LING 520: Computational Analysis of English

Semester: FALL '16

Instructor: Sowmya Vajjala

Iowa State University, USA

27 October 2016

Class Outline

- Reminder: Assignment 4 submission due on Saturday
- Comment: The dictionary saving example I showed - is just one simple way of doing it. There are several other ways (e.g., using numpy library)
- Today's topics:
  - Natural language parsing: an introduction
  - NLTK exercises

Natural Language Parsing - Introduction

What is parsing?

- In the context of NLP, parsing refers to converting a sentence into some form of syntactic structure.
- syntactic structure includes: grammatical relations (subject-object etc), grouping constituents together (NP, VP etc), governor-dependent relationships etc.

Why parse at all?

- Can you think of some reasons to parse the syntactic structure? Where is it useful?

Why parse at all?

- Can you think of some reasons to parse the syntactic structure? Where is it useful?
- Some imaginary examples where parsing the syntax is useful:
  1. You may want to study how certain grammatical structures are used in applied linguistics vs biology research articles
  2. or how Arabic L1 people write English compared to Chinese L1, but beyond simple vocabulary based patterns.

Why parse at all?

- Can you think of some reasons to parse the syntactic structure? Where is it useful?
- Some imaginary examples where parsing the syntax is useful:
  1. You may want to study how certain grammatical structures are used in applied linguistics vs biology research articles
  2. or how Arabic L1 people write English compared to Chinese L1, but beyond simple vocabulary based patterns.
  3. You may want to design a system to understand the meaning of a sentence (automatically). Why?
  4. You may want to extract 'who did what to whom' kind of information automatically from a novel.

Why parse at all?

- Can you think of some reasons to parse the syntactic structure? Where is it useful?
- Some imaginary examples where parsing the syntax is useful:
  1. You may want to study how certain grammatical structures are used in applied linguistics vs biology research articles
  2. or how Arabic L1 people write English compared to Chinese L1, but beyond simple vocabulary based patterns.
  3. You may want to design a system to understand the meaning of a sentence (automatically). Why?
  4. You may want to extract 'who did what to whom' kind of information automatically from a novel.
  5. Although ngrams and POS tags are relatively straight forward to extract, they cannot give you this information.

How to parse?

- Clearly, we need POS tags, but also something beyond them get this kind of information.
- There are two main ways to show syntactic structure in NLP: Constituency structure and Dependency structure.
- Constituency structure shows the sentence as relations between its constituents (phrases), based on a pre-defined grammar.
- Dependency structure shows modifier-modified, argument relationships etc. between words.

Examples of parsed sentences

Please enter a sentence to be parsed:

My dog also likes eating sausage.

Language: English Sample Sentence Parse

Your query

My dog also likes eating sausage.

Tagging

```
My/PRP$  dog/NN  also/RB  likes/VBZ  eating/VBG  sausage/NN  ./.
```

Parse

```
(ROOT
  (S
    (NP (PRP$ My) (NN dog))
    (ADVP (RB also))
    (VP (VBZ likes)
      (S
        (VP (VBG eating)
          (NP (NN sausage)))))
    (. .)))
```

Universal dependencies

```
nmod:poss(dog-2, My-1)
nsubj(likes-4, dog-2)
advmod(likes-4, also-3)
root(ROOT-0, likes-4)
xcomp(likes-4, eating-5)
dobj(eating-5, sausage-6)
```

Constituency Structure: Context Free Grammars

- CFG is a mathematical way to model constituent structure in a language.
- A CFG consists of:
  1. a set of symbols in the language (lexicon)
  2. a set of rules or productions which describe how symbols in that language can be grouped together

Constituency Structure: Context Free Grammars

- CFG is a mathematical way to model constituent structure in a language.
- A CFG consists of:
  1. a set of symbols in the language (lexicon)
  2. a set of rules or productions which describe how symbols in that language can be grouped together
- Example symbols: NP, VP, PP, DT, NN, the, car etc.
- Example rules: S $->$ NP VP; $->$ DT N; Noun $->$ car etc.
- Symbols in CFGs are of two types: terminals (words) and non-terminals (NP, N etc)

Context Free Grammars - Purpose

1. To assign structure to a given sentence
2. To generate sentences automatically

CFG Parsing in NLP: brief history

- Compile a grammar (rules), have a lexicon, and use both to derive parses for sentences.
- Problem: low coverage (all constructions cannot be covered), combinations of rules can create several parses - what is the best parse?
- 90s: growth of annotated data such as Penn Tree Bank, which resulted in the development of statistical parsers.

Constituency structure - Example

- Let us say this is my grammar and lexicon together:
  1. S $->$ NP VP (Sentence contains Noun Phrase, Verb Phrase)
  2. VP $->$ V NP | V NP PP | V S | V ADJ (| is or like in regex)
  3. PP $->$ P NP
  4. V $->$ "saw" | "ate" | "walked"
  5. NP $->$ "John" | "Mary" | "Bob" | Det N | Det N PP | PropN
  6. Det -- $>$ "a" | "an" |"the" | "my"
  7. N $->$ "man" | "dog" | "cat" | "telescope" | "park"
  8. P $->$ "in" | "on" | "by" | "with"
- What will the constituency structure (or phrase structure tree) look like for the sentence: "John saw a man with the telescope"?

Recursion in CFGs

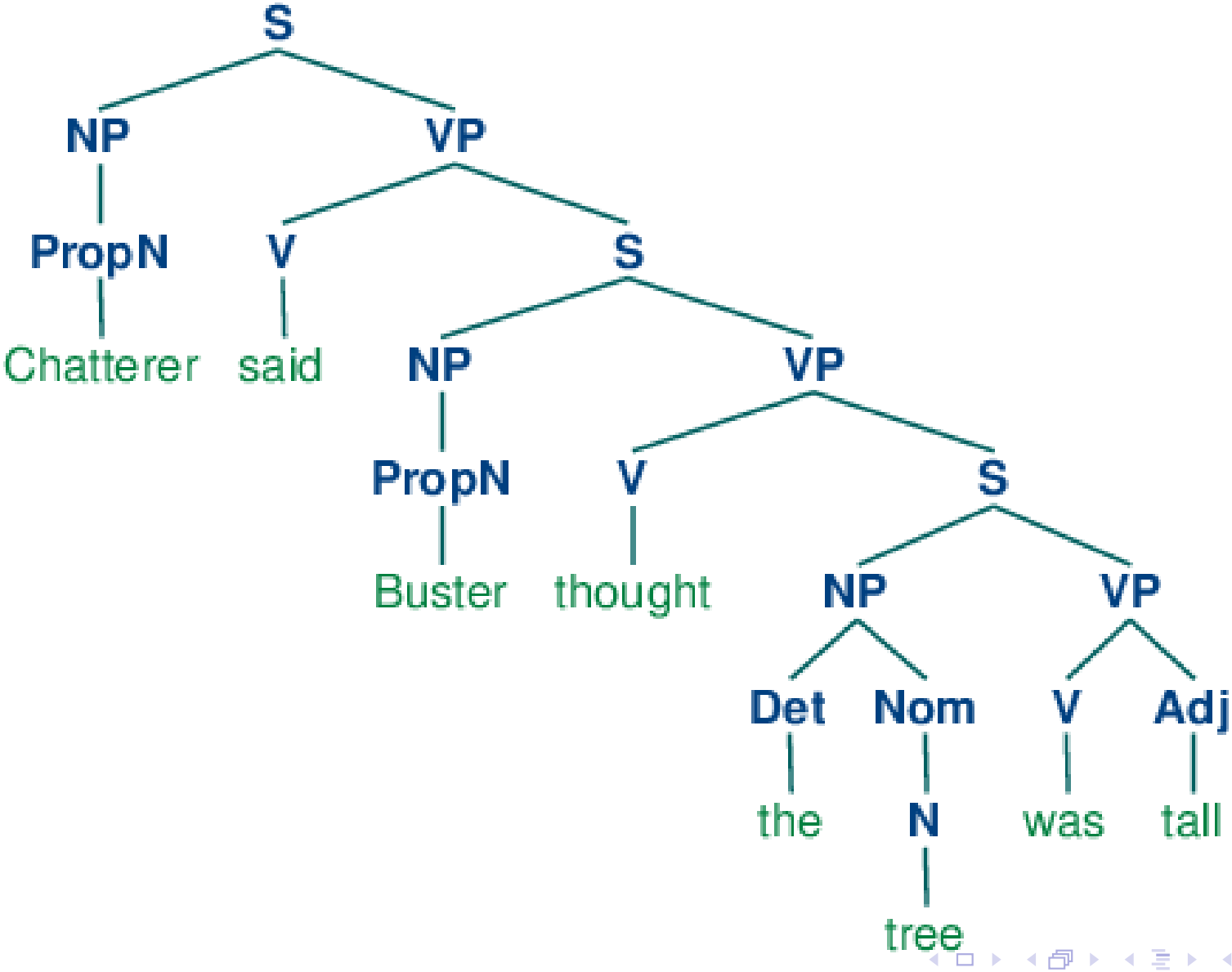
Source: NLTK book, Chapter 8

Note: CFGs can be recursive (What does that mean??)

Recursion in CFGs

Source: NLTK book, Chapter 8

Note: CFGs can be recursive (What does that mean??)



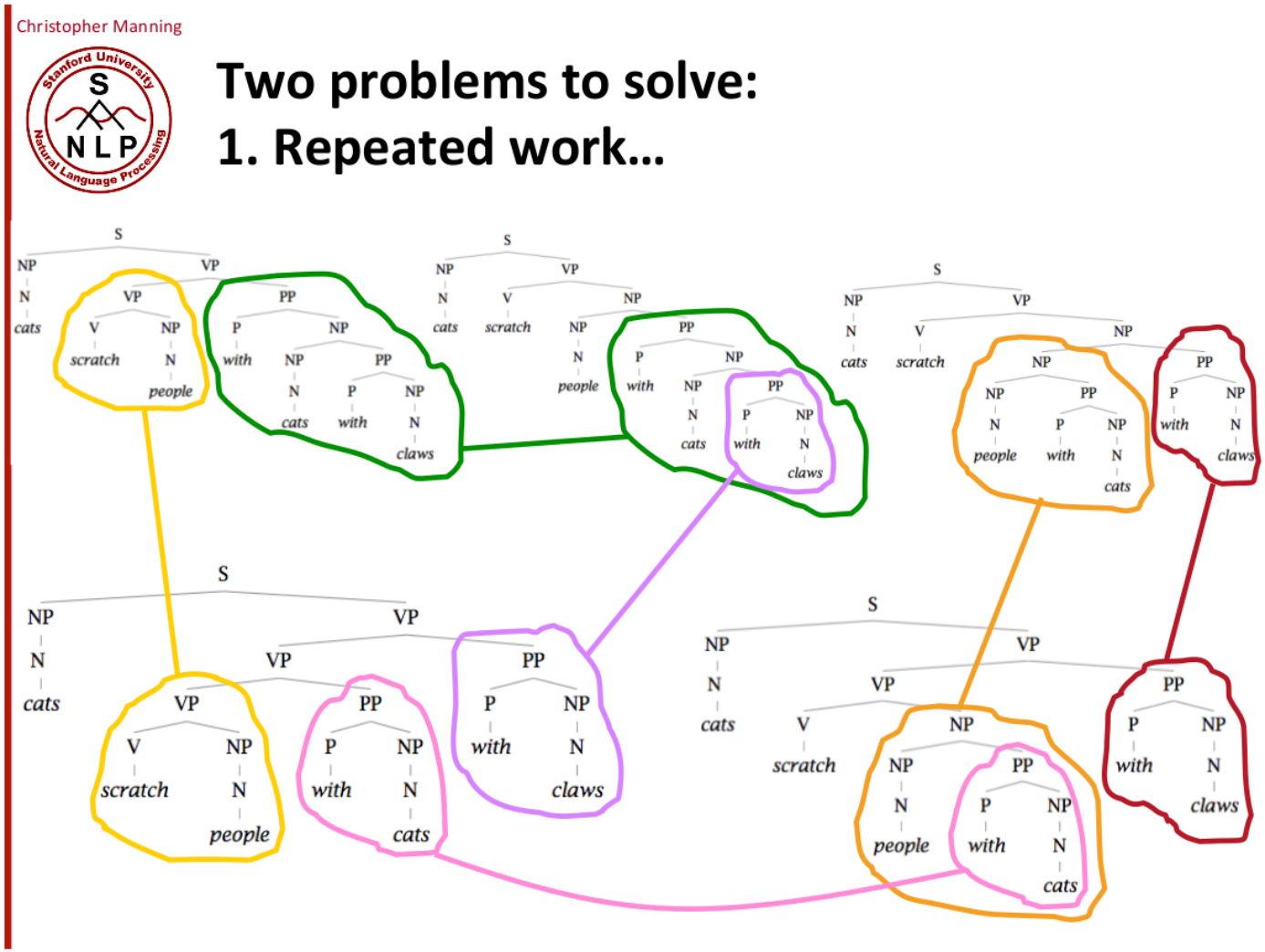
CFG Parsing in NLP: key issues

- Attachment ambiguities: where to attach a PP, a conjunction etc.
- Two problems arise due to this:
  1. You should explore multiple parses and pick the most likely one.
  2. In this process, there are many sub-processes that repeat, and we should somehow preserve this information to not start from scratch for each possibility.

CFG Parsing in NLP: key issues

Source: Jurafsky and Manning's Coursera lectures

For a sentence: cats scratch people with cats with claws.



CFG Parsing: Methods

- Recursive Descent Parsing (Top-down)
- Shift Reduce parsing (bottom up)
- left-corner parsing
- chart parsing
- ...

.. overview on Tuesday.

Probabilistic CFG Parsing

Source: Jurafsky and Manning's Coursera lectures

Relies on treebank data to get these probabilities for each rule.

Christopher Manning

A PCFG

S → NP VP	1.0	N → *people*	0.5
VP → V NP	0.6	N → *fish*	0.2
VP → V NP PP	0.4	N → *tanks*	0.2
NP → NP NP	0.1	N → *rods*	0.1
NP → NP PP	0.2	V → *people*	0.1
NP → N	0.7	V → *fish*	0.6
PP → P NP	1.0	V → *tanks*	0.3
		P → *with*	1.0

Exercise -1

Source: Exercise 12.2 in J&M

Draw tree structures for the following sentences, after creating a grammar together.

- I would like to fly on American airlines.
- Please repeat that.
- Does American 487 have a first class section?
- I need to fly between Philadelphia and Atlanta
- Does American airlines have a flight between five a.m. and six a.m.?
- What is the fare from Atlanta to Denver?
- Is there an American airlines flight from Philadelphia to Dallas?

Exercise-2: NLTK

Source: Chapter 8 in NLTK Book

1. Follow the groucho grammar example in Section 1.2 and simple grammar example in Section 3.1 that uses recursive descent parser.
2. After that, use the grammar in Section 3.3 instead of groucho grammar, and try to parse examples 10 (a) and (b) in the textbook with this grammar.
3. Finally: Figure out how to make Example 3.2 work on your computer, with your own custom created grammar.

Another Exercise

Figure out how to use Stanford parser in NLTK. I will ask about this in Tuesday's class. Work outside the class if needed and find a solution.

Next Week

1. Topics:
   - CFG parsing algorithms overview
   - Dependency grammars
   - using dependency parsers in NLTK
   - Constituency vs Dependency parsing - which is more useful and when?
2. Readings: Chapter 8 in NLTK (Mandatory). Chapter 12–14 in J&M (Optional)
3. Video lectures (optional): Week 5 Lectures in Jurafsky and Manning's course or Weeks 4 and 5 lectures in Radev's course.