ILLINOIS INSTITUTE OF TECHNOLOGY

Context-free Grammars and Syntax

CS-585

Natural Language Processing

Sonjia Waxmonsky

Based on slides from Kai Shu and Derrick Higgins

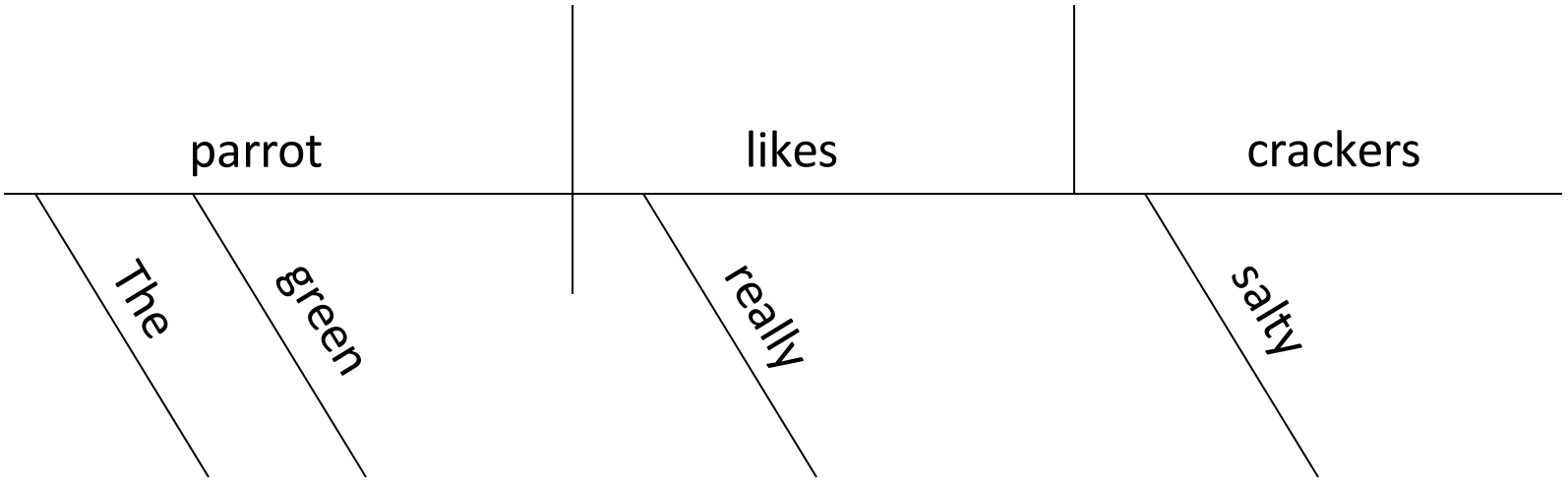
GRAMMATICAL REPRESENTATIONS

Grammatical representations

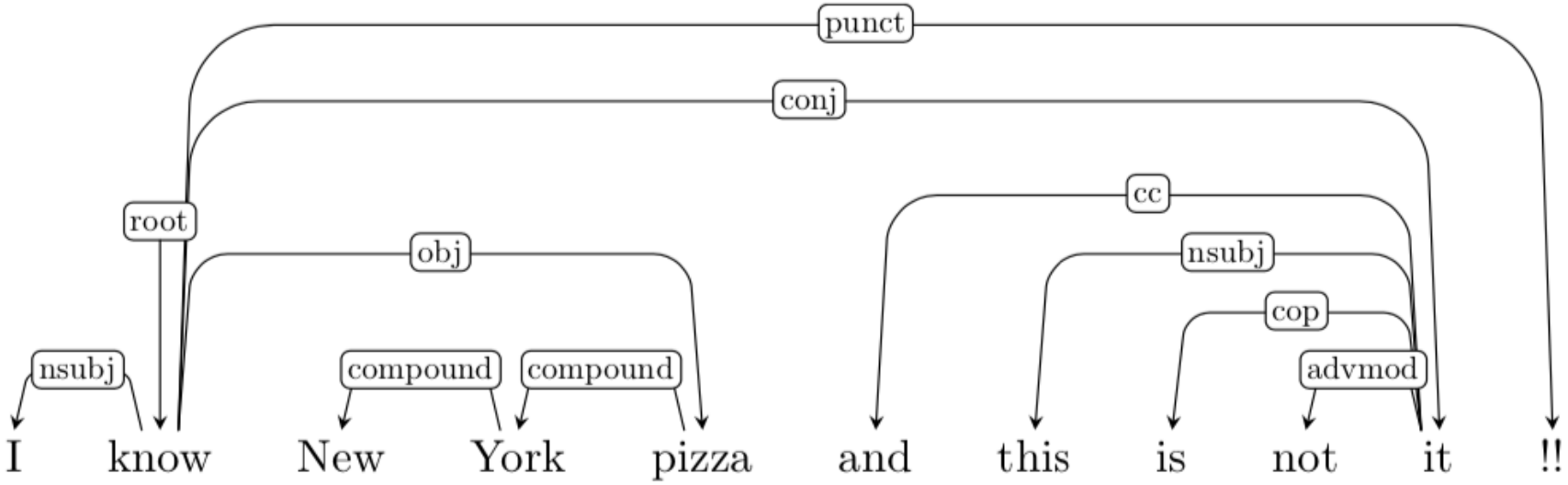
- Grammatical study has been going on for centuries
- But it's only been in the last couple of centuries that people have been representing linguistic structure using **trees** or other **hierarchical** representations

Sentence diagrams

Reed & Kellogg. 1877. *Higher Lessons in English.*



Dependency trees



E-NLP Fig 11.3 (p260)

Generative Grammar and Phrase Structure

- Chomsky, Noam. 1957. Syntactic structures
  - Treat human languages such as English and Sankrit like formal languages such as predicate logic
  - A **generative** process that determines admissible sequences of symbols (and by exclusion, inadmissible sequences)
  - Languages defined by **production rules** that indicate options for expressing sentence parts:

S → NP VP

VP → V NP

NP → DT N

NP → N

VP → V

V → *saw*

V → *heard*

DT → *the*

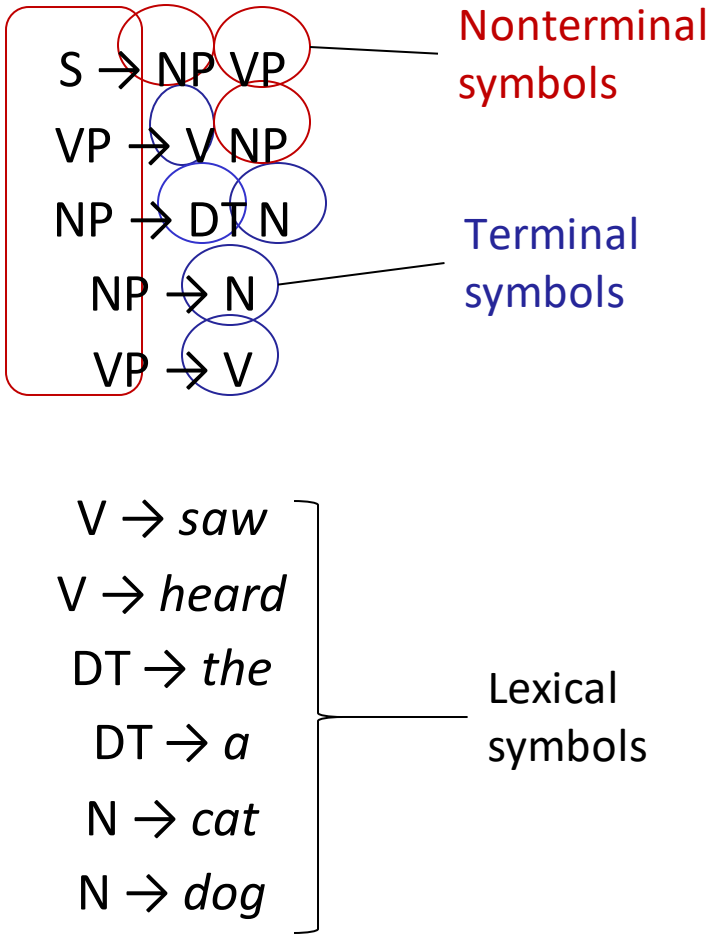
DT → *a*

N → *cat*

N → *dog*

Generative Grammar and Phrase Structure

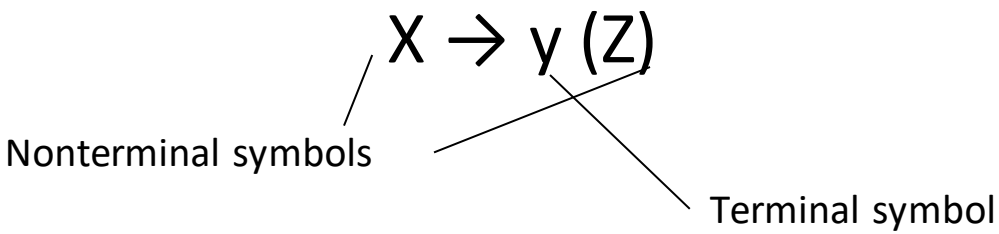
- Chomsky, Noam. 1957. Syntactic structures
  - Treat human languages such as English and Sankrit like formal languages such as predicate logic
  - A **generative** process that determines admissible sequences of symbols (and by exclusion, inadmissible sequences)
  - Languages defined by **production rules** that indicate options for expressing sentence parts:



GRAMMAR TYPES

Regular (Finite-State) Grammars

- Production rules of form



- "Root" symbol **S**
- Same class as regular expressions
- Limited expressivity – for example, incapable of capturing a language such as

$$A^nB^n$$

Context-free Grammars

- Production rules of form

$$X \rightarrow y\ Z$$

$$X \rightarrow y$$

$$X \rightarrow Y\ Z$$

Terminal symbols [x, y, z,] …]

Nonterminal symbols [X, Y, Z, …]

- For example:

$$S \rightarrow NP\ VP$$

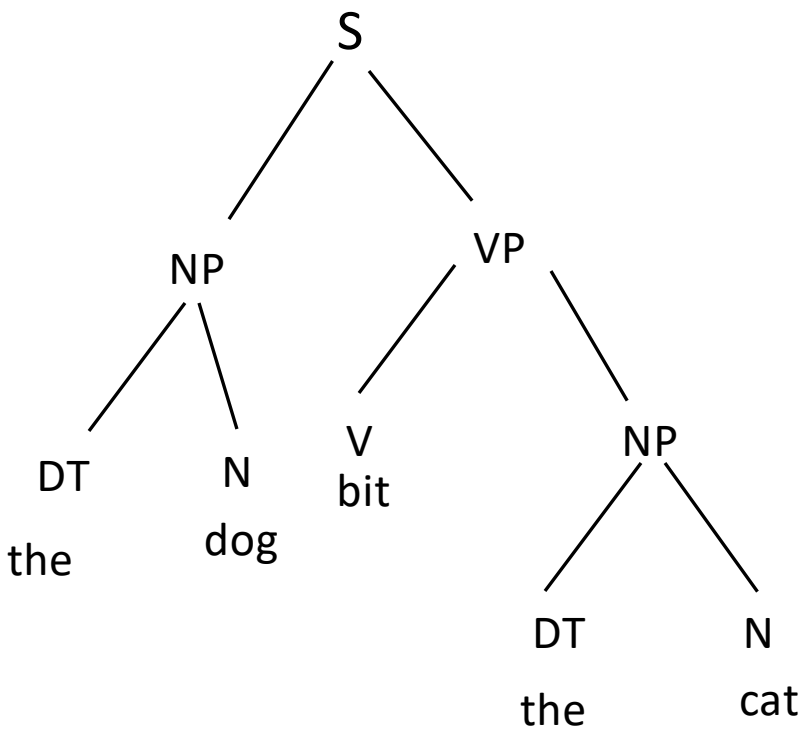
$$NP \rightarrow DT\ N$$

$$DT \rightarrow the$$

$$N \rightarrow dog$$

…

Context-Free Grammars



S → NP VP

NP → DT N

VP → V NP

NP → DT N

DT → the

N → dog

V → bit

DT → the

N → cat

Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S
NP VP

S → NP VP

Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S	
NP VP	S → NP VP
DT N VP	NP → DT N

Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S	
NP VP	S → NP VP
DT N VP	NP → DT N
DT N **V NP**	VP → V NP

Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S	
NP VP	S → NP VP
DT N VP	NP → DT N
DT N **V NP**	VP → V NP
DT N V **DT N**	NP → DT N

Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S	
NP VP	S → NP VP
DT N VP	NP → DT N
DT N **V NP**	VP → V NP
DT N V **DT N**	NP → DT N
the N V DT N	DT → the

Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S	
NP VP	S → NP VP
DT N VP	NP → DT N
DT N **V NP**	VP → V NP
DT N V **DT N**	NP → DT N
the N V DT N	DT → the
the **dog** V DT N	N → dog

Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S	
NP VP	S → NP VP
DT N VP	NP → DT N
DT N **V NP**	VP → V NP
DT N V **DT N**	NP → DT N
the N V DT N	DT → the
the **dog** V DT N	N → dog
the dog **bit** DT N	V → bit

Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S	
NP VP	S → NP VP
DT N VP	NP → DT N
DT N **V NP**	VP → V NP
DT N V **DT N**	NP → DT N
the N V DT N	DT → the
the **dog** V DT N	N → dog
the dog **bit** DT N	V → bit
the dog bit **the** N	DT → the

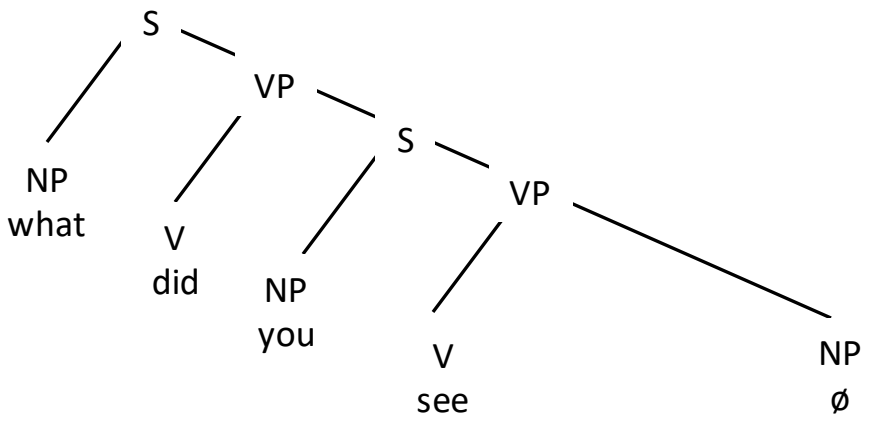
Context-Free Grammars

A sentence is **generated** by a series of rewrite operations

S	
NP VP	S → NP VP
DT N VP	NP → DT N
DT N **V NP**	VP → V NP
DT N V **DT N**	NP → DT N
the N V DT N	DT → the
the **dog** V DT N	N → dog
the dog **bit** DT N	V → bit
the dog bit **the** N	DT → the
the dog bit the **cat**	N → cat

Transformational Grammars

- After production rules apply to create a syntactic tree, transformation operations sensitive to tree structure can further modify it
- Motivated by phenomena such as “wh-extraction”
  - Grammar is simplified if we assume the existence of a ”missing” noun phrase



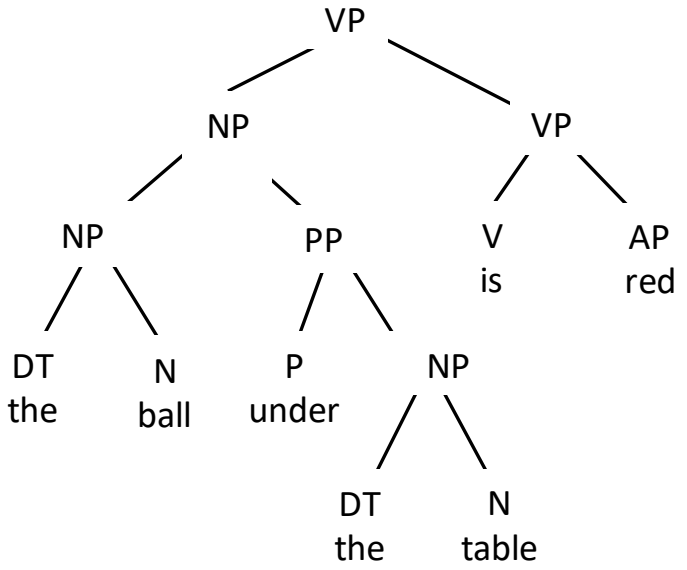
Grammatical Representations for NLP

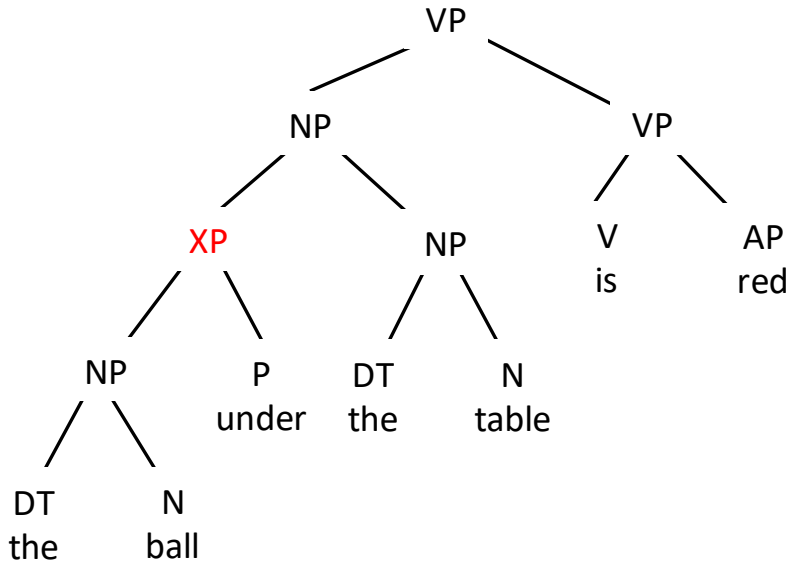
- Transformations are powerful and expressive, but **computationally intractable**, and not a helpful static representation of structure
- So almost all NLP work on syntax is done in terms of **context-free phrase structure** representations
  - Widely used resources such as the Penn Treebank for training and evaluating models
  - Well-established methodologies for CFG parsing

CONSTITUENCY AND PHRASES

Constituents

- There are many ways we could potentially break down a sentence into grammatical units (phrases or *constituents*)





- What is the "correct" way?

Constituency Tests

Conjunction: Only constituents can be conjoined.

The ball [under the table] and [near the door] is red.

** [The ball under] and [the truck on] the table are red.*

Furthermore, conjoined phrases must be of the same category

** The ball [under the table] and [that has my initials on it] is red.*

Constituency Tests

- Fragments: Only constituents can stand alone as incomplete sentence utterances

I saw the ball under the table.

Where?

[Under the table]

I saw the ball under the table.

What/how/where?

** [The ball under]*

Phrase Types

- Noun phrases
  - Headed by a noun
  - May start with a determiner, include modifying adjectives, prepositional phrases and postnominal relative clauses
  - Serves as arguments of prepositions or verbs, or as sentential subject

[$_{NP}$ The fellow I met last week]

[$_{NP}$ The horse with the bit in its teeth]

[$_{NP}$ Four quarters]

[$_{NP}$ We]

Phrase Types

- Verb phrases
  - Headed by a verb
  - Auxiliary verbs head VPs with recursive structure
  - May include adverbial or PP modifiers
  - Includes verb object arguments, but not sentence subject

[$_{VP}$ Finally took a week-long vacation]

[$_{VP}$ Saw the moon with a telescope]

[$_{VP}$ Will [$_{VP}$ have [$_{VP}$ been [$_{VP}$ winning]]]]

[$_{VP}$ Left]

Phrase Types

- Prepositional phrases
  - Headed by a preposition
  - Include noun phrase object
  - Often adjoined to VPs or NPs; may also be verbal complements

[$_{PP}$ On the moon]

[$_{PP}$ Between here and eternity]

[$_{PP}$ One week ago]

Phrase Types

- Adjective phrases
  - Headed by an adjective
  - Various complex adjectival structures

[$_{AP}$ Red and gold]

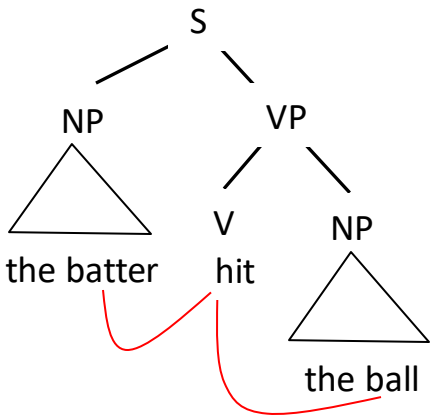
[$_{AP}$ Certain to succeed]

[$_{AP}$ Bigger than a breadbox]

[$_{VP}$ Happy as a clam]

Phrasal Heads

- The *head* of a phrase is the word that determines its attributes
  - Typically, of the same category as the phrase: the head of a noun phrase is a noun, the head of a prepositional phrase is a preposition, etc.
  - Attributes of the head (e.g., tense in the case of verbs, number and case in the case of nouns) are shared by the phrase as a whole
  - Relationships between heads of phrases are strongly predictive for parsing



SYNTACTIC STRUCTURE

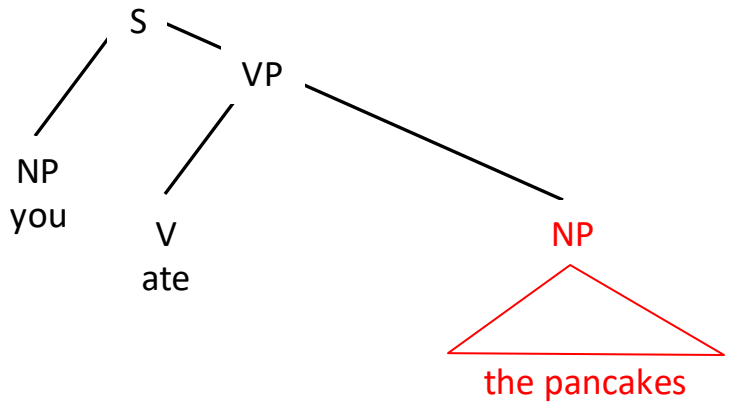
Complements and Adjuncts

- Complement: Syntactic elements that are **necessary to complete the meaning** of a sentence in a particular context
  - Category determined by a specific lexical item in the phrase
  - Limited number of slots to fill
- Adjunct: **Optional** elements added to phrases of a particular category
  - Largely independent of the lexical head of the phrase
  - Possibly more than one

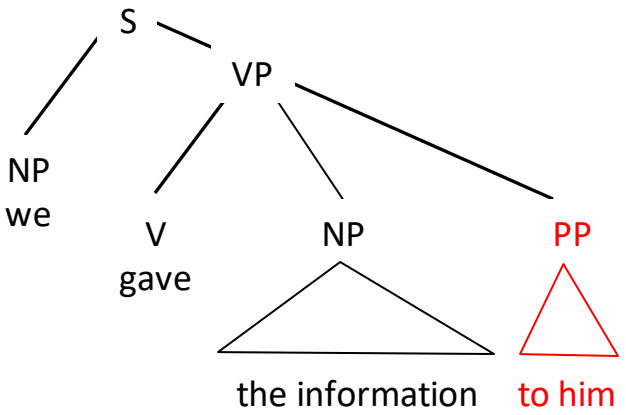
Complements

For example

- Direct objects



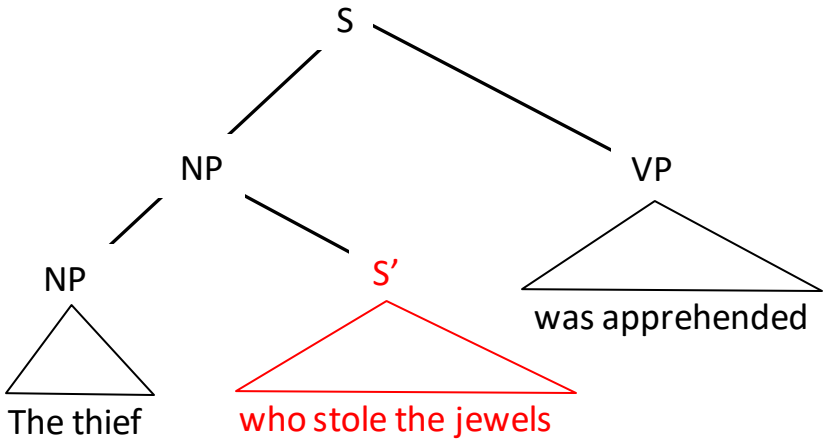
- Prepositional phrase complements of ditransitive verbs



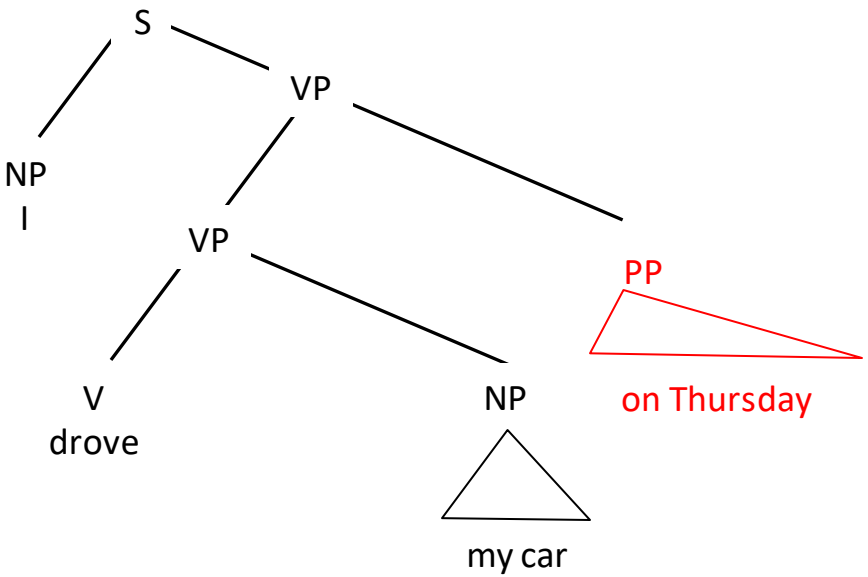
Adjuncts

For example

- Relative clauses

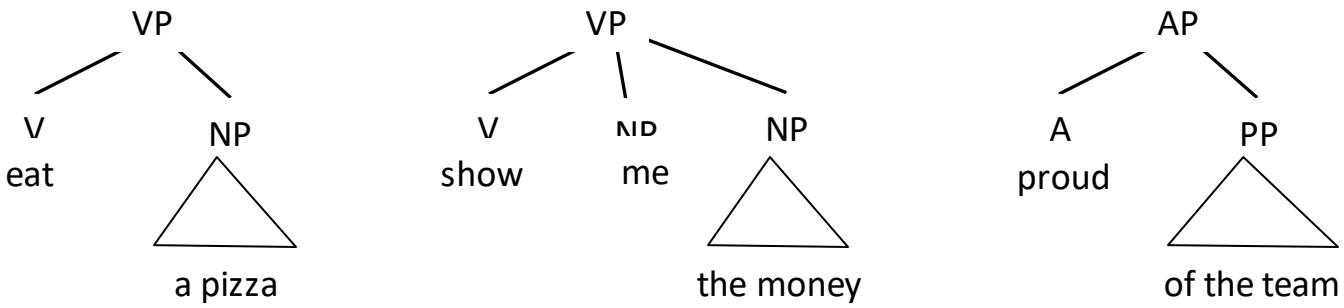


- Adverbial prepositional phrases



Subcategorization

- Subcategorization is the relationship between a **syntactic** head word and the **dependents** it requires
  - A transitive verb like “eat” subcategorizes for a single noun phrase
  - A ditransitive verb like “show” subcategorizes for two noun phrases
  - The adjective “proud” subcategorizes for a prepositional phrases headed by “of”

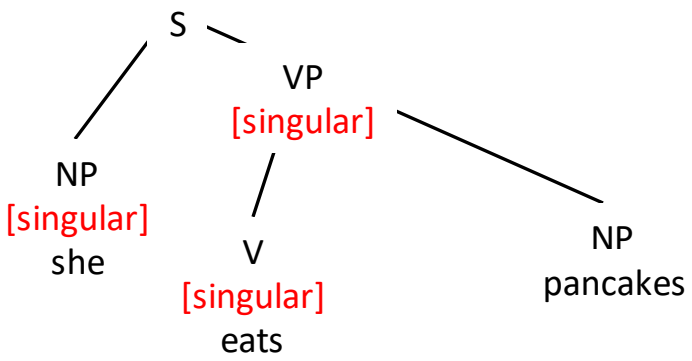


Agreement

- *Agreement* phenomena: words in a phrase that take different forms based on the number, case, person, etc. of a noun
- Motivates feature-based syntactic representations

$S \rightarrow NP[\alpha \text{ number}]\ VP[\alpha \text{ number}]$

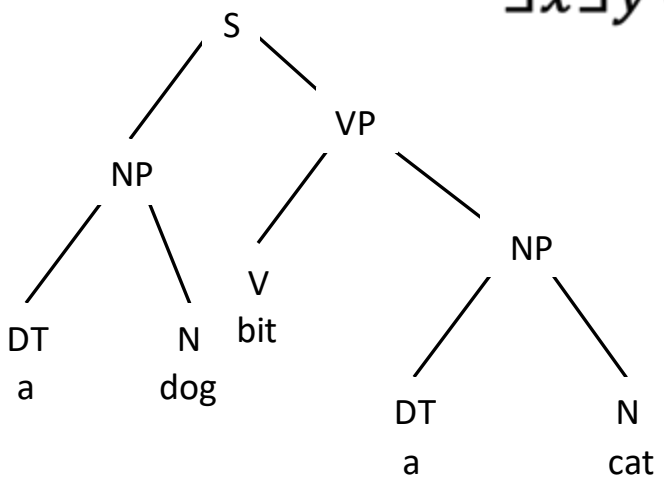
$VP[\alpha \text{ number}] \rightarrow V[\alpha \text{ number}]\ NP$



SYNTAX AND SEMANTICS

Compositionality

- Compositionality is that semantic structure should mirror syntactic structure
  - Each phrase has a meaning
  - The meaning of larger units is a function of the meaning of smaller units and the way in which they are combined



$$\exists x \exists y \, \mathbf{dog}'(x) \wedge \mathbf{cat}'(y) \wedge \mathbf{bit}'(x, y)$$

$$\exists x \, \mathbf{dog}'(x)$$

$$\lambda x \lambda y \, \mathbf{bit}'(x, y)$$

$$\exists y \, \mathbf{cat}'(y)$$

Beyond bag of words for sentiment

Slide from Session 11

- Neural models are a natural fit
- Socher et al. (2013)
  - Phrase-level sentiment scores for over 215K phrases (≈12K sentences)
  - Recursive architecture predicts sentiment for each constituent of a syntactic structure, until tree root (full sentence) is reached
  - Detailed analysis of how linguistic cues to sentiment are captured by the model
  - Full-featured demo, code, and corpus at the project site: https://nlp.stanford.edu/sentiment/

Socher, Richard, et al. 2013. *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*. In EMNLP.

Beyond bag of words for sentiment

Slide from Session 11

Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.7**	**87.6**	**85.4**

Table 1: Accuracy for fine grained (5-class) and binary predictions at the sentence level (root) and for all nodes.

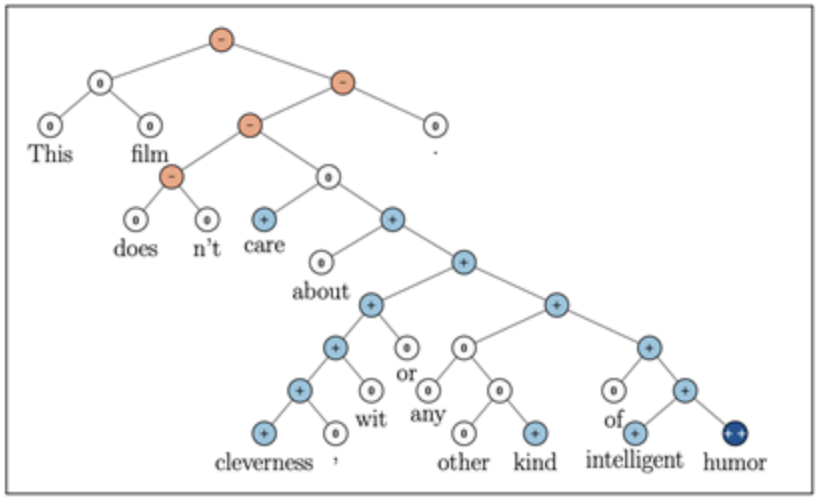


Figure 1: Example of the Recursive Neural Tensor Network accurately predicting 5 sentiment classes, very negative to very positive (– –, –, 0, +, + +), at every node of a parse tree and capturing the negation and its scope in this sentence.

Remember these phrase-level sentiment predictions?

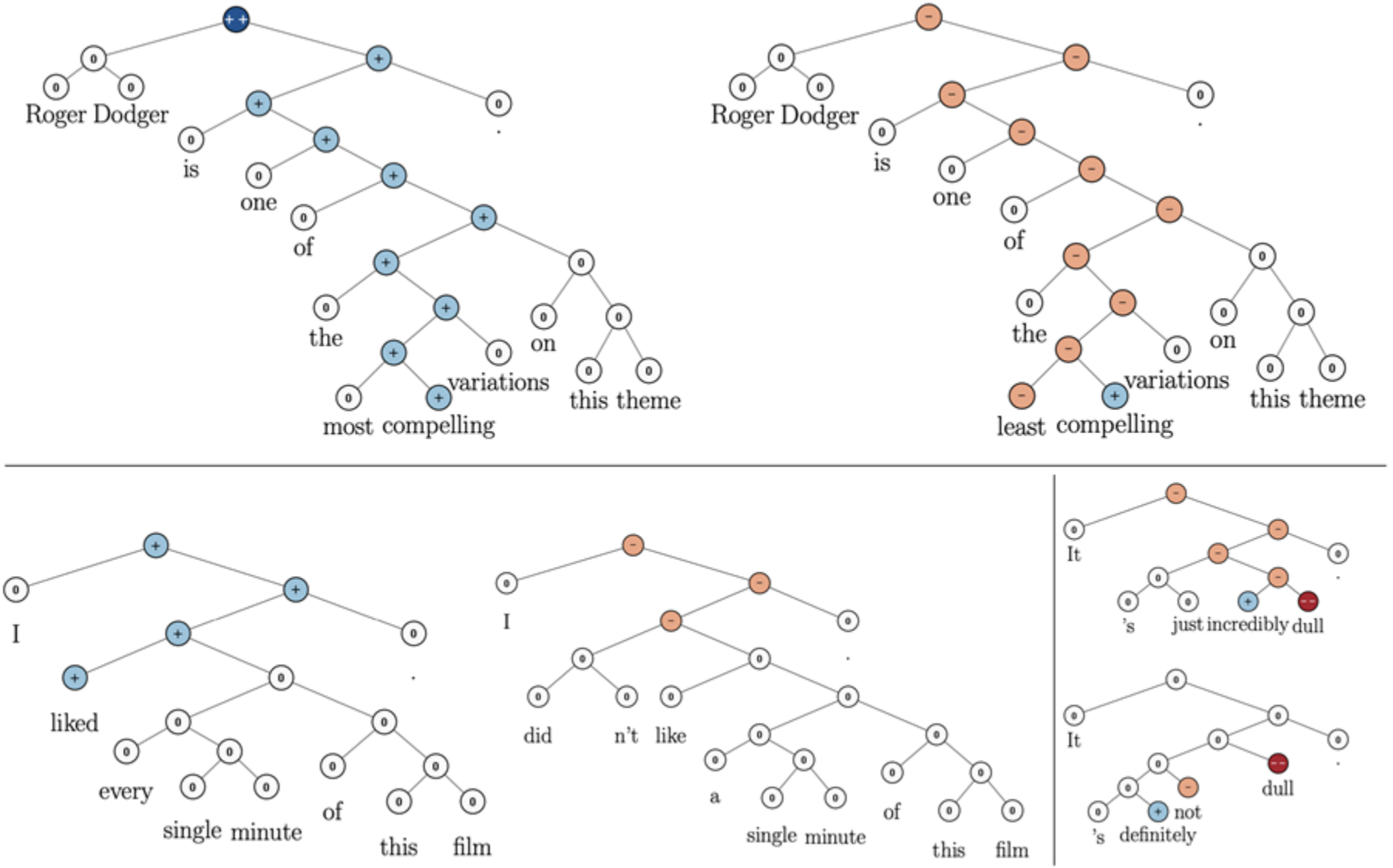
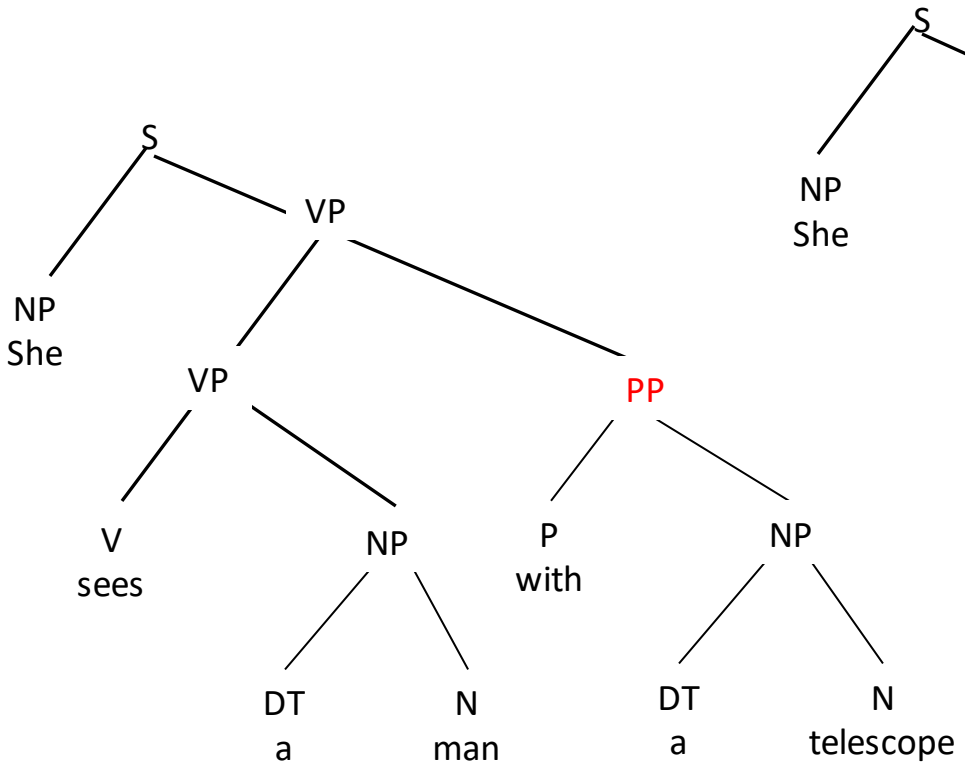
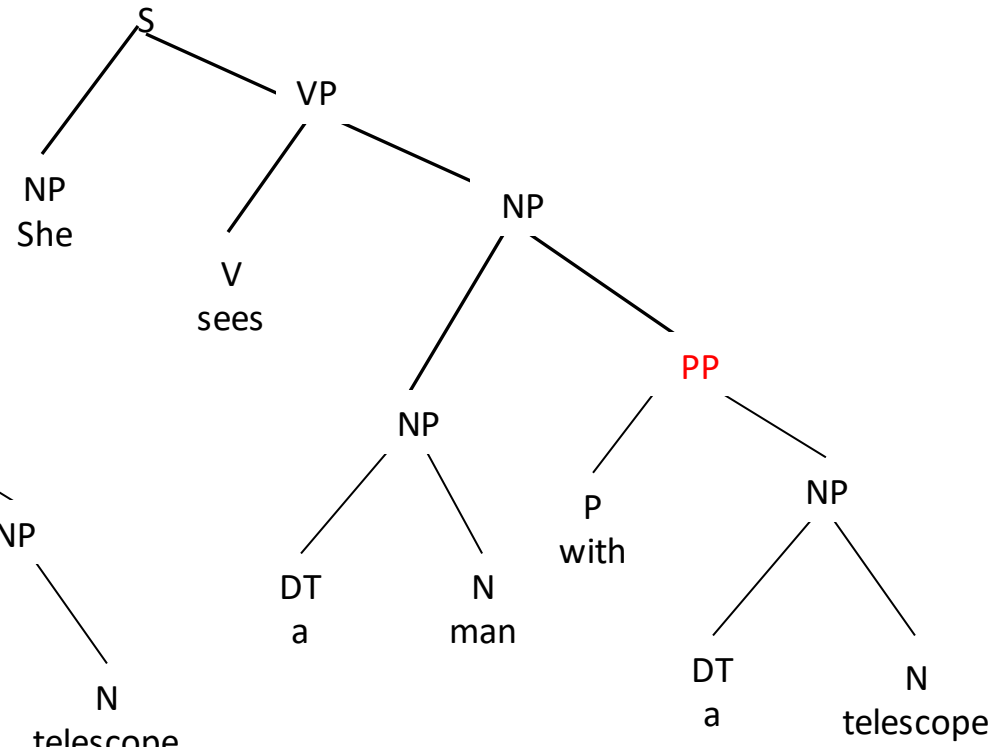


Figure 9: RNTN prediction of positive and negative (bottom right) sentences and their negation.

Structural Ambiguity

- Related to compositionality – sometimes meaning differences can be traced to differences in syntactic structures





TREEBANK DATASETS

Treebanks

- A **treebank** is a set of linguistic tree structures that humans have assigned to texts following a set of **annotation guidelines**
- Used for **training** and **evaluation** of syntactic parsing models
- The best-known treebank is the **Penn Treebank**
  - About 7 million words from newswire text, and assorted fiction and nonfiction genres
  - Includes both Part-of-Speech (POS) tags and syntax parse
- There are others:
  - Stuttgart TIGER treebank of German
  - Penn Chinese Treebank
  - Penn Arabic Treebank
  - UAM Treebank of Spanish

Penn Treebank

1.2.2 SINV

The SINV label is used for subject-auxiliary inversion in the case of negative inversion, conditional inversion, locative inversion, and some topicalizations. (SINV is not used with questions. See section 1.2.6 and section 1.2.5 for the treatment of subject-auxiliary inversion in the case of yes/no questions and *wh*-questions, respectively.) Inverted auxiliaries are unlabeled.

```
(SINV (ADVP-TMP Never)
      had
      (NP-SBJ I)
      (VP seen
          (NP such a place)))
```

When the inversion results in a conditional clause (i.e., when it is equivalent to (SBAR-ADV if...), the SINV is enclosed in SBAR-ADV).

```
(S (SBAR-ADV (SINV had
                   (NP-SBJ Casey)
                   (VP thrown
                       (NP the ball)
                       (ADVP-MNR harder))))
   ,
   (NP-SBJ it)
   (VP would
       (VP have
           (VP reached
               (NP home plate)
               (PP-TMP in
                       (NP time))))))
```

318 pages of instructions!

Treebank Bracketing

- Diagrammatic trees are nice graphical representations of syntactic structure, but we need **computer-readable** representations
- The typical convention is to
  - Show constituent structure as nested brackets, with spaces inserted between adjacent constituents
  - Label the opening bracket of a constituent with its category

Treebank Bracketing

```
(S (NP she)
   (VP (VP (V sees)
           (NP (DT a)
               (N man)
           )
       )
       (PP (P with)
           (NP (DT a)
               (N telescope)
           )
       )
   )
)
```