Natural Language Processing
ECS763P

Feb 6th 2017

Mehrnoosh Sadrzadeh

Topics Covered in lectures: Feb 6, 13, 27.

1- Formal Grammar of English

2- Syntactic Parsing

3- Statistical Parsing

4- Formal Semantics (might carry over to March 6th)

Chapters 12-14 and 18 of text book

Jurafsky and Martin

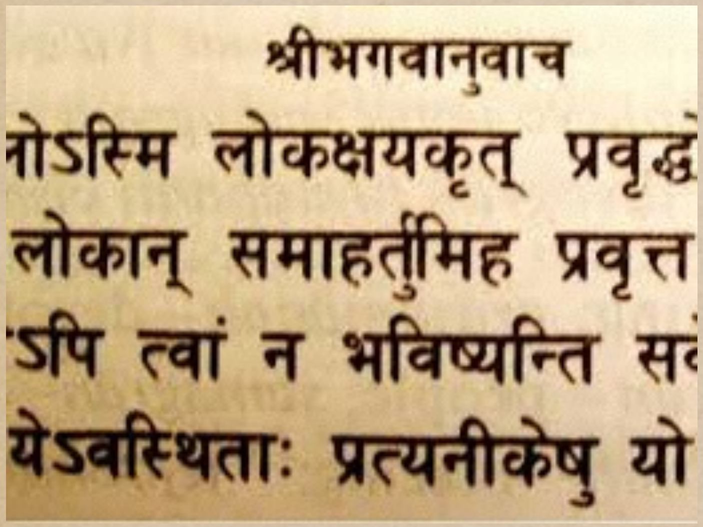
Pearson International Edition,2nd edition,

copy right 2009

Formal Grammar

A bit of history of Formal Grammar:

The first formal grammar was written over 2000 years ago for Sanskrit by Panini. But it is still referenced today when teaching Sanskrit and studying its grammar.



श्रीभगवानुवाच

नोऽस्मि लोकक्षयकृत् प्रवृद्ध

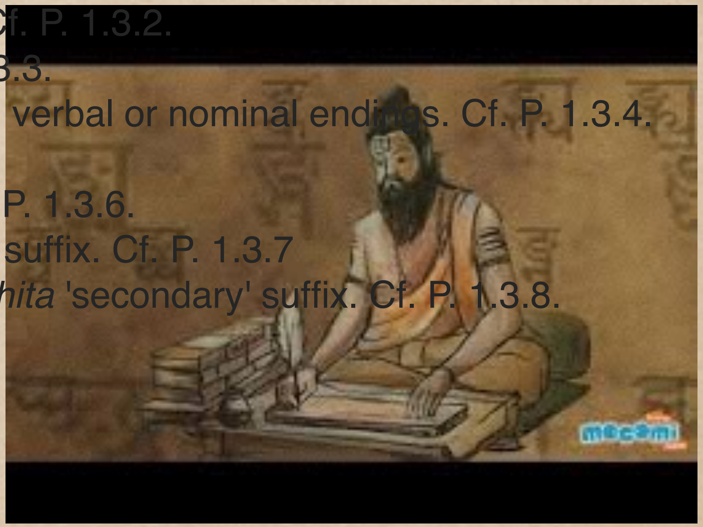
लोकान् समाहर्तुमिह प्रवृत्त

ऽपि त्वां न भविष्यन्ति स

येऽवस्थिताः प्रत्यनीकेषु यो

1 Nasalized vowels, e.g. *bhañjO*. Cf. P. 1.3.2.
2 A final consonant (*haL*). Cf. P. 1.3.3.
3 2. (a) except a dental, *m* and *s* in verbal or nominal endings. Cf. P. 1.3.4.
4 Initial *ñi ṭu ḍu*. Cf. P 1.3.5
5 Initial *ṣ* of a suffix (*pratyaya*). Cf. P. 1.3.6.
6 Initial palatals and cerebrals of a suffix. Cf. P. 1.3.7
7 Initial *l*, *ś*, and *k* but not in a *taddhita* 'secondary' suffix. Cf. P. 1.3.8.

1.1.1: {*ā*, *ai*, *au*} are called *vṛddhi*.
1.1.2: {*a*, *e*, *o*} are called *guṇa*.



Formal Grammar

A bit of history of Formal Grammar:

Formal grammar is sometimes referred to as the study of "syntax" (versus semantics or pragmatics). The word "syntax" originates from the Greek word "**SYNTAXIX**",which meant :

setting out together

arrangement

In a linguistic context, this word is used to refer to "the ways words are arranged together", e.g. in the sentences and other constructions of natural language.

Formal Grammar

There are three main ideas in studying formal grammar:

1- Constituency

2- Grammatical Relations

3- Dependency

Formal Grammar

Constituency

Groups of words that behave as a single unit are called a constituent.

Example: a noun phrase, which is a group of words that acts as a unit. It can be just a single word, naming an individual such as she or Michael. It can also be a phrase, such as the house, Russian Hill, the deep blue sky.

We will introduce a formalism called Context Free Grammars, which allows us to study such constituency facts, e.g. what are constituents and how do they behave.

Formal Grammar

Grammatical Relations

These are relationships between the constituents.
Examples are Subject and Object.

For example in the sentence
“She adores the deep blue sky”,
she and deep blue sky are noun phrase constituents that are the subject and the object of the adores.

Formal Grammar

Dependency Relations

These are special type of relations between the words and phrases.

For example, the verb want can be followed by an infinitive, e.g. in the sentence I want to sleep.

It can also be followed by a noun phrase, e.g. in I want a sleeping bag.

This is not the case for all verbs, for example the verb find, cannot be followed by an infinitive. One cannot say I find to fly to Edinburgh. etc.

Context Free Grammars

In order to describe facts about grammar of English (or any language), we need a formal tool to be able to say things like " noun phrases can occur before verbs to form sentences". Note that not all the words in a noun phrase can occur before verb, for example we can say

"three parties from London arrived"

but we cannot say

* "three parties London from arrived".

The word "from" cannot occur before a verb.

Other examples: "the is, as attracts, spot sat, ..."

Context Free Grammars

Similarly, we can say:

“On Sept 17th, three parties from London arrived.”

This phrase can be placed in different locations,e.g.

“Three parties from London arrived on Sept 17th.”

But the individual words within this phrase, do not have the same property.

For example we cannot say

*“On Sept, three parties from London arrived 17th.”

or

*“On, three parties from London arrived Sept 17th”.

Context Free Grammars

Context free grammars or CFG's are also called Phrase Structure Grammars.

The idea behind describing grammar using constituency structure goes back to the work of psychologist Wilhem Wundt in 1900.

This idea for was formalised by Noam Chomsky in 1956 and also independently by Backus in 1959.

Context Free Grammars

A CFG has:

- a set of production rules: how symbols of language are grouped and ordered together.
- a lexicon: a set of rules encoding words of language.

For example, the following rules:

NP -> Det Nominal

NP -> ProperNoun

Nominal -> Noun|

express that a noun phrase NP can be composed of either a ProperNoun or a determiner followed by Nominal, where a Nominal can be one of more Nouns.

Context Free Grammars

A CFG can be embedded in a hierarchy, for example we can combine the previous rules with the following ones, expressing facts about the lexicon: “a” can be a determiner, the word “the” can be a determiner, and the word “flight” can be a Noun.

Det -> a

Det -> the

Noun -> flight

The symbols on the left hand side is the lexical category of the word.

Context Free Grammars

Rules with the same left hand side can also be denoted using the delimiter |, to save space. This form is often used for lexical rules, examples are as follows:

Det -> a | the | this| that
Noun -> flight | morning | star

Context Free Grammars

The symbols of a CFG are classified into two groups:

1- Terminals:

These correspond to the words of language.

The words are introduced via these rules in the lexicon.

e.g. flight, morning, star, a , the, this, that

2- Non-Terminals:

Symbols that express generalisations of these.

a.g. S, NP, VP, Noun, Det

Context Free Grammars

A CFG can be thought of in two ways:

1- Generating sentences of language

in the lexicon.

2- Giving structure to a given sentence.

In generator rule, the rules are treated as rewrite rules.

For example,

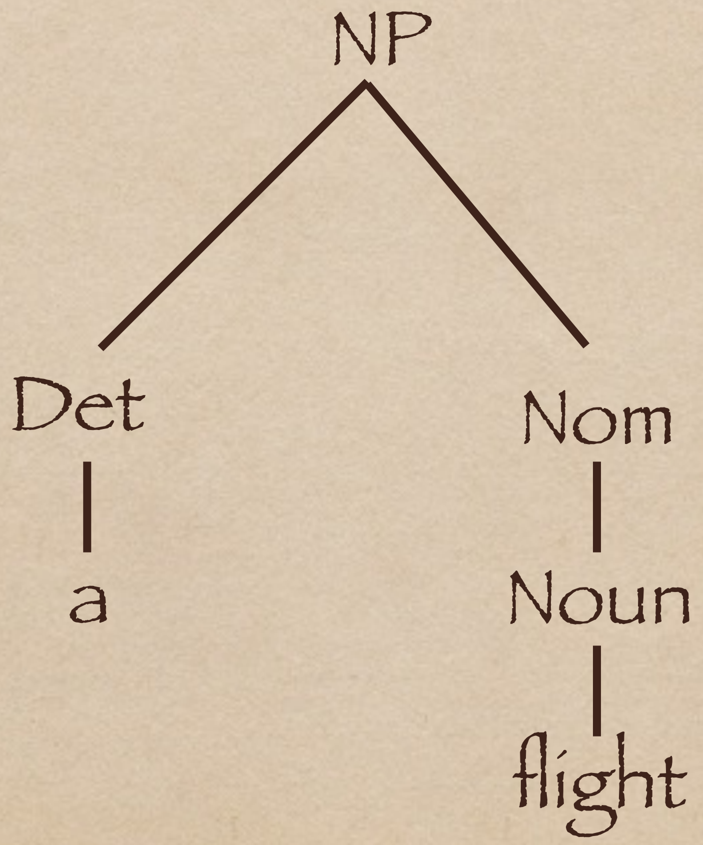
NP->Det Nom	rewrites NP to Det Nom
Nominal -> Noun	rewrites Det Nom to Det Noun
Noun -> flight	rewrites Det Noun to Det flight
Det -> a	rewrites Det flight to a flight

We say the string "a flight" can be derived from non-terminal NP. The sequence of rules is called a derivation.

Context Free Grammars

A derivation is commonly represented by a parse tree.

For example the tree of the previous derivation is:



NP

Det

Nom

a

Noun

flight

Context Free Grammars

A few more rules for the grammar of English

S -> NP VP
VP -> Verb NP
VP -> Verb NP PP
VP -> Verb PP
PP -> Preposition NP
Pronoun -> me| I| you| it

e.g.

I prefer a morning star
prefer a morning star
leave London at noon
Leave on Sunday

From London
on Wednesday
On July 16th

Context Free Grammars

Provide a parse tree for the sentence:

“I prefer a morning star.”

S -> NP VP

NP -> Det Nominal

NP -> ProperNoun

NP -> Pronoun

Nominal -> Noun

Nominal -> Nominal Noun

VP -> Verb NP

VP -> Verb NP PP

VP -> Verb PP

VP -> Verb

Det -> a | an | the | this | that

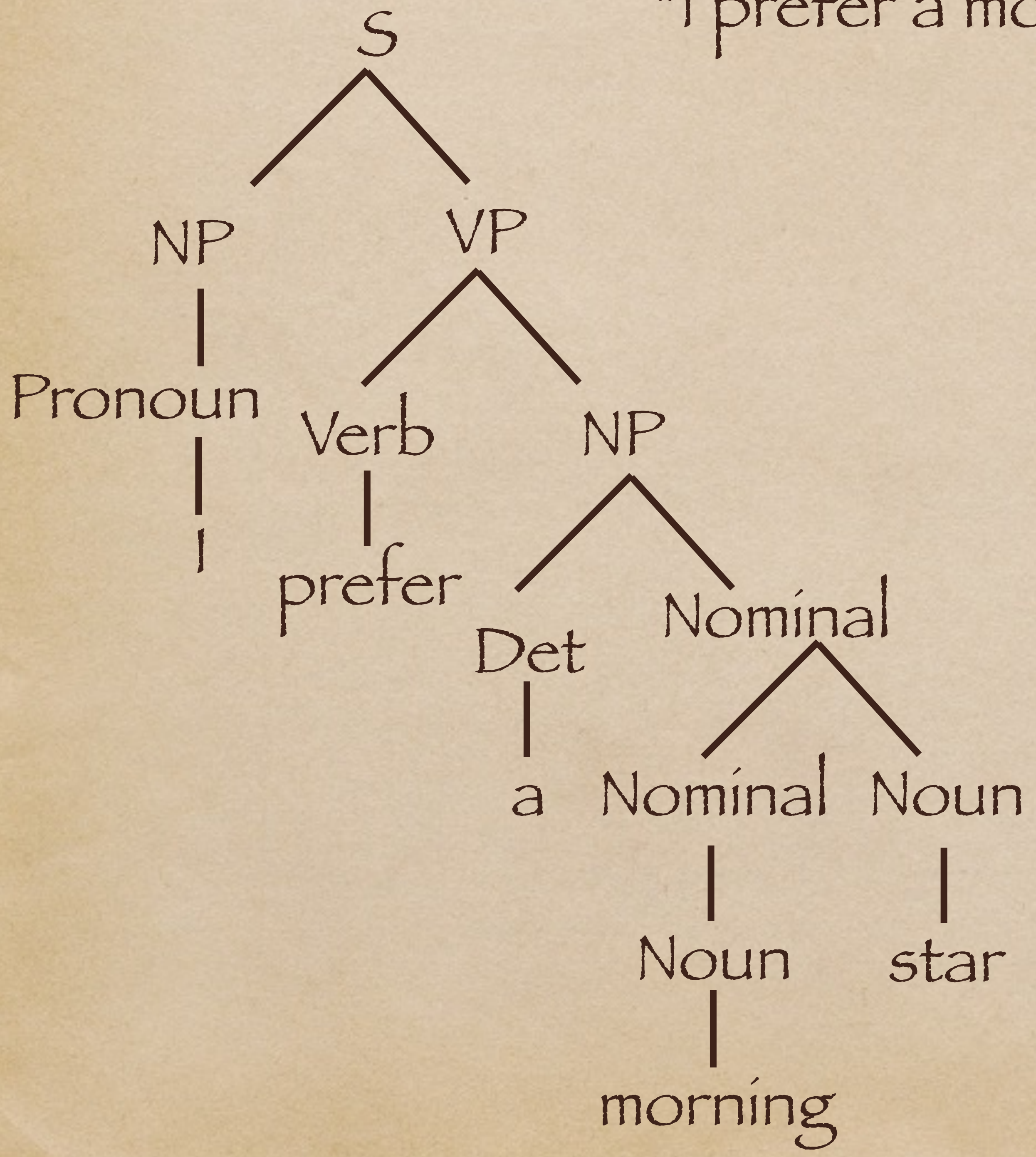
Noun -> flight | start | morning

Verb -> is | prefer | like | want

Pronoun -> me | I | you | it

Context Free Grammars

"I prefer a morning star."



S

NP

VP

Pronoun

Verb

NP

I

prefer

Det

Nominal

a

Nominal

Noun

Noun

star

morning

Context Free Grammars

Formal definition of a CFG:

$$(N, \Sigma, R, S)$$

N a set of non-terminal symbols

Σ a set of terminal symbols (disjoint from N)

S a designated start symbol

R a set of production rules of the form $\alpha \to \beta$

- α a non-terminal
- β a string os symbols from the strings $(\Sigma \cup N)^*$

Context Free Grammars

A language is defined through the concept of derivation. A string derives another if it an be rewritten as the second one by a series of rule applications.

If $A \to \beta$ is a production rule generating P and α and γ are any two strings in $(\Sigma \cup N)^*$, then we say:

$\alpha A \gamma$ directly derives $\alpha \beta \gamma$

This is more formally denoted by:

$$\alpha A \gamma \implies \alpha \beta \gamma$$

obtained by substituting A by β .

Context Free Grammars

A derivation is a generalisation of a direct derivation. If we have $\alpha_1 \Longrightarrow \alpha_2, \alpha_2 \Longrightarrow \alpha_3, \cdots, \alpha_{n-1} \Longrightarrow \alpha_n$ then, we say α_1 derives α_n and formally write $\alpha_1 \overset{*}{\Longrightarrow} \alpha_n$.

The language generated by a CFG is the set of strings composed of terminals that can be derived from the designated start symbol.

$$\mathcal{L}_{CFG} = \left\{ w \mid w \in \Sigma^* \quad \text{and} \quad S \overset{*}{\Longrightarrow} w \right\}$$

Parsing is the problem of mapping a string of words to its derivation.

Tree Bank

CFG's can in principle be used to assign a parse tree to any given sentence.

Given a corpus, we can "annotate" each of its sentences with a parse tree.

A corpus thus annotated is called a Tree Bank.

Tree Banks are widely used in empirical investigations of syntactic phenomena.

Tree Bank

How to build a Tree Bank:

1- Use an automatic parser

2- Use linguistics to hand -correct the parser

Example: Pen Tree Bank

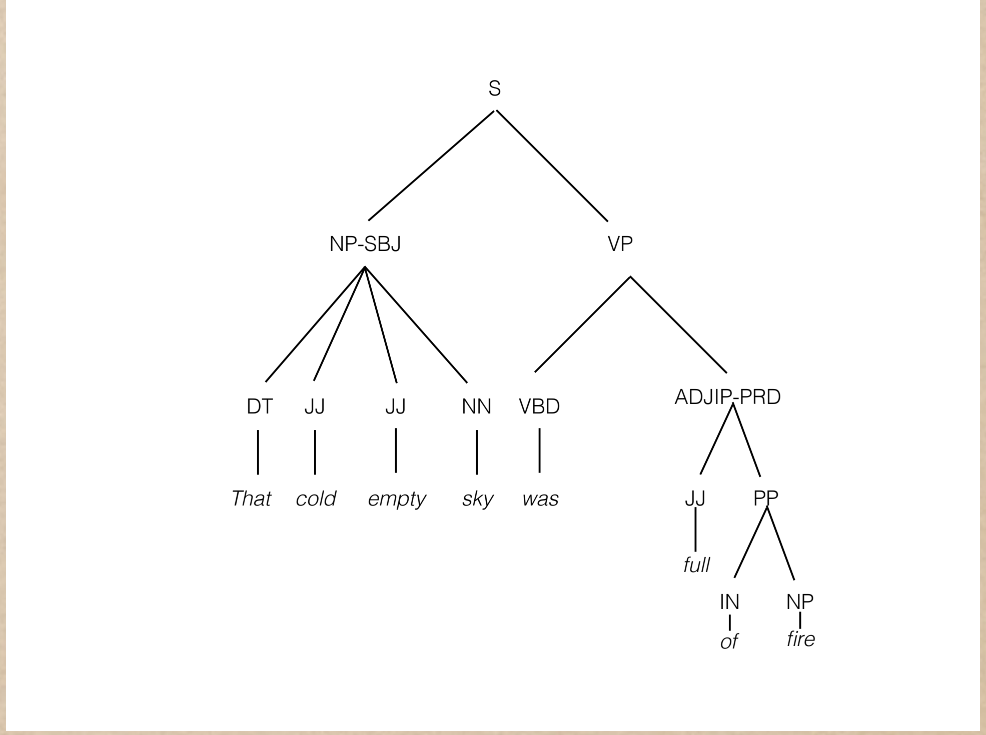
Produced from: Brown, ATIS, Wall Street Journal: English

Other languages such as Arabic and Chinese

A compact bracketed notation to denote the parse trees of a treebank

```
((S
  (NP-SBJ (DT The)
 (JJ long)  ( , , )
(JJ lonely)  (NN night))
(VP  (VBD is)
(ADJP-PRD  (JJ full)
(PP  (IN of)
(NP  (NN stars)
(CC and)
(NN moonlight) ))))
(. . )  ))
```

The long, lonely night is full of stars and moonlight.


S
NP-SBJ
VP
DT
JJ
JJ
NN
VBD
ADJIP-PRD
That
cold
empty
sky
was
JJ
PP
full
IN
NP
of
fire