Syntax: Context-free Grammars

Ling 571
Deep Processing Techniques for NLP
January 5, 2011

Roadmap

- Motivation: Applications
- Context-free grammars (CFGs)
  - Formalism
  - Grammars for English
  - Treebanks and CFGs
  - Speech and Text

Applications

- Shallow techniques useful, but limited
- Deeper analysis supports:
  - Grammar-checking – and teaching
  - Question-answering
  - Information extraction
  - Dialogue understanding

Grammar and NLP

- Grammar in NLP is NOT prescriptive high school grammar
  - Explicit rules
  - Split infinitives, etc

- Grammar in NLP tries to capture structural knowledge of language of a native speaker
  - Largely implicit
  - Learned early, naturally

Representing Syntax

- Context-free grammars
- CFGs: 4-tuple
  - A set of terminal symbols: Σ
  - A set of non-terminal symbols: N
  - A set of productions P: of the form A -> α
    - Where A is a non-terminal and α in $(\Sigma \cup N)^*$
  - A designated start symbol S

CFG Components

- Terminals:
  - Only appear as leaves of parse tree
  - Right-hand side of productions (rules) (RHS)
  - Words of the language
    - Cat, dog, is, the, bark, chase
- Non-terminals
  - Do not appear as leaves of parse tree
  - Appear on left or right side of productions (rules)
  - Constituents of language
    - NP, VP, Sentence, etc

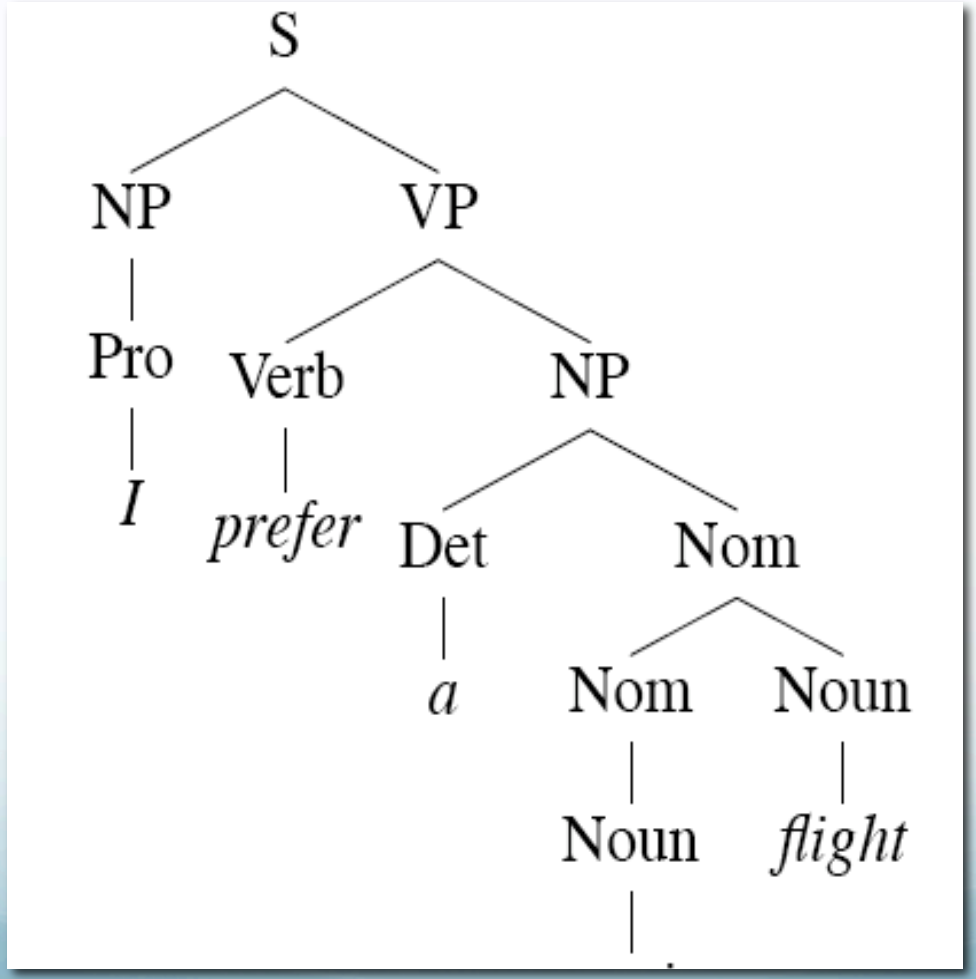
CFG Components

- Productions
  - Rules with one non-terminal on LHS and any number of terminals and non-terminals on RHS

  - S -> NP VP
  - VP -> V NP PP | V NP
  - Nominal -> Noun | Nominal Noun
  - Noun -> dog | cat | rat
  - Det -> the

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

Parse Tree



Parsing Goals

Parsing Goals

- Accepting:
  - Legal string in language?
    - Formally: rigid

Parsing Goals

- Accepting:
  - Legal string in language?
    - Formally: rigid
    - Practically: degrees of acceptability

Parsing Goals

- Accepting:
  - Legal string in language?
    - Formally: rigid
    - Practically: degrees of acceptability
- Analysis
  - What structure produced the string?
  - What sequence of rule applications derives this string

Parsing Goals

- Accepting:
  - Legal string in language?
    - Formally: rigid
    - Practically: degrees of acceptability
- Analysis
  - What structure produced the string?
  - What sequence of rule applications derives this string
    - Produce one (or all) parse trees for the string

Parsing Goals

- Accepting:
  - Legal string in language?
    - Formally: rigid
    - Practically: degrees of acceptability
- Analysis
  - What structure produced the string?
  - What sequence of rule applications derives this string
    - Produce one (or all) parse trees for the string
- Generation
  - Given a grammar, produce all legal strings of language

Some English Grammar

- Sentences:

Some English Grammar

- Sentences:
  - Declarative: S -> NP VP
    - I want a flight from Ontario to Chicago

Some English Grammar

- Sentences:
  - Declarative: S -> NP VP
    - I want a flight from Ontario to Chicago
  - Imperative: S -> VP
    - Show me the cheapest fare.

Some English Grammar

- Sentences:
  - Declarative: S -> NP VP
    - I want a flight from Ontario to Chicago
  - Imperative: S -> VP
    - Show me the cheapest fare.
  - S -> Aux NP VP
    - Can you give me the same information for United?

Some English Grammar

- Sentences:
  - Declarative: S -> NP VP
    - I want a flight from Ontario to Chicago
  - Imperative: S -> VP
    - Show me the cheapest fare.
  - S -> Aux NP VP
    - Can you give me the same information for United?
  - S -> Wh-NP VP
    - What airlines fly from Burbank to Denver?

Some English Grammar

- Sentences: Full sentence or clause; a complete thought
  - Declarative: S -> NP VP
    - I want a flight from Ontario to Chicago
  - Imperative: S -> VP
    - Show me the cheapest fare.
  - S -> Aux NP VP
    - Can you give me the same information for United?
  - S -> Wh-NP VP
    - What airlines fly from Burbank to Denver?
  - S -> Wh-NP Aux NP VP
    - What flights do you have from Chicago to Baltimore?

The Noun Phrase

The Noun Phrase

- NP -> Pronoun | Proper Noun (NNP) | Det Nominal
  - Head noun + pre-/post-modifiers
  - It , Flight 852,...

The Noun Phrase

- NP -> Pronoun | Proper Noun (NNP) | Det Nominal
  - Head noun + pre-/post-modifiers
- Determiners:

The Noun Phrase

- NP -> Pronoun | Proper Noun (NNP) | Det Nominal
  - Head noun + pre-/post-modifiers
- Determiners:
  - Det -> DT
    - the, this, a, those

The Noun Phrase

- NP -> Pronoun | Proper Noun (NNP) | Det Nominal
  - Head noun + pre-/post-modifiers
- Determiners:
  - Det -> DT
    - the, this, a, those
  - Det -> NP ‘s
    - United’s flight, Chicago’s airport

In and around the Noun

- Nominal -> Noun
  - PTB POS: NN, NNS, NNP, NNPS
  - flight, dinner, airport

In and around the Noun

- Nominal -> Noun
  - PTB POS: NN, NNS, NNP, NNPS
  - flight, dinner, airport

- NP -> (Det) (Card) (Ord) (Quant) (AP) Nominal
  - The least expensive fare, one flight, the first route

In and around the Noun

- Nominal -> Noun
  - PTB POS: NN, NNS, NNP, NNPS
  - flight, dinner, airport
- NP -> (Det) (Card) (Ord) (Quant) (AP) Nominal
  - The least expensive fare, one flight, the first route
- Nominal -> Nominal PP
  - The flight from Chicago

Verb Phrase and Subcategorization

- Verb phrase includes Verb, other constituents
  - Subcategorization frame: what constituent arguments the verb requires

Verb Phrase and Subcategorization

- Verb phrase includes Verb, other constituents
  - Subcategorization frame: what constituent arguments the verb requires
  - VP -> Verb disappear

Verb Phrase and Subcategorization

- Verb phrase includes Verb, other constituents
  - Subcategorization frame: what constituent arguments the verb requires

  - VP -> Verb disappear
  - VP -> Verb NP book a flight

Verb Phrase and Subcategorization

- Verb phrase includes Verb, other constituents
  - Subcategorization frame: what constituent arguments the verb requires

VP -> Verb	disappear
VP -> Verb NP	book a flight
VP -> Verb PP PP	fly from Chicago to Seattle

Verb Phrase and Subcategorization

- Verb phrase includes Verb, other constituents
  - Subcategorization frame: what constituent arguments the verb requires

• VP -> Verb	disappear
• VP -> Verb NP	book a flight
• VP -> Verb PP PP	fly from Chicago to Seattle
• VP -> Verb S	I think I want that flight

Verb Phrase and Subcategorization

- Verb phrase includes Verb, other constituents
  - Subcategorization frame: what constituent arguments the verb requires

VP -> Verb	disappear
VP -> Verb NP	book a flight
VP -> Verb PP PP	fly from Chicago to Seattle
VP -> Verb S	I think I want that flight
VP -> Verb VP	I want to arrange three flights

CFGs and Subcategorization

- Issues?

CFGs and Subcategorization

- Issues?
  - I prefer United has a flight.

CFGs and Subcategorization

- Issues?
  - I prefer United has a flight.
- How can we solve this problem?

CFGs and Subcategorization

- Issues?
  - I prefer United has a flight.

- How can we solve this problem?
  - Create explict subclasses of verb
    - Verb-with-NP
    - Verb-with-S-complement, etc...

CFGs and Subcategorization

- Issues?
  - I prefer United has a flight.
- How can we solve this problem?
  - Create explict subclasses of verb
    - Verb-with-NP
    - Verb-with-S-complement, etc...
- Is this a good solution?

CFGs and Subcategorization

- Issues?
  - I prefer United has a flight.
- How can we solve this problem?
  - Create explict subclasses of verb
    - Verb-with-NP
    - Verb-with-S-complement, etc...
- Is this a good solution?
  - No, explosive increase in number of rules
  - Similar problem with agreement

Treebanks

- Treebank:
  - Large corpus of sentences all of which are annotated syntactically with a parse
    - Built semi-automatically
      - Automatic parse with manual correction
  - Examples:
    - Penn Treebank (largest)
      - English: Brown (balanced); Switchboard (conversational speech); ATIS (human-computer dialogue); Wall Street Journal; Chinese; Arabic
    - Korean

Treebanks

- Include wealth of language information
  - Traces, grammatical function (subject, topic, etc), semantic function (temporal, location)
- Implicitly constitutes grammar of language
  - Can read off rewrite rules from bracketing
  - Not only presence of rules, but frequency
  - Will crucial in building statistical parsers

Treebank WSJ Example

```
( (S (`` ``)
    (S-TPC-2
      (NP-SBJ-1 (PRP We) )
      (VP (MD would)
        (VP (VB have)
          (S
            (NP-SBJ (-NONE- *-1) )
            (VP (TO to)
              (VP (VB wait)
                (SBAR-TMP (IN until)
                  (S
                    (NP-SBJ (PRP we) )
                    (VP (VBP have)
                      (VP (VBN collected)
                        (PP-CLR (IN on)
                          (NP (DT those)(NNS assets)))))))))))))
    (, ,) (`` ``)
    (NP-SBJ (PRP he) )
    (VP (VBD said)
      (S (-NONE- *T*-2) ))
    (. .) ))
```

Treebanks & Corpora

- Many corpora on patas
- patas$ ls /corpora
  - birkbeck enron_email_dataset grammars LEAP TREC
  - Coconut europarl ICAME med-data treebanks
  - Conll europarl-old JRC-Acquis.3.0 nltk
  - DUC framenet LDC proj-gutenberg
- Many large corpora from LDC
- Many corpus samples in nltk

Treebank Issues

Treebank Issues

- Large, expensive to produce

Treebank Issues

- Large, expensive to produce
- Complex
  - Agreement among labelers can be an issue

Treebank Issues

- Large, expensive to produce
- Complex
  - Agreement among labelers can be an issue
- Labeling implicitly captures theoretical bias
  - Penn Treebank is ‘bushy’, long productions

Treebank Issues

- Large, expensive to produce
- Complex
  - Agreement among labelers can be an issue
- Labeling implicitly captures theoretical bias
  - Penn Treebank is ‘bushy’, long productions
- Enormous numbers of rules
  - 4,500 rules in PTB for VP
    - VP-> V PP PP PP
  - 1M rule tokens; 17,500 distinct types – and counting!

Spoken & Written

- Can we just use models for written language directly?

Spoken & Written

- Can we just use models for written language directly?
- No!

Spoken & Written

- Can we just use models for written language directly?
- No!
- Challenges of spoken language
  - Disfluency
    - Can I um uh can I g- get a flight to Boston on the 15th?
      - 37% of Switchboard utts > 2 wds

Spoken & Written

- Can we just use models for written language directly?
- No!
- Challenges of spoken language
  - Disfluency
    - Can I um uh can I g- get a flight to Boston on the 15th?
      - 37% of Switchboard utts > 2 wds
  - Short, fragmentary
    - Uh one way

Spoken & Written

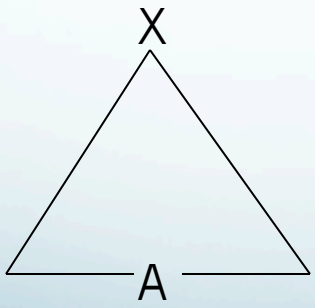
- Can we just use models for written language directly?
- No!
- Challenges of spoken language
  - Disfluency
    - Can I um uh can I g- get a flight to Boston on the 15th?
      - 37% of Switchboard utts > 2 wds
  - Short, fragmentary
    - Uh one way
  - More pronouns, ellipsis
    - That one

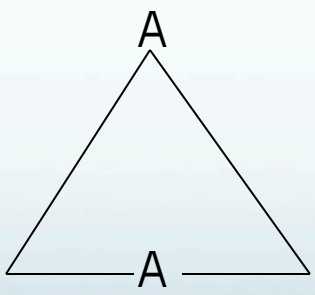
Grammar Equivalence and Form

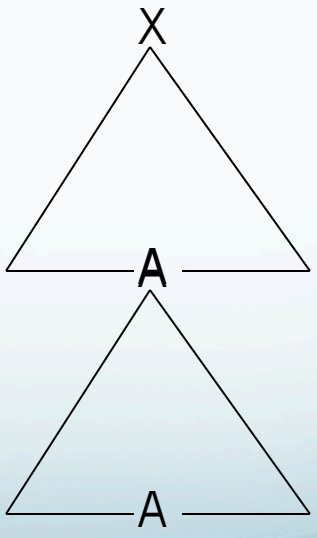
- Grammar equivalence
  - Weak: Accept the same language, May produce different analyses
  - Strong: Accept same language, Produce same structure
- Canonical form:
  - Chomsky Normal Form (CNF)
    - All CFGs have a weakly equivalent CNF
    - All productions of the form:
      - A-> B C where B,C in N, or
      - A->a where a in Σ

Tree Adjoining Grammars

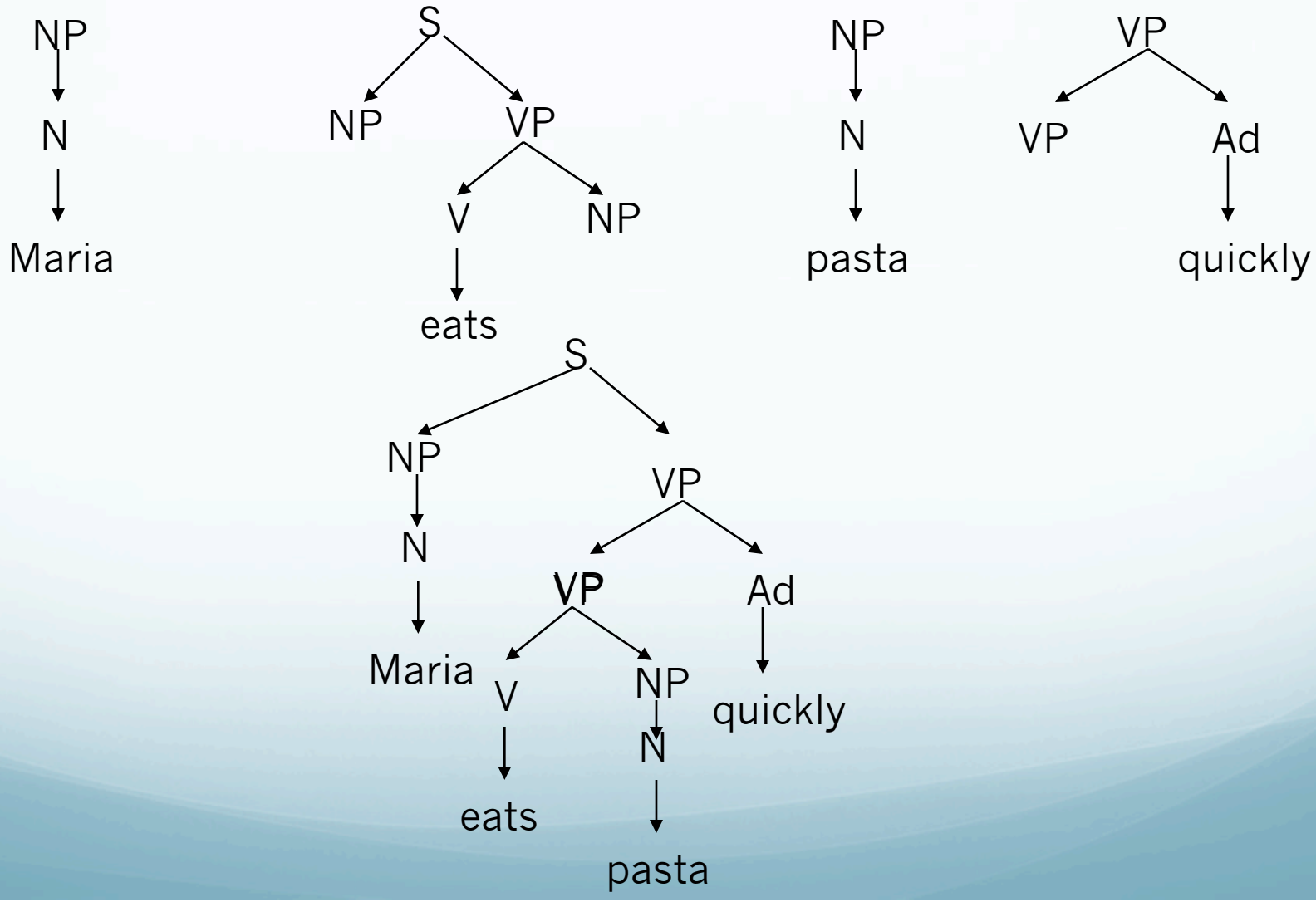
- Mildly context-sensitive (Joshi, 1979)
  - Motivation:
    - Enables representation of crossing dependencies
- Operations for rewriting
  - "Substitution" and "Adjunction"







TAG Example


NP
N
Maria
S
NP
VP
V
NP
eats
NP
N
pasta
VP
VP
Ad
quickly
S
NP
N
Maria
VP
VP
Ad
V
NP
N
eats
pasta
quickly

Computational Parsing

- Given a grammar, how can we derive the analysis of an input sentence?
  - Parsing as search
  - CKY parsing
  - Earley parsing

- Given a body of (annotated) text, how can we derive the grammar rules of a language, and employ them in automatic parsing?

  - Treebanks & PCFGS