Parsing

Luciano Barbosa

(baseado nos slides do curso de PLN de Stanford e livro Speech and Language Processing)

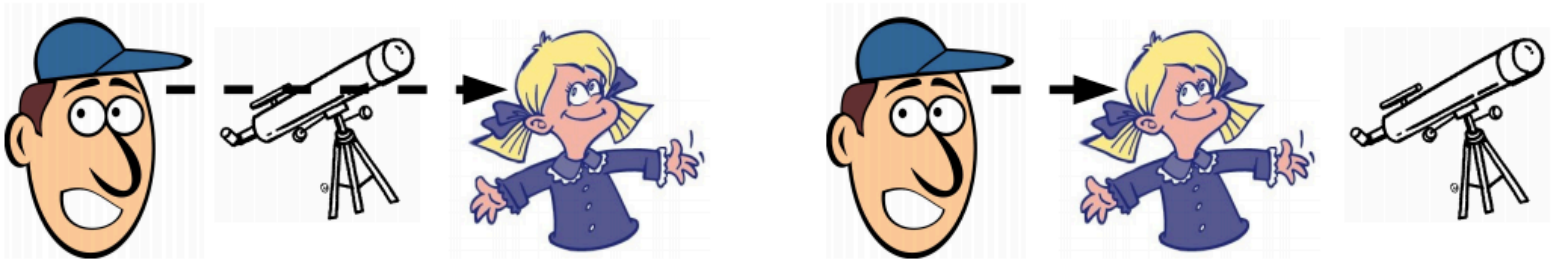
Centro de Informática UFPE

Universidade Federal de Pernambuco

Ambiguidades em Interpretar Linguagem

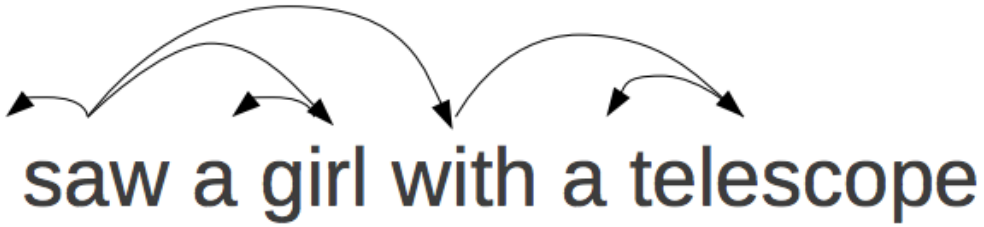
- Parsing resolve ambiguidade estrutural de uma maneira formal

I saw a girl with a telescope

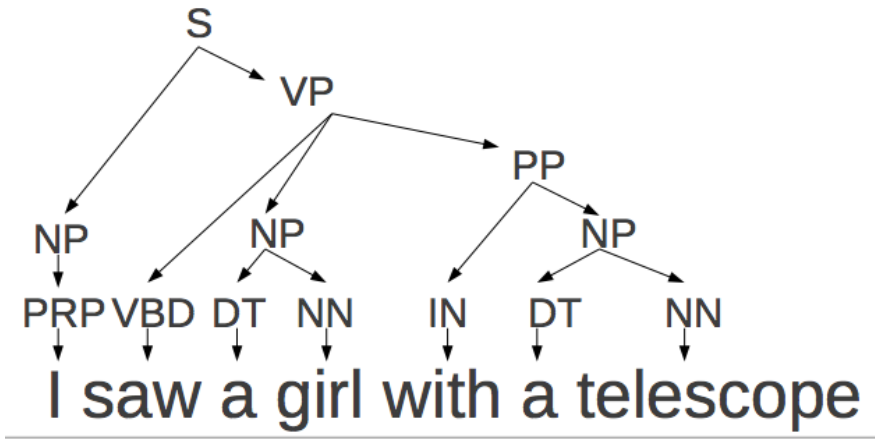


Dois Tipos de Parsing

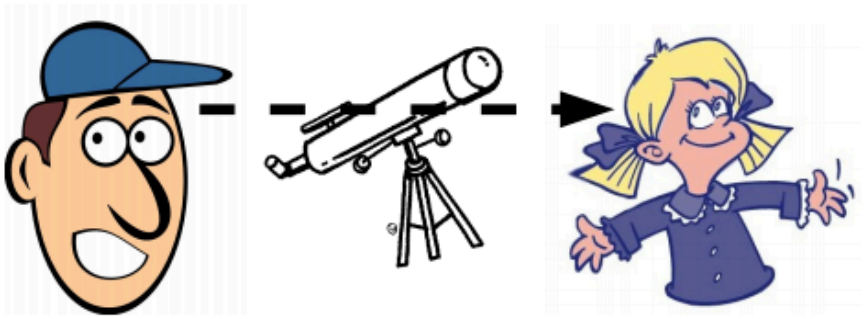
- Dependency: relações entre palavras

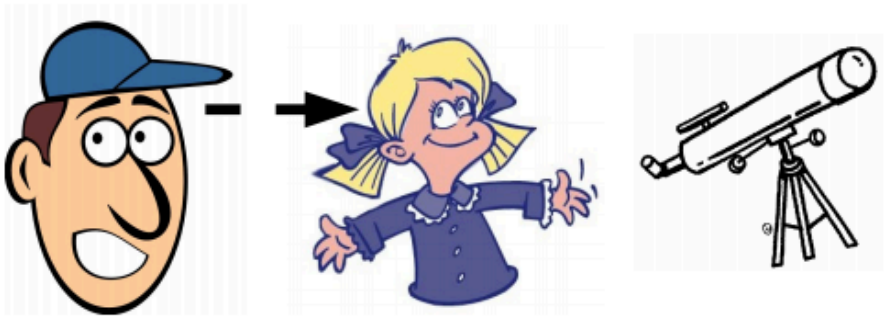


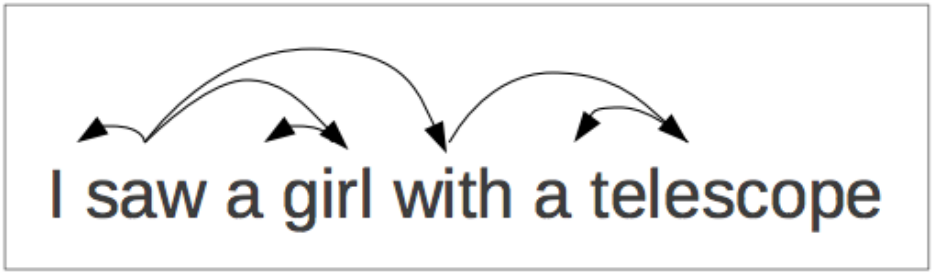
- Constituency: identifica frases e sua estrutura recursiva

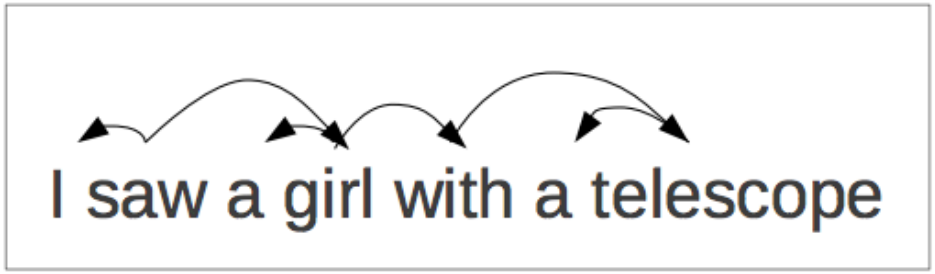


Resolvendo Ambiguidade com Dependências



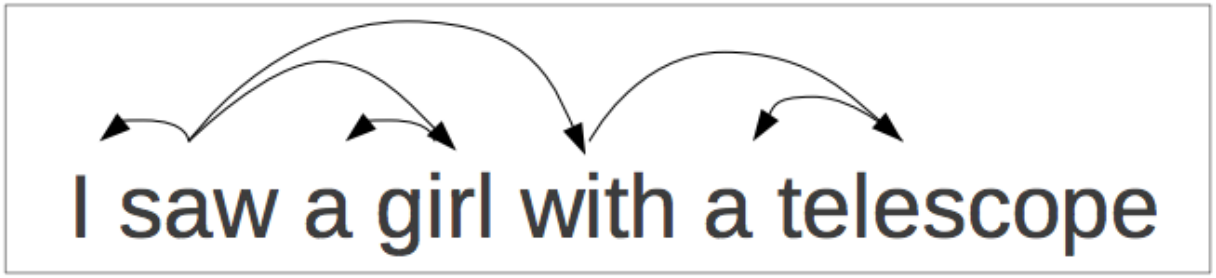






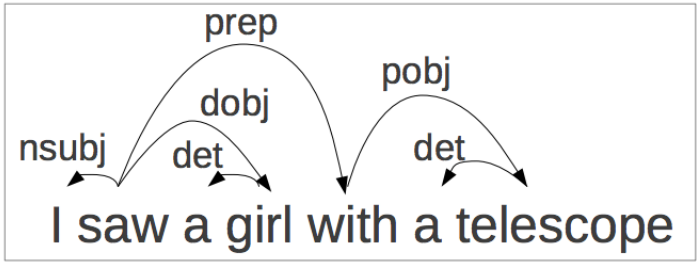
Tipos de Dependências

- Não-tipadas: somente dependências



Tipos de Dependências

- Tipadas: rótulos indicam o relacionamento entre palavras



Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Tipos de Dependências

- Tipadas: rótulos indicam o relacionamento entre palavras

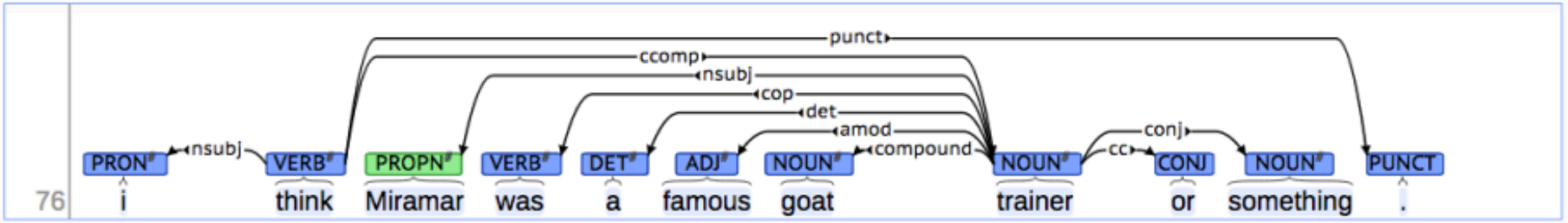
Relation	Examples with *head* and **dependent**
NSUBJ	**United** *canceled* the flight.
DOBJ	United *diverted* the **flight** to Reno.
	We *booked* her the first **flight** to Miami.
IOBJ	We *booked* **her** the flight to Miami.
NMOD	We took the **morning** *flight*.
AMOD	Book the **cheapest** *flight*.
NUMMOD	Before the storm JetBlue canceled **1000** *flights*.
APPOS	*United*, a **unit** of UAL, matched the fares.
DET	**The** *flight* was canceled.
	Which *flight* was delayed?
CONJ	We *flew* to Denver and **drove** to Steamboat.
CC	We flew to Denver **and** *drove* to Steamboat.
CASE	Book the flight **through** *Houston*.

Preferências na Dependência

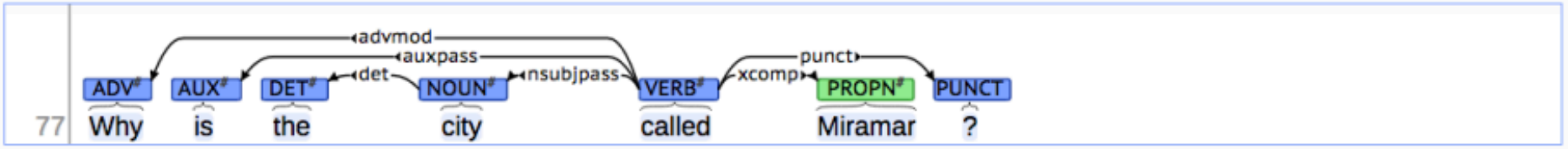
- Distância: palavras próximas
- Raramente vai além dos verbos e pontuações
- Não há ciclos (estrutura de árvore)
- Tem uma raiz

Dados Anotados: Treebank

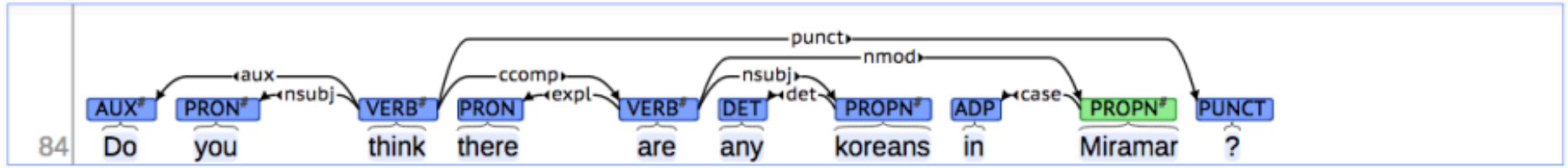
[context] [conllu]



[context] [conllu]



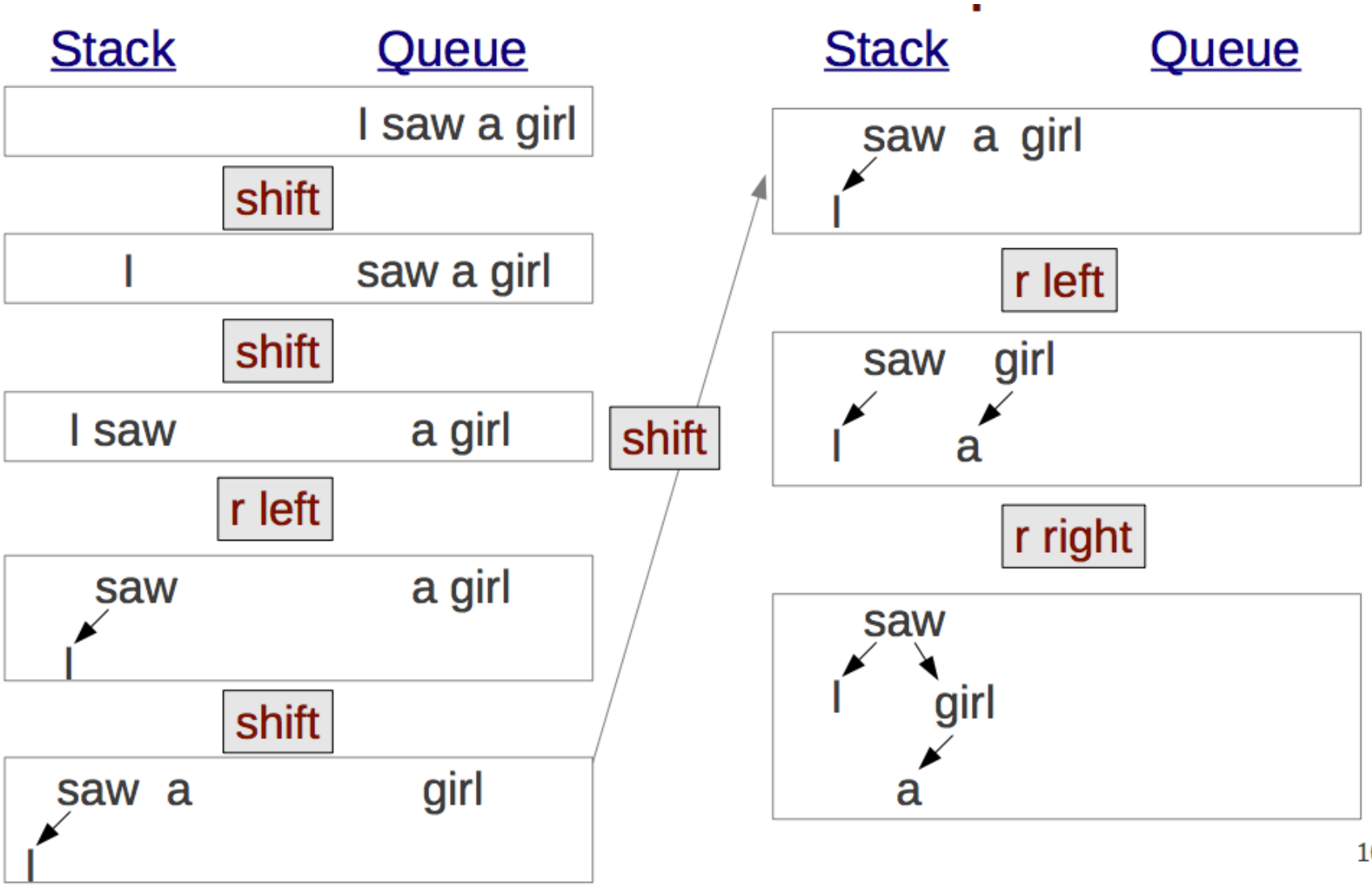
[context] [conllu]



Algoritmo Shift-Reduce

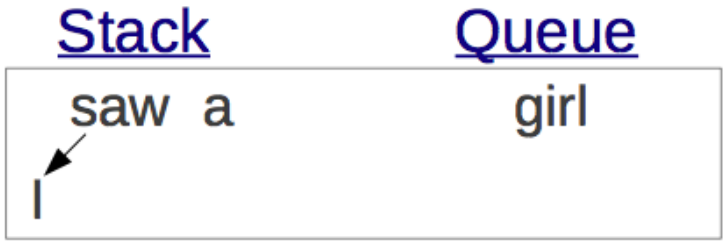
- Processa uma palavra por vez da esquerda para a direita
- Duas estruturas
  - Queue: palavras não processadas
  - Stack: palavras parcialmente processadas
- Em cada ponto escolha
  - shift: move uma palavra do queue para o stack
  - reduce left: palavra no topo do stack é a cabeça da segunda palavra
  - reduce right: segunda palavra no stack é a cabeça da palavra no topo
- Aprender cada tarefa utilizando um classificador

Exemplo do Shift Reduce



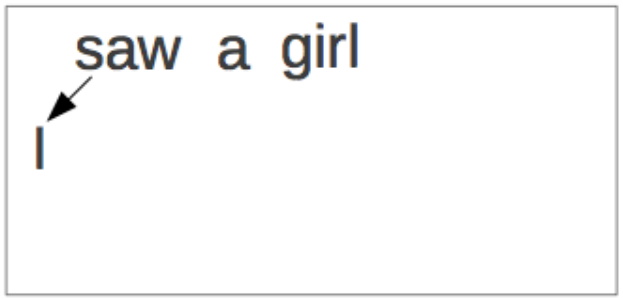
Classificação em Shift-Reduce

- Dado o estado:

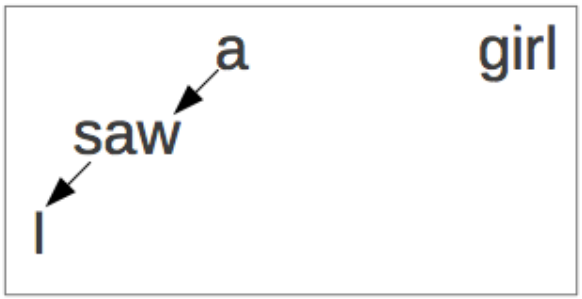


- Qual a próxima ação

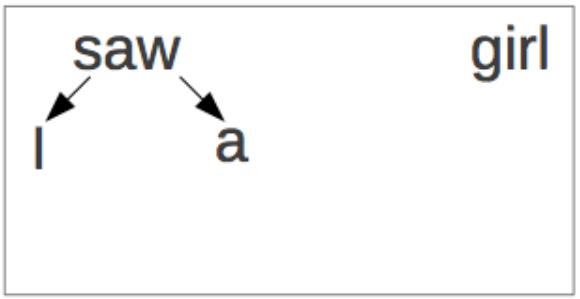
shift ?



r left ?



r right ?



Features

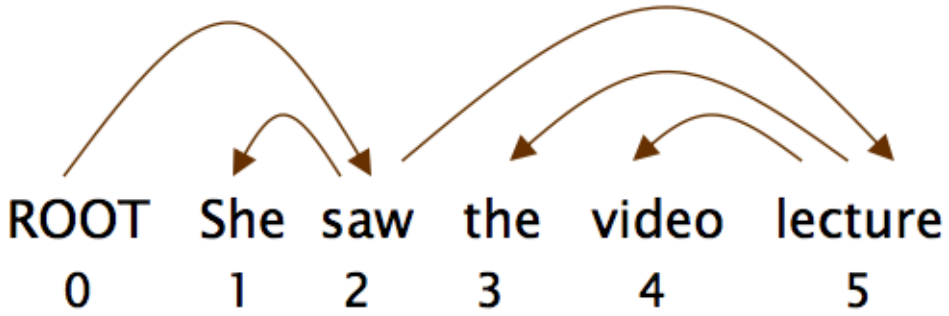
- POS
- Lema
- Palavra no topo do stack
- Primeira palava no queue

	stack[-2]	*stack*[-1]	*queue*[0]
Word:	saw	a	girl
POS:	VBD	DET	NN

(-2 → second-to-last)
(-1 → last)
(0 → first)

Avaliação

- Acurácia rotulada (LAS)e não rotulada (UAS)



$$\text{Acc} = \frac{\text{\# correct deps}}{\text{\# of deps}}$$

UAS = 4 / 5 = 80%
LAS = 2 / 5 = 40%

Gold			
1	2	She	nsubj
2	0	saw	root
3	5	the	det
4	5	video	nn
5	2	lecture	obj

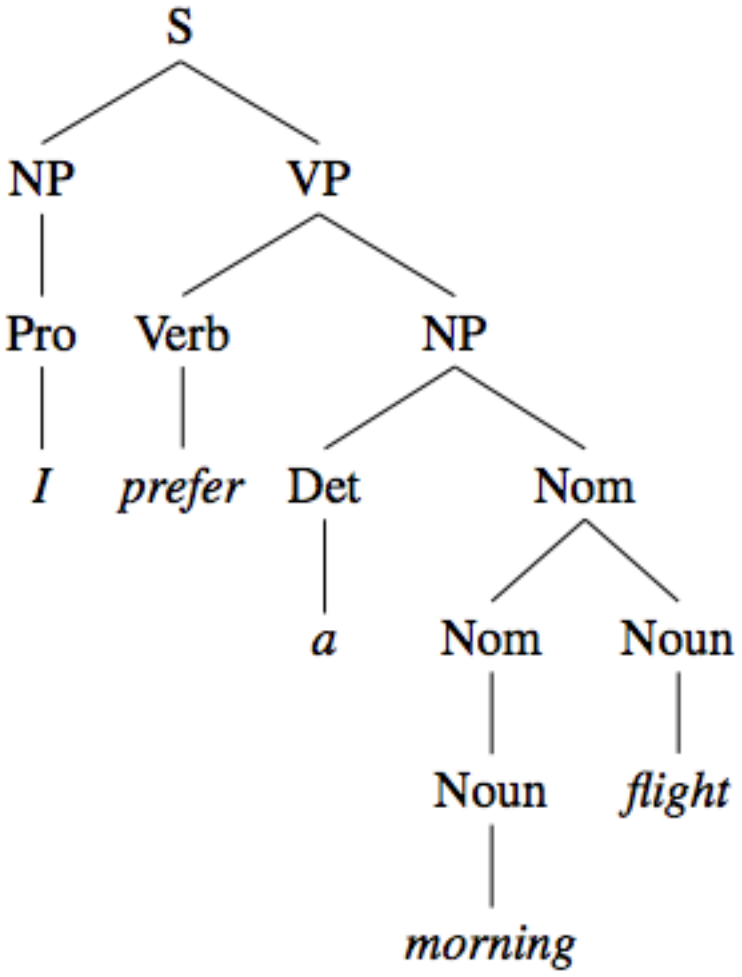
Parsed			
1	2	She	nsubj
2	0	saw	root
3	4	the	det
4	5	video	nsubj
5	2	lecture	ccomp

Context-Free Grammar (CFG)

Noun	→	*flights* | *breeze* | *trip* | *morning*
Verb	→	*is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective	→	*cheapest* | *non-stop* | *first* | *latest* | *other* | *direct*
Pronoun	→	*me* | *I* | *you* | *it*
Proper-Noun	→	*Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American*
Determiner	→	*the* | *a* | *an* | *this* | *these* | *that*
Preposition	→	*from* | *to* | *on* | *near*
Conjunction	→	*and* | *or* | *but*

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

Context-Free Grammar (CFG)

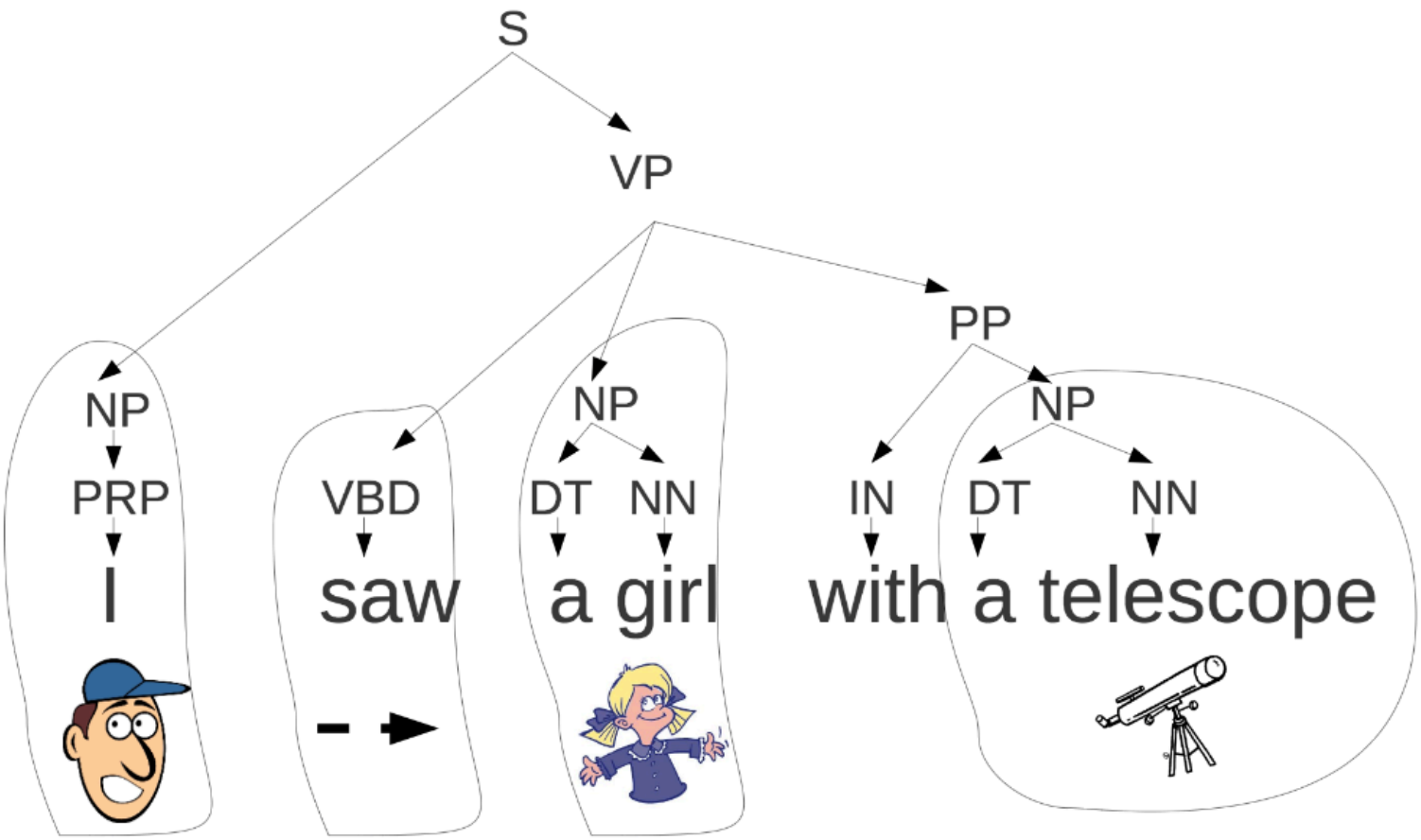


Treebanks

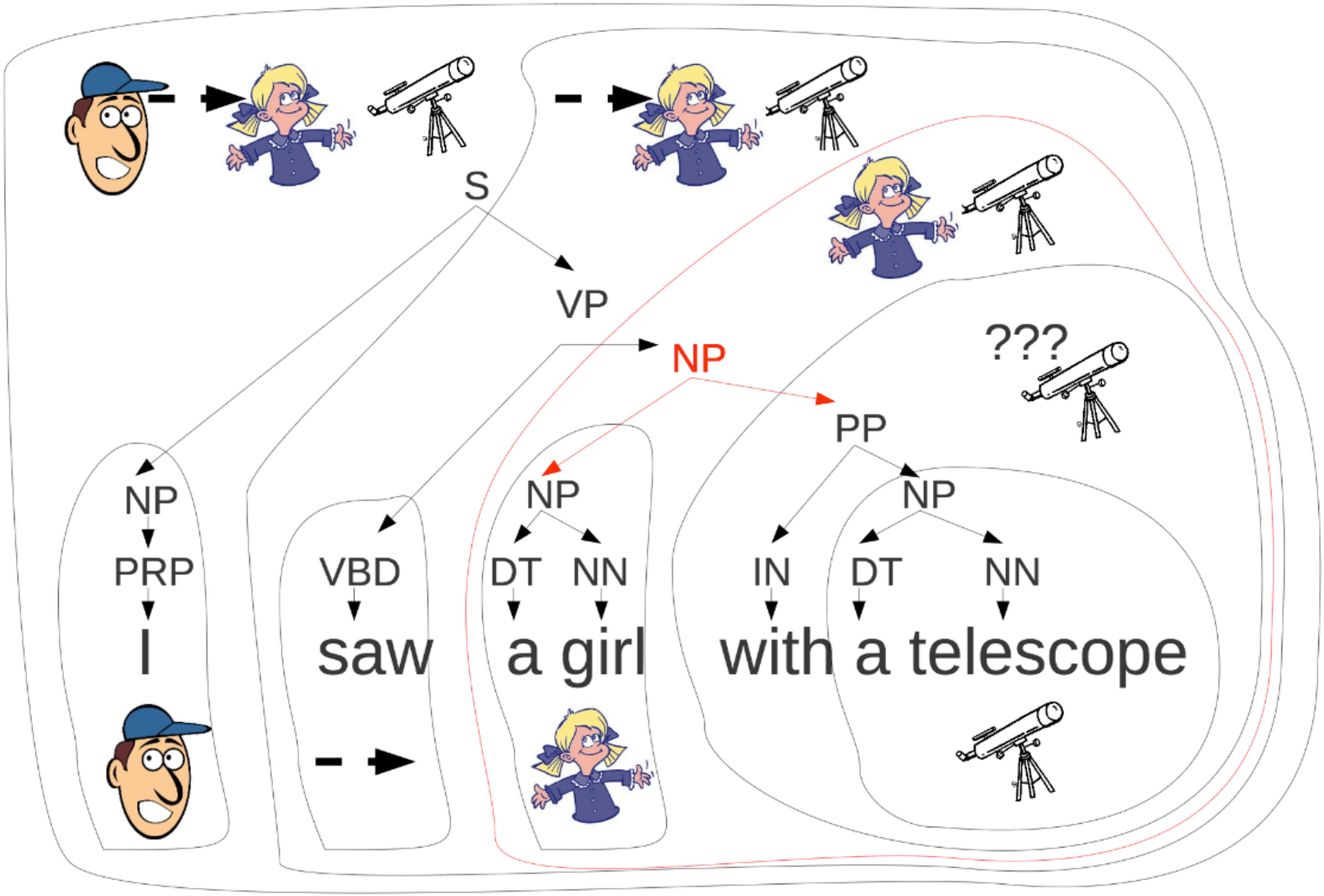
- Cada sentença com parse tree anotada
- Penn Treebank para inglês

```
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN director) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
```

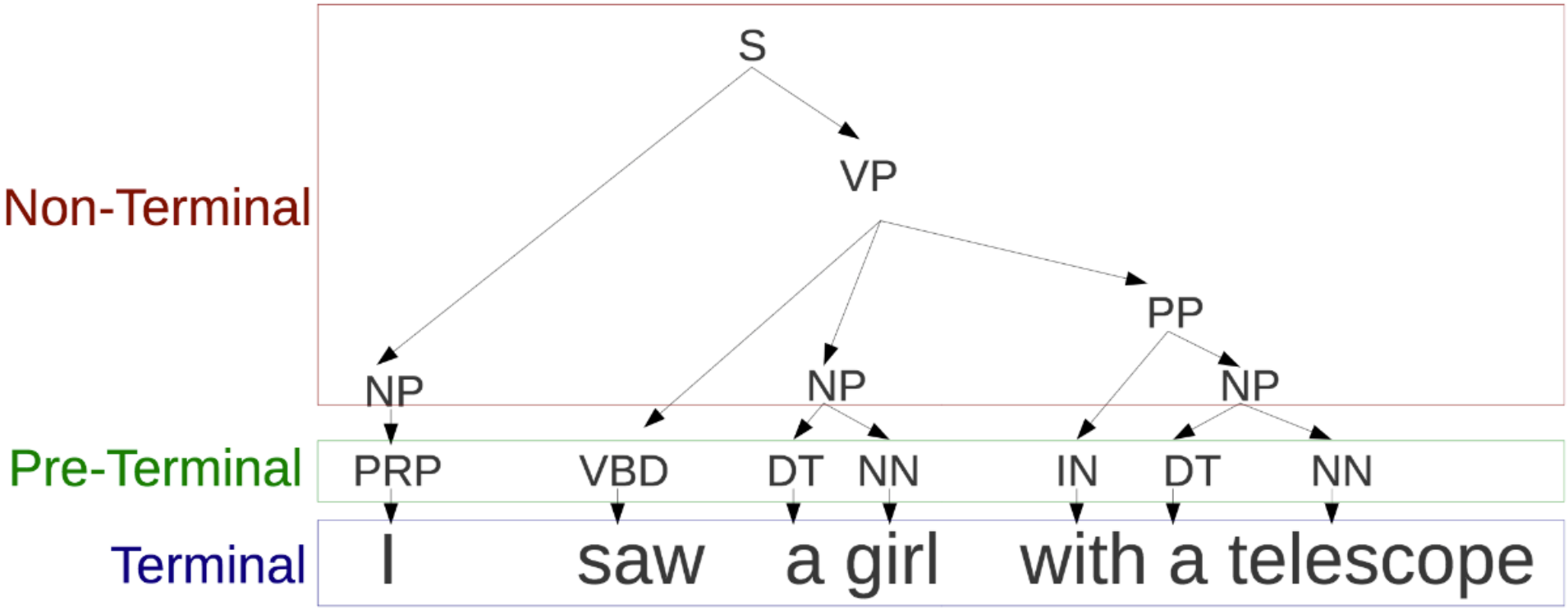
Ambiguidade



Ambiguidade

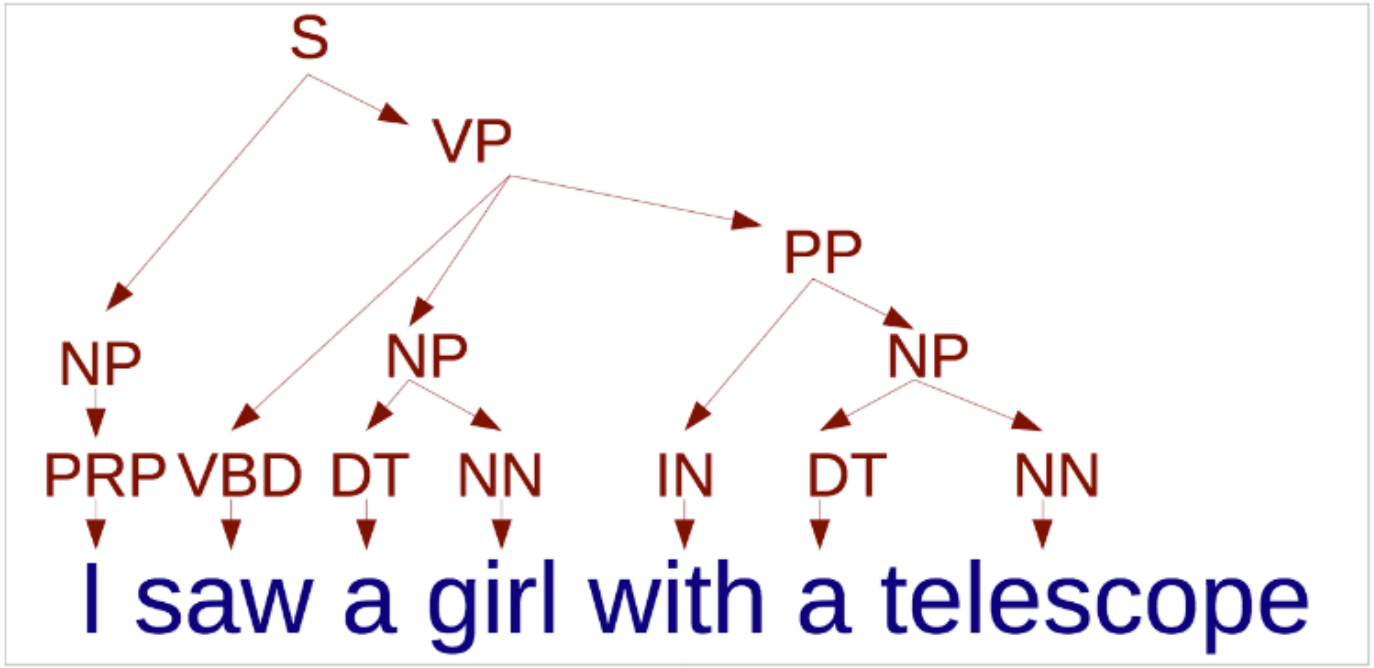


Componentes



Aprendendo uma Parse Tree

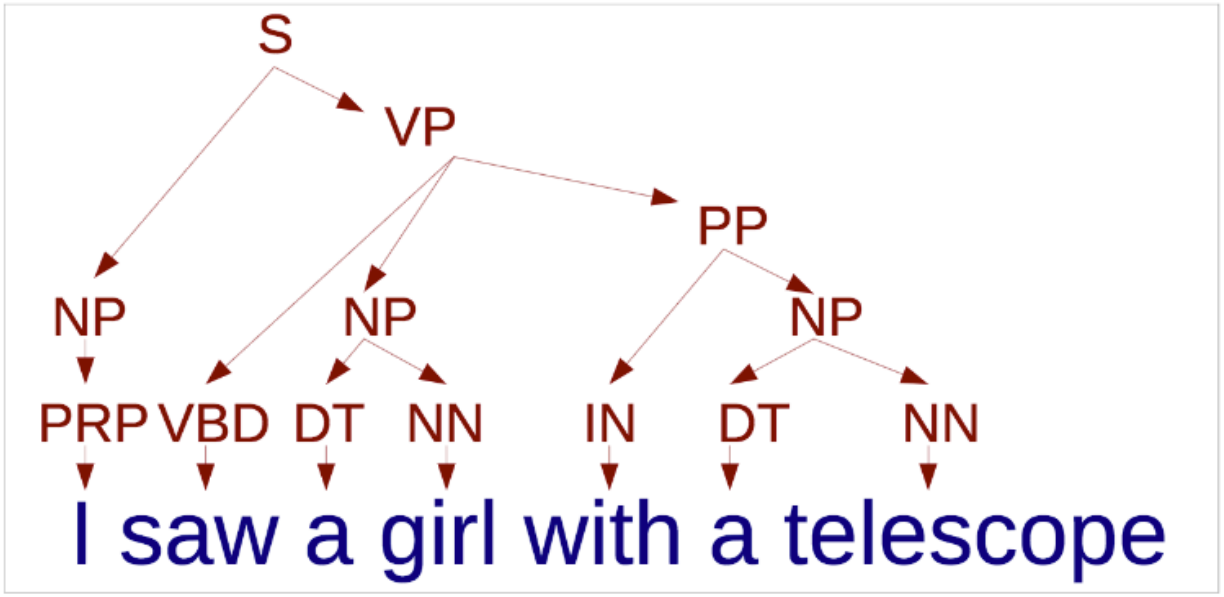
- Dada uma sentença X, predizer a parse tree



- Predição estruturada (similar a POS)

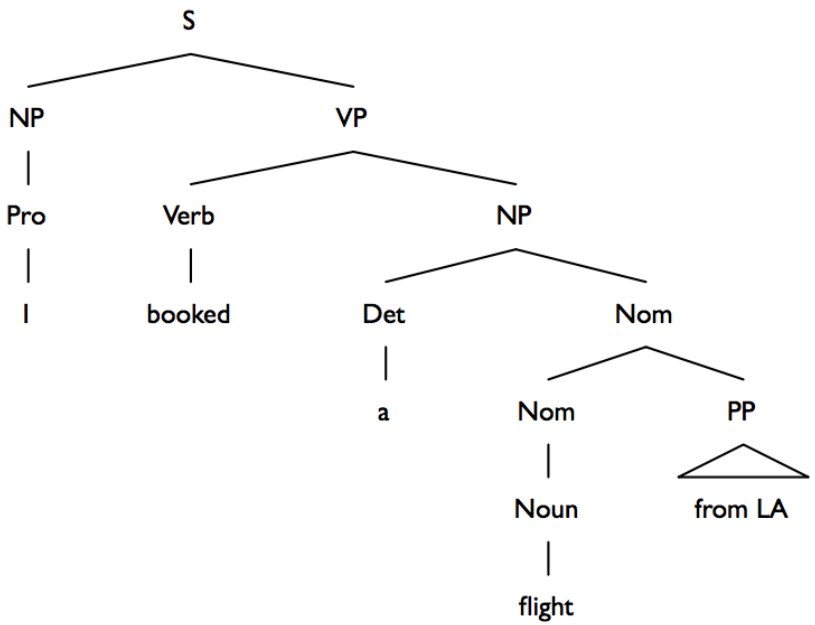
Abordagem Probabilística

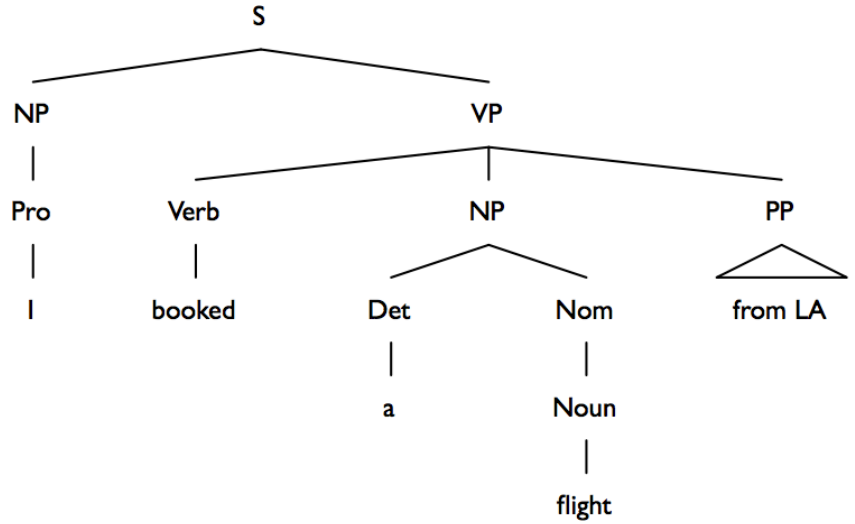
- Dada a sentença X, predizer a parse tree Y mais provável



$$\underset{Y}{\operatorname{argmax}} P(Y|X)$$

Escolha da Melhor Parse Tree

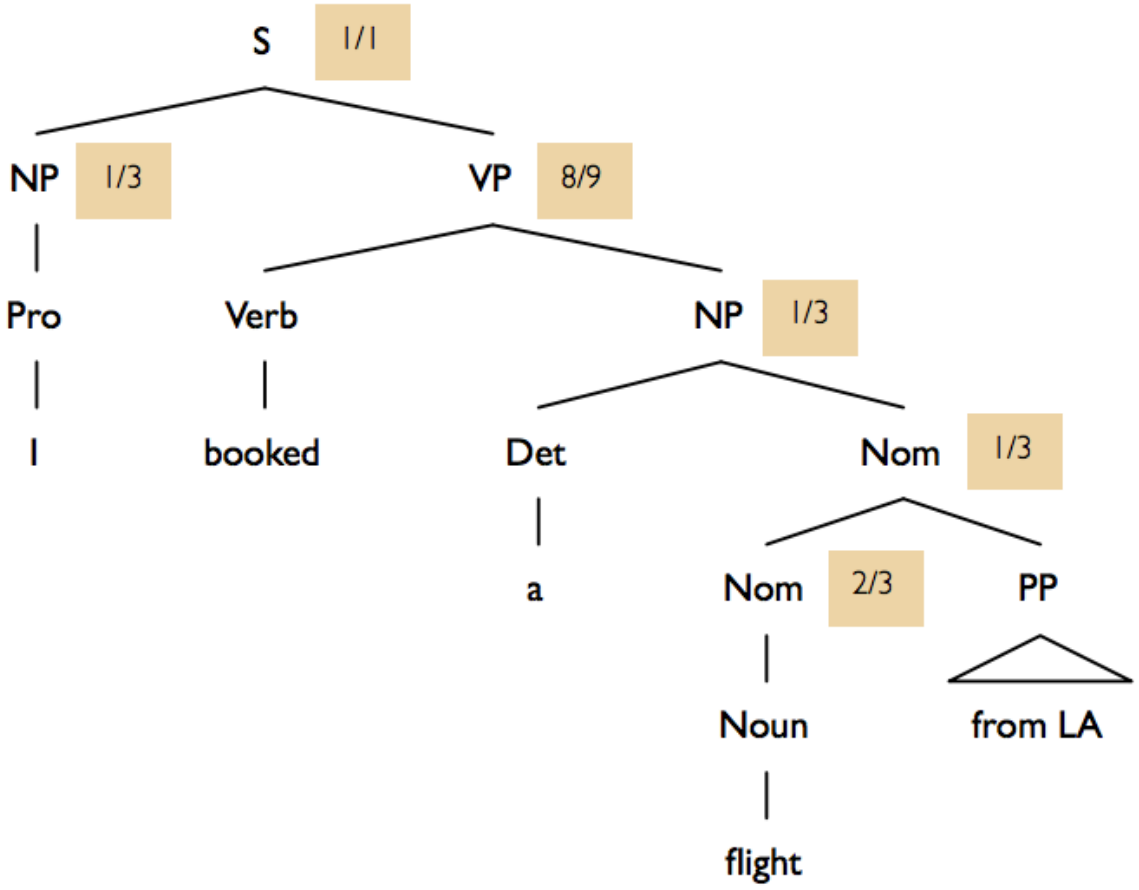




Rule	Probability
S → NP VP	1
NP → Pronoun	1/3
NP → Proper-Noun	1/3
NP → Det Nominal	1/3
Nominal → Nominal PP	1/3
Nominal → Noun	2/3
VP → Verb NP	8/9
VP → Verb NP PP	1/9
PP → Preposition NP	1

https://cl.lingfil.uu.se/~nivre/master/NLP-SyntaxParsing.pdf

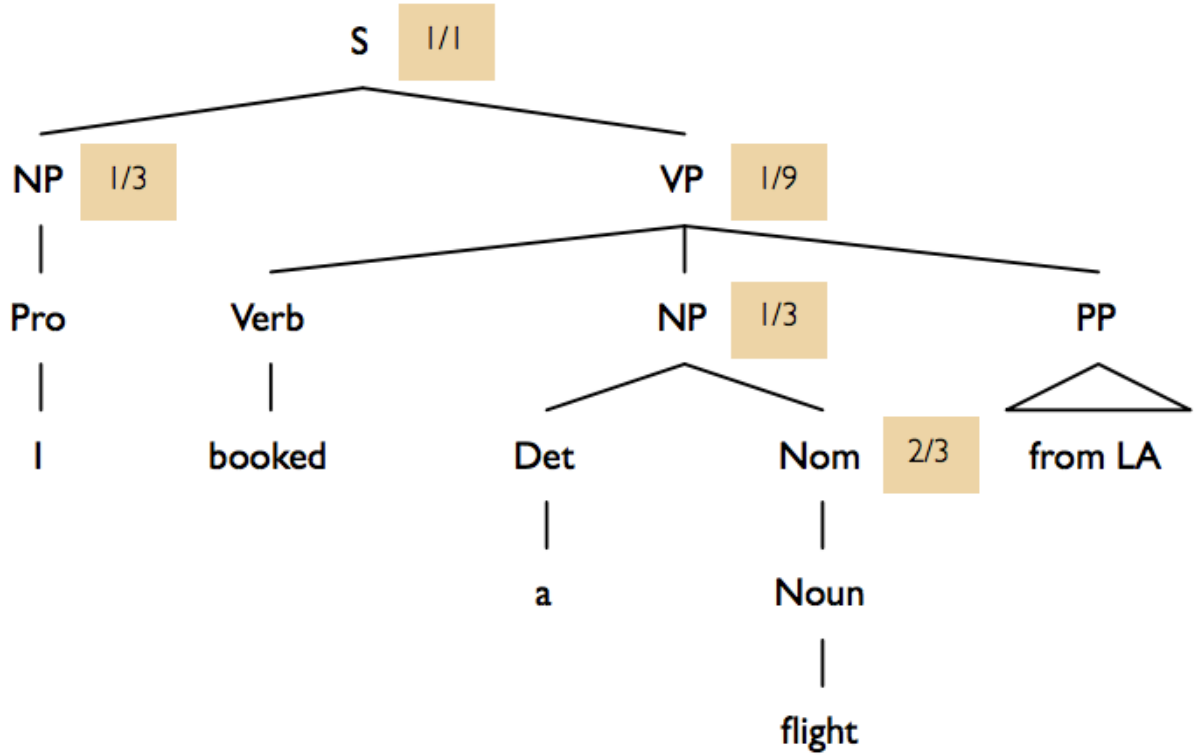
Escolha da Melhor Parse Tree



Probability: 16/729

https://cl.lingfil.uu.se/~nivre/master/NLP-SyntaxParsing.pdf

Escolha da Melhor Parse Tree



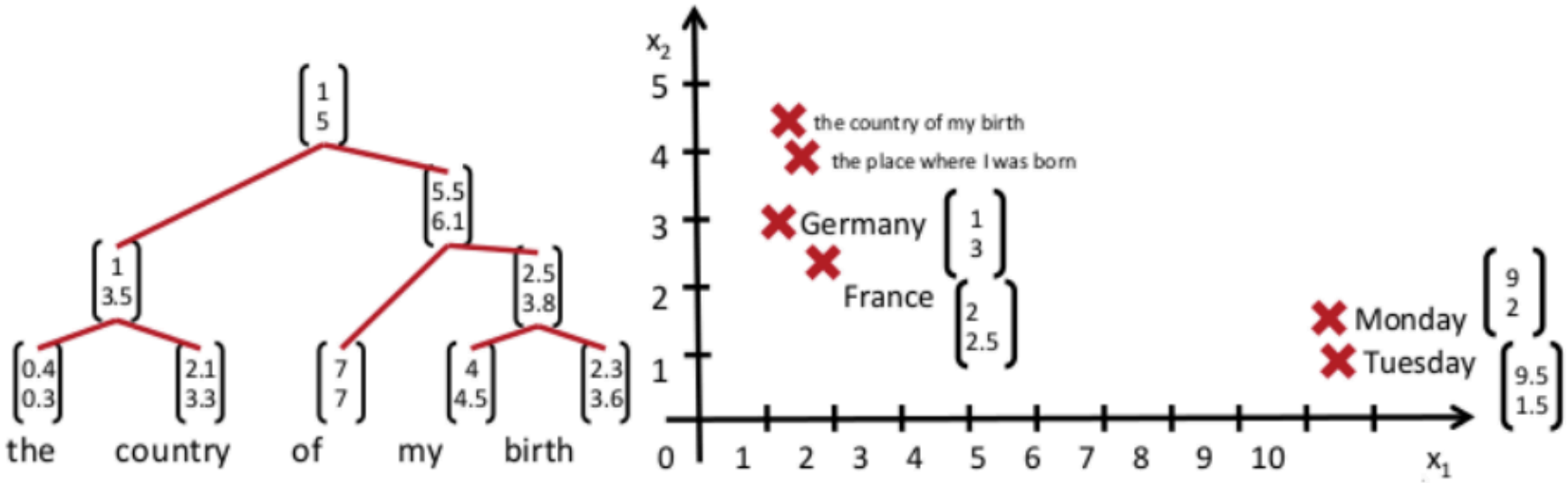
Probability: 6/729

https://cl.lingfil.uu.se/~nivre/master/NLP-SyntaxParsing.pdf

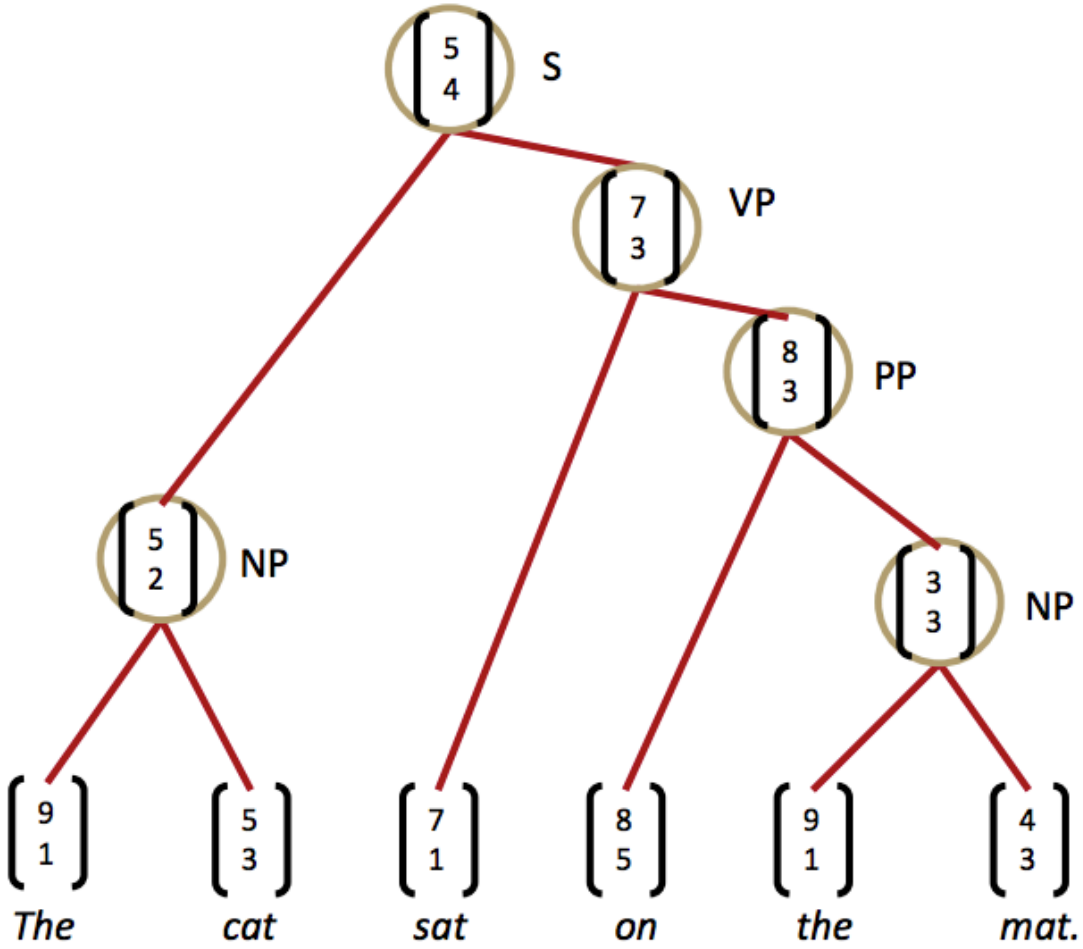
cin.ufpe.br

Representando Sentenças

- Significado de uma sentença
  - Significado das palavras
  - Regras que as combinam

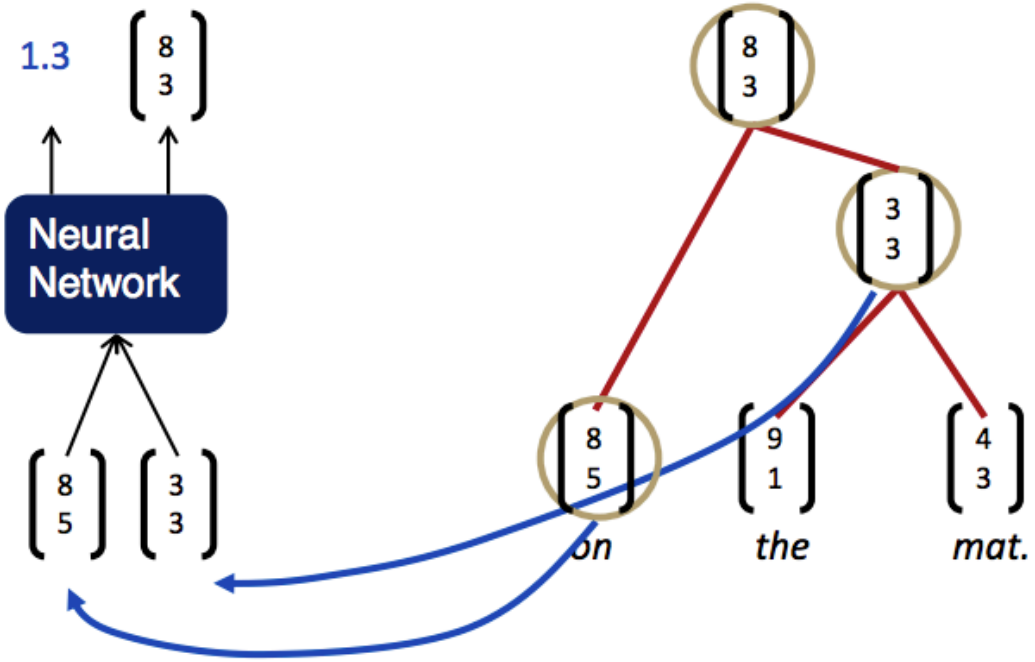


Representando Nós



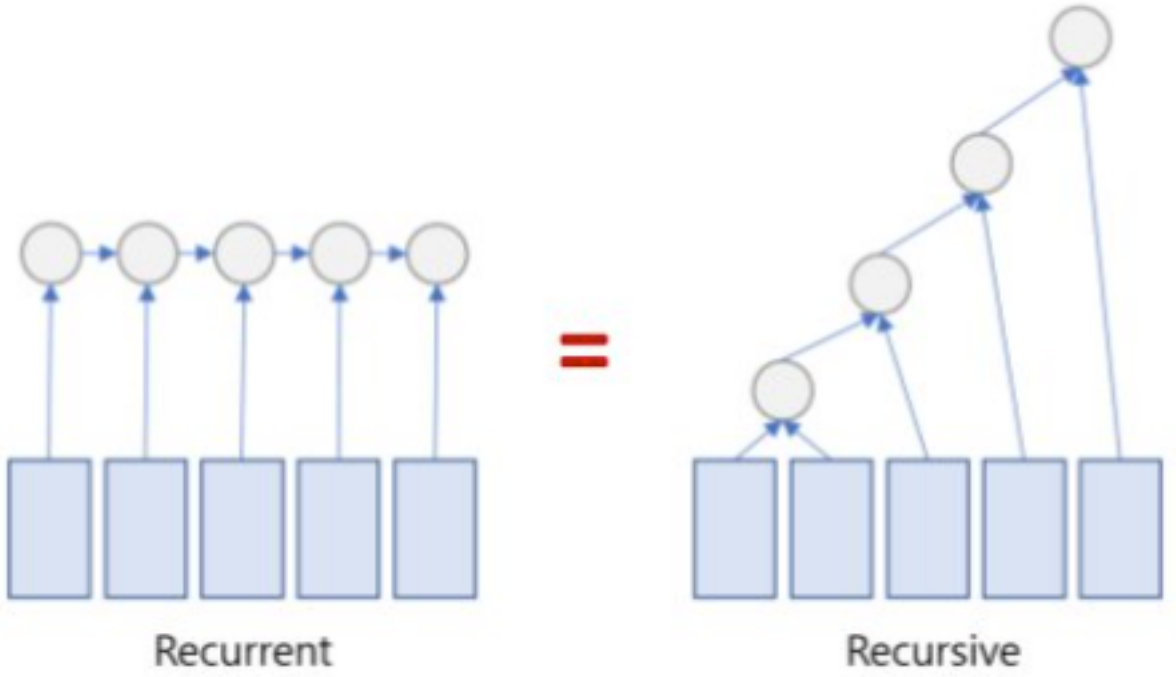
Recursive Neural Networks

- Entrada: representação dos nós filhos
- Saída:
  - A representação semântica dos nós
  - Score de quão plausível é essa junção

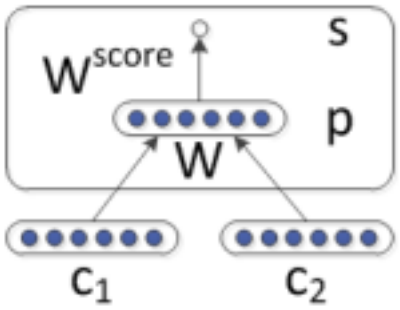


Recursive vs Recurrent

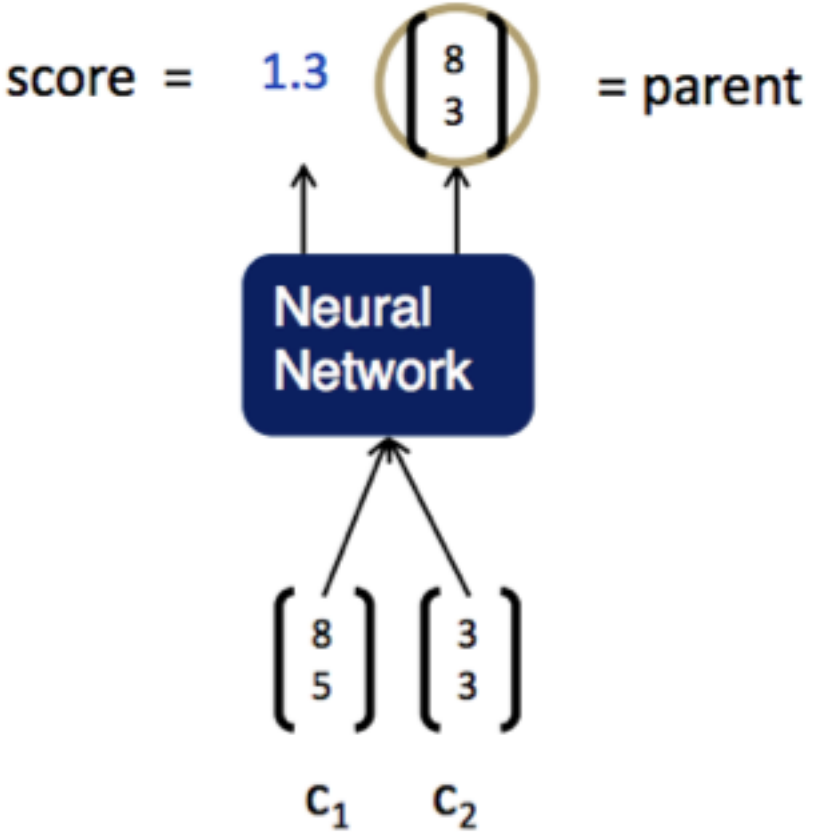
- Tipo de RNN



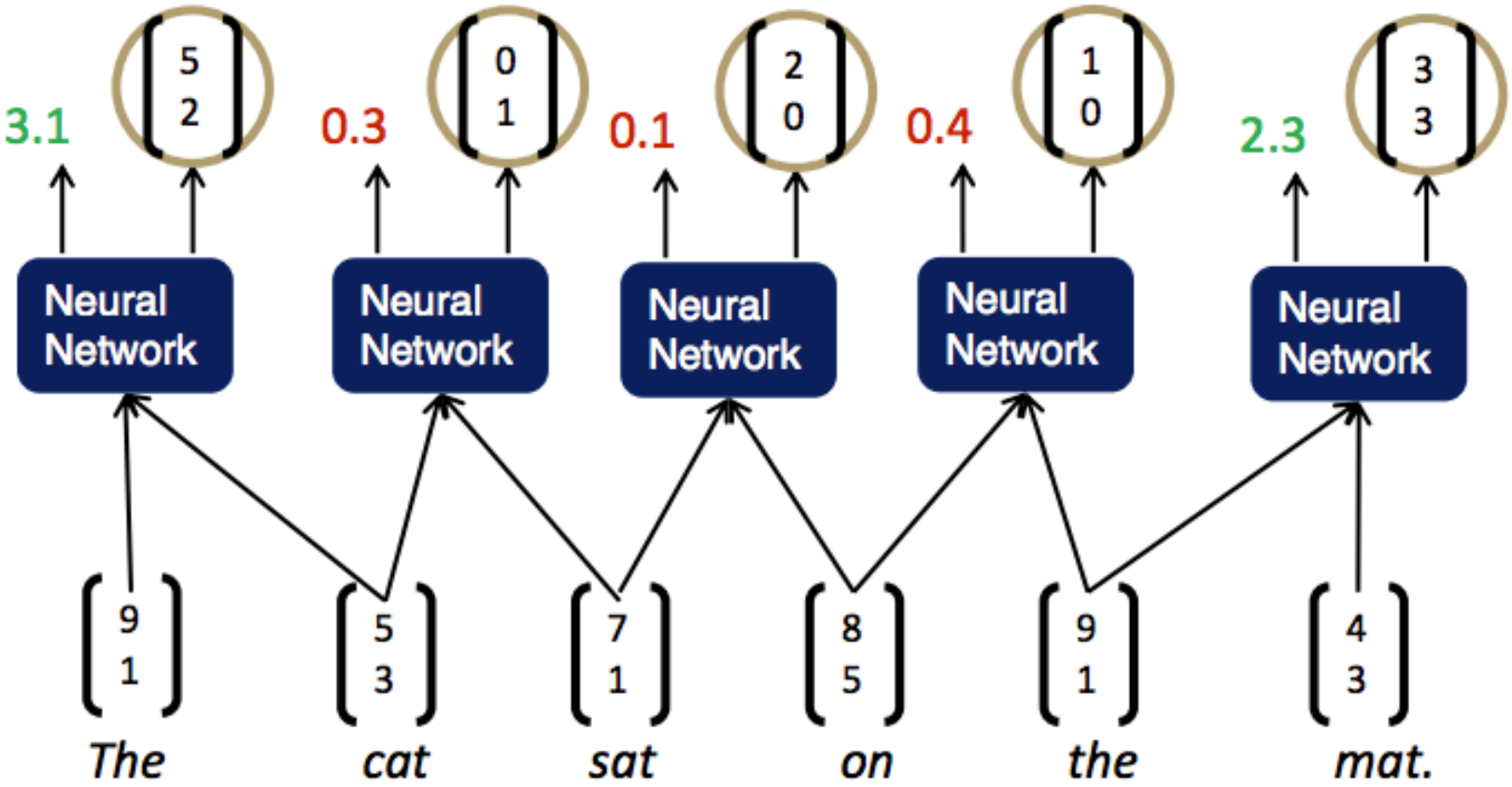
Recursive Neural Networks



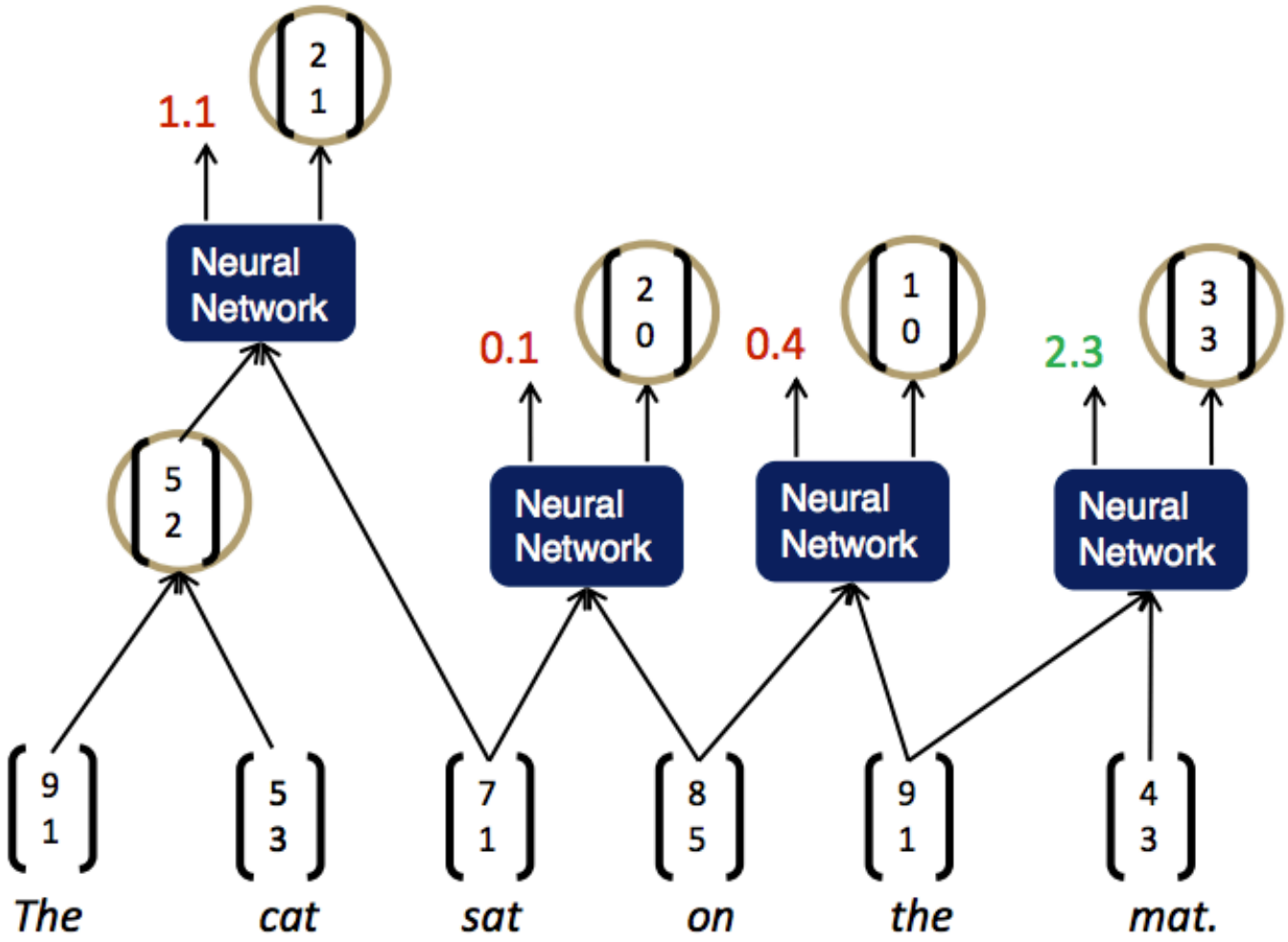
$$s = W^{score} p$$
$$p = f(W[c_1; c_2] + b)$$



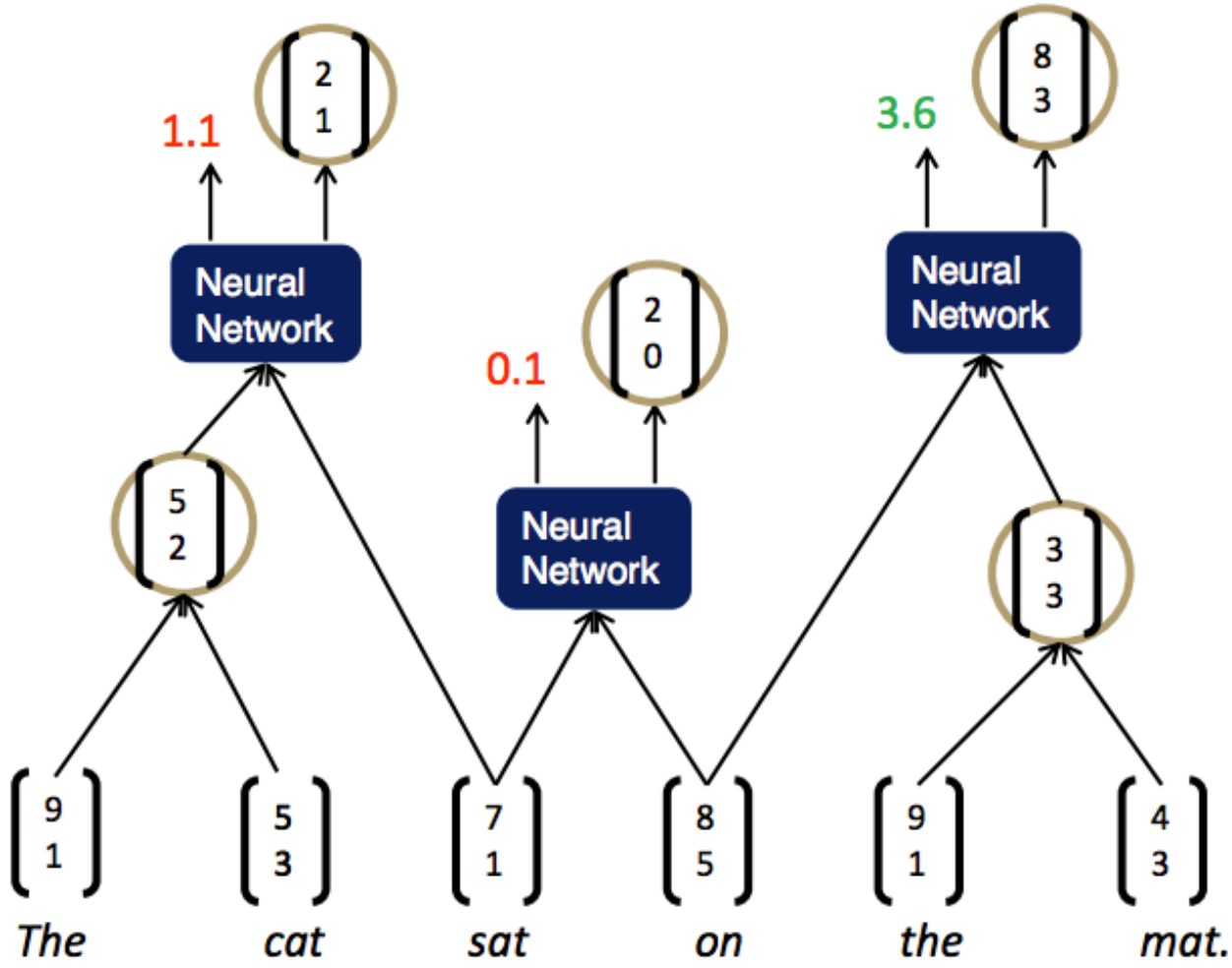
Recursive Neural Networks



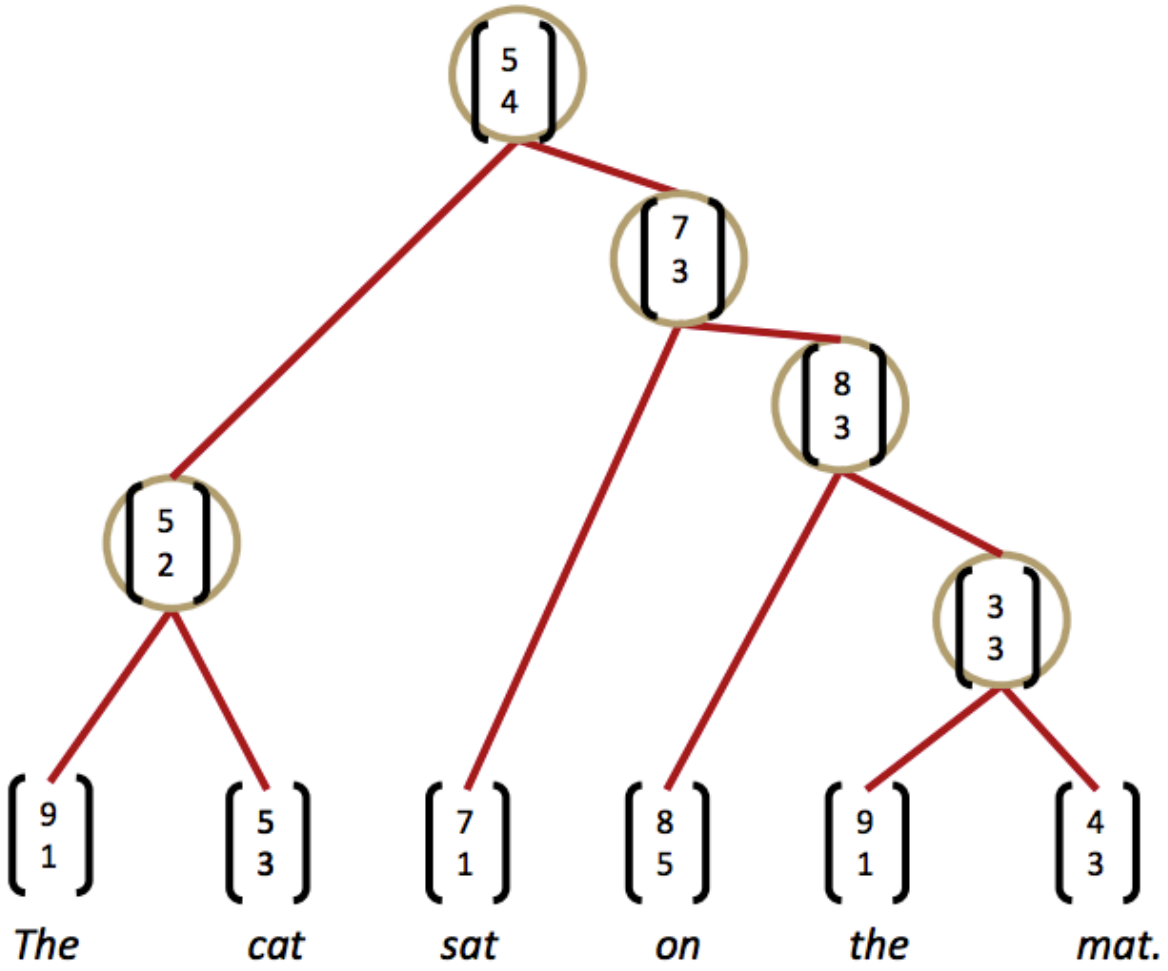
Recursive Neural Networks



Recursive Neural Networks

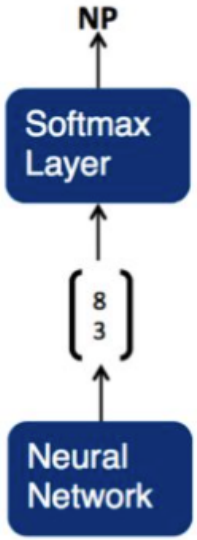


Recursive Neural Networks



Classificando Nós

- Categorias são os tipos gramaticais



Partial Parsing: Chunking

- Criação de parse trees completas pode ser custoso
- Chunking: Identifica as frases de uma sentença

[$_{NP}$ The morning flight] [$_{PP}$ from] [$_{NP}$ Denver] [$_{VP}$ has arrived.]

- Pode focar em um tipo específico de frase

[$_{NP}$ The morning flight] from [$_{NP}$ Denver] has arrived.

- Utiliza sequence labeling para classificar as frases

The	*morning*	*flight*	*from*	*Denver*	*has*	*arrived*
B_NP	I_NP	I_NP	B_PP	B_NP	B_VP	I_VP