

Topics in Natural Language Processing

Shay Cohen

Institute for Language, Cognition and Computation

University of Edinburgh

Lecture 5



Administrativa

- Information has been sent over the weekend – anything unclear?

Last class

Prior conjugacy:

The likelihood and the prior are conjugate if the posterior is from the prior family as well for any choice of prior from \mathcal{P} .

family \mathcal{P}

Maximum a Posteriori Estimation:

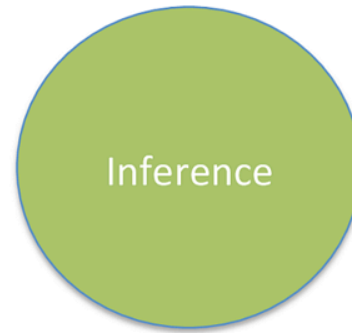
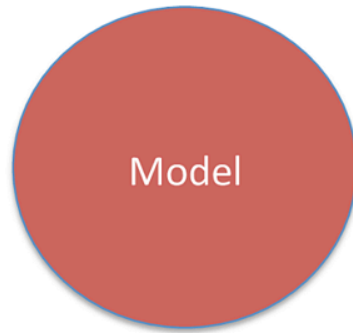
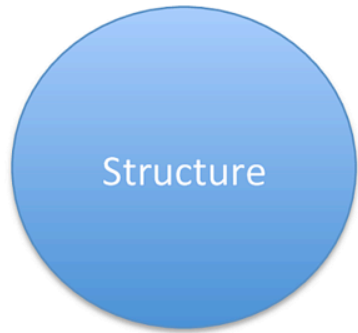
$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}} \underbrace{\log p(\theta)}_{\text{prior}} + \underbrace{\sum_{i=1}^n \log p(w_i | \theta)}_{\text{ll}}$$

Today's class

- Structure in NLP
- Grammars in NLP (models for structure)
 - Context-free grammars and their canonical form
 - How to add probabilities?
 - How to do inference with them?

Solving an NLP problem

When modelling a new problem in NLP, need to address four issues:



Today's class

What is our Ω ?

Today's class

What is our Ω ?

Examples:

- Finite sets of symbols (such as a set of words)
- Sequences
- Trees - “dependency” and others
- Graphs and hypergraphs
- Miscellaneous - tailored to a specific problem

Bag of words

$$\Omega = \{\text{documents}\} \quad d = (w:c)_{w \in V} \leftarrow \text{vocabulary}$$

- Does not have much structure
- Still, a very useful way to decompose the space of documents
- Especially when interested in “content” and not “syntax”
- We will re-visit this model later

Segmentation

$$\Omega = \{ \text{sentences} \} \quad \{ \text{paragraphs} \}$$

Useful for:

- Segmentation of languages such as Chinese
- Identifying co-locations (New York)
- Tokenisation
- Sentence segmentation (a “solved” problem)
- Morphological segmentation (for example, Turkish)

Sequence labelling

$$\Omega = \{ (s, t) \}$$

When is it useful?

- Part-of-speech tagging
 - POS tagging using majority vote: 90%
 - POS tagging using sequence labelling: 97%
- Whenever context is needed to decipher an observation

Chunking

$\Omega =$

When is it useful?

- Shallow parsing (or as a precursor to full parsing)
- Identifying named entities
- Connection to sequence labelling

B - I - O

B - P

I - P

O

B - L

President

Obama

visited

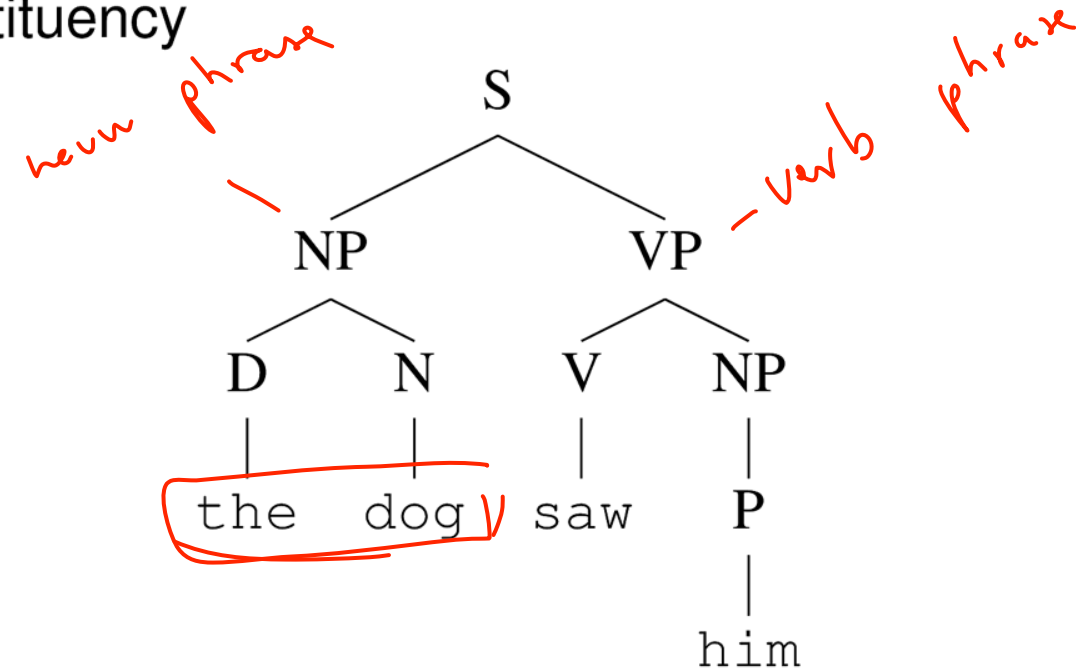
Scotland

Parsing

$$\Omega = \{ (s, t) \}$$

Two main types of parsing structures:

- Constituency

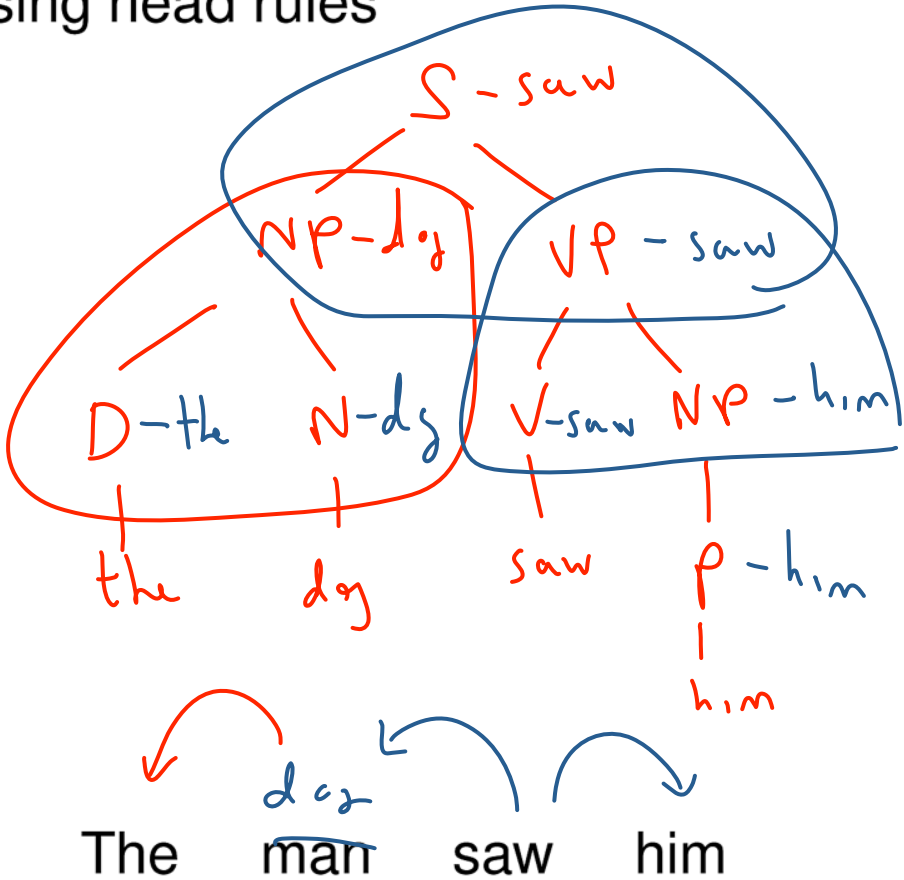


- Dependency



Conversion of dependency to constituency bracketing

Can be done by using head rules



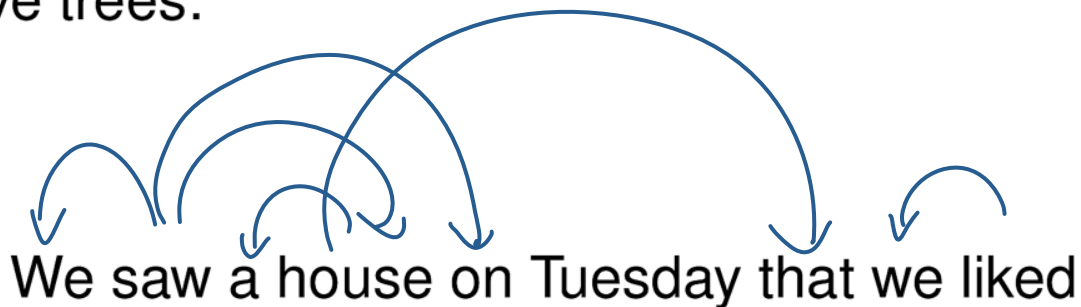
Projective vs. non-projective parsing

Projective trees:

If you draw all edges above the tree, they
are never going to cross.

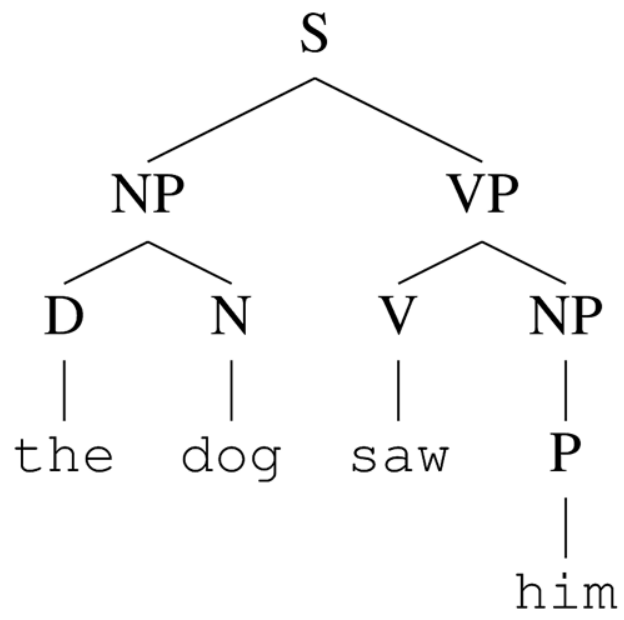
Non-projective trees:

We saw a house on Tuesday that we liked

A diagram illustrating non-projective parsing. The sentence "We saw a house on Tuesday that we liked" is written in black text. Above the text, several blue arcs represent dependencies between words. The arcs are: a small arc from "We" to "saw"; a large arc from "saw" to "liked"; a small arc from "a" to "house"; a small arc from "on" to "Tuesday"; a large arc from "Tuesday" to "liked"; and a small arc from "that" to "we". The arcs from "saw" to "liked" and "Tuesday" to "liked" cross each other, demonstrating that the dependency structure is non-projective.

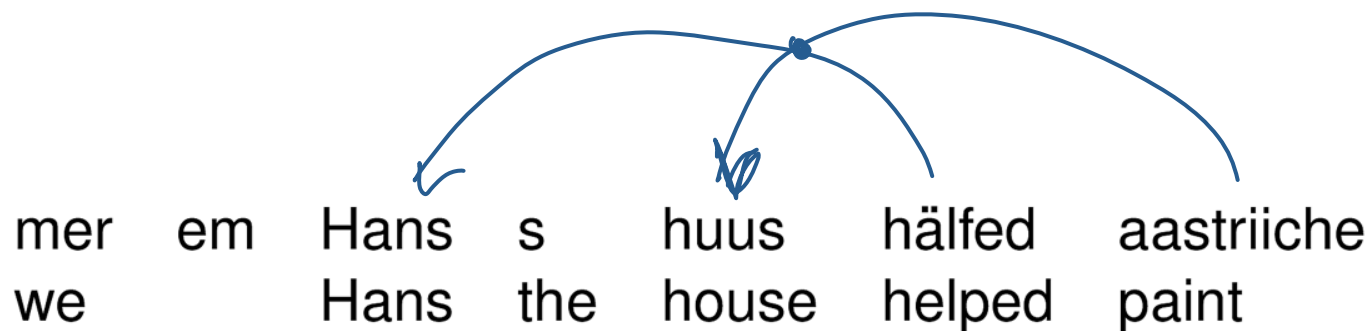
Any constituent tree induces a projective tree

Why do constituent trees induce only projective trees?



Non-projectivity

Not so common in English. Very common in languages such as Dutch and Swiss-German:



Sentence in English:

We helped Hans paint the house

(Called “cross-serial dependencies”)

Clustering

$$\Omega = \{ (\text{set of elements, partition}) \}$$

Useful for:

- Word clustering
- Coreference resolution
 - Algorithms for such resolution usually use specialised methods
 - Enforcing transitivity

Alignments

$\Omega = \{ (\text{pair of sentences, alignment between the words}) \}$

Most useful for machine translation

mer	em	Hans	s	huus	hähfed	aastrische
we	helped	Hans	paint	the	house	

Useful for extracting phrases (phrase-based translation)

Most common models for alignment were developed by IBM (“IBM models 1–4”)

Modelling new problems

How do we represent our space of inputs/outputs?

- What is the space used for?
- What are the available inference algorithms?
- Is there a linguistic theory that can help?

Natural language representations

Representations are a moving target

- Linguistic theories develop
- Changes in data and domains
- New algorithms develop, make hard representations feasible

Grammars

What is a grammar?

Grammars

What is a grammar?

A system of rules that govern the production of a language

What is a formal grammar?

Grammars

What is a grammar?

A system of rules that govern the production of a language

What is a formal grammar?

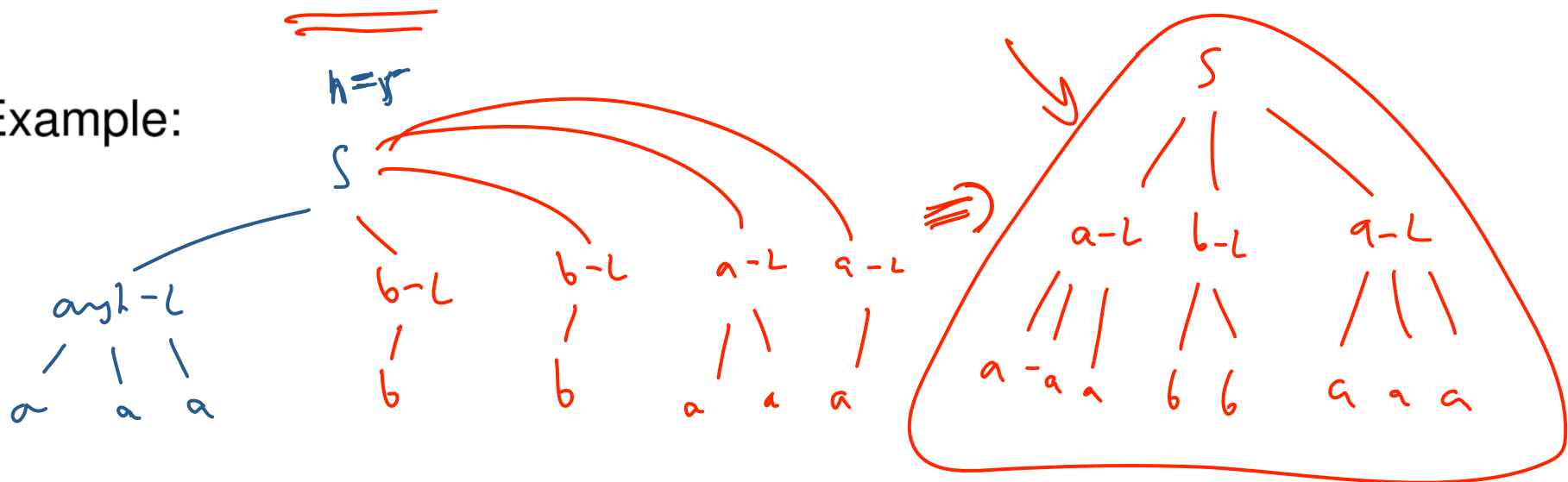
A formal system of rules that govern the production of a language

Usually describes a step-by-step process

A pre-historic grammar

- Choose a number n
- Generate n labels that can be “argh-phrase” or “blah-phrase”
- Each argh-phrase and blah-phrase generate 1-3 “argh” or “blah”
- Optional: merge identically-labelled consecutive phrases

Example:



Components in a formal grammar

Language:

any sequence of a's and b's

Structure:

Derivations:

Context-free grammars

What is a CFG?

Nonterminals N

Terminals T

Production

$A \rightarrow \alpha$

$A \in N$

$\alpha \in (N \cup T)^*$

Context-free grammars

What is a CFG?

- A set of nonterminals N
- A set of vocabulary terminals V
- A start symbol $S \in N$
- A set of production rules R , $a \rightarrow \alpha$ is such that $a \in N$ and $\alpha \in (V \cup N)^*$

CFGs: basic terminology

How to represent a partial derivation?

$S \Rightarrow \underline{NP} \quad VP \Rightarrow \underline{DT} \quad \underline{NN} \quad VP \Rightarrow$

$\Rightarrow \underline{the} \quad \underline{NN} \quad VP \Rightarrow the \quad cat \quad VP \Rightarrow$

$\Rightarrow the \quad cat \quad VB \quad NP \Rightarrow the \quad cat \quad chased \quad NP \Rightarrow$

$\Rightarrow the \quad cat \quad chased$

~~$DT \quad NN$~~ the dog

Two relations: \Rightarrow and $\boxed{\Rightarrow^*}$

$S \Rightarrow^* the \quad NN \quad VP$

$S \Rightarrow^* the \quad cat \quad chased \quad NP$