

# Python Course: Lecture 16

February 9, 2006

## 1 Computational Syntax

- Now this course will take a look at computational syntax.
- Syntax is (roughly) the study of the way that words combine into more complicated structures in order to convey meaning.
- We'll introduce a few basic syntactic concepts from theoretical linguistics and discuss how to encode them formally in Python.
- We'll also look at ways syntax can be modeled statistically using techniques similar to what we used for N-grams.

## 2 Theoretical Syntax

### 2.1 Grammatical vs. Ungrammatical Distinction

- Theoretical syntax concerns itself with *sentences*.
- For purposes of this class, we'll rely on your intuition about what comprises a sentence—e.g. a short sequence of words expressing a cohesive thought possessing certain structural characteristics like a subject and a verb, etc.
- In theoretical syntax we make a distinction between sentences that are *grammatical* and those that are *ungrammatical*.
- Grammatical sentences are ones that a typical English speaker would regard as well-formed. Ungrammatical ones sound ill-formed or somehow incorrect.
- By convention we mark ungrammatical sentences with an asterisk.
  - (1) John gave the pen to Mary.
  - (2) \*Pen give Mary John a the to.

- Note that ungrammatical doesn't mean merely nonsensical (“Colorless green ideas sleep furiously”), aesthetically unpleasing (“It was a dark and stormy night . . .”), hard to understand (“Difference is never in itself a sensible plenitude”)<sup>1</sup> or something frowned upon by high school English teachers that people nevertheless say all the time (“Where is he at?” “To boldly go where no man has gone before.”). There has to be a sense that the language has been used incorrectly, not that it has been used correctly for obscure or unpleasant means.
- If this all seems vague, it is. In practice grammaticality judgments are highly subjective, and you can often invent a context that makes an “ungrammatical” sentence sound “grammatical”.
- At the same time, however, it is possible to come up with sentence pairs that appear to have distinctly different degrees of grammaticality, all other things being equal. This class will focus on giving accounts for those sentence pairs.

## 2.2 Classic Syntactic Effects

Here are examples of a few basic syntactic mechanisms that can be reflected in grammaticality judgments.

As with all grammaticality judgments, the distribution of asterisks may not line up with your personal intuitions, and it is possible to come up with contexts that goose a ungrammatical sentence into a grammatical one or vice-versa.

The point here is to illustrate the kinds of phenomena that syntacticians are interested in accounting for, and to show what sorts of theoretical machinery we need to be able to express them.

- *Agreement*

There is a notion of agreement in English—you have to be consistent in your use of, say, singular vs. plural word forms.

(3) The boys love Led Zeppelin.

(4) Mike loves Led Zeppelin.

(5) \*The boys loves Led Zeppelin.

(6) \*Mike love Led Zeppelin.

(7) \*The boys I met last week in Oswego loves Led Zeppelin.

What's going on here is that the subject of the sentence has to agree with the main verb in number. So we need to be able to express the structural notions of “subject” and “main verb”.

---

<sup>1</sup><http://itre.cis.upenn.edu/~myl/languagelog/archives/000024.html>

- *Case*

Different English pronouns get used depending on whether they function as the subject or direct object of a verb.

(8) I gave the book to him.

(9) \*I gave the book to he.

(10) He gave the book to me.

(11) \*Him gave the book to me.

We need to be able to express the notions of “subject” and “direct object”.

- *Subcategorization*

Certain verbs appear to require certain objects. For example “kill” and “devour” want a direct object, but “ate” is okay by itself.

(12) Eddie killed the bug.

(13) \*Eddie killed.

(14) Jane ate.

(15) \*Jane devoured.

Again, we need some notion of a verb’s “object”.

- *Anaphora*

Certain noun forms like “herself”, “she”, “he”, “himself”, etc. don’t have referents themselves, but get their referents from other nouns in the sentence, called *antecedents*. This phenomenon is called *anaphora*. In any particular instance of anaphora, there can be subtle rules for which terms can serve as antecedents. Here coindexing is used to indicate whether or not a particular word can serve as an antecedent.

(16) John<sub>*i*</sub> likes himself<sub>*i*</sub>.

(17) \*John<sub>*i*</sub> likes himself<sub>*j*</sub>.

(18) \*John<sub>*i*</sub> likes him<sub>*i*</sub>.

(19) John<sub>*i*</sub> likes him<sub>*j*</sub>.

(20) The people who know John<sub>*i*</sub> like him<sub>*i*</sub>.

(21) The people who know John<sub>*i*</sub> like him<sub>*j*</sub>.

The exact relationship between “John” and “him” and “himself” in these sentences is hard to pin down.

### 3 Syntactic Trees

- We need a way of accounting for the contrasts illustrated in section (2.1).
- Merely looking at the linear order of words (as we did for N-grams) won't be sufficient. The phenomena are too complex.
- Given a string of words, we need to map them into some more sophisticated data structure that can be used in making these distinctions.
- The typical structure that linguists have relied on since the 1950s is a branching tree structure.
- This is nice for programmers, since the computer science literature is rife with examples of how to work with trees.
- But a data structure isn't true merely because it's convenient, so let's do a high-level motivation of tree structure as a meaningful way of describing a sentence. Take the following sentence as an example.

(22) He put the book on the table.

- The individual words in the sentences appear to fall into rough equivalence classes in the sense that they can be replaced with other words, only some of which harm the grammaticality of the sentence.

(23) She put the book on the table.

(24) Mary put the book on the table.

(25) He put the book under the table.

(26) \*He put the book she the table.

(27) \*She put the book on Mary table.

Call these equivalence classes *parts of speech*. We'll use the familiar ones like noun (N), verb (V), preposition (P), and determiner (D).

- We can represent (3) and its part of speech information like so.

N	V	D	N	P	D	N
He	put	the	book	on	the	table

- A similar substitution/equivalence class argument indicates that there are certain substrings of that go together more than others.

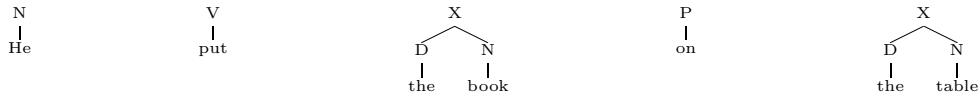
(28) He put it on the table.

(29) \*He put it the table.

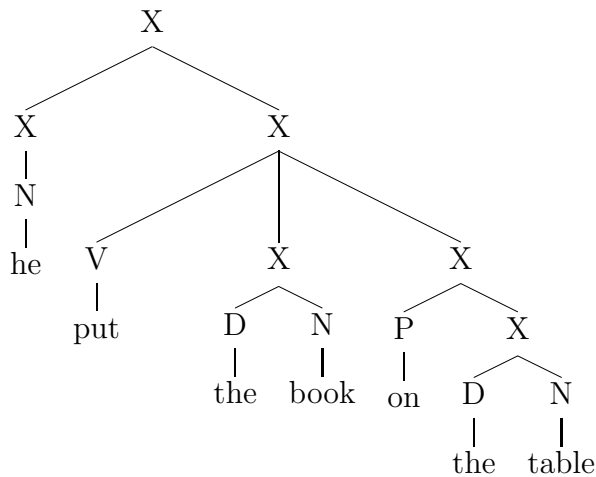
- (30) He put the book there.
- (31) The librarian put the book on the table.
- (32) \*The librarian the book on the table.

Making a less empirically-motivated appeal to your intuition, there is also a sense in which certain substrings do a better job by themselves of expressing a coherent thought than others. For example “the book” is an entity and “on the table” is a location, but “on the” isn’t really anything.<sup>2</sup>

These arguments incline us to add a further grouping of the words, which can be represented as parent nodes of the part of speech nodes.



- These composed elements could serve as building blocks for even larger components called *constituents*, until structure is assigned to the entire sentence.

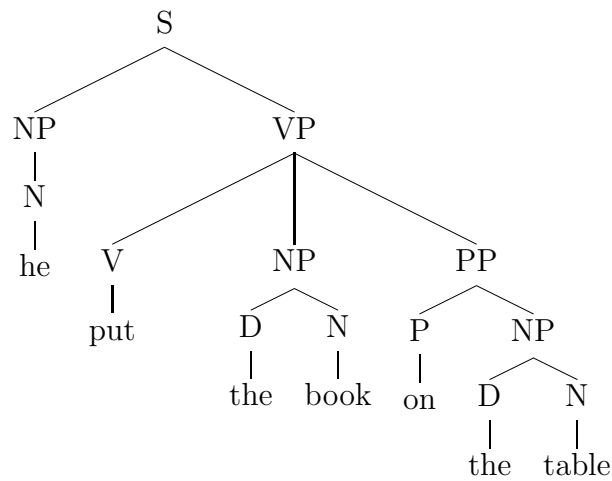


This isn’t the only way you could draw a tree above these nodes. (In fact a fair amount of theoretical syntax involves debates about what sorts of trees we should be drawing.) However, it is a reasonable picture given the motivating arguments above.

- Returning to semantic intuition, there is a sense in which the different constituents belong to equivalence classes similar to the parts of speech. For example, “the book” is a noun-like entity. By virtue of being a location, “on the table” is a preposition-like entity and so on. We can represent that intuition in the tree structure by labeling the higher-level nodes as “noun phrases” (abbreviated NP), “verb phrases” and so on.

---

<sup>2</sup>Except of course a high-frequency bigram.



Note that we are hypothesizing that the structure is *compositional*—a node can only be part of a constituent that dominates it in the tree. For example the token “he” can’t be part of the prepositional phrase “on the table”.

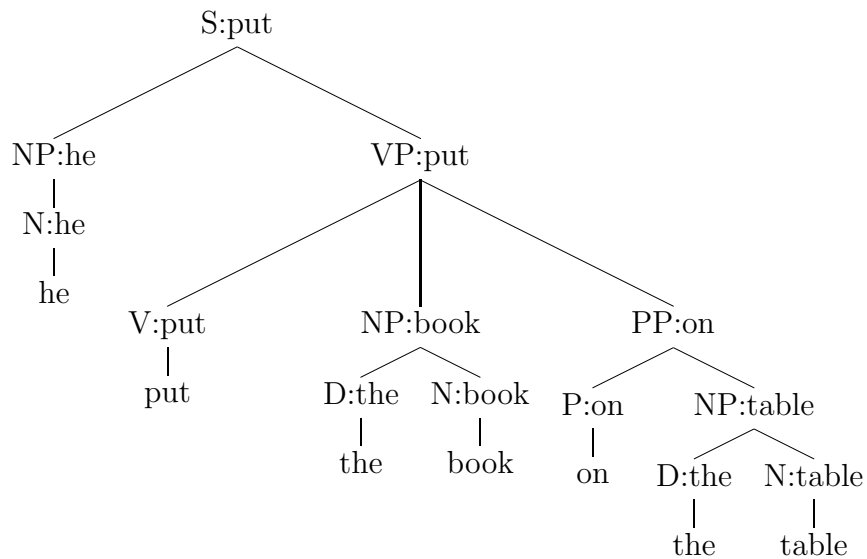
Compositionality is a reasonable hypothesis. It is also a simplifying assumption. If we abandoned it, we’d have to draw data structures more complicated than these trees.<sup>3</sup>

- Finally, notice that we’ve chosen our node labels such that each constituent is of the same “type” as one of its subnodes. For example, the noun phrase “the book” contains the constituent “book”, and the prepositional phrase “on the table” contains the preposition “on”.

The choice reflects the hypothesis the syntactic constituents have *heads* which are the most important words they contain. You can think of the non-head words in a syntactic constituent as being modifiers of the head. Alternately, you can think of the head as “projecting” upwards into more elaborated structures. Either way, it is sometimes useful to annotate the constituents with their lexical heads.

---

<sup>3</sup>An equivalent way of stating the compositionality hypothesis is to posit that the lines in a tree structure cannot cross one another.



Again, there are many theoretical assertions implicit in the picture I’ve draw above. For example, I have decided that the entire sentence is headed by the main verb “put” instead of some other component. Also, there is a perennial debate in linguistics about whether noun phrases should be headed by nouns or determiners. I’ve chosen the former here, but you could do the latter.

- All the preceding is a simplified version of a branch of theoretical linguistics called *X-bar Theory*. This brief account doesn’t do X-bar Theory justice, but the basic structural components—trees, nodes, and heads—are things we’ll need to be able to capture in code if we’re going to use Python to do X-bar like syntax.