, consisting of a set of P-markers and a transformational history.

*Miller and Chomsky (1963: p.480)*
(4) the man who persuaded John to be examined by a specialist was fired

The “transformational history” of (4) by which it is derived from its basis might be represented, informally, by the diagram (5).

(5) (1) \( T_E - T_R - T_P - T_{AD} \)

(2) \( T_E - T_D - T_{io} \)

(3) \( -T_P \)
(4) the man who persuaded John to be examined by a specialist was fired

The “transformational history” of (4) by which it is derived from its basis might be represented, informally, by the diagram (5).

(5) (1) $T_E - T_R - T_P - T_{AD}$

(2) $T_E - T_D - T_{io}$

(3) $- T_P$

Differences these days:

- We’ll have things like merge and move at the internal nodes instead of $T_P, T_E$, etc.
- We’ll have lexical items at the leaves rather than base-derived trees.
Outline

1. What we want to do with grammars
2. How to get grammars to do it
3. Derivations and representations
4. Information-theoretic complexity metrics
Surprisal and entropy reduction

Why these complexity metrics?

- Partly just for concreteness, to give us a goal.
- They are **formalism neutral** to a degree that others aren’t.
- They are **mechanism neutral** (Marr level one).
- The pieces of the puzzle that we need to get there (e.g. probabilities) seem likely to be usable in other ways.
Surprisal and entropy reduction

Why these complexity metrics?

- Partly just for concreteness, to give us a goal.
- They are formalism neutral to a degree that others aren’t.
- They are mechanism neutral (Marr level one).
- The pieces of the puzzle that we need to get there (e.g. probabilities) seem likely to be usable in other ways.
Interpretation functions

\[
\begin{align*}
S & \quad NP \quad VP \\
& \quad John \quad V \quad NP \\
& \quad loves \quad Mary \\
& \quad SEM \quad PHON \\
& \quad L(m)(j) \quad 5 \\
& \quad John loves Mary
\end{align*}
\]

\[
\begin{align*}
S & \quad NP \quad VP \\
& \quad John \quad V \quad NP \\
& \quad loves \quad everyone \\
& \quad SEM \quad PHON \\
& \quad \forall x \ L(x)(j) \quad 5 \\
& \quad John loves everyone
\end{align*}
\]

\[
\begin{align*}
S & \quad NP \quad VP \\
& \quad everyone \quad V \\
& \quad loves \quad NP \\
& \quad SEM \quad PHON \\
& \quad \exists y \forall x \ L(x)(y) \quad 7 \\
& \quad someone loves everyone
\end{align*}
\]

Caveats:

- Maybe we’re interested in the finite specification of the set
- Maybe there’s no clear line between observable and not
- Maybe some evidence is based on relativities among interpretations
Interpretation functions

John loves Mary

John loves everyone

Someone loves everyone

Caveats:

- Maybe we’re interested in the finite specification of the set
- Maybe there’s no clear line between observable and not
- Maybe some evidence is based on relativities among interpretations
Surprisal

Given a sentence $w_1 w_2 \ldots w_n$:

$$\text{surprisal at } w_i = - \log P(W_i = w_i \mid W_1 = w_1, W_2 = w_2, \ldots, W_{i-1} = w_{i-1})$$
Surprisal

<table>
<thead>
<tr>
<th>Probability</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>John ran</td>
</tr>
<tr>
<td>0.15</td>
<td>John saw it</td>
</tr>
<tr>
<td>0.05</td>
<td>John saw them</td>
</tr>
<tr>
<td>0.25</td>
<td>Mary ran</td>
</tr>
<tr>
<td>0.1</td>
<td>Mary saw it</td>
</tr>
<tr>
<td>0.05</td>
<td>Mary saw them</td>
</tr>
</tbody>
</table>

What predictions can we make about the difficulty of comprehending ‘John saw it’?
What predictions can we make about the difficulty of comprehending ‘John saw it’?

\[
\text{surprisal at ‘John’} = -\log P(W_1 = \text{John}) \\
= -\log(0.4 + 0.15 + 0.05) \\
= -\log 0.6 \\
= 0.74
\]
Surprisal

What predictions can we make about the difficulty of comprehending ‘John saw it’?

surprisal at ‘John’ = \(- \log P(W_1 = \text{John})\)
\[= - \log(0.4 + 0.15 + 0.05)\]
\[= - \log 0.6\]
\[= 0.74\]

surprisal at ‘saw’ = \(- \log P(W_2 = \text{saw} \mid W_1 = \text{John})\)
\[= - \log \frac{0.15 + 0.05}{0.4 + 0.15 + 0.05}\]
\[= - \log 0.33\]
\[= 1.58\]
What predictions can we make about the difficulty of comprehending ‘John saw it’?

surprisal at ‘John’ = − log P(W₁ = John)
= − log(0.4 + 0.15 + 0.05)
= − log 0.6
= 0.74

surprisal at ‘saw’ = − log P(W₂ = saw | W₁ = John)
= − log \frac{0.15 + 0.05}{0.4 + 0.15 + 0.05}
= − log 0.33
= 1.58

surprisal at ‘it’ = − log P(W₃ = it | W₁ = John, W₂ = saw)
= − log \frac{0.15}{0.15 + 0.05}
= − log 0.75
= 0.42
Accurate predictions made by surprisal

(Hale 2001)
Accurate predictions made by surprisal

(8) The reporter [who ____ attacked the senator] left the room.  
(9) The reporter [who the senator attacked ____] left the room.

(Levy 2008)
An important distinction

Using surprisal as a complexity metric says nothing about the form of the knowledge that the language comprehender is using!

- We’re asking “what’s the probability of $w_i$, given that we’ve seen $w_1 \ldots w_{i-1}$ in the past”.
- This does not mean that the comprehender’s knowledge takes the form of answers to this kind of question.
- The linear nature of the metric reflects the task, not the knowledge being probed.
Probabilistic CFGs

1.0  $S \rightarrow NP \ VP$
0.3  $NP \rightarrow John$
0.7  $NP \rightarrow Mary$
0.2  $VP \rightarrow ran$
0.5  $VP \rightarrow V \ NP$
0.3  $VP \rightarrow V \ S$
0.4  $V \rightarrow believed$
0.6  $V \rightarrow knew$

$$P(\text{Mary believed John ran}) = 1.0 \times 0.7 \times 0.3 \times 1.0 \times 0.3 \times 0.2 = 0.00504$$
Probabilistic CFGs

1.0  \( S \rightarrow \text{NP VP} \)
0.3  \( \text{NP} \rightarrow \text{John} \)
0.7  \( \text{NP} \rightarrow \text{Mary} \)
0.2  \( \text{VP} \rightarrow \text{ran} \)
0.5  \( \text{VP} \rightarrow \text{V NP} \)
0.3  \( \text{VP} \rightarrow \text{V S} \)
0.4  \( \text{V} \rightarrow \text{believed} \)
0.6  \( \text{V} \rightarrow \text{knew} \)

\[
P(\text{Mary believed John ran}) = 1.0 \times 0.7 \times 0.3 \times 0.4 \times 1.0 \times 0.3 \times 0.2
= 0.00504
\]
Surprisal with probabilistic CFGs

**Goal:** Calculate step-by-step surprisal values for ‘Mary believed John ran’

surprisal at ‘John’ = $-\log P(W_3 = \text{John} \mid W_1 = \text{Mary}, W_2 = \text{believed})$
**Goal:** Calculate step-by-step surprisal values for ‘Mary believed John ran’

\[
surprisal \text{ at ‘John’} = - \log P(W_3 = \text{John} \mid W_1 = \text{Mary}, W_2 = \text{believed})
\]

<table>
<thead>
<tr>
<th>surprisal</th>
<th>phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.098</td>
<td>Mary believed Mary</td>
</tr>
<tr>
<td>0.042</td>
<td>Mary believed John</td>
</tr>
<tr>
<td>0.012348</td>
<td>Mary believed Mary knew Mary</td>
</tr>
<tr>
<td>0.01176</td>
<td>Mary believed Mary ran</td>
</tr>
<tr>
<td>0.008232</td>
<td>Mary believed Mary believed Mary</td>
</tr>
<tr>
<td>0.005292</td>
<td>Mary believed Mary knew John</td>
</tr>
<tr>
<td>0.005292</td>
<td>Mary believed John knew Mary</td>
</tr>
<tr>
<td>0.00504</td>
<td>Mary believed John ran</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

There are an infinite number of derivations consistent with input at each point!
Surprisal with probabilistic CFGs

**Goal:** Calculate step-by-step surprisal values for ‘Mary believed John ran’

surprisal at ‘John’ = \(-\log P(W_3 = \text{John} \mid W_1 = \text{Mary}, W_2 = \text{believed})\)

<table>
<thead>
<tr>
<th>surprisal</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.098</td>
<td>Mary believed Mary</td>
</tr>
<tr>
<td>0.042</td>
<td>Mary believed John</td>
</tr>
<tr>
<td>0.012348</td>
<td>Mary believed Mary knew Mary</td>
</tr>
<tr>
<td>0.01176</td>
<td>Mary believed Mary ran</td>
</tr>
<tr>
<td>0.008232</td>
<td>Mary believed Mary believed Mary</td>
</tr>
<tr>
<td>0.005292</td>
<td>Mary believed Mary knew John</td>
</tr>
<tr>
<td>0.005292</td>
<td>Mary believed John knew Mary</td>
</tr>
<tr>
<td>0.00504</td>
<td>Mary believed John ran</td>
</tr>
</tbody>
</table>

... ...

There are an **infinite number of derivations** consistent with input at each point!

surprisal at ‘John’ = \(-\log P(W_3 = \text{John} \mid W_1 = \text{Mary}, W_2 = \text{believed})\)

\[
= -\log \frac{0.042 + 0.005292 + 0.00504 + \ldots}{0.098 + 0.042 + 0.12348 + 0.01176 + 0.008232 + \ldots}
\]
Intersection grammars

\[
\begin{align*}
1.0 & \quad S \rightarrow NP \ VP \\
0.3 & \quad NP \rightarrow John \\
0.7 & \quad NP \rightarrow Mary \\
0.2 & \quad VP \rightarrow ran \\
0.5 & \quad VP \rightarrow V \ NP \\
0.3 & \quad VP \rightarrow V \ S \\
0.4 & \quad V \rightarrow believed \\
0.6 & \quad V \rightarrow knew \\
\end{align*}
\]

\[
G_2
\]

\[
\begin{tikzpicture}[node distance=2cm,>=latex]
  \node (0) [state] {0};
  \node (1) [state, right of=0] {1};
  \node (2) [state, above of=1] {2};
  \path[->]
  (0) edge [loop above, out=90, in=90] node {Mary} (0)
  (0) edge node {believed} (1)
  (1) edge node {\*} (2)
  (2) edge [loop below] node {\*} (2);
\end{tikzpicture}
\]
Intersection grammars

\[
G_2 = \{ \text{Mary believed} \}
\]

\[
G_3 = \{ \text{Mary believed John} \}
\]

1.0 \( S \rightarrow \text{NP VP} 
0.3 \( \text{NP} \rightarrow \text{John} 
0.7 \( \text{NP} \rightarrow \text{Mary} 
0.2 \( \text{VP} \rightarrow \text{ran} 
0.5 \( \text{VP} \rightarrow \text{V NP} 
0.3 \( \text{VP} \rightarrow \text{V S} 
0.4 \( \text{V} \rightarrow \text{believed} 
0.6 \( \text{V} \rightarrow \text{knew} 

\]

\[
G_2 = \{ \text{Mary believed} \}
\]

\[
G_3 = \{ \text{Mary believed John} \}
\]

\[
G_2 \cap G_3 = \{ \text{Mary believed John} \}
\]

\[
G_2 \cap G_3 = \{ \text{Mary believed John} \}
\]
Intersection grammars

1.0 \( S \rightarrow NP \ VP \)
0.3 \( NP \rightarrow John \)
0.7 \( NP \rightarrow Mary \)
0.2 \( VP \rightarrow ran \)
0.5 \( VP \rightarrow V \ NP \)
0.3 \( VP \rightarrow V \ S \)
0.4 \( V \rightarrow believed \)
0.6 \( V \rightarrow knew \)

\[
0 \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \frac{0.0672}{0.224} = 0.3045
\]

surprisal at ‘John’ = \(- \log P(W_3 = John \mid W_1 = Mary, W_2 = believed)\)

= \(- \log \frac{\text{total weight in } G_3}{\text{total weight in } G_2}\)

= \(- \log \frac{0.0672}{0.224}\)

= 1.74
Grammar intersection example (simple)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>1.0</td>
</tr>
<tr>
<td>NP → John</td>
<td>0.3</td>
</tr>
<tr>
<td>NP → Mary</td>
<td>0.7</td>
</tr>
<tr>
<td>VP → ran</td>
<td>0.2</td>
</tr>
<tr>
<td>VP → V NP</td>
<td>0.5</td>
</tr>
<tr>
<td>VP → V S</td>
<td>0.3</td>
</tr>
<tr>
<td>V → believed</td>
<td>0.4</td>
</tr>
<tr>
<td>V → knew</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Diagram:

```
0 ----> Mary ----> 1 ----> believed ----> 2
```

NB: Total weight in this grammar is not one! (What is it? Start symbol is $S_{0,2}$.)

Each derivation has the weight "it" had in the original grammar.
Grammar intersection example (simple)

NB: Total weight in this grammar is not one! (What is it? Start symbol is $S_{0,2}$.) Each derivation has the weight “it” had in the original grammar.
Grammar intersection example (more complicated)

\[
\begin{align*}
S & \rightarrow \text{NP} \ \text{VP} \\
\text{VP} & \rightarrow \text{V} \ \text{NP} \\
\text{NP} & \rightarrow \text{DET} \\
\text{NP} & \rightarrow \text{DET} \ \text{N} \\
\text{NP} & \rightarrow \text{ADJ} \ \text{N}
\end{align*}
\]

\[
\begin{align*}
\text{V} & \rightarrow \text{fish} \\
\text{V} & \rightarrow \text{damaged} \\
\text{DET} & \rightarrow \text{these} \\
\text{N} & \rightarrow \text{fish} \\
\text{ADJ} & \rightarrow \text{damaged}
\end{align*}
\]

These fish damaged …
Grammar intersection example (more complicated)

- **S → NP VP**
- **V → fish**
- **VP → V NP**
- **V → damaged**
- **NP → DET**
- **DET → these**
- **NP → DET N**
- **N → fish**
- **NP → ADJ N**
- **ADJ → damaged**

**Derivation Trees:**

- **S → NP VP VP**
- **NP → DET DET → these**
- **NP → DET N → fish**
- **NP → ADJ N → damaged**

**Diagram:**

- **S**
  - **NP**
    - **VP**
      - **DET**
        - **these**
      - **fish**
  - **VP**
    - **V**
      - **damaged**

**Information-theoretic complexity metrics**

**NP → DET N → fish**

**NP → ADJ N → damaged**

**NP → DET N → fish**

**NP → ADJ N → damaged**
Intersection grammars

1.0 $S \rightarrow NP \ VP$
0.3 $NP \rightarrow John$
0.7 $NP \rightarrow Mary$
0.2 $VP \rightarrow ran$
0.5 $VP \rightarrow V \ NP$
0.3 $VP \rightarrow V \ S$
0.4 $V \rightarrow believed$
0.6 $V \rightarrow knew$

$\cap$

surprise at ‘John’ $= - \log P(W_3 = John \mid W_1 = Mary, W_2 = believed)$

$= - \log \frac{\text{total weight in } G_3}{\text{total weight in } G_2}$

$= - \log \frac{0.0672}{0.224}$

$= 1.74$
Computing sum of weights in a grammar (“partition function”)

\[ Z(A) = \sum_{A \rightarrow \alpha} \left( p(A \rightarrow \alpha) \cdot Z(\alpha) \right) \]

- \( Z(\epsilon) = 1 \)
- \( Z(a\beta) = Z(\beta) \)
- \( Z(B\beta) = Z(B) \cdot Z(\beta) \) where \( \beta \neq \epsilon \)

1.0 \( S \rightarrow NP \ VP \)
0.3 \( NP \rightarrow John \)
0.7 \( NP \rightarrow Mary \)
0.2 \( VP \rightarrow ran \)
0.5 \( VP \rightarrow V NP \)
0.4 \( V \rightarrow believed \)
0.6 \( V \rightarrow knew \)

\[ Z(V) = 0.4 + 0.6 = 1.0 \]
\[ Z(NP) = 0.3 + 0.7 = 1.0 \]
\[ Z(VP) = 0.2 + (0.5 \cdot Z(V) \cdot Z(NP)) \]
\[ = 0.2 + (0.5 \cdot 1.0 \cdot 1.0) = 0.7 \]
\[ Z(S) = 1.0 \cdot Z(NP) \cdot Z(VP) \]
\[ = 0.7 \]

(Nederhof and Satta 2008)
Computing sum of weights in a grammar (“partition function”)

\[
Z(A) = \sum_{A \rightarrow \alpha} \left( p(A \rightarrow \alpha) \cdot Z(\alpha) \right)
\]

\[
Z(\epsilon) = 1
\]

\[
Z(a\beta) = Z(\beta)
\]

\[
Z(B\beta) = Z(B) \cdot Z(\beta) \quad \text{where } \beta \neq \epsilon
\]

(Nederhof and Satta 2008)

<table>
<thead>
<tr>
<th>Production</th>
<th>Weight</th>
<th>Calculation</th>
</tr>
</thead>
</table>
| S → NP VP        | 1.0    | \[
Z(S) = 1.0 \cdot Z(NP) \cdot Z(VP)
\]  |
| NP → John        | 0.3    | \[
Z(NP) = 0.3 + 0.7 = 1.0
\]  |
| NP → Mary        | 0.7    | \[
Z(NP) = 0.3 + 0.7 = 1.0
\]  |
| VP → ran         | 0.2    | \[
Z(VP) = 0.2 + (0.5 \cdot Z(V) \cdot Z(NP))
\]  |
| VP → V NP        | 0.5    | \[
Z(VP) = 0.2 + (0.5 \cdot Z(V) \cdot Z(NP)) + (0.3 \cdot Z(V) \cdot Z(S))
\]  |
| V → believed     | 0.4    | \[
Z(V) = 0.4 + 0.6 = 1.0
\]  |
| V → knew         | 0.6    | \[
Z(V) = 0.4 + 0.6 = 1.0
\]  |
Things to know

Technical facts about CFGs:

- Can intersect with a “prefix FSA”
- Can compute the total weight (and the entropy)
Things to know

Technical facts about CFGs:

- Can intersect with a “prefix FSA”
- Can compute the total weight (and the entropy)

More generally:

- Intersecting a grammar with a prefix produces a new grammar which is a representation of the comprehender’s sentence-medial state
- So we can construct a sequence of grammars which represents the comprehender’s sequence of knowledge-states
- Ask “what changes” (or “how much changes”, etc.) at each step

The general approach is compatible with many very different grammar formalisms (any grammar formalism?) — provided the technical tricks can be pulled off.

(Hale 2006)
Looking ahead

Wouldn’t it be nice if we could do all that for minimalist syntax?

The average syntax paper shows *illustrative derivations*, not a *fragment*.

What would we need?

- An explicit characterization of the set of possible derivations
- A way to “intersect” that with a prefix
- A way to define probability distributions over the possibilities

This will require certain idealizations. (But what’s new?)


