Speech and Language Processing

Constituency Grammars

Chapter 11

Review

- POS Decoding
  - What does this mean?
  - What representation do we use?
  - What algorithm do we use, and why?
- Fish sleep example
  - Posted on syllabus
  - Next page

Hidden Markov Models

- States $Q = q_1, q_2 \ldots q_{N;}$
- Observations $O= o_1, o_2 \ldots o_{N;}$
  - Each observation is a symbol from a vocabulary V = $\{v_1, v_2, \ldots v_V\}$
- Transition probabilities
  - Transition probability matrix $A = \{a_{ij}\}$

    $$a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \le i, j \le N$$
- Observation likelihoods
  - Output probability matrix $B=\{b_i(k)\}$

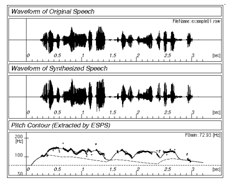
    $$b_i(k) = P(X_t = o_k \mid q_t = i)$$

    $$\pi_i = P(q_1 = i) \quad 1 \le i \le N$$
- Special initial probability vector π

Today

- Formal Grammars
  - Context-free grammar
  - Grammars for English
  - Treebanks
  - Dependency grammars

Simple View of Linguistic Analysis

Phonology  ⇔ /waddyasai/

Morphology /waddyasai/ ⇔ what did you say

Syntax what did you say ⇔ say (subj: you, obj: what)

Semantics say (subj: you, obj: what) ⇔ P[λx. say(you, x)]

Syntax

- Grammars (and parsing) are key components in many applications
  - Grammar checkers
  - Dialogue management
  - Question answering
  - Information extraction
  - Machine translation

Syntax

- Key notions that we'll cover
  - Constituency
  - Grammatical relations and Dependency
    - Heads
- Key formalism
  - Context-free grammars
- Resources
  - Treebanks

Types of Linguistic Theories

- Prescriptive theories: how people ***ought*** to talk
- Descriptive theories: how people ***actually*** talk
  - Most appropriate for NLP applications

Constituency

- The basic idea here is that groups of words within utterances can be shown to act as single units.
- And in a given language, these units form coherent classes that can be be shown to behave in similar ways
  - With respect to their internal structure
  - And with respect to other units in the language

Constituency

- Internal structure
  - We can describe an internal structure to the class (might have to use disjunctions of somewhat unlike sub-classes to do this).
- External behavior
  - For example, we can say that noun phrases can come before verbs

Constituency

- For example, it makes sense to the say that the following are all ***noun phrases*** in English...

Harry the Horse	a high-class spot such as Mindy's
the Broadway coppers	the reason he comes into the Hot Box
they	three parties from Brooklyn

- Why? One piece of evidence is that they can all precede verbs.
  - This is external evidence

Grammars and Constituency

- Of course, there's nothing easy or obvious about how we come up with right set of constituents and the rules that govern how they combine...
- That's why there are so many different theories of grammar and competing analyses of the same data.
- The approach to grammar, and the analyses, adopted here are very generic (and don't correspond to any modern linguistic theory of grammar).

Context-Free Grammars

- Context-free grammars (CFGs)
  - Also known as
    - Phrase structure grammars
    - Backus-Naur form
- Consist of
  - Rules
  - Terminals
  - Non-terminals

Context-Free Grammars

- Terminals
  - We'll take these to be words (for now)
- Non-Terminals
  - The constituents in a language
    - Like noun phrase, verb phrase and sentence
- Rules
  - Rules are equations that consist of a single non-terminal on the left and any number of terminals and non-terminals on the right.

Some NP Rules

- Here are some rules for our noun phrases

$$
\begin{aligned}
NP &\rightarrow Det\ Nominal \\
NP &\rightarrow ProperNoun \\
Nominal &\rightarrow Noun \mid Nominal\ Noun
\end{aligned}
$$

- Together, these describe two kinds of NPs.
  - One that consists of a determiner followed by a nominal
  - And another that says that proper names are NPs.
  - The third rule illustrates two things
    - An explicit disjunction
      - Two kinds of nominals
    - A recursive definition
      - Same non-terminal on the right and left-side of the rule

L0 Grammar

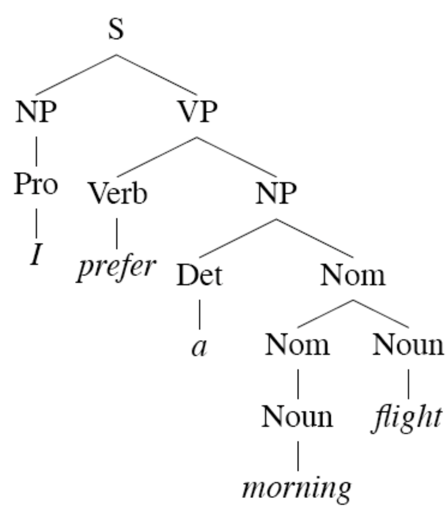
Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles

Generativity

- As with n-grams, you can view these rules as either analysis or synthesis machines
  - Generate strings in the language
  - Reject strings not in the language
  - Impose structures (trees) on strings in the language

Derivations

- A derivation is a sequence of rules applied to a string that *accounts* for that string
  - Covers all the elements in the string
  - Covers only the elements in the string



Definition

- More formally, a CFG consists of

N	a set of **non-terminal symbols** (or **variables**)
Σ	a set of **terminal symbols** (disjoint from N)
R	a set of **rules** or productions, each of the form $A \rightarrow \beta$,
	where A is a non-terminal,
	β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$
S	a designated **start symbol**

Parsing

- Parsing is the process of taking a string and a grammar and returning a (multiple?) parse tree(s) for that string
  - There are languages we can capture with CFGs that we can't capture with regular expressions
  - There are properties that we can capture that we can't capture with n-grams