Homework Three: Natural Language Processing

Quin'darius Lyles-Woods

October 30, 2021

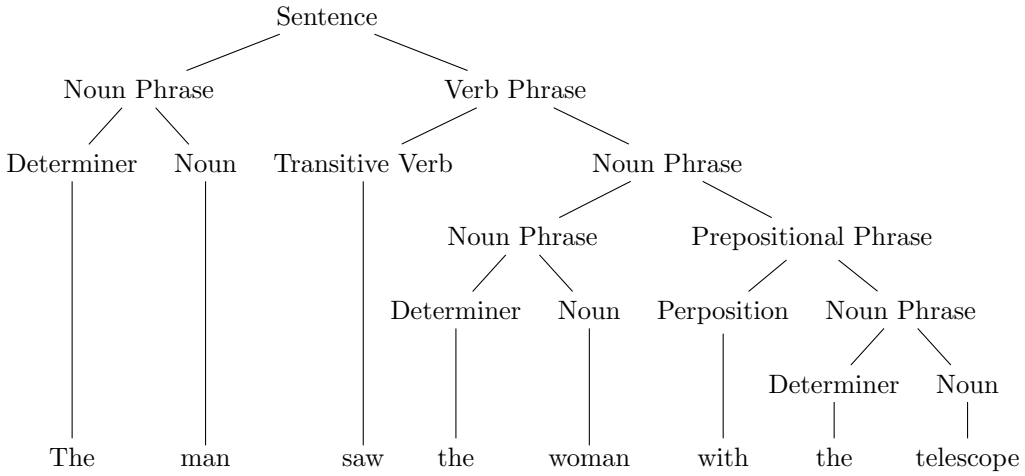
1 Probabilistic Parse Trees

$$
\begin{aligned}
\text{SENTENCE} &\mapsto \text{NOUN PHRASE} \quad \text{VERB PHRASE} \\
\text{NOUN PHRASE} &\mapsto \text{NOUN PHRASE} \quad \text{PREPOSITIONAL PHRASE} \\
\text{NOUN PHRASE} &\mapsto \text{DETERMINER} \quad \text{NOUN} \\
\text{VERB PHRASE} &\mapsto \text{TRANSITIVE VERB} \quad \text{NOUN PHRASE} \\
\text{VERB PHRASE} &\mapsto \text{VERB PHRASE} \quad \text{PREPOSITIONAL PHRASE} \\
\text{VERB PHRASE} &\mapsto \text{INTRANSITIVE VERB} \\
\text{PREPOSITIONAL PHRASE} &\mapsto \text{PREPOSITION} \quad \text{NOUN PHRASE} \\
\text{NOUN} &\mapsto \text{man} \\
\text{NOUN} &\mapsto \text{woman} \\
\text{NOUN} &\mapsto \text{telescope} \\
\text{INTRANSITIVE VERB} &\mapsto \text{sleeps} \\
\text{TRANSITIVE VERB} &\mapsto \text{saw} \\
\text{DETERMINER} &\mapsto \text{the} \\
\text{PREPOSITION} &\mapsto \text{with} \\
\text{PREPOSITION} &\mapsto \text{in}
\end{aligned}
$$

1.1 Parse Tree

Sentence

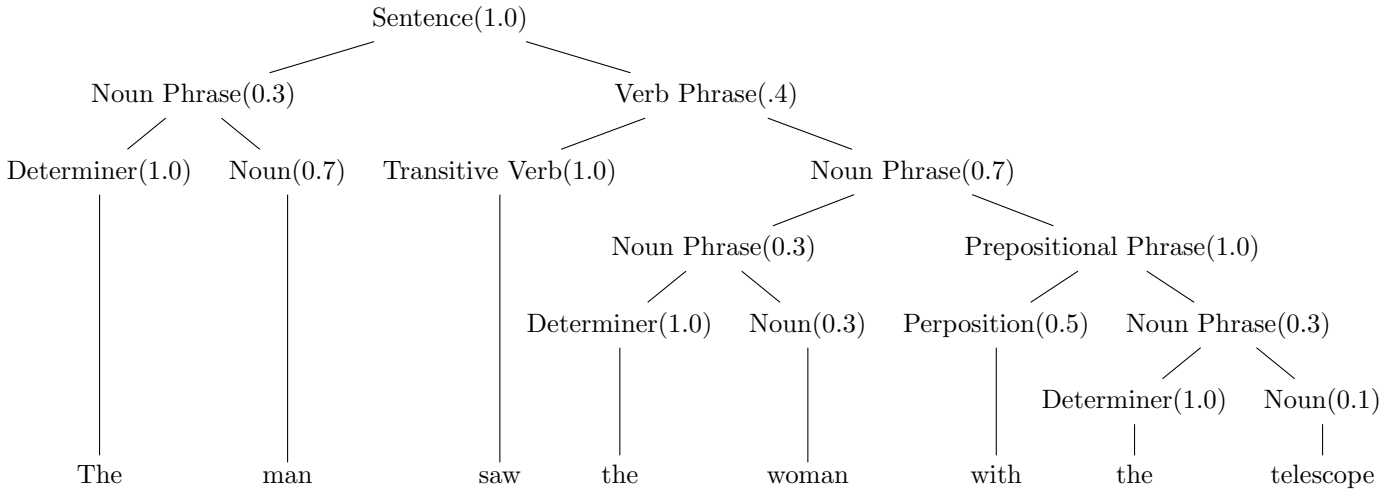
The man saw the woman with the telescope



1.2 Probabilistic Parse Tree

Sentence One

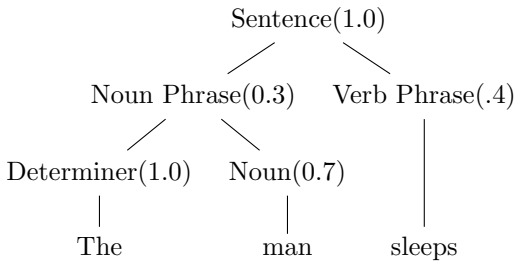
The man saw the woman with the telescope



$$Probabilistic\,Tree = 1.0 * 0.3 * 1.0 * 0.7 * 0.4 *$$
$$1.0 * 0.7 * 0.3 * 1.0 * 0.3 * 1.0 * 0.5 *$$
$$0.3 * 1.0 * 0.1 = 0.00007938$$

Sentence Two

The man sleeps



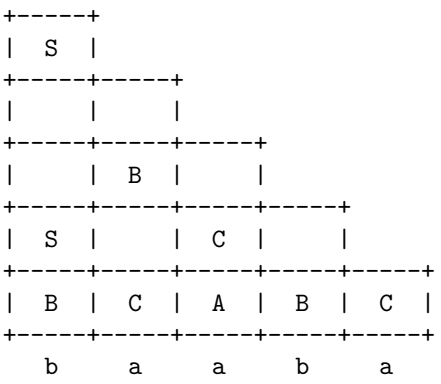
$$Probabilistic \quad Tree \quad = \quad 1.0 * 0.3 * 1.0 * 0.7 * 0.4 \quad = \quad 0.084$$

2 Grammar Legality with Cocke-Younger-Kasami Algorithm

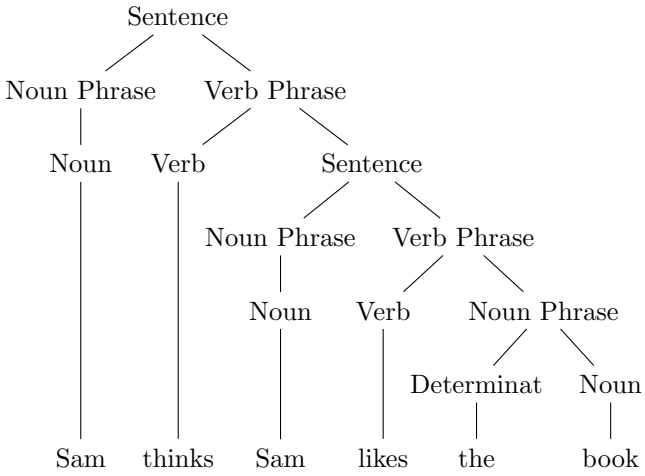
$$\begin{aligned} \text{SENTENCE} &\mapsto \text{AB} \\ &\mapsto \text{BC} \\ \text{A} &\mapsto \text{BA} \\ &\mapsto \text{a} \\ \text{B} &\mapsto \text{CC} \\ &\mapsto \text{b} \\ \text{C} &\mapsto \text{AB} \\ &\mapsto \text{a} \end{aligned}$$

Sentence

baaba



3 Bracketed Notation



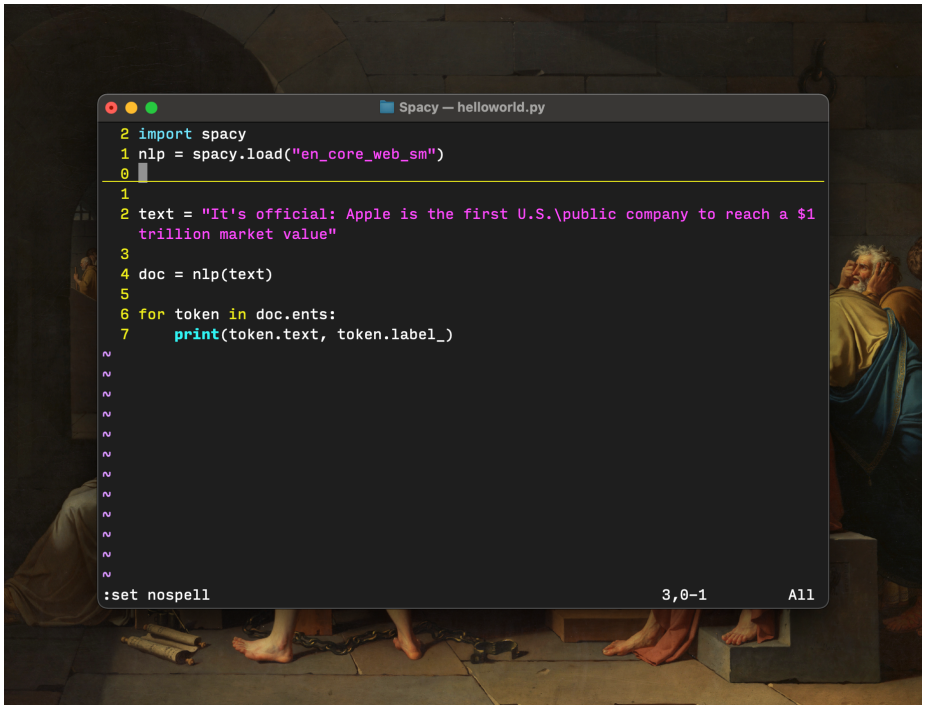
$$[Sentence[NounPhrase[Noun[Sam]]][VerbPhrase[Verb[thinks]][Sentence [NounPhrase[Noun[Sam]]][VerbPhrase[Verb[likes]] [NounPhrase[Determinat[the]][Noun[book]]]]]]]$$

4 Tokenization with Spacy

4.1 String Processing

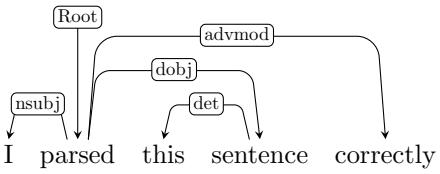
In Spacy, text are processed by taking a string and converting the in a Doc. The process of turning them into a doc varies from program to program depending upon on what pipeline you have set up in your program. You can create custom pipelines for the processing your document.

4.2 Text Tokenization



5 Transition Dependency Parsing

Sentence



States

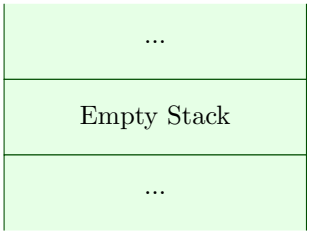
- Start State
  - Stack intialized with Root node
  - Input buffer with words of the sentence
  - Dependency relation set is empty
- Transition State
  - Producing a new configuation

- End State
  - Stack and word list are empty
  - Dependency relation set is finalized

Transition Operators

- LEFT-ARC
  - Create a head-dependent relation[1] between word at the top of the stack and the word under the top.
  - Remove 2nd word from the stack
- RIGHT-ARC
  - Create a head dependent relation between the word on the bottom of the top and the top of the stack.
  - Remove the word at the top of the stack
- SHIFT
  - Remove the word at the head of the input buffer.
  - Push that word onto the stack

Stack



Input Buffer

I parsed this sentence correctly

Operations

Relations

I parsed this sentence correctly

[1] In linguistics, the head or nucleus of a phrase is the word that determines the syntactic category of that phrase.

Stack

...
I
...

Input Buffer

parsed this sentence correctly

Operations

1. SHIFT

Relations

I parsed this sentence correctly

Stack

...
parsed
I
...

Input Buffer

this sentence correctly

Operations

1. SHIFT
2. SHIFT
3. left-arc

Relations

I parsed this sentence correctly

Stack

...
sentence
this
parsed
...

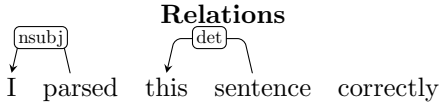
Input Buffer

correctly

Operations

1. SHIFT
2. SHIFT
3. left-arc
4. SHIFT
5. SHIFT
6. left-arc

Relations



Stack

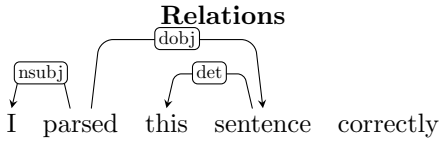
...
correctly
parsed
...

Input Buffer

Operations

1. SHIFT
2. SHIFT
3. left-arc
4. SHIFT
5. SHIFT
6. left-arc
7. right-arc
8. SHIFT

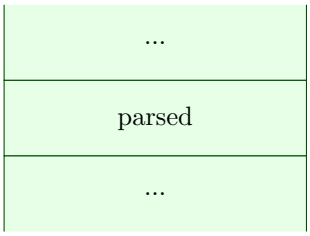
Relations



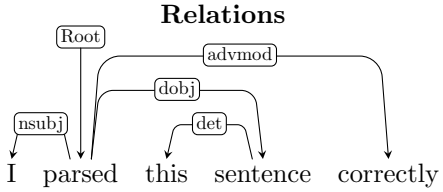
Stack **Input Buffer**

Operations

1. SHIFT
2. SHIFT
3. left-arc
4. SHIFT
5. SHIFT
6. left-arc
7. right-arc
8. SHIFT
9. right-arc



Relations



6 Comparison of Constituency and Dependency Parsing

6.1 Constituency Parsing

When you are Constituency parsing a sentence or a corpus you are breaking it down into constituents[2]. Deals mainly with the *syntax* of the given sentence or corpus.

6.2 Dependency Parsing

When you are dependency parsing you are deriving than *semantic* relationships rather than the *syntax* This is done by finding what words are dependent on which words creating binary relationships between words.

[2]In linguistics this is a word or construction that is part of a larger construction.