CKY Parsing

Ling571
Deep Processing Approaches to NLP
January 18, 2017

Roadmap

- CKY parsing algorithm
  - Dynamic programming structure
  - Algorithm and example
  - Analysis & Discussion
- Probabilistic CFGs
  - Motivation: Ambiguity
  - Augmenting CFGs with probabilities

CKY

- Given an input string S of length n,
  - Build table (n+1) x (n+1)
  - Indexes correspond to inter-word positions
    - E.g., 0 Book 1 That 2 Flight 3
- Cells [i,j] contain SETS of non-terminals of ALL constituents spanning i,j
  - [j-1,j] contains pre-terminals
  - If [0,n] contains Start, the input is recognized

CKY Algorithm

```
function CKY- PARSE(words, gra1nmar) returns table

  for j f--- from 1 to LE GTH(words) do
      table[ j - 1  j] f--- {A | A ---t  wo,ds[J ] E  gramn ɿar }
      for i f--- fro1n j - 2 do"nto O do
          for k f--- i + 1 to J - 1 do
              table[ i j]+- table[i j] U
                              {A | A ---+  BC  E  gran.rnar,
                                  B  E  tab!e[i  k]
                                  C  E  table [k  j]}
```

Is this a parser?

CKY Algorithm

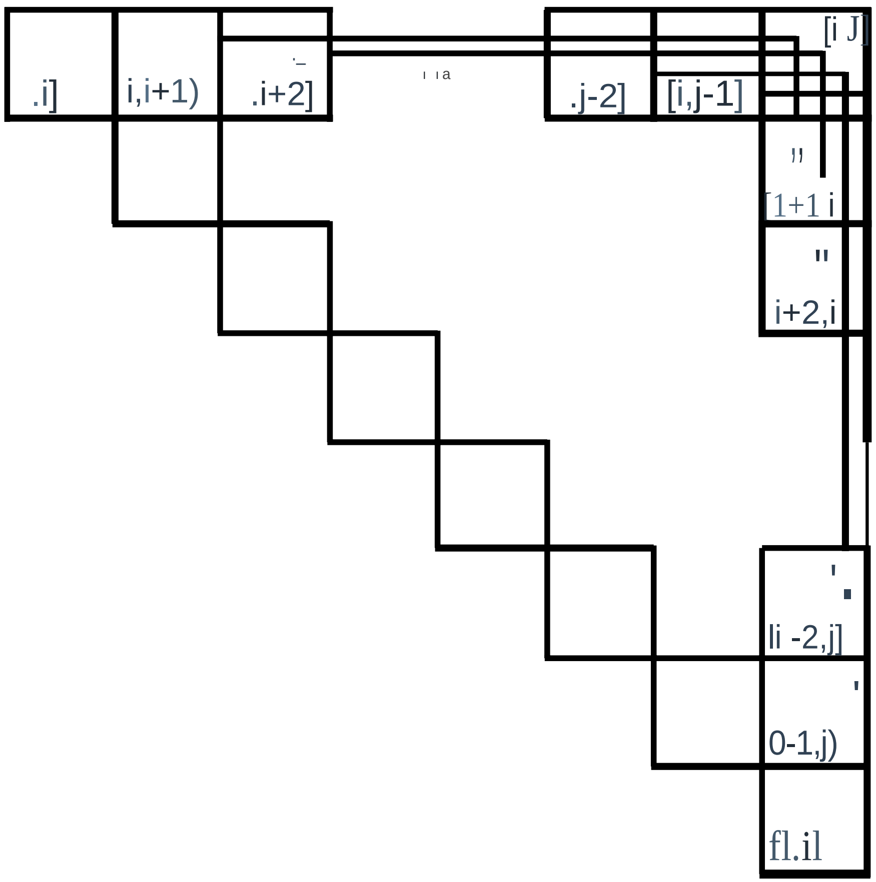
- Table fills:
  - Column-by-column
  - Left-to-right
  - Bottom-to-top

- Why?
  - Necessary info available (below and left)
  - Allows online sentence analysis
    - Works across input string as it arrives

CKY Table

Book	*the*	*flight*	*through*	*Houston*
S, VP, Verb Nominal, Noun [O,1]	[0,2]	S,VP,X2 [0,3]	10,4]	S, VP [0,5]
	Det [1,2]	NP [1,3]	['1,4]	NP [1,5]
		Nominal, Noun 12 31	12 41	Nominal 12 51
			P rep 13,4]	pp [3,5)
				NP, Proper-Noun f4 .5l

Filling CKY cell



0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]					

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]					
	Verb,VP,S [1,2]				

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]				
	Verb,VP,S [1,2]				

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]				
	Verb,VP,S [1,2]				
		Det [2,3]			

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]				
	Verb,VP,S [1,2]	[1,3]			
		Det [2,3]			

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]			
	Verb,VP,S [1,2]	[1,3]			
		Det [2,3]			
			NN, Nom [3,4]		

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]			
	Verb,VP,S [1,2]	[1,3]			
		Det [2,3]	NP [2,4]		
			NN, Nom [3,4]		

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]			
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]		
		Det [2,3]	NP [2,4]		
			NN, Nom [3,4]		

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]	S [0,4]		
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]		
		Det [2,3]	NP [2,4]		
			NN, Nom [3,4]		

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]	S [0,4]	[0.5]	
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]	[1,5]	
		Det [2,3]	NP [2,4]	[2,5]	
			NN, Nom [3,4]	[3,5]	
				Prep [4,5]	

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]	S [0,4]	[0.5]	
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]	[1,5]	
		Det [2,3]	NP [2,4]	[2,5]	
			NN, Nom [3,4]	[3,5]	
				Prep [4,5]	
					NNP, NP [5,6]

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]	S [0,4]	[0.5]	
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]	[1,5]	
		Det [2,3]	NP [2,4]	[2,5]	
			NN, Nom [3,4]	[3,5]	
				Prep [4,5]	PP [4,6]
					NNP, NP [5,6]

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]	S [0,4]	[0.5]	
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]	[1,5]	
		Det [2,3]	NP [2,4]	[2,5]	NP [2,6]
			NN, Nom [3,4]	[3,5]	Nom [3,6]
				Prep [4,5]	PP [4,6]
					NNP, NP [5,6]

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]	S [0,4]	[0.5]	
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]	[1,5]	VP,XP2,S [1,6]
		Det [2,3]	NP [2,4]	[2,5]	NP [2,6]
			NN, Nom [3,4]	[3,5]	Nom [3,6]
				Prep [4,5]	PP [4,6]
					NNP, NP [5,6]

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]	S [0,4]	[0.5]	S [0,6]
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]	[1,5]	VP,XP2,S [1,6]
		Det [2,3]	NP [2,4]	[2,5]	NP [2,6]
			NN, Nom [3,4]	[3,5]	Nom [3,6]
				Prep [4,5]	PP [4,6]
					NNP, NP [5,6]

From Recognition to Parsing

- Limitations of current recognition algorithm:
  - Only stores non-terminals in cell
    - Not rules or cells corresponding to RHS
  - Stores SETS of non-terminals
    - Can't store multiple rules with same LHS

- Parsing solution:
  - All repeated versions of non-terminals
  - Pair each non-terminal with pointers to cells
    - Backpointers
  - Last step: construct trees from back-pointers in [0,n]

0 I 1 prefer 2 a 3 flight 4 on 5 TWA 6

I	prefer	a	flight	on	TWA
NP,Pronoun [0,1]	S [0,2]	[0,3]	S [0,4]	[0.5]	S_1, S_2, S_3 [0,6]
	Verb,VP,S [1,2]	[1,3]	VP,XP2,S [1,4]	[1,5]	VP_1,VP_2,VP_3, ,XP2,S_1,S_2,S_3 [1,6]
		Det [2,3]	NP [2,4]	[2,5]	NP [2,6]
			NN, Nom [3,4]	[3,5]	Nom [3,6]
				Prep [4,5]	PP [4,6]
					NNP, NP [5,6]

CKY Discussion

- Running time:
  - $O(n^3)$ where *n* is the length of the input string
  - Inner loop grows as square of # of non-terminals
- Expressiveness:
  - As implemented, requires CNF
    - Weakly equivalent to original grammar
    - Doesn't capture full original structure
      - Back-conversion?
        - Can do binarization, terminal conversion
        - Unit non-terminals require change in CKY

Parsing: PCFGs

Ling 571
Deep Processing Techniques for NLP
January 18, 2017

Roadmap

- Probabilistic Context-free Grammars (PCFGs)
  - Motivation: Ambiguity
  - Approach:
    - Definition
    - Disambiguation
    - Parsing
    - Evaluation
    - Enhancements

- What about ambiguity?
- CKY can represent it
- Can’t resolve it

Probabilistic Parsing

- Provides strategy for solving disambiguation problem
  - Compute the probability of all analyses
  - Select the most probable

- Employed in language modeling for speech recognition
  - N-gram grammars predict words, constrain search
  - Also, constrain generation, translation

PCFGs

- Probabilistic Context-free Grammars
  - Augmentation of CFGs

N	a set of **non-terminal symbols** (or **variables**)
Σ	a set of **terminal symbols** (disjoint from N)
R	a set of **rules** or productions, each of the form $A \rightarrow \beta\ \boxed{[p]}$, where A is a non-terminal, β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$, and p is a number between 0 and 1 expressing $P(\beta\|A)$
S	a designated **start symbol**

PCFGs

- Augment each production with probability that LHS will be expanded as RHS
  - P(A→→B) or P(A→→B|A), P(RHS|LHS)
  - Sum over all possible expansions is 1

$$\sum_{\beta} P(A \rightarrow \beta) = 1$$

- A PCFG is consistent if sum of probabilities of all sentences in language is 1.
  - Recursive rules often yield inconsistent grammars

Example PCFG

Grammar

S ---+ NP VP	[.80)
S ---+ Aux NP VP	[.15]
S ---+ V P	[.05]
NP ---+ Pronoun	[.35)
NP ---+ Proper-Noun	[.30)
NP ---+ Det No1ninal	[.20]
NP ---+ Nonzinal	[.15]
Norninal ---+ Noun	[.75)
Norninal ---+ Nornina l Noun	[.20]
Norninal ---+ Nonzinal PP	[.05]
VP ---+ Verb	[.35]
VP ---+ Verb l\TP	[.20]
VP ---+ Verb NP PP	[.10]
VP ---+ Verb PP	[.15]
VP ---+ Verb NP NP	[.05)
VP ---+ VP PP	[.15)
PP ---+ Prep osition NP	[1.0]

Lexicon

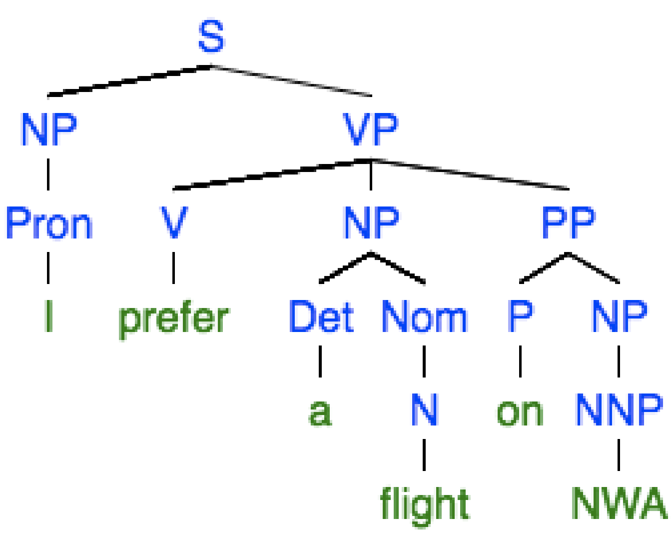
Det ---+ that [.10] | *a* [.30) | *the* [.60]
Noun ---+ book [.10] | *flight* [.30]
| *rneal* [.15] | *rnoney* [.05]
| *flights* [.40) | *dinner* [.10]
Verb ---+ book [.30) | *include* [.30)
| *prefer;* [.40]
Pron oun ---+ I [.40] | *she* [.05]
| *nle* [.15] | *you* [.40]
P,oper-Noun ---+ Houston [.60]
| *NWA* [.40]
Aux ---+ does [.60] | *can* [40]
Prepos;tion ---+ front [.30) | *to* [.30]
| *on* [.20) | *near* [.15)
| *through* [.05]

Disambiguation

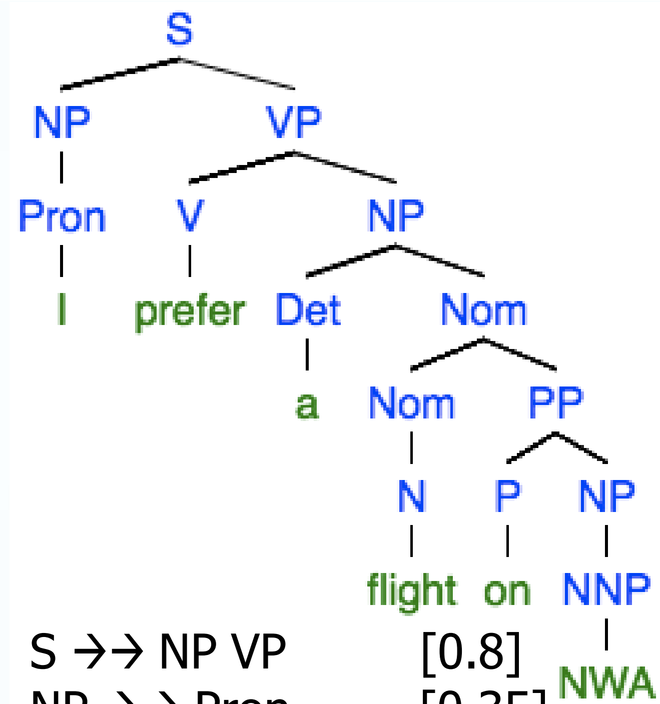
- A PCFG assigns probability to each parse tree T for input S.
  - Probability of T: product of all rules to derive T

$$P(T,S) = \prod_{i=1}^{n} P(RHS_i \mid LHS_i)$$

$$P(T,S) = P(T)P(S \mid T) = P(T)$$



S →→ NP VP	[0.8]
NP →→ Pron	[0.35]
Pron →→ I	[0.4]
VP →→ V NP PP	[0.1]
V →→ prefer	[0.4]
NP →→ Det Nom	[0.2]
Det →→ a	[0.3]
Nom →→ N	[0.75]
N →→ flight	[0.3]
PP →→ P NP	[1.0]
P →→ on	[0.2]
NP →→ NNP	[0.3]
NNP →→ NWA	[0.4]



S →→ NP VP	[0.8]
NP →→ Pron	[0.35]
Pron →→ I	[0.4]
VP →→ V NP	[0.2]
V →→ prefer	[0.4]
NP →→ Det Nom	[0.2]
Det →→ a	[0.3]
Nom →→ Nom PP	[0.05]
Nom →→ N	[0.75]
N →→ flight	[0.3]
PP →→ P NP	[1.0]
P →→ on	[0.2]
NP →→ NNP	[0.3]
NNP →→ NWA	[0.4]

Parsing Problem for PCFGs

- Select T such that:

$$\hat{T}(S) = \underset{T s.t, S = yield(T)}{\operatorname{argmax}} P(T)$$

  - String of words S is *yield* of parse tree over S
  - Select tree that maximizes probability of parse