English Syntax and Context Free Grammars

JHU Summer Workshop

Instructor: Jackie CK Cheung

June 11, 2019

About Me

Education:

- BSc in Computer Science (UBC) 2004-2008
- MSc / PhD in Computer Science (Toronto) 2008-2014
- Assistant professor at McGill 2015-

Member of Mila Quebec AI Institute

Research topics in my lab:

- Natural language generation
- Automatic summarization
- Computational discourse
- Computational semantics
- Commonsense reasoning

Syntax

How words can be arranged together to form a **grammatical** sentence.

- *This is a valid sentence.*
- **A sentence this valid is.*

An asterisk is used to indicate **ungrammaticality**.

One view of syntax:

Generate all and exactly those sentences of a language which are grammatical

The First Grammarian

Panini (Pāṇini) from the 4th century B.C. developed a grammar for Sanskrit.

51. The affixes ktvâ, क्त and क्तवतु optionally get इट् after पू ॥

As पूत्वा or पवित्वा, सोमोतिपूतः, सोमोतिपवितः पूतवान् or पवितवान् ॥ This allows option where by VII. 2. 11 there would have been prohibition. See I. 2. 22.

वसतिक्षुधोरिट् ॥ ५२ ॥ पदानि ॥ वसति, क्षुधोः, इट् ॥

वृत्तिः ॥ वसतेः क्षुधेश्च क्त्वानिष्ठयोरिडागमो भवति ।

52. The affix ktvâ, kta and ktavatu always receive the augment इट् after वस् (वसति) and क्षुध् ॥

As उषित्वा, उषितः and उषितवान्, क्षुधित्वा, क्षुधितः, क्षुधितवान् ॥ The वस् of the Adâdi class will get इट् as it is enumerated in the list of सेट् roots. The repetition of इट् shows that the rule is invariable, the 'optionally' of the preceding sûtra does not affect it.

अञ्चेः पूजायाम् ॥ ५३ ॥ पदानि ॥ अञ्चेः, पूजायाम् ॥

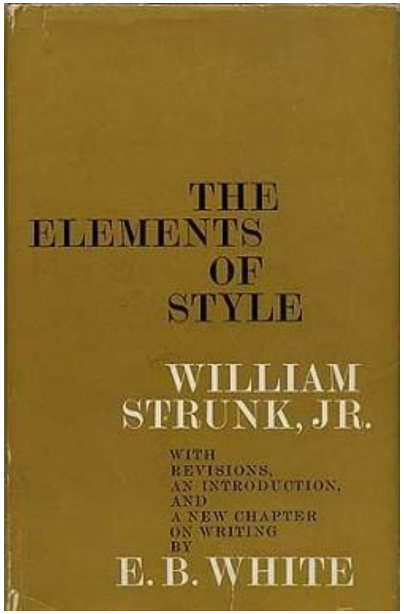
वृत्तिः ॥ अञ्चेः पूजायामर्थे क्त्वानिष्ठयोरिडागमो भवति ।

Source: https://archive.org/details/ashtadhyayitrans06paniuoft

What We Don’t Mean by Grammar

Rules or guides for how to write properly

e.g.,



These style guides are **prescriptive**. We are concerned with **descriptive** grammars of naturally occurring language.

Basic Definitions Terms

- grammaticality
- prescriptivism vs descriptivism
- constituency
- grammatical relations
- subcategorization

Constituency

A group of words that behave as a unit

Noun phrases:

- *computational linguistics, it, Justin Trudeau, three people on the bus, "Jean-Claude Van Damme, the Muscles from Brussels"*

Adjective phrases:

- *blue, purple, very good, ridiculously annoying and tame*

Tests for Constituency

1. They can appear in similar syntactic environments.

I saw …

it

Jean-Claude Van Damme, the Muscles from Brussels

three people on the bus

**Van*

**on the*

Tests for Constituency

2. They can be placed in different positions or replaced in a sentence as a unit.

[Jean-Claude Van Damme, the Muscles from Brussels], beat me up.

It was [Jean-Claude Van Damme, the Muscles from Brussels], who beat me up.

I was beaten up by [Jean-Claude Van Damme, the Muscles from Brussels].

He beat me up. (i.e., J-C V D, the M from B)

Tests for Constituency

3. It can be used to answer a question.

Who beat you up?

[Jean-Claude Van Damme, the Muscles from Brussels]

**[the Muscles from]*

Grammatical Relations

Relationships between different constituents

Subject

- *Jean-Claude Van Damme relaxed.*
- *The wallet was stolen by a thief.*

(Direct) object

- *The boy kicked the ball.*

Indirect object

- *She gave him a good beating.*

There are many other grammatical relations.

Subcategorization

Notice that different verbs seem to require a different number of **arguments**:

relax	1	subj
steal*	2	subj, dobj
kick	2	subj, dobj
give	3	subj, iobj, dobj

*the passive changes the subcategorization of the verb

More Subcategorization

Some other possibilities:

want	2	subj, inf. clause

- *I want to learn about computational linguistics.*

apprise	3	subj, obj, pobj with of

- *The minister apprised him of the new developments.*

different	2	subj, pobj with from/than/to

- *This course is different [from/than/to] what I expected.*

Short Exercise

Identify the prepositional phrase in the following sentence. Give arguments for why it is a constituent.

The next assignment is due on Tuesday, October 16th.

Formal Grammars

Since we are computational linguists, we will use a formal computational model of grammar to account for these and other syntactic concerns.

Formal grammar

Rules that generate a set of strings that make up a **language**.

(In this context, language simply refers to a set of strings.)

Why?

- Formal understanding lets us develop appropriate algorithms for dealing with syntax.
- Implications for cognitive science/language learning

FSAs and Regular Grammars

FSAs can be used to describe aspects of English morphology

- An FSA generates a **regular language**
- FSAs correspond to a class of formal grammars called **regular grammars**

To describe the syntax of natural languages (with multiple constituents, subcategorization, etc.), we need a more powerful class of formal grammars – **context free grammars** (**CFG**s).

Context Free Grammars (CFG)s

Rules that describe what possible sentences are:

S → NP VP

NP → this

VP → V

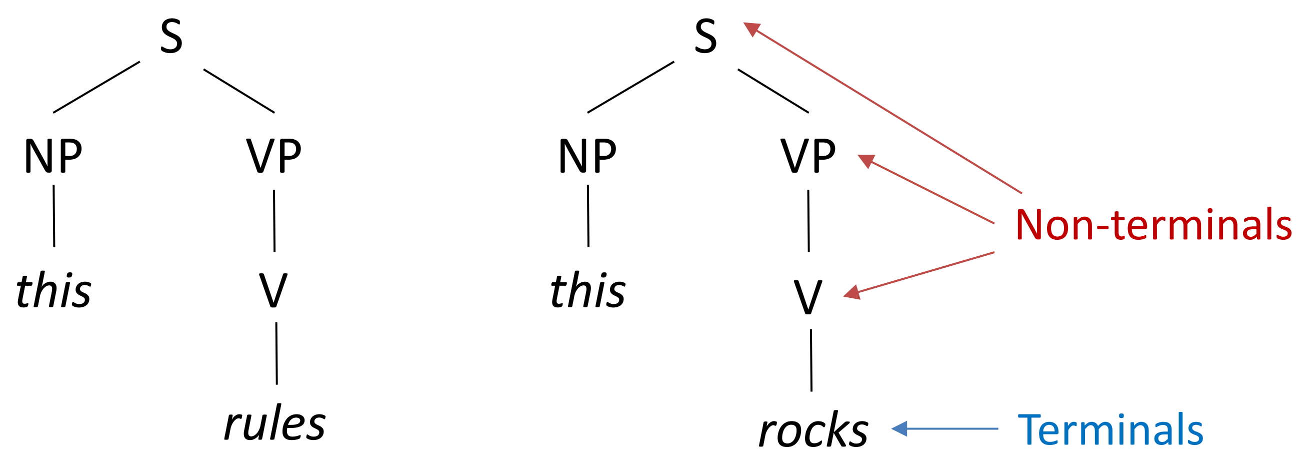
V → is | kicks| jumps | rocks

Constituent Tree

Trees (and sentences) generated by the previous rules:

S → NP VP NP → *this*

VP → V V → *is* | *rules*| *jumps* | *rocks*



Formal Definition of a CFG

A 4-tuple:

- N set of **non-terminal** symbols
- Σ set of **terminal** symbols
- R set of **rules** or **productions** in the form $A \rightarrow (\Sigma \cup N)^*$, and $A \in N$
- S a designated **start symbol**, $S \in N$

Extended Example

Let's develop a CFG that can account for verbs with different subcategorization frames:

intransitive verbs	*relax*	1	subj
transitive verbs	*steal, kick*	2	subj, dobj
ditransitive verbs	*give*	3	subj, iobj, dobj

Undergeneration and Overgeneration

Problems with above grammar:

Undergeneration: misses valid English sentences

- *The boy kicked the ball softly*.
- *The thief stole the wallet with ease.*

Overgeneration: generates ungrammatical sentences

- **The boy kick the ball.*
- **The thieves steals the wallets.*

Extension 1

Let’s add adverbs and prepositional phrases to our grammar

Recursion

Consider the following sentences:

- *The dog barked.*
- *I know that the dog barked.*
- *You know that I know that the dog barked.*
- *He knows that you know that I know that the dog barked.*
- …

In general:

S -> NP VP

VP -> Vintr

Vintr -> *barked*

VP -> Vthat Sthat

Vthat-> *know*

Sthat -> *that* **S**

Recursion

This recursion in the syntax of English means that sentences can be infinitely long (theoretically).

- For a given sentence S, you can always make it longer by adding [I/you/he know(s) that S].

In practice, the length is limited because we have limited attention span/memory/processing power.

Exercise

Let’s try to fix the subject-verb agreement issue:

Present tense:

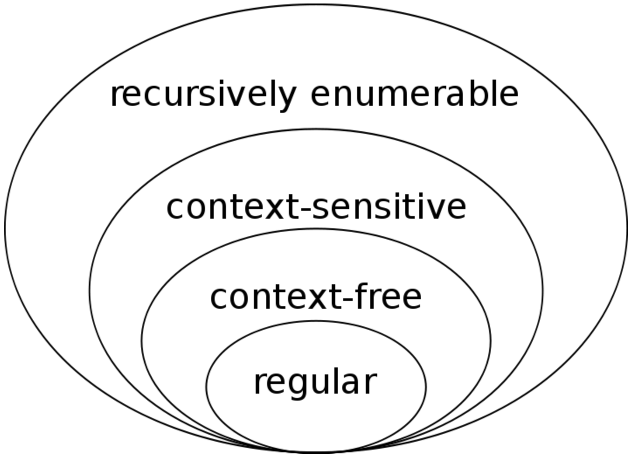
- Singular third-person subject -> verb has affix of –s or –es
- Otherwise -> base form of verb
- (*to be* is an exception, along with other irregular verbs)

Are Natural Languages CFGs?

Recall that a formal language is defined to be a set of strings constructed over a specified vocabulary

Are natural languages CFGs? i.e., can we define each natural language (e.g., English, French, German, etc.) as a CFG?

Other possibilities: **Chomsky hierarchy**



https://en.wikipedia.org/wiki/Chomsky_hierarchy

Cross-serial Dependencies

Swiss German (Shieber, 1985) and Bambara (Culy, 1985) have structures that generate strings which cannot be captured by CFGs (**cross-serial dependencies**):

$$a^m b^n c^m d^n$$

Relies on following assumption:

- m and n can be arbitrarily large values
- strings are either in a language or not (grammatical or ungrammatical)

May not be the most useful question to ask after all

Parsing

Input sentence, grammar → output parse tree

Parsing into a CFG: **constituent parsing**

Parsing into a dependency representation: **dependency parsing**

Difficulty: need an efficient way to search through plausible parse trees for the input sentence

Parsing into a CFG

Given:

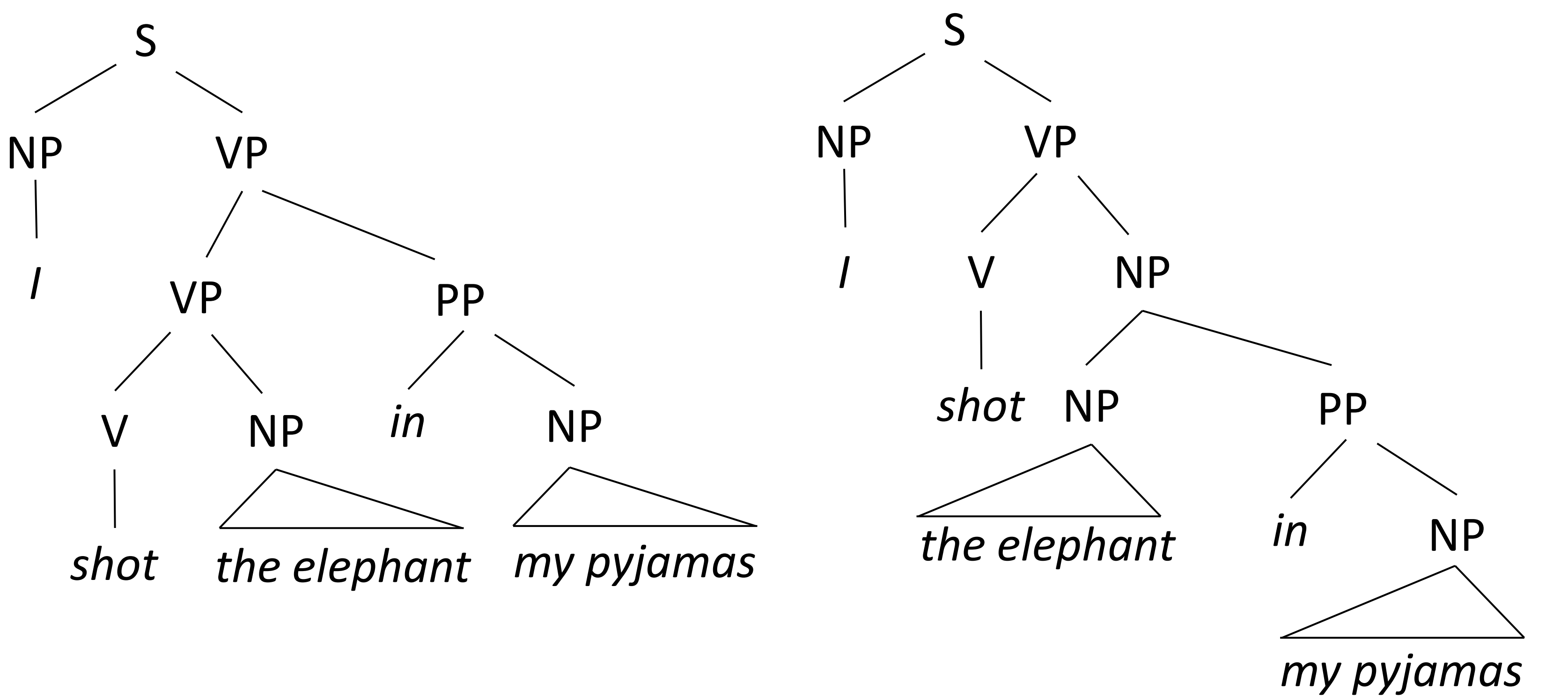
1. CFG
2. A sentence made up of words that are in the terminal vocabulary of the CFG

Task: Recover all possible parses of the sentence.

Why *all* possible parses?

Syntactic Ambiguity

I shot the elephant in my pyjamas.



Types of Parsing Algorithms

Top-down

Start at the top of the tree, and expand downwards by using rewrite rules of the CFG to match the tokens in the input string

e.g., Earley parser

Bottom-up

Start from the input words, and build ever-bigger subtrees, until a tree that spans the whole sentence is found

e.g., **CYK algorithm**, shift-reduce parser

Key to efficiency is to have an efficient search strategy that avoids redundant computation

CYK Algorithm

Cocke-Younger-Kasami algorithm

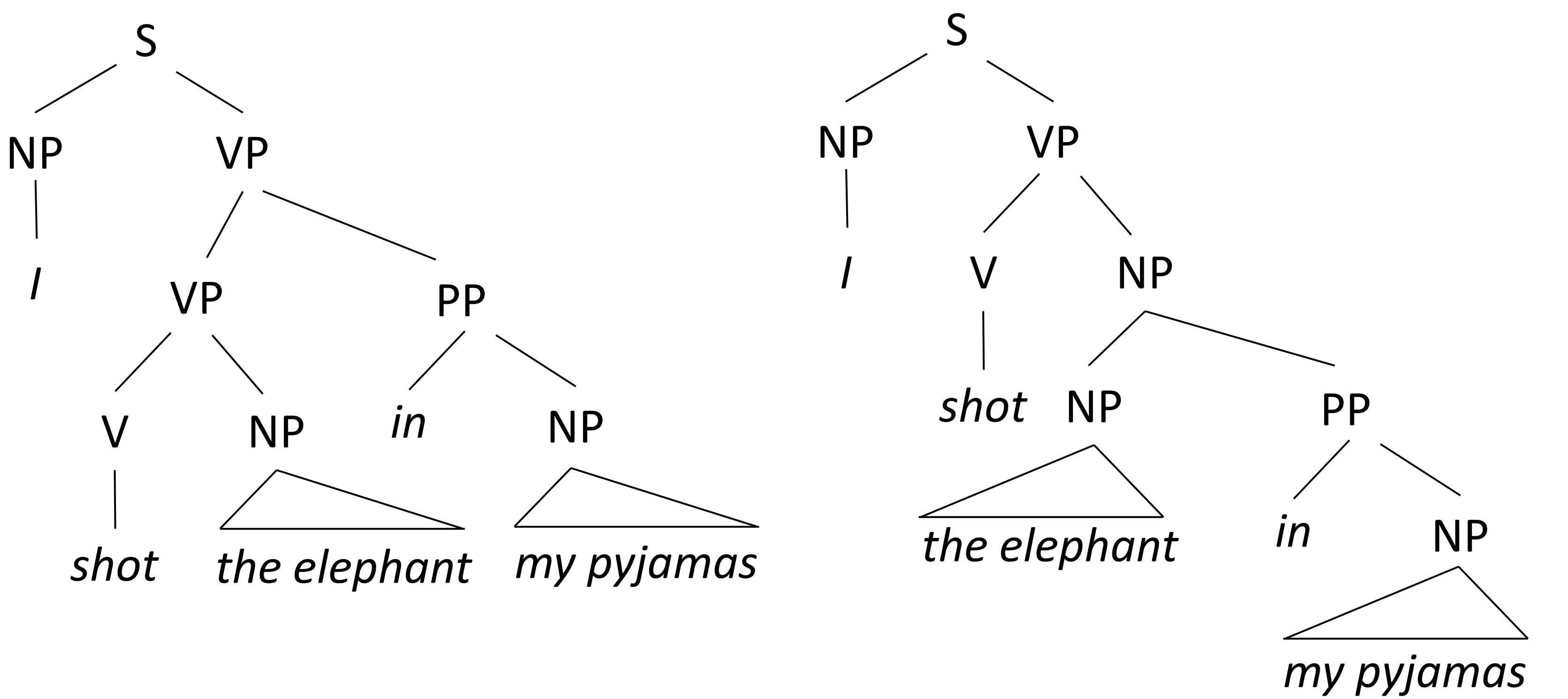
- A **dynamic programming** algorithm – partial solutions are stored and efficiently reused to find all possible parses for the entire sentence.
- Also known as the CKY algorithm

Steps:

1. Convert CFG to an appropriate form
2. Set up a table of possible constituents
3. Fill in table
4. Read table to recover all possible parses

Dealing with Syntactic Ambiguity

In practice, one of these is more likely than the other:



How to distinguish them?

Probabilistic CFGs

Associate each rule with a probability:

e.g.,

NP $\rightarrow$ NP PP	0.2
NP $\rightarrow$ Det N	0.4
NP $\rightarrow$ *I*	0.1
…	
V $\rightarrow$ *shot*	0.005

Probability of a parse tree for a sentence is the product of the probabilities of the rules in the tree.

Formally Speaking

For each nonterminal $A \in N$,

$$\sum_{\alpha \to \beta \in R \; s.t. \alpha = A} \Pr(\alpha \to \beta) = 1$$

- i.e., rules for each LHS form a probability distribution

If a tree t contains rules $\alpha_1 \to \beta_1, \alpha_2 \to \beta_2, \ldots,$

$$\Pr(t) = \prod_i \Pr(\alpha_i \to \beta_i)$$

- Tree probability is product of rule probabilities

Probabilistic Parsing

Goal: recover the best parse for a sentence, along with its probability

For a sentence, sent,

let $\tau(\text{sent})$ be the set of possible parses for it,

we want to find

$$\underset{t \in \tau(\text{sent})}{\operatorname{argmax}} \Pr(t)$$

Can extend CYK to keep track of probabilities in table

How to Train a PCFG?

Derive from a treebank, such as WSJ.

Simplest version:

- each LHS corresponds to a categorical distribution
- outcomes of the distributions are the RHS
- MLE estimates:

$$\Pr(\alpha \to \beta) = \frac{\#(\alpha \to \beta)}{\#\alpha}$$

- Can smooth these estimates in various ways