Basic Syntax

Micha Elsner

February 20, 2014
Basics

- Syntax describes language structure above the word level
- The syntax rules for a language are called its grammar
  - Sometimes people use this term for morphological rules etc. as well
- Syntactic theories explain two kinds of data:
  - What you “can say” in a language
  - And how structure relates to meaning

Grammaticality: what you can say

- *Toothbrush the is blue
- The * means ungrammatical
- Sentences with no clear meaning can still be grammatical...
- Chomsky: “Colorless green ideas sleep furiously”

(btw, parts of this lecture follow Carnie: “Syntax, a generative introduction”)

Phrases

English lets you replace long word strings with a pronoun...

- If “cats sit” is grammatical, “they sit” is too
- “the cat sat” ... “it sat”
- “my fuzzy orange cat sat” ... “it sat”
- “my mother’s fuzzy orange cat, which is always wandering around the neighbor’s yard, sat” ... “it sat”

This works for objects, too:

- “The T-Rex ate a linguist” ... “The T-Rex ate her”
- “The T-Rex ate a famous linguist” ... “The T-Rex ate her”
- “The T-Rex ate acclaimed MIT Professor Noam Chomsky, author of Syntactic Structures” ... “The T-Rex ate him”
Patterns

Compare the things you can replace with a pronoun… with the things you can elicit with questions:

▶ “What sits?” … “The cat”
▶ “Who ate a linguist?” … “The T-Rex”
▶ “Who did the T-Rex eat?” … “A famous linguist”
A particular example
The T-Rex ate a famous linguist in Harvard Yard

- List all the substrings we can extract with questions
- Or pronouns
A particular example

The T-Rex ate a famous linguist in Harvard Yard

- List all the substrings we can extract with questions
- Or pronouns

- “Who ate...” “The T-Rex”
- “Where did the T-Rex eat...” “In Harvard Yard” (pro. “there”)
- “What did the T-Rex do?” “Ate a....” (pro. “did”)
  - As in, “The raptor ate a linguist and the T-Rex did too”
- “What did the T-Rex do in Hvd. Yard?” “Ate a famous linguist”
- “What did the T-Rex eat?” ... “A famous linguist”
- “The T-Rex ate” is marginal, since it leaves out content...
  - Doesn’t work with verbs like “killed”
- “*What did the T-Rex linguist?” “Ate a famous”
- “*Who did the T-Rex eat Harvard Yard?” “A linguist in”
Constituents

These “replaceable” components are called constituents

- A critical discovery is that they form a nested structure
- The theoretical justification for syntax trees

The T-Rex ate a famous linguist in Harvard Yard

A syntax tree organizes these constituents into a hierarchy:
Labeling the tree

- Many of these constituents have one-part-of-speech versions
  - “The T-Rex” can be replaced with “T-Rexes”
    - By replaced, I mean the new version would be grammatical
  - “ate a famous linguist...” can be replaced with “roared” or “slept”
  - “in Harvard Yard” ... “outside”, “downtown”, etc

We’ll call the multiword constituents “phrases” and name them after the POS tag we would assign to the single-word version

- “T-Rexes” is a proper noun: “The T-Rex” is a noun phrase
  - Usually phrases are abbreviated: NP
- “outside” is a preposition: PP
- “slept” is a verb: VP
- By convention, the whole sentence is S (sentence)
  - Theoretical arguments over whether it can be considered an XP of some kind
The tree
Bracketed “LISP” notation

(S (NP (DT the) (NN T-Rex))
 (VP (VBD ate)
   (NP (DT a) (JJ famous) (NN linguist))
   (PP (IN in)
     (NP (NNP Harvard) (NNP Yard))))))

\[
(S \ (NP \ (DT \ the) \ (NN \ T-Rex)) \\
(VP \ (VBD \ ate) \\
 (NP \ (DT \ a) \ (JJ \ famous) \ (NN \ linguist)) \\
 (PP \ (IN \ in) \\
   (NP \ (NNP \ Harvard) \ (NNP \ Yard))))))
\]
Compositionality

Interpretation of meaning processes each subtree separately

A major claim in many theories of syntax/semantics

“T-Rex” : \( x : \text{isa}(x, \text{TRex}) \)

“linguist” : \( x : \text{isa}(x, \text{linguist}) \)

“famous linguist” : \( x : \text{isa}(x, \text{linguist}) \land \text{famous}(x) \)

“ate a famous linguist” : \( e(x) : \text{event}(e(x)) \land \text{wherein}(e, \text{ate}(x, y)) \land \text{isa}(y, \text{linguist}) \land \text{famous}(y) \)

“the T-Rex ate a famous linguist” :
\( e(x) : \text{event}(e(x)) \land \text{wherein}(e, \text{ate}(x, y)) \land \text{isa}(y, \text{linguist}) \land \text{famous}(y) \land \text{isa}(\text{TRex}, x) \)
Heads and modifiers

- Each phrase has a *head* element
- Usually of the type that we named the phrase after (eg N for an NP)
- A direct child of the phrase node
- Tells us the main content of the phrase
  - T-Rex for “the T-Rex”
  - ate for “ate a linguist”
  - “linguist” for “a famous linguist”
  - “in” for “in Harvard Yard”
- The other stuff in the phrase is split into *arguments*
  - Usually required; tell us what the head applies to
- And *modifiers*
  - Usually optional; tell us a detail about the head
Context-free grammars

We can represent the structure of such a tree as a context-free grammar

- Another formal language, like regular expressions
- Although more powerful
- Grammar is being used in a slightly different sense
  - Here means “rules for a formal language”, not “rules of English”

CFG

A start symbol: S

- Rules of the form: \( L \rightarrow R_1 R_2 \ldots R_k \)
- \( L \) is a nonterminal symbol
- The \( R \) are either more nonterminals, or terminal symbols
  - The “words”
A simple grammar

S → NP VP
NP → DT NN
DT → the
DT → a
NN → T-Rex
NN → linguist
VP → V NP
V → ate
Deriving a sentence

S → NP VP
NP → DT NN
DT → the
DT → a
NN → T-Rex
NN → linguist
VP → V NP
V → ate

▶ S
Deriving a sentence

S → NP VP
NP → DT NN
DT → the
DT → a
NN → T-Rex
NN → linguist
VP → V NP
V → ate

- S
- NP VP
Deriving a sentence

S → NP VP
NP → DT NN
DT → the
DT → a
NN → T-Rex
NN → linguist
VP → V NP
V → ate

▶ S
▶ NP VP
▶ DT NN VP
▶ a NN VP
▶ a linguist VP
Deriving a sentence

S → NP VP
NP → DT NN
DT → the
DT → a
NN → T-Rex
NN → linguist
VP → V NP
V → ate

- S
- NP VP
- DT NN VP
- a NN VP
- a linguist VP
- a linguist V NP
- a linguist ate NP
Deriving a sentence

\[
S \rightarrow NP \ VP \\
NP \rightarrow DT \ NN \\
DT \rightarrow \text{the} \\
DT \rightarrow \text{a} \\
NN \rightarrow \text{T-Rex} \\
NN \rightarrow \text{linguist} \\
VP \rightarrow V \ NP \\
V \rightarrow \text{ate}
\]

- S
- NP VP
- DT NN VP
- a NN VP
- a linguist VP
- a linguist V NP
- a linguist ate NP
- a linguist ate a linguist
Some implications

- The rule that inserts the NP node...
- And the rule that expands the NP node...
- Don’t interact, except in that they both deal with an NP
- A version of the Markov property

In particular, although this grammar was designed only for generating one sentence, it generates 8:

\(((a|the) \ (linguist|T-Rex)) \ ate \ ((a|the) \ (linguist|T-Rex))\)
Weaseling out of the Markov property

If we wanted to encode a proper understanding of the food chain, we could flatten the tree:

$$S \rightarrow DT \text{ T-Rex ate DT linguist}$$

But this loses all our insights about constituents...

We can also expand the grammar:

- $$S \rightarrow NP_{predator} \ VP$$
- $$NP_{predator} \rightarrow DT \ NN_{predator}$$
- $$VP \rightarrow V \ NP_{prey}$$
- $$NN_{predator} \rightarrow \text{T-Rex}$$
- $$NN_{prey} \rightarrow \text{linguist}$$

Changes like this increase accuracy, but also sparsity

- We’ll see many examples later on
Prepositional phrases

Let’s add PPs to our grammar.

S → NP VP
NP → DT NN
DT → the
DT → a
NN → T-Rex
NN → linguist
VP → V NP
V → ate
PP → IN NP
Prepositional phrases

Let’s add PPs to our grammar.

- Under VP, to cover “Kim slept [PP on the couch]”

\[
\begin{align*}
S & \rightarrow \text{NP VP} \\
\text{NP} & \rightarrow \text{DT NN} \\
\text{DT} & \rightarrow \text{the} \\
\text{DT} & \rightarrow \text{a} \\
\text{NN} & \rightarrow \text{T-Rex} \\
\text{NN} & \rightarrow \text{linguist} \\
\text{VP} & \rightarrow \text{V NP} \\
\text{V} & \rightarrow \text{ate} \\
\text{PP} & \rightarrow \text{IN NP} \\
\text{VP} & \rightarrow \text{V NP PP}
\end{align*}
\]
Prepositional phrases

Let’s add PPs to our grammar.

- Under VP, to cover “Kim slept [PP on the couch]”
- Under NP, to cover “Kim has [NP a cat [PP with black fur]]”

S → NP VP
NP → DT NN
DT → the
DT → a
NN → T-Rex
NN → linguist
VP → V NP
V → ate
PP → IN NP
VP → V NP PP
NP → NP PP
Ambiguity

Sam saw a moose with a telescope

or

VP

saw

NP

a moose

PP

with a telescope

or

VP

saw

NP

NP

a moose

PP

with a telescope
Different interpretations

Compositionality implies that these correspond to different meanings:
saw... with a telescope vs moose with a telescope
Long-distance effects

- Another pattern of alternations
- If X is grammatical, Y is grammatical
- Subject questions
  - “Kim bit Robin”: “Who bit Robin?”
  - “Sandy shouted at Kim”: “Who shouted at Kim?”
- Implies “who” can fill an NP slot in the grammar
Non-subject questions

- The “who” doesn’t need to be a subject
  - “Kim bit Robin”: “Who(m) did Kim bite?”
  - “Sandy shouted at Kim”: “Who(m) did Sandy shout at?”

First attempt to add this to our tiny grammar:

\[ SQ \rightarrow WHNP \text{ did } VP \]
Non-subject questions

- The “who” doesn’t need to be a subject
  - “Kim bit Robin”: “Who(m) did Kim bite?”
  - “Sandy shouted at Kim” : “Who(m) did Sandy shout at?”

First attempt to add this to our tiny grammar:

```
SQ  →  WHNP did VP
```

Issue:

- This generates sentences like: “Who did Kim bite Sandy”

What we need is more like:

```
SQ  →  WHNP did VP-missing-an-NP
```
Living with the consequences

We can continue trying to write out the grammar from here

- We’ll borrow from Categorial Grammar and write VP-missing-an-NP as VP/NP

\[
\begin{align*}
\text{VP/NP} & \rightarrow \text{V (no NP)} \\
\text{VP/NP} & \rightarrow \text{V NP PP/NP} \\
\text{PP/NP} & \rightarrow \text{IN (no NP)}
\end{align*}
\]

- Recursive rules have to keep passing down the /NP
- Coping with these make the grammar much larger
- Often hypothesized that what’s really going on in the mind isn’t quite context-free

Some syntactic theories believe the (no NP) is actually “there”

- Called a \textit{trace}: represented in the Penn Treebank by \textit{*T*}
Other long-distance effects

- “It seems that Kim likes cheese” : “Kim seems to (*) like cheese”
- “Sandy wants herself to go home” : “Sandy wants (*) to go home”
- “We pet the cats” : “It is the cats that we pet (*)”
Quick digression: Chomsky

Chomsky’s analysis of these kinds of constructions involves movement:

- Grammar first generates statement
  - “We pet the cats”
- Then post-processes by applying *transformations*
  - Remove “cats” and leave behind a trace
  - Move “cats” left to some new position in the tree
  - Add the “it is” and “that” to fix everything up
- Details depend on specific era of Chomsky
- Theories often assume lots of usually-empty nodes in which you can put the moved stuff
- ...and fill in the “fix-up” stuff
Quick digression: Chomsky

Chomsky’s analysis of these kinds of constructions involves movement:

- Grammar first generates statement
  - “We pet the cats”
- Then post-processes by applying transformations
  - Remove “cats” and leave behind a trace
  - Move “cats” left to some new position in the tree
  - Add the “it is” and “that” to fix everything up
- Details depend on specific era of Chomsky
- Theories often assume lots of usually-empty nodes in which you can put the moved stuff
  - ...and fill in the “fix-up” stuff

Computational linguists don’t like “movement” much

- Algorithms are complicated and inefficient

Prefer to pursue the VP/NP style account.
Our intuitions about constituency, replacement suggest the sentence is hierarchical

Motivate a tree representation...

...which we can write as a CFG

CFG has the Markov property

Which is computationally helpful (we’ll see soon)

But makes some grammatical phenomena hard to capture