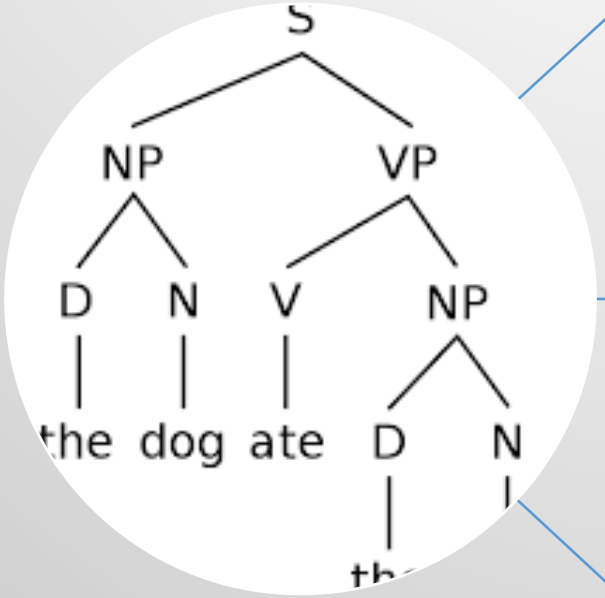
Natural Language Processing

Dr. Karen Mazidi

Part Three: Sentences



Topics

- Syntax parsers
  - PSG
  - dependency
  - SRL

Quizzes

- No quiz

Homework

- Portfolio: Parsing

Parsing natural language

- Formal grammars rely on rules; natural language requires a probabilistic approach
- Issues:
  - POS ambiguity, example:
    - My spidey sense senses danger.
  - Structural ambiguity:

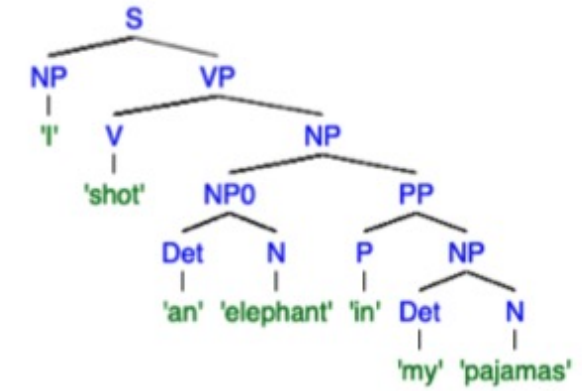


Figure 10.1: The elephant is in the pajamas

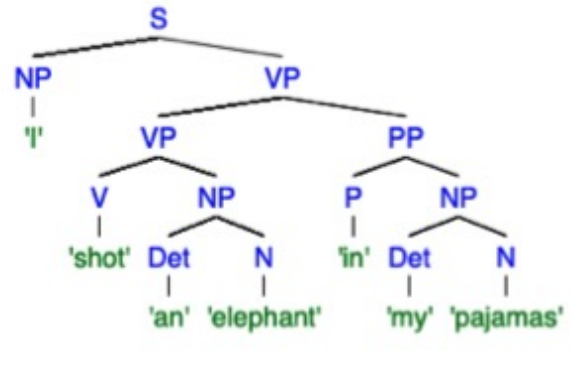


Figure 10.2: The shooter is in the pajamas

Structural ambiguity

- Attachment ambiguity – where to place a phrase in the syntax tree (elephant example)
- Coordination ambiguity – coordinating conjunctions
  - Old men and dogs

Types of syntax parsers

- Phrase structure grammar parse (PSG)
- Dependency parse
- Semantic role label parse

PSG parse

aka constituent parsing

PSG parse

- Organizes sentence in hierarchy of constituents (phrases)
  - Similar to CFG in that top level is S
  - Next-to bottom level is POS
  - Bottom level is the tokens
- PSG parsers are trained on millions of sentences
- Visualized in bracket notation or a tree
- Sometimes called constituency parsing

PSG

- http://mshang.ca/syntree/

```
[S [NP [DT The] [NN butler]]
     [VP [VBN murdered]
         [NP [NNP John]]
            [PP [IN in] [DT the] [NP [NN living] [NN room]]]
```

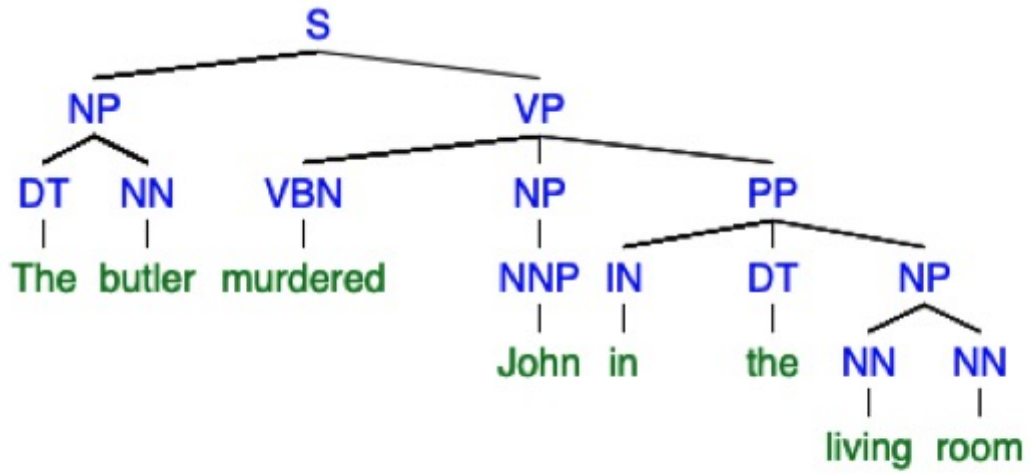


Figure 10.3: Phrase Structure Parse

AllenNLP

- Expanding boxes instead of tree
- https://demo.allennlp.org/

Figure 10.4: Top Level Phrase Structure

Expanding

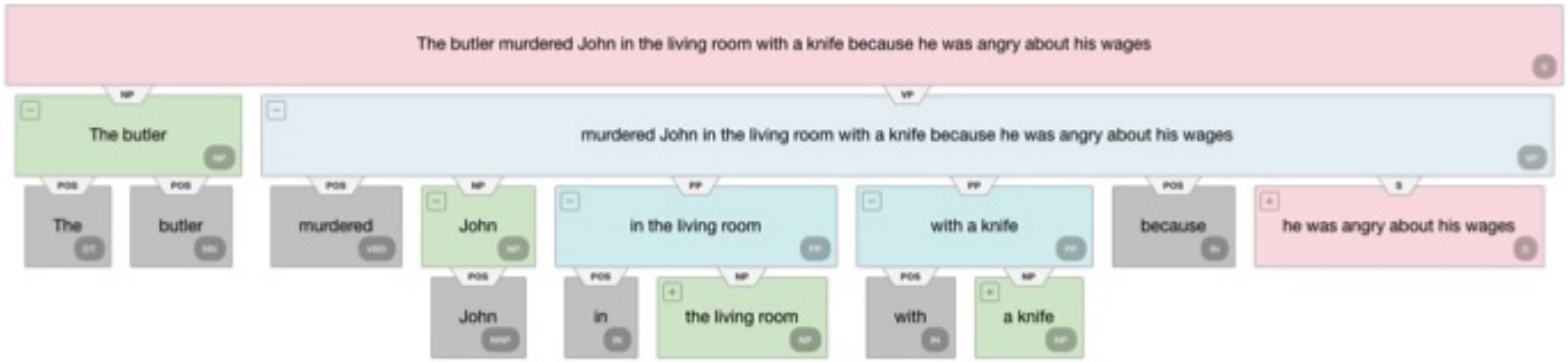


Figure 10.5: Adding Levels to the Phrase Structure

Expanding

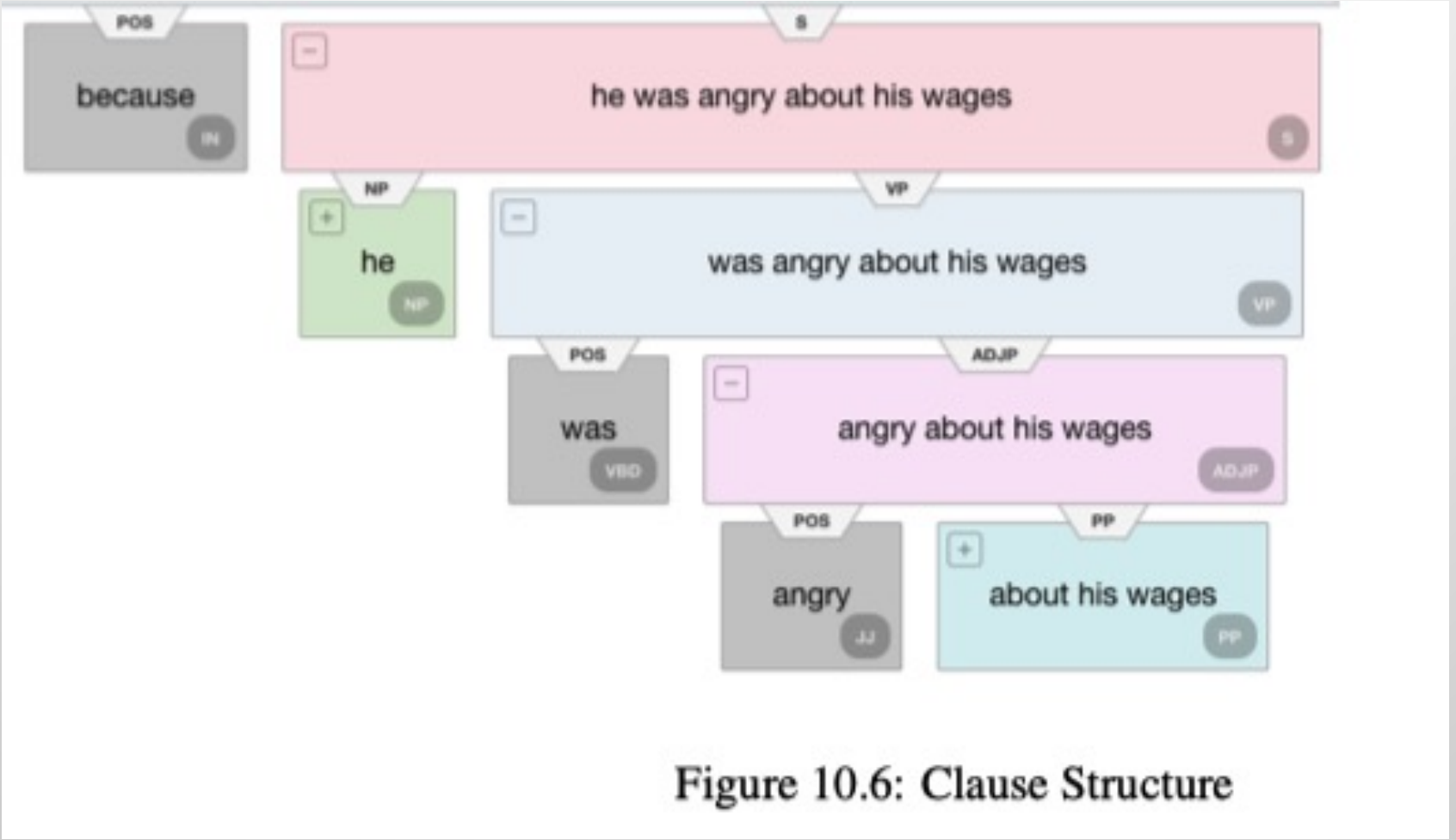
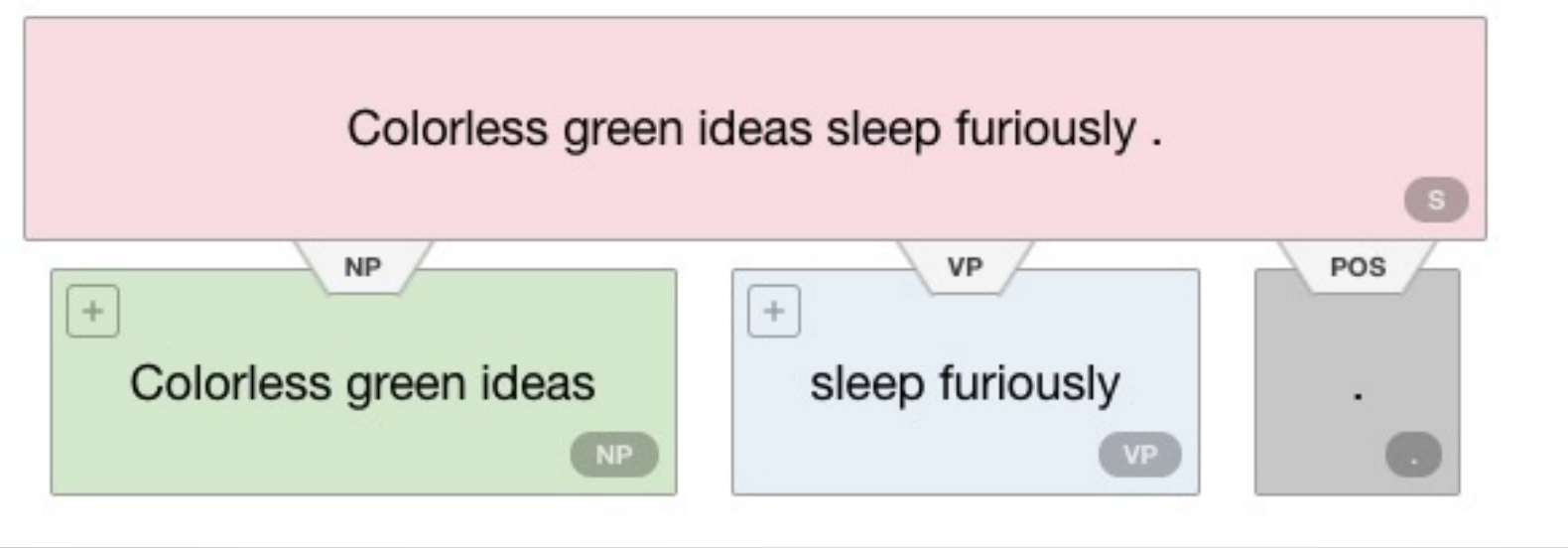


Figure 10.6: Clause Structure

Example

- Colorless green ideas sleep furiously.



Dependency parse

DAG

Dependency parse

- Words are in an acyclic graph, usually with the main predicate as the root node
- Stanford dependencies in Figure 10.11
- Run at: https://corenlp.run/

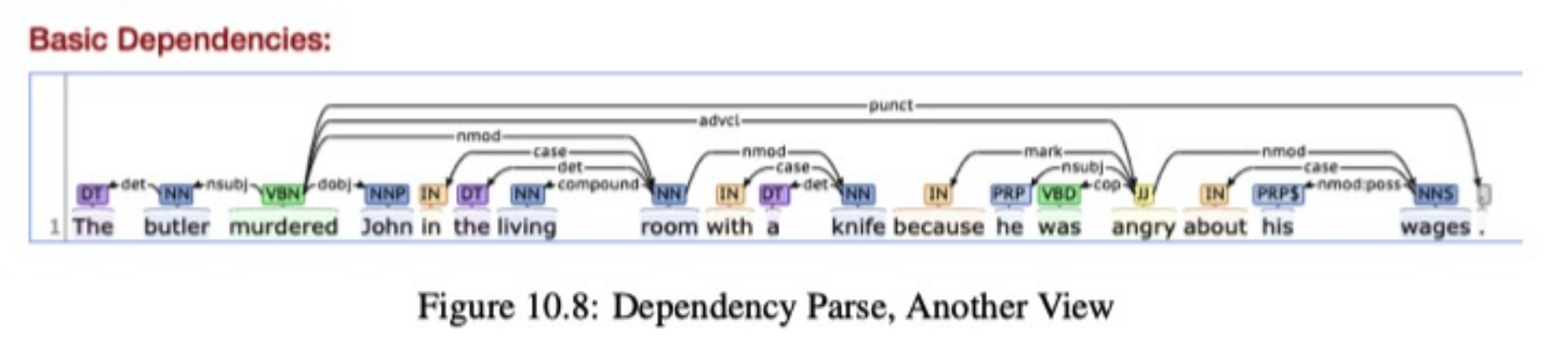


Figure 10.8: Dependency Parse, Another View

Allen NLP

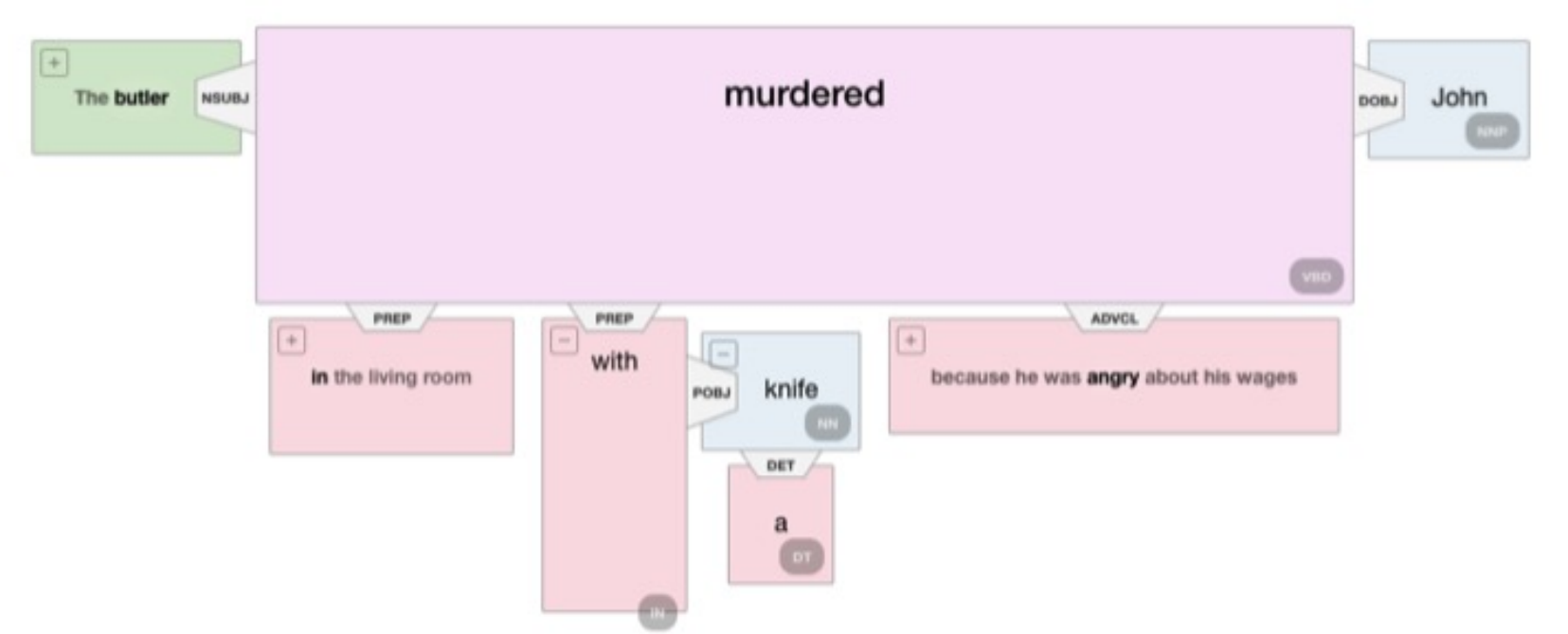
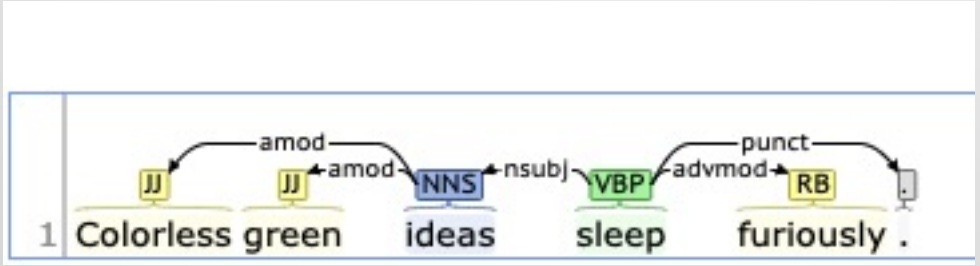


Figure 10.7: Dependency Parse

Example

- Colorless green ideas sleep furiously.
- AllenNLP demo – problem for this sentence
- Stanford demo:



Copular clauses

- John is handsome.
  - Links subject with adjective
- John is a lawyer.
  - Links subject with noun

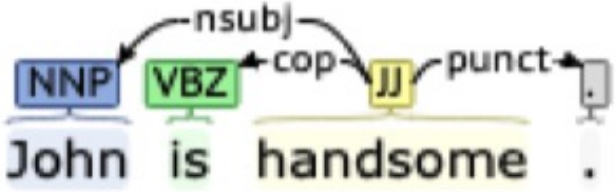


Figure 10.9: The predicate is JJ

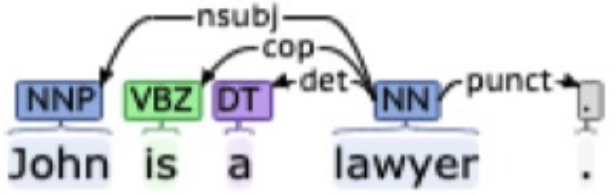


Figure 10.10: The predicate is NN

Copular clauses

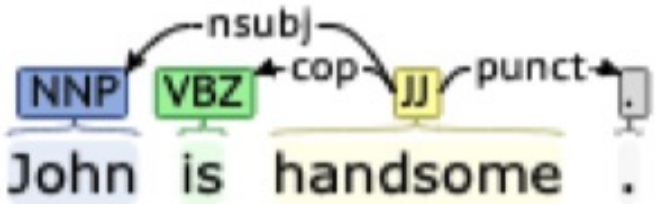


Figure 10.9: The predicate is JJ

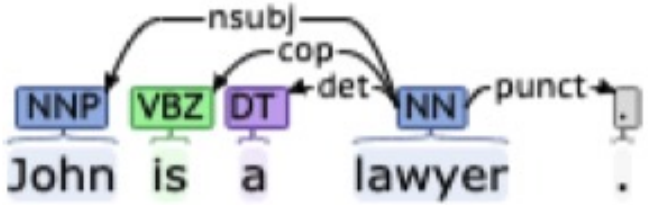


Figure 10.10: The predicate is NN

- Why isn't "is" the predicate?
- Evidence yes:
  - Verb "is" changes tense
  - Verb "is" matches subject
- Evidence no:
  - Move copular clause to subordinate position

```
Mary thinks that John is handsome.
Mary thinks John handsome.
```

Stanford dependencies

- See p. 118 of the Stanford manual
- Note the structure:
  - root
  - dep
    - aux
    - arg
      - agent
      - comp
      - subj

Universal dependencies

- Dependencies that work across languages
- https://www.asc.ohio-state.edu/demarneffe.1/papers/depling.pdf

- Stanford now uses universal dependencies by default:
- https://nlp.stanford.edu/software/stanford-dependencies.html

Common sentence patterns

- In expository text

Pattern	Example	Comment
S-V	The Cowboys lost.	no object or complement
S-V-dobj	The Cowboys lost the game.	direct object
S-V-iobj-dobj	The Cowboys handed them the game.	both direct and indirect objects
S-V-acomp	He looked tired.	'tired' completes the meaning
S-V-ccomp	I hope that you get the job.	'you' is the internal subject
S-V-xcomp	I love to take long walks.	no internal subject in clause
S-V-xcomp	I love walking in the woods.	no internal subject in clause

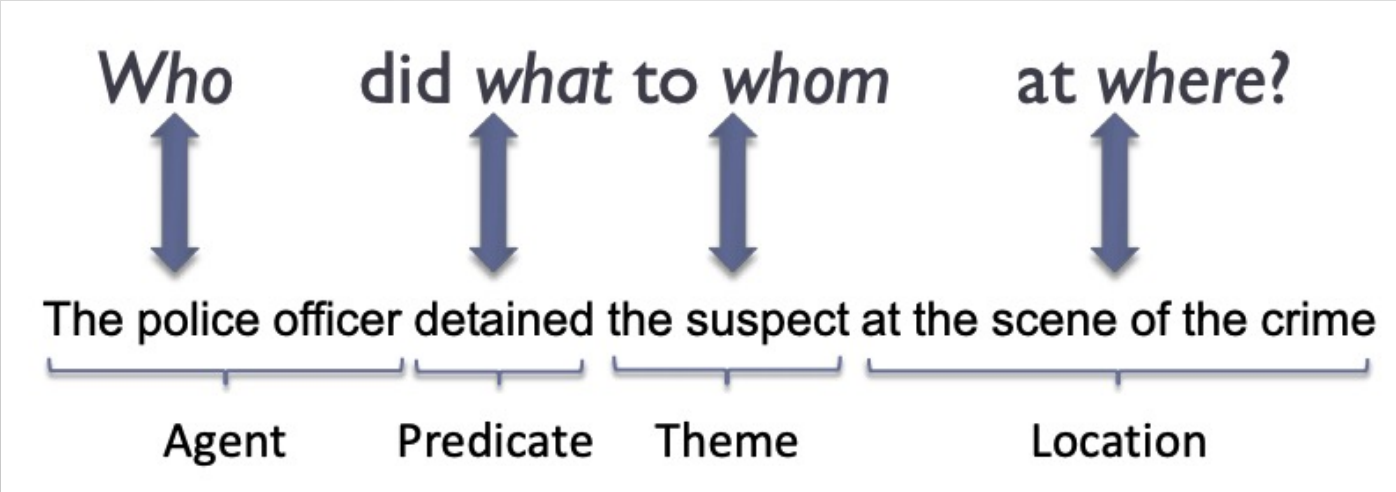
Table 10.1: Common Sentence Patterns

SRL parse

shallow semantics

SRL Parse

- Semantic role label parse – determines a role for each constituent relative to the predicate
- Aka shallow semantic parse
- Two categories of labels:
  - Arguments: actors in the sentence
  - Modifiers: more details such as time and place



SRL

- AllenNLP

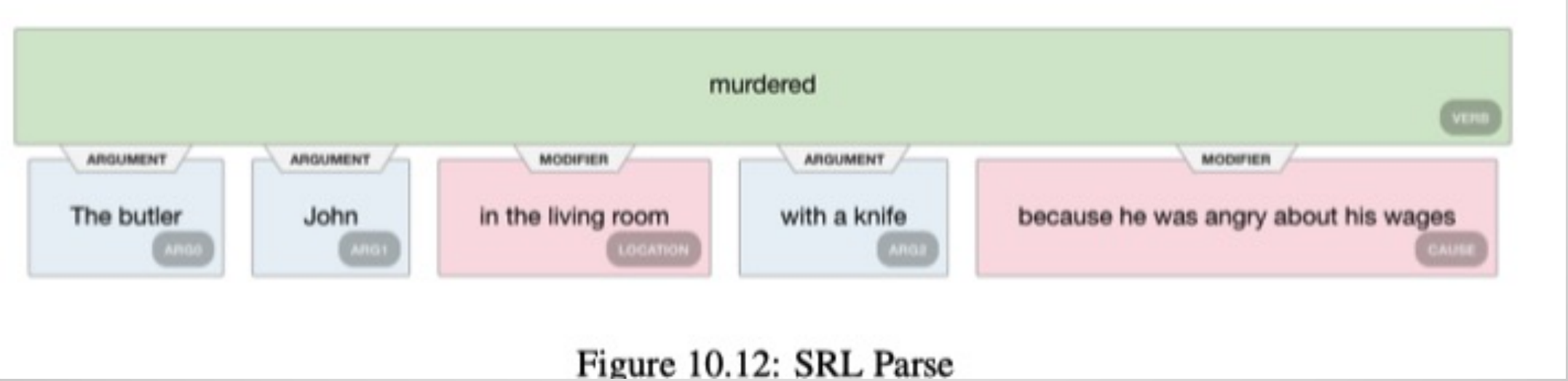
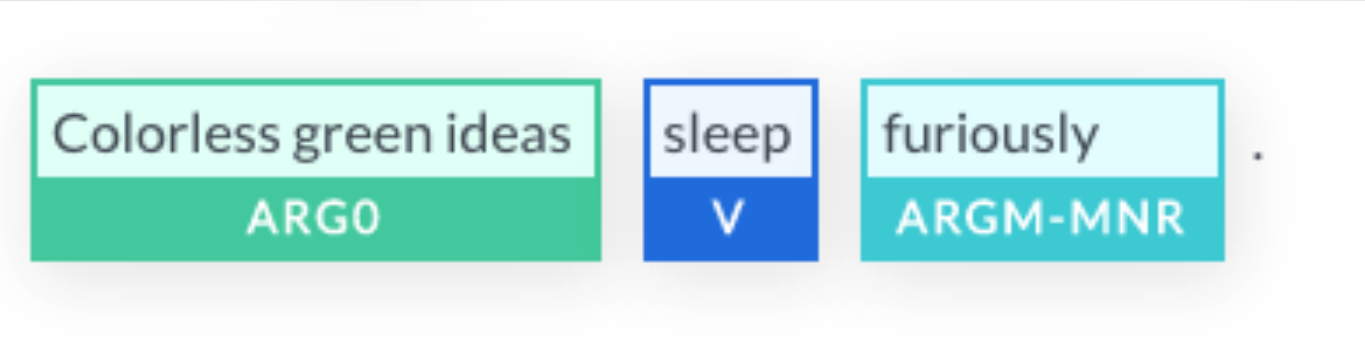


Figure 10.12: SRL Parse

SRL

- AllenNLP



SRL arguments

- Numbered from 0 to 6
- Arguments available for a particular verb, particular sense of the word, vary
- Arg0, A0, is the agent of the sentence, the one doing the action
- Arg1, A1, is often the passive actor
- Arg1 vs. Arg0 has nothing to do with “subject”

```
Arg0 [The butler] murdered Arg1 [John].
Arg1 [John] was murdered by Arg0 [the butler].
```

SRL arguments

- Arg2, A2, is often the ‘instrument’
  - With a knife
  - An argument can be a PP as well as a NP

Arg	**Meaning**	textbfExample
Agent	Entity doing the action	**John** broke the window.
Patient	Entity that is acted upon	John broke the **window**.
Instrument	Entity used in action	John broke the window with a **hammer**.
Beneficiary	Entity recipient	John gave the ring to **Mary**.

Table 10.2: Common Arguments

SRL args

- More arguments

Thematic Role	Definition	Example
AGENT	The volitional causer of an event	*The waiter* spilled the soup.
EXPERIENCER	The experiencer of an event	*John* has a headache.
FORCE	The non-volitional causer of the event	*The wind* blows debris from the mall into our yards.
THEME	The participant most directly affected by an event	Only after Benjamin Franklin broke *the ice*...
RESULT	The end product of an event	The city built a *regulation-size baseball diamond*...
CONTENT	The proposition or content of a propositional event	Mona asked *"You met Mary Ann at a supermarket?"*
INSTRUMENT	An instrument used in an event	He poached catfish, stunning them *with a shocking device*...
BENEFICIARY	The beneficiary of an event	Whenever Ann Callahan makes hotel reservations *for her boss*...
SOURCE	The origin of the object of a transfer event	I flew in *from Boston*.
GOAL	The destination of an object of a transfer event	I drove *to Portland*.

- Problem: hard to create a standard set of arguments for all verbs
- PropBank: fewer roles

PropBank

- Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics, 31(1):71–106
- Some roles:
    - Arg0: proto-agent
    - Arg1: proto-patient
    - Arg2: often benefactive, instrument, attribute, end state
    - Arg3: start point, benefactive, instrument, attribute
    - Arg4: the end point

SRL modifiers

- Not arguments, optional content that adds more info
- Common modifiers:

Mod	Meaning	Example
DIR	Motion along a path	John threw the papers **in the trash.**
LOC	Where the action happened	John was born **in Texas**.
MNR	How the action was performed	John broke the window **violently**.
TMP	When the action happened	John was born **in 1960**.
CAU	Reason for action	John moved to NY **because of his job**.
PNC	Motivation for an action	John saved money **for his move**.

PropBank Verb Frames

- Annotation of major verbs and major senses
- Example:

fall.01

Arg1: Logical subject, patient, thing falling
Arg2: Extent, amount fallen
Arg3: start point
Arg4: end point, end state of arg1
Ex1: [$_{Arg1}$ Sales] *fell* [$_{Arg4}$ to $25 million] [$_{Arg3}$ from $27 million].
Ex2: [$_{Arg1}$ The average junk bond] *fell* [$_{Arg2}$ by 4.2%].

PropBank Verb Frames

- Example verb frame:

increase.01 "go up incrementally"
Arg0: causer of increase
Arg1: thing increasing
Arg2: amount increased by, EXT, or MNR
Arg3: start point
Arg4: end point

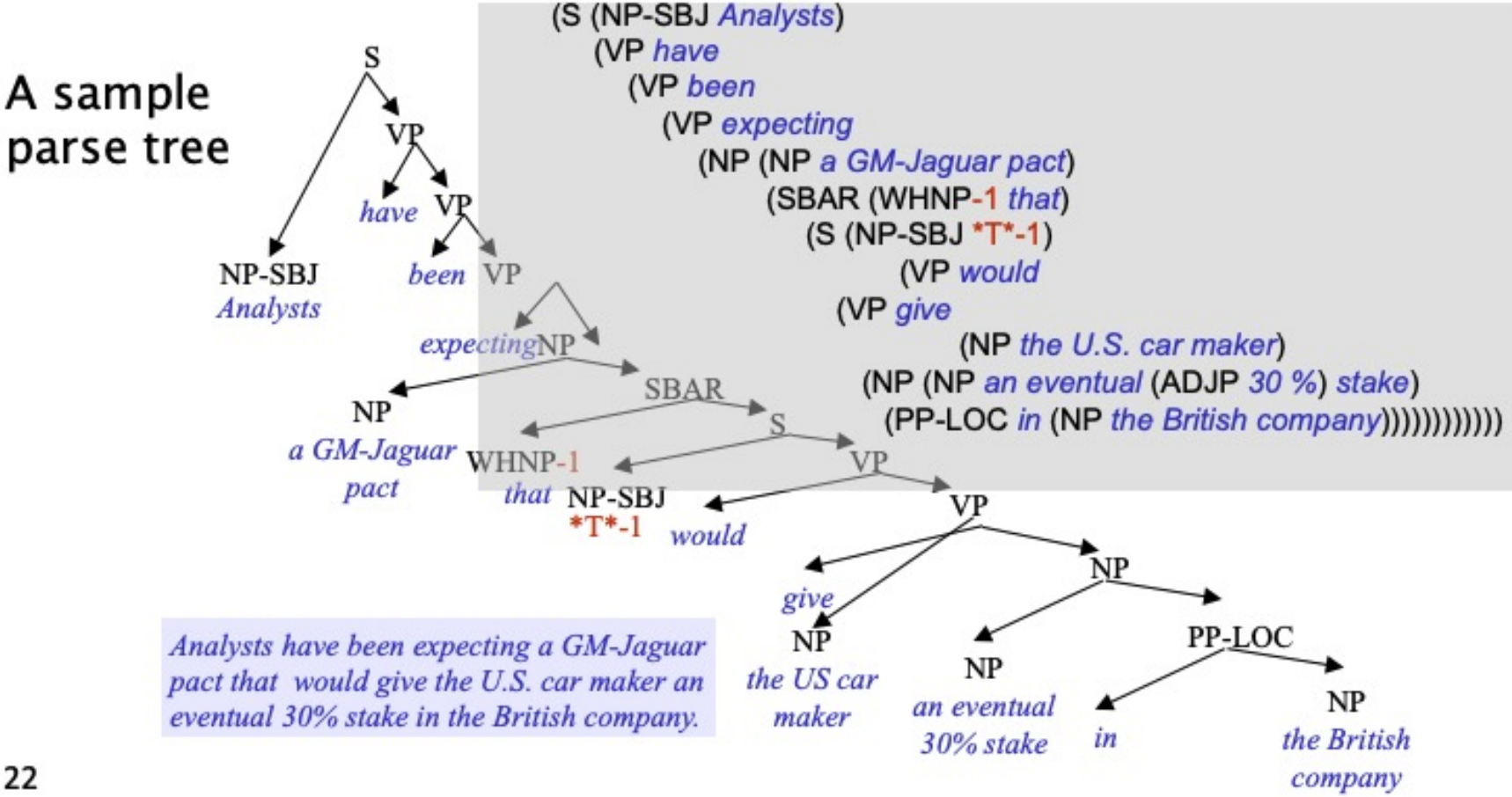
- Clarifies commonalities here:

[$_{Arg0}$ Big Fruit Co.] increased [$_{Arg1}$ the price of bananas].
[$_{Arg1}$ The price of bananas] was increased again [$_{Arg0}$ by Big Fruit Co.]
[$_{Arg1}$ The price of bananas] increased [$_{Arg2}$ 5%].

PropBanking a Sentence

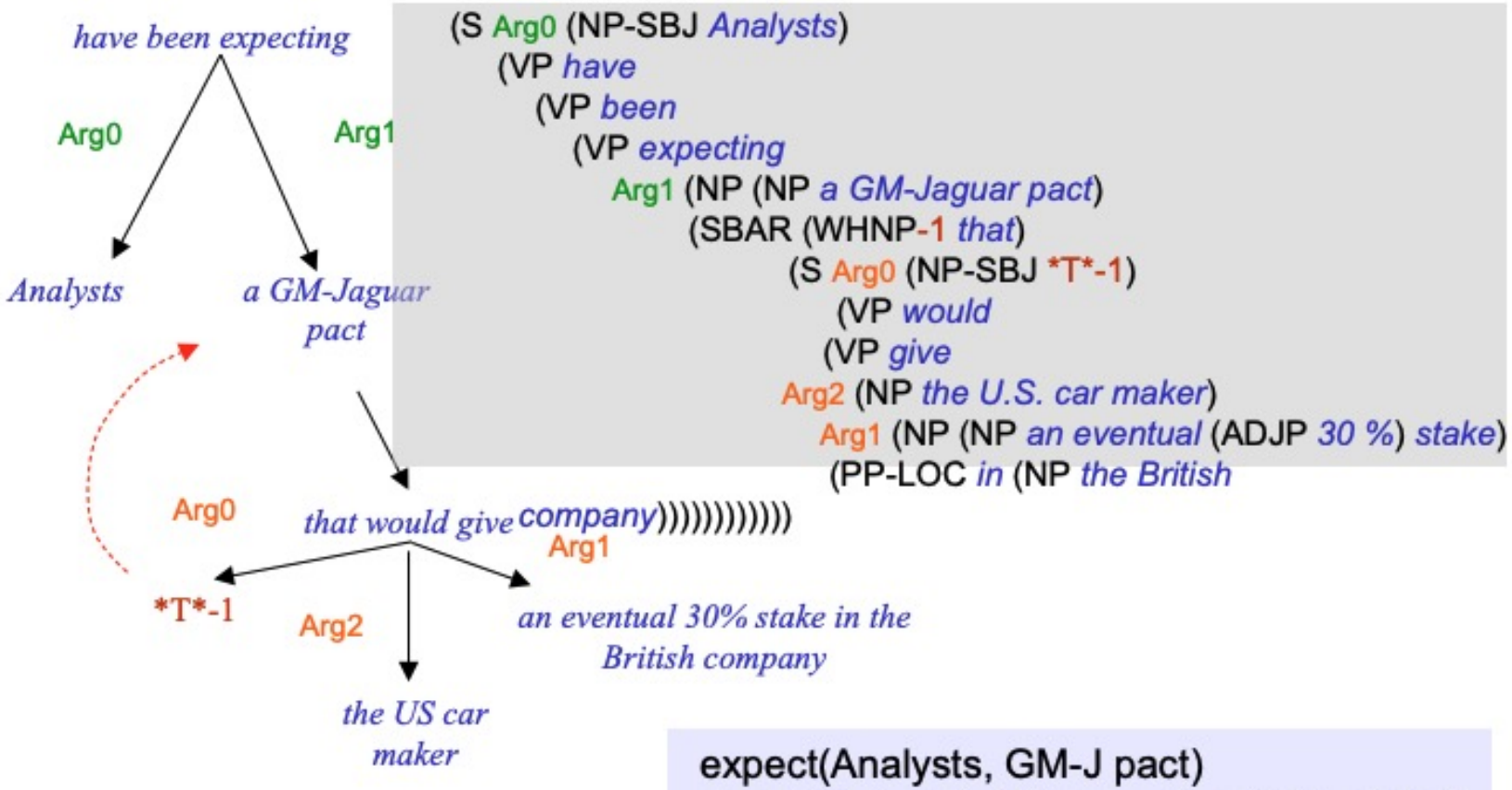
Martha Palmer 2013

A sample parse tree



The same parse tree PropBanked

Martha Palmer 2013



expect(Analysts, GM-J pact)
give(GM-J pact, US car maker, 30% stake)

SRL

- SRL gets closer to the semantics of the sentence
- Some have used SRL parses to extract logical propositions from text

Sasha broke the window

Pat opened the door

$$\exists e, x, y \; Breaking(e) \wedge Breaker(e, Sasha) \wedge BrokenThing(e, y) \wedge Window(y)$$

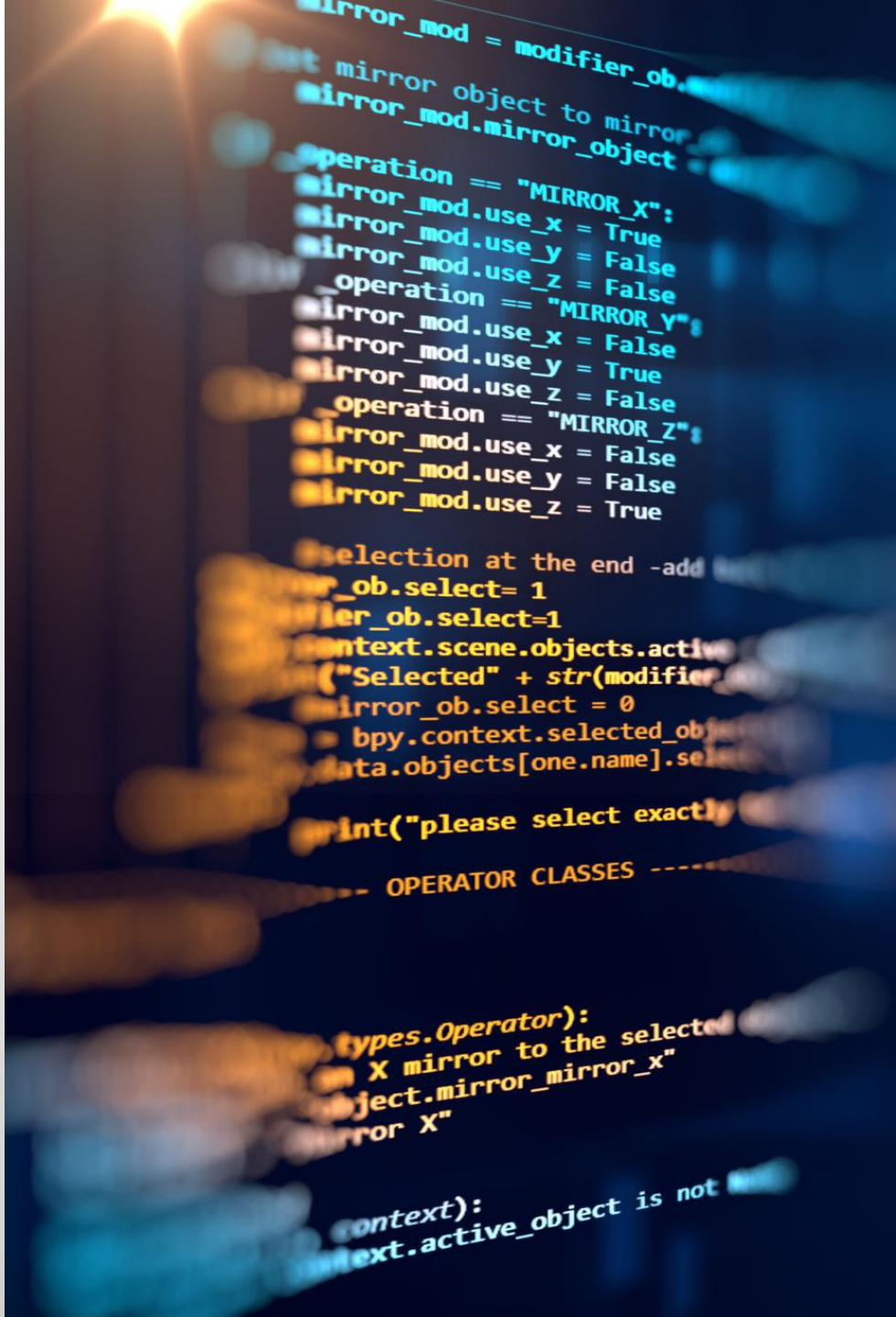
$$\exists e, x, y \; Opening(e) \wedge Opener(e, Pat) \wedge OpenedThing(e, y) \wedge Door(y)$$

SRL parsers

- Allen NLP
- SENNA

Other Software Tools

- Stanford
- spaCy



Stanford CoreNLP

- Toolkit from Stanford NLP group
- Many annotations, including tokenization, POS tagging, dependency parsing, and more
- To use Stanford CoreNLP:
  - 1. download CoreNLP
  - 2. Unzip the download
  - 3. Update classpath variables to point to the expanded directory

Stanford CoreNLP

- Setting up annotations in Java

Code 10.4.1 — Stanford CoreNLP. Java implementation

```
// Select the annotations
Properties props = new Properties();
props.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner, parse,
            dcoref, sentiment");

// Set up the pipeline
StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

// Run all the selected Annotators on this text
pipeline.annotate(annotation);
```

Python Wrapper: Stanza

- install:

```
$pip3 install stanza
$stanza.download('en')
```

Code 10.4.2 — Stanford CoreNLP. Stanza Python wrapper

```
import stanza

# set up the pipeline
nlp = stanza.Pipeline('en')

# set up the doc object on text
text = "Barack Obama was born in Hawaii.  He was elected president in 2008."
doc = nlp(text)

for sentence in doc.sentences:
    sentence.print_dependencies()
```

```
('Barack', 4, 'nsubj:pass')
('Obama', 1, 'flat')
('was', 4, 'aux:pass')
('born', 0, 'root')
('in', 6, 'case')
('Hawaii', 4, 'obl')
('.', 4, 'punct')
('He', 3, 'nsubj:pass')
('was', 3, 'aux:pass')
('elected', 0, 'root')
('president', 3, 'xcomp')
('in', 6, 'case')
('2008', 3, 'obl')
('.', 3, 'punct')
```

spaCY

- See Notebook https://github.com/kjmazidi/NLP/blob/master/Part_3-Sentences/Chapter_10_Parsing/Dependency_parse_spaCy.ipynb

Practice

- The quick brown fox jumped over the lazy river.

Essential points to note

- NLP practitioners have three types of sentences syntax parsers:
- PSG
- Dependency
- SRL
- And many NLP toolkits to use
- Sentence parsing is an important preprocessing step for many NLP applications

Next topic

Documents