COMP 790.139 (Fall 2017)
Natural Language Processing

Lecture 4: Syntactic Parsing (Constituent, Dependency, CCG, etc.)

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Mohit Bansal

(various slides adapted/borrowed from courses by Dan Klein, Richard Socher, Chris Manning, JurafskyMartin-SLP3, others)

Announcements

- Chapter section summary due Sunday Sep24 midnight
- Coding-HW1 (on word vector training+evaluation_ +visualization) will be release in 1-2 days – TA Yixin will give overview of the homework today!
- TA Yixin Nie's office hours: 2.30-3.30pm Wednesdays (SN-372; might move to 2^{nd} floor reading room)

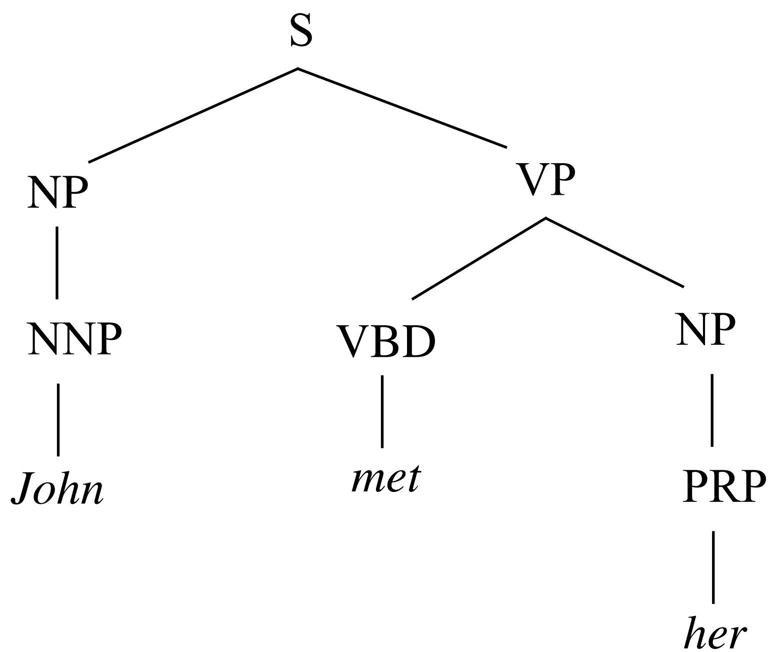
Coding HW1 (TA Yixin Nie’s presentation)

Syntactic Parsing

Constituent Parsing

Syntactic Parsing -- Constituent

- Phrase-structure parsing or Bracketing



- Demos: http://tomato.banatao.berkeley.edu:8080/parser/parser.html

Probabilistic Context-free Grammars

- A context-free grammar is a tuple <*N, T, S, R*>

N : the set of non-terminals

Phrasal categories: S, NP, VP, ADJP, etc.

Parts-of-speech (pre-terminals): NN, JJ, DT, VB

T : the set of terminals (the words)

S : the start symbol

Often written as ROOT or TOP

Not usually the sentence non-terminal S

R : the set of rules

Of the form $X \rightarrow Y_1\ Y_2 \ldots Y_k$, with $X, Y_i \in N$

Examples: $S \rightarrow NP\ VP$, $VP \rightarrow VP\ CC\ VP$

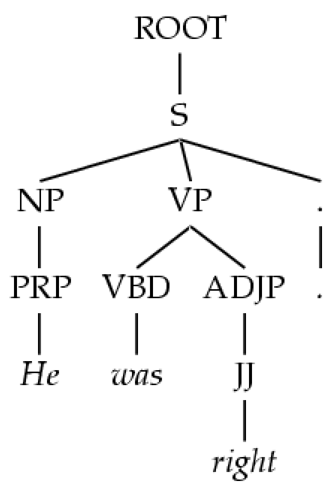
Also called rewrites, productions, or local trees

Probabilistic Context-free Grammars

- A PCFG:
  - Adds a top-down production probability per rule $P(Y_1\ Y_2 \ldots Y_k \mid X)$
  - Allows us to find the 'most probable parse' for a sentence
  - The probability of a parse is just the product of the probabilities of the individual rules

Treebank PCFG

- Can just count the frequency of each rule and normalize (but not very effective)



ROOT → S	1
S → NP VP .	1
NP → PRP	1
VP → VBD ADJP	1
…..	

Model	*F1*
Baseline	72.0

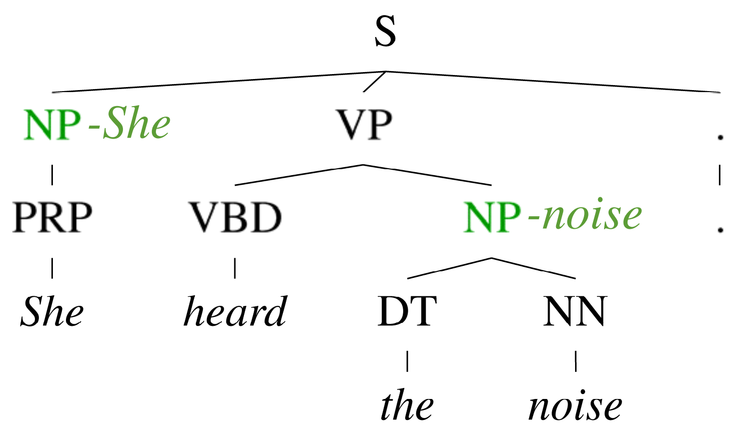
[Charniak, 1996]

Real Treebank Examples

- Long, complex sentences with several clauses, nested prepositions, etc.

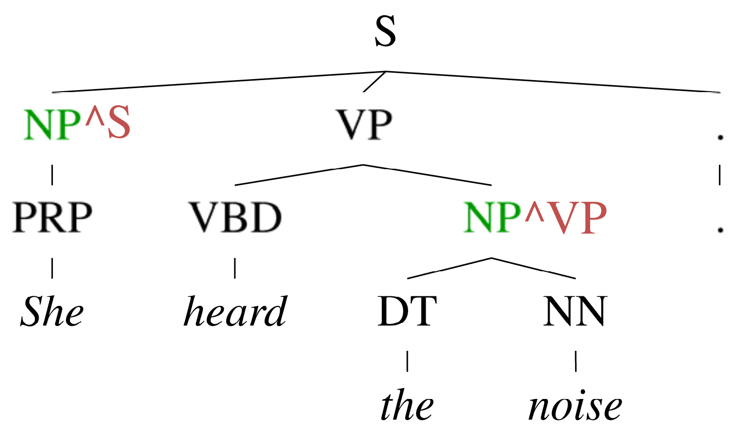
```
( (S (NP-SBJ The move)
     (VP followed
         (NP (NP a round)
             (PP of
                 (NP (NP similar increases)
                     (PP by
                         (NP other lenders))
                     (PP against
                         (NP Arizona real estate loans)))))
         ,
         (S-ADV (NP-SBJ *)
                (VP reflecting
                    (NP (NP a continuing decline)
                        (PP-LOC in
                                (NP that market))))))
     .))
```

Grammar Refinement



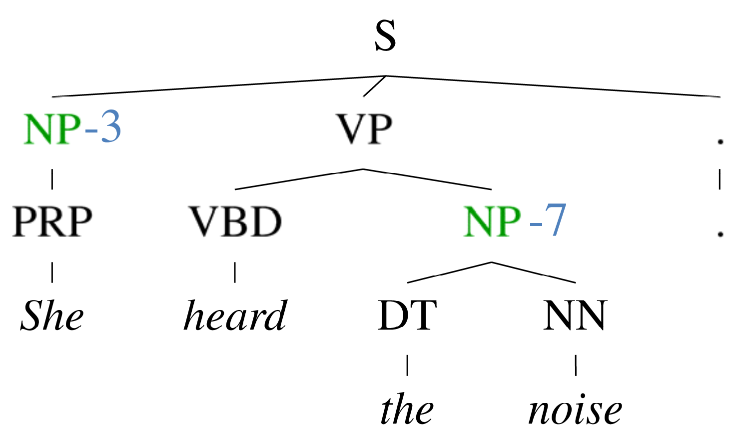
- Conditional independence assumptions often too strong! Not every NP expansion can fill every NP slot
- Better results by enriching the grammar e.g.,
  - Lexicalization [Collins, 1999; Charniak, 2000]

Grammar Refinement



- Conditional independence assumptions often too strong! Not every NP expansion can fill every NP slot
- Better results by enriching the grammar e.g.,
  - Lexicalization [Collins, 1999; Charniak, 2000]
  - Markovization, Manual Tag-splitting [Johnson, 1998; Klein & Manning, 2003]

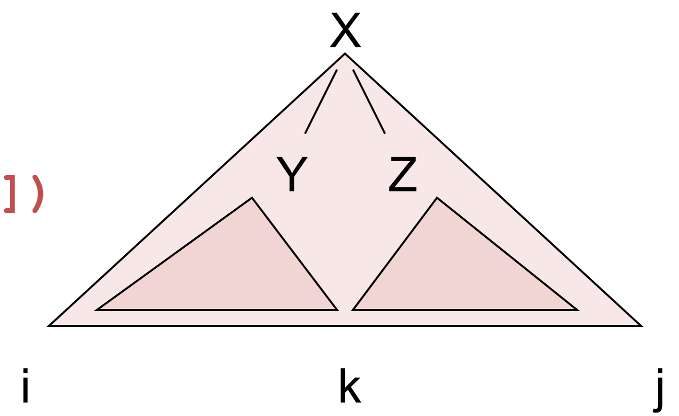
Grammar Refinement



- Conditional independence assumptions often too strong! Not every NP expansion can fill every NP slot
- Better results by enriching the grammar e.g.,
  - Lexicalization [Collins, 1999; Charniak, 2000]
  - Markovization, Manual Tag-splitting [Johnson, 1998; Klein & Manning, 2003]
  - Latent Tag-splitting [Matsuzaki et al., 2005; Petrov et al., 2006]

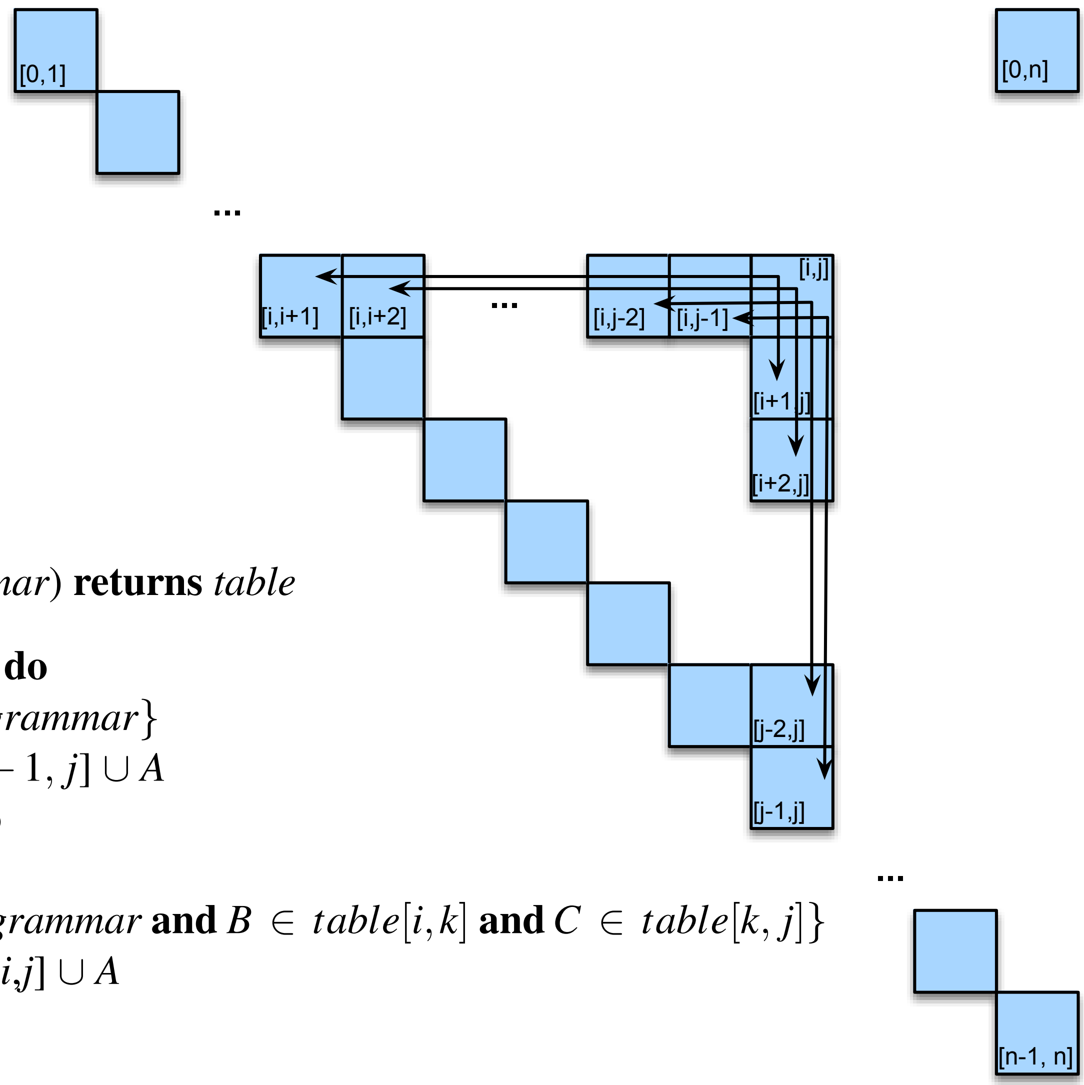
CKY (or CYK) Parsing Algorithm (Bottom-up)

```
bestScore(s)
 for (i : [0,n-1])
   for (X : tags[s[i]])
     score[X][i][i+1] = tagScore(X,s[i])
 for (diff : [2,n])
   for (i : [0,n-diff])
     j = i + diff
     for (X->YZ : rule)
       for (k : [i+1, j-1])
         score[X][i][j] = max{score[X][i][j], score(X->YZ)
                                              *score[Y][i][k]
                                              *score[Z][k][j]}
```



[Cocke, 1970; Kasami, 1965; Younger, 1967]

CKY Parsing Algorithm (Bottom-up)

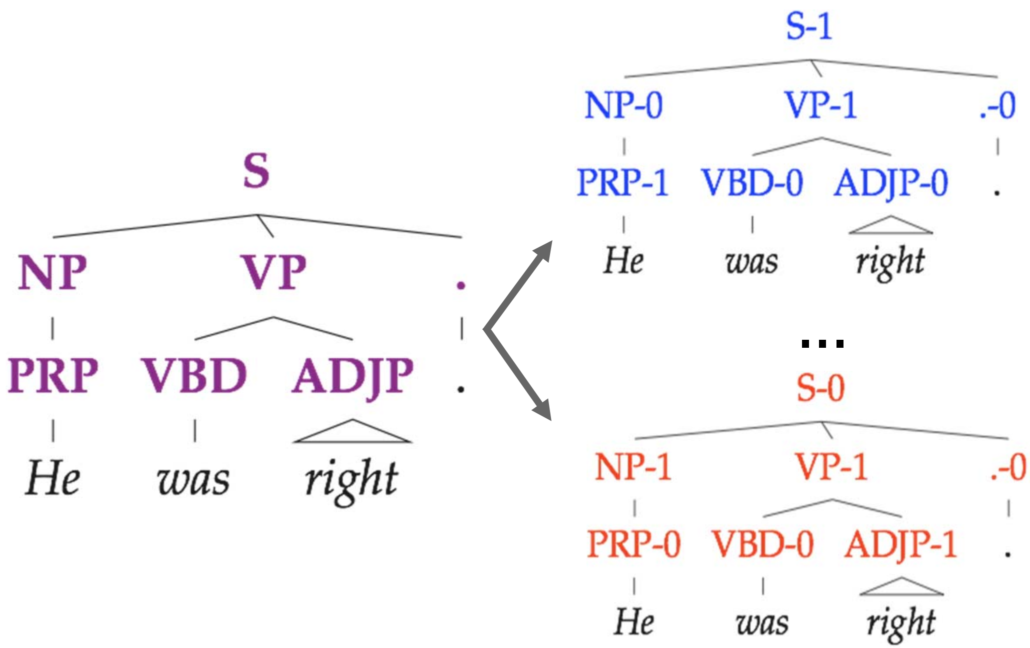


function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  for all $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      for all $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

[Jurafsky-Martin-SLP3]

Latent Variable Grammars



Grammar G	
$S_0 \rightarrow NP_0\ VP_0$	?
$S_0 \rightarrow NP_1\ VP_0$	?
$S_0 \rightarrow NP_0\ VP_1$	?
$S_0 \rightarrow NP_1\ VP_1$	?
$S_1 \rightarrow NP_0\ VP_0$	?
...	
$S_1 \rightarrow NP_1\ VP_1$	?
...	
$NP_0 \rightarrow PRP_0$	?
$NP_0 \rightarrow PRP_1$	?
...	

Lexicon	
$PRP_0 \rightarrow$ She	?
$PRP_1 \rightarrow$ She	?
...	
$VBD_0 \rightarrow$ was	?
$VBD_1 \rightarrow$ was	?
$VBD_2 \rightarrow$ was	?
...	

Parse Tree T

Sentence w

Derivations $t : T$

Parameters θ

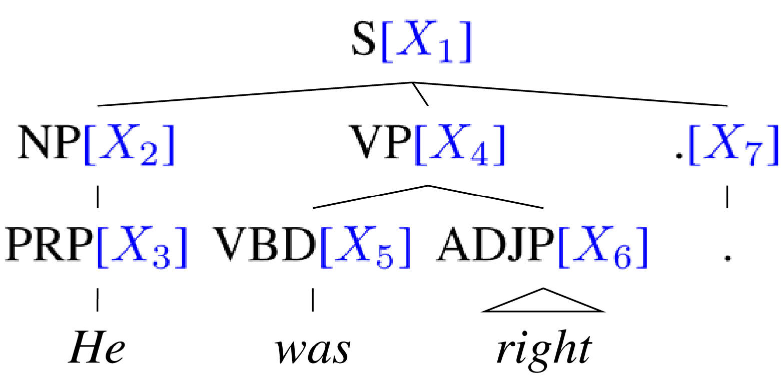
[Petrov et al., 2006]

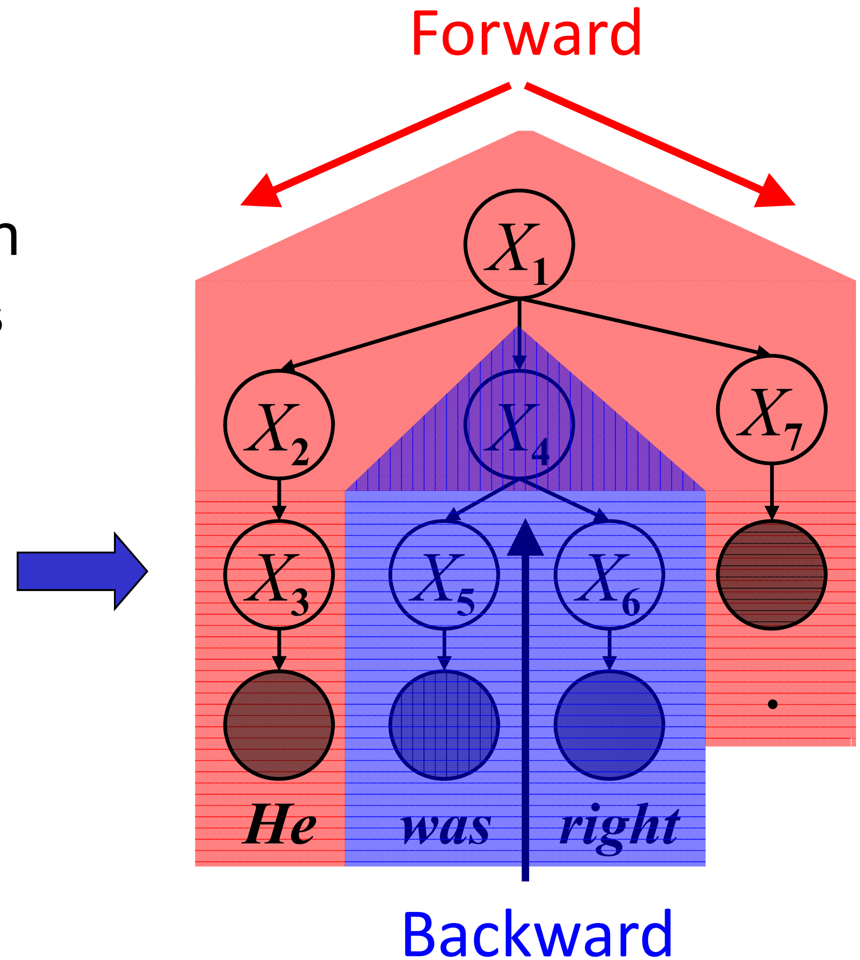
Learning Latent Splits (Inside-Outside)

- Forward-backward (last week) but for trees

EM algorithm:

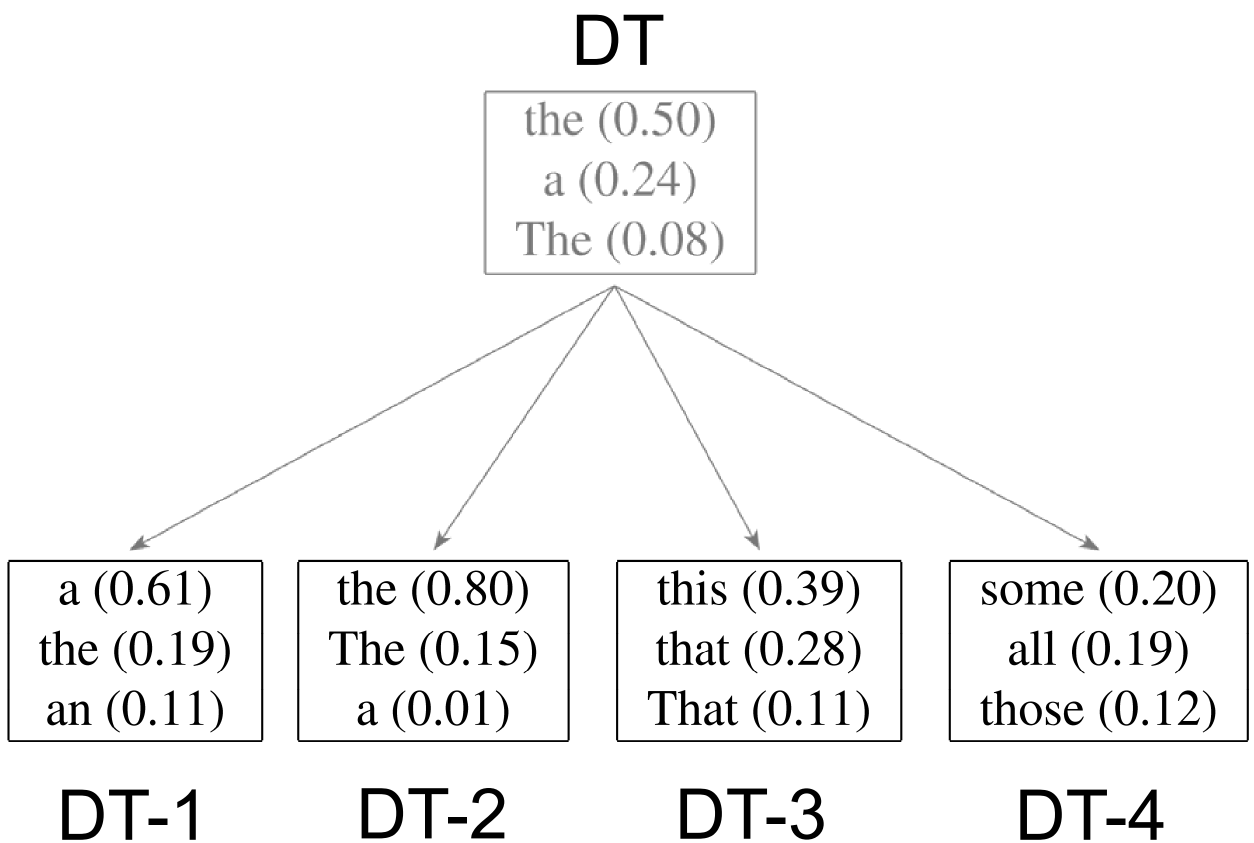
- Brackets are known
- Base categories are known
- Only induce subcategories





[Petrov et al., 2006]

DT Tag Splits Example



[Petrov et al., 2006]

Other Learned Splits

- Proper Nouns (NNP):

NNP-14	Oct.	Nov.	Sept.
NNP-12	John	Robert	James
NNP-2	J.	E.	L.
NNP-1	Bush	Noriega	Peters
NNP-15	New	San	Wall
NNP-3	York	Francisco	Street

- Personal pronouns (PRP):

PRP-0	It	He	I
PRP-1	it	he	they
PRP-2	it	them	him

[Petrov et al., 2006]

Other Learned Splits

- Relative adverbs (RBR):

RBR-0	further	lower	higher
RBR-1	more	less	More
RBR-2	earlier	Earlier	later

- Cardinal Numbers (CD):

CD-7	one	two	Three
CD-4	1989	1990	1988
CD-11	million	billion	trillion
CD-0	1	50	100
CD-3	1	30	31
CD-9	78	58	34

[Petrov et al., 2006]

Latent PCFG Results

		≤ 40 words F1	all F1
ENG	Charniak&Johnson '05 (generative)	90.1	89.6
	Split / Merge	**90.6**	**90.1**
GER	Dubey '05	76.3	-
	Split / Merge	**80.8**	**80.1**
CHN	Chiang et al. '02	80.0	76.6
	Split / Merge	**86.3**	**83.4**

[Petrov et al., 2006]

Evaluating Constituent Parsers

$$\textbf{labeled recall:} = \frac{\text{\# of correct constituents in hypothesis parse of } s}{\text{\# of correct constituents in reference parse of } s}$$

$$\textbf{labeled precision:} = \frac{\text{\# of correct constituents in hypothesis parse of } s}{\text{\# of total constituents in hypothesis parse of } s}$$

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R} \qquad F_1 = \frac{2PR}{P+R}$$

cross-brackets: the number of constituents for which the reference parse has a bracketing such as ((A B) C) but the hypothesis parse has a bracketing such as (A (B C)).

[Jurafsky-Martin-SLP3]

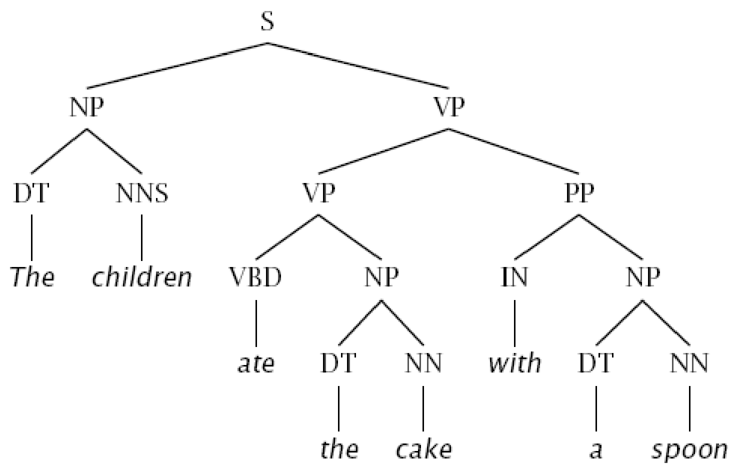
Other Results

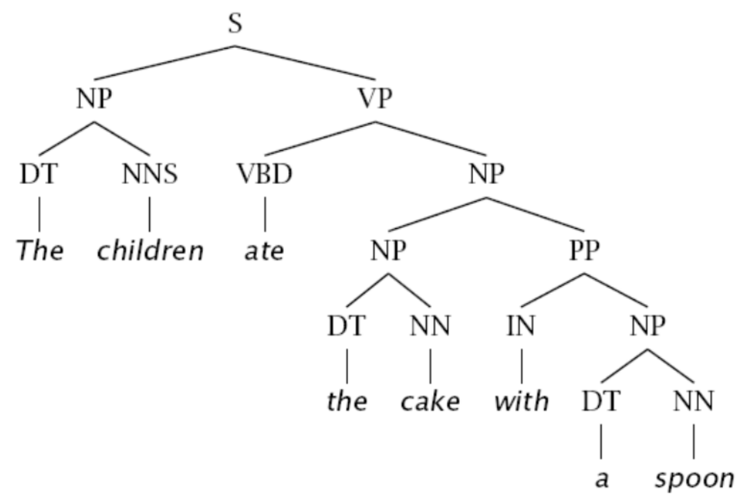
- Collins, 1999 → 88.6 F1 (generative lexical)
- Charniak and Johnson, 2005 → 89.7 / 91.3 F1 (generative lexical / reranking)
- Petrov et al., 2006 → 90.7 F1 (generative unlexical)
- McClosky et al., 2006 – 92.1 F1 (generative + reranking + self-training)

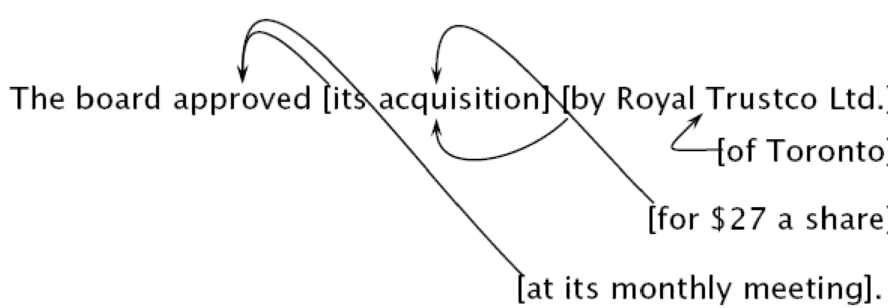
Syntactic Ambiguities

- I saw the old man with a telescope
- I shot an elephant in my pajamas
- I cleaned the dishes in my pajamas
- I cleaned the dishes in the sink

Real-Data PP Attachment Ambiguities







Attachment Ambiguity Types

- Prepositional phrases:
  They cooked the beans in the pot on the stove with handles.
- Particle vs. preposition:
  The puppy tore up the staircase.
- Complement structures
  The tourists objected to the guide that they couldn't hear.
  She knows you like the back of her hand.
- Gerund vs. participial adjective
  Visiting relatives can be boring.
  Changing schedules frequently confused passengers.

Attachment Ambiguity Types

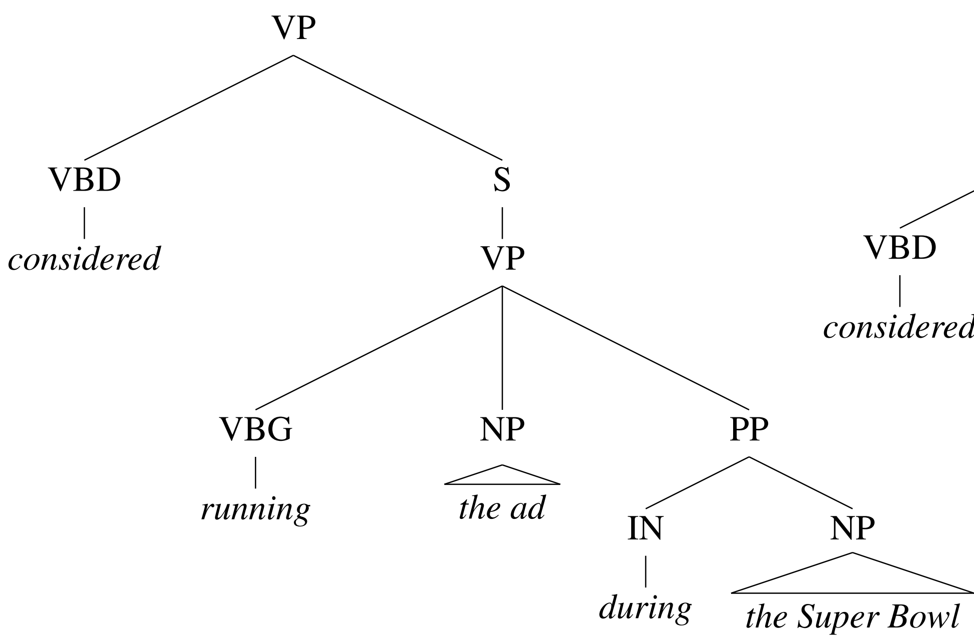
- Modifier scope within NPs
  impractical design requirements
  plastic cup holder
- Multiple gap constructions
  The chicken is ready to eat.
  The contractors are rich enough to sue.
- Coordination scope:
  Small rats and mice can squeeze into holes or cracks in the wall.

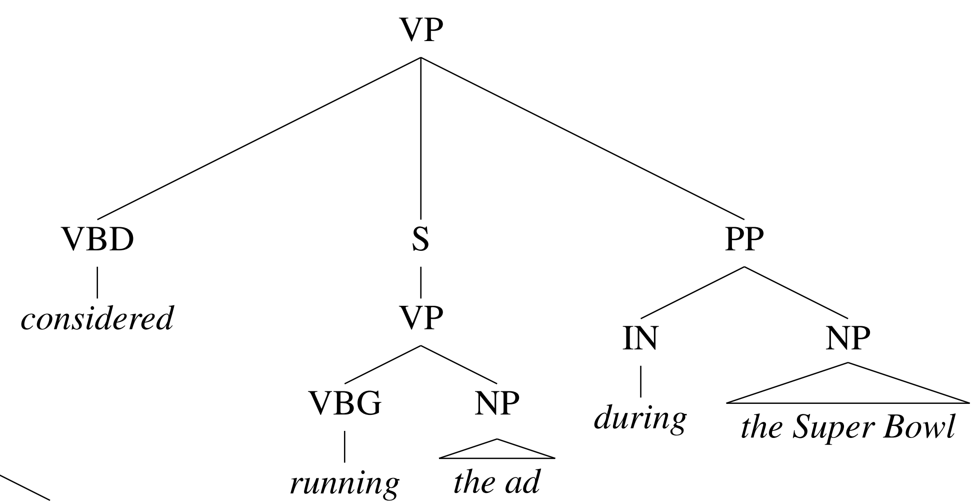
World Knowledge is Important

Web Features for Syntactic Parsing

Dependency:

Constituent:





[Nakov and Hearst 2005; Pitler et al., 2010; Bansal and Klein, 2011]

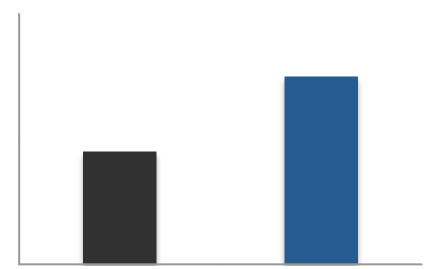
Web Features for Syntactic Parsing

They considered running the ad during the Super Bowl.

Web Ngrams

count(*running it during*) > count(*considered it during*)

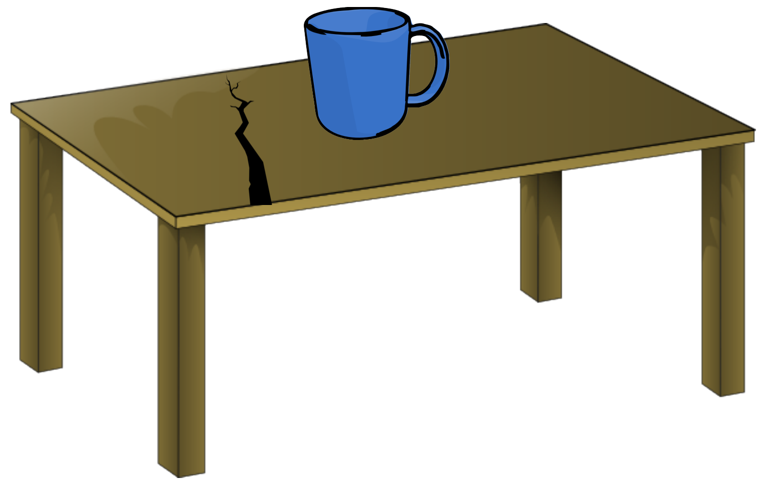
- 7-10% relative error reduction over 90-92% parsers



[Bansal and Klein, 2011]

Visual Recognition Cues

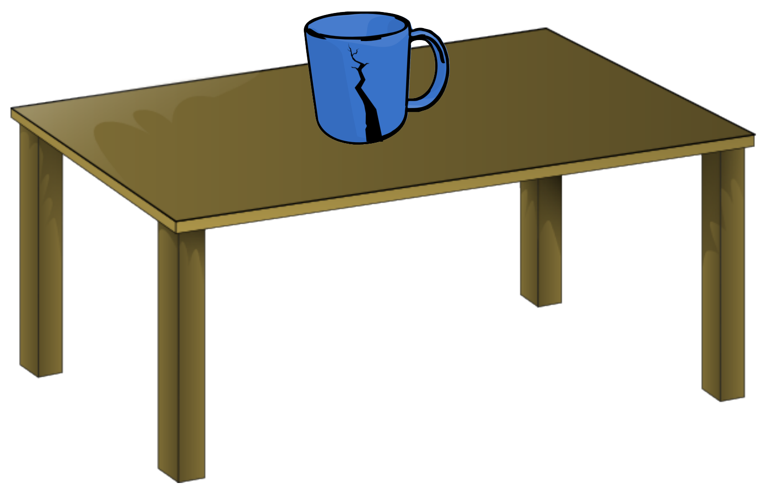
- Joint parsing and image recognition



the mug on the table with a crack

Visual Recognition Cues

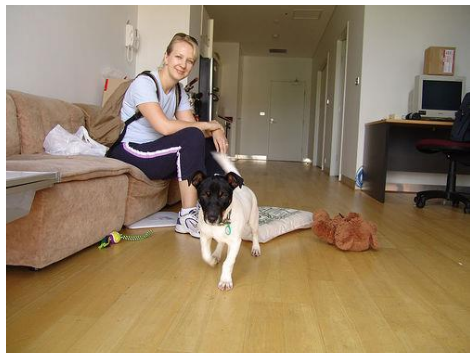
- Joint parsing and image recognition



the mug on the table with a crack

red chair and table
light green table

Visual Recognition Cues



“A dog is standing next to a woman on a couch”

Ambiguity:
(dog next to woman) on couch
vs dog next to (woman on couch)

Vision: Semantic Segmentation

Labels: Chairs, desks, etc.

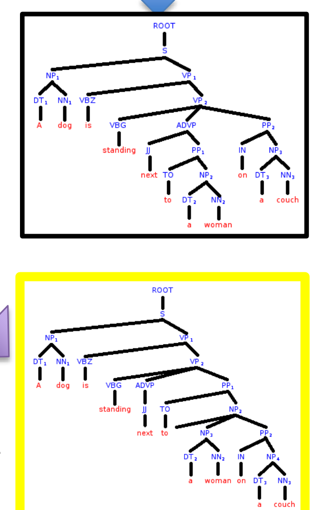
NLP: Sentence Parsing

Output: Parse Tree

Solution #1

Solution #M

Consistent



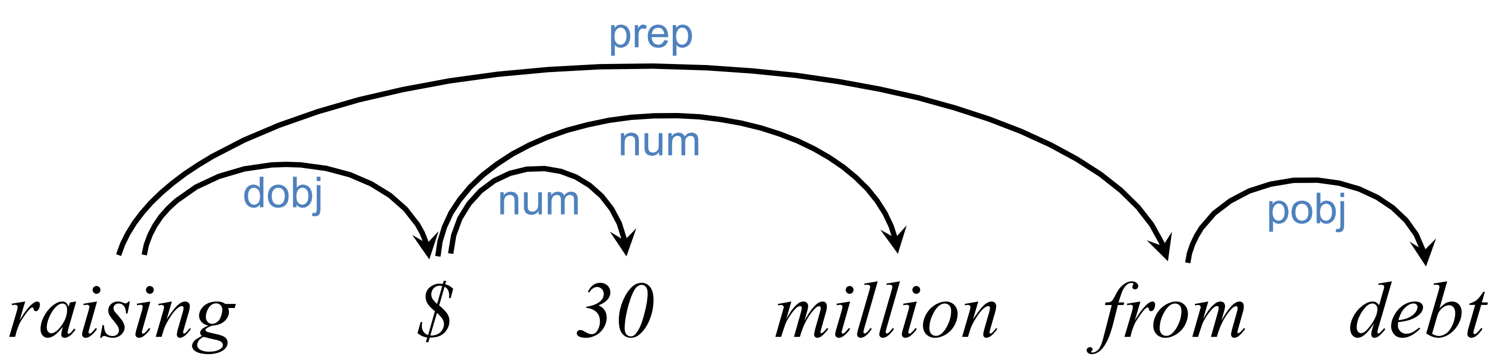
PASCAL Sentence Dataset

[Christie et al., 2016]

Dependency Parsing

Dependency Parsing

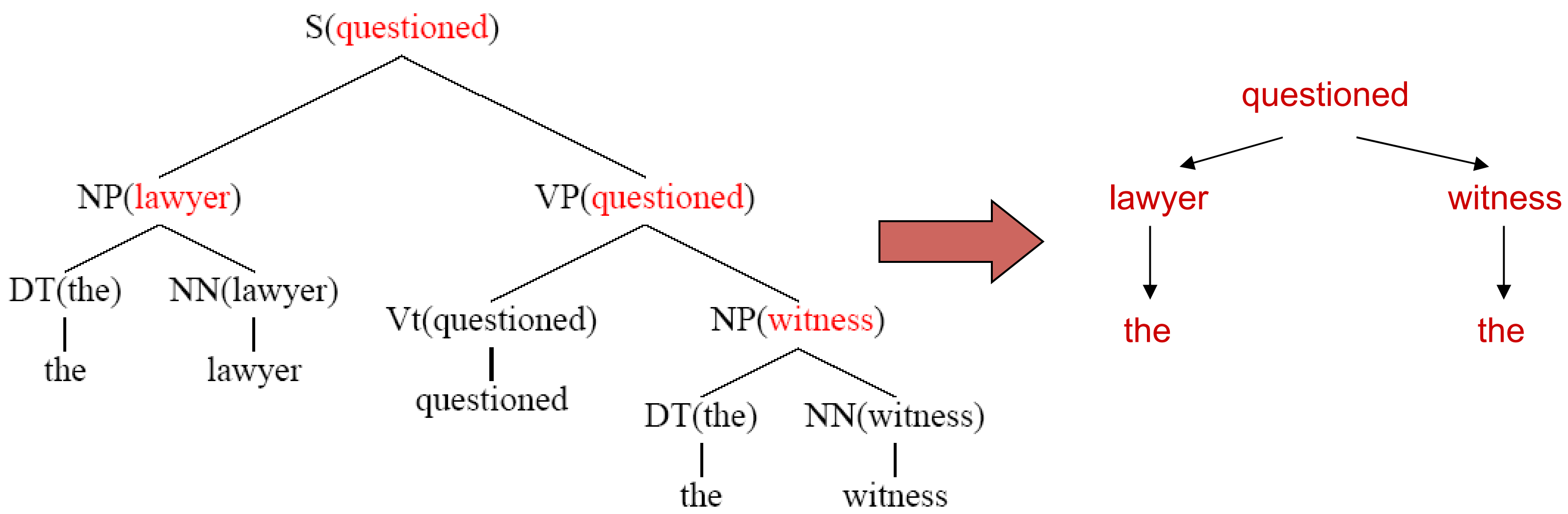
- Predicting directed head-modifier relationship pairs



- Demos: http://nlp.stanford.edu:8080/corenlp/

Dependency Parsing

- Can convert (lexicalized) constituent tree to dependency tree (each local binary tree gives us a dependency attachment from head to modifier)



Constituent Parse (with head words)

Dependency Parse

Dependency Parsing

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

[Jurafsky-Martin-SLP3]

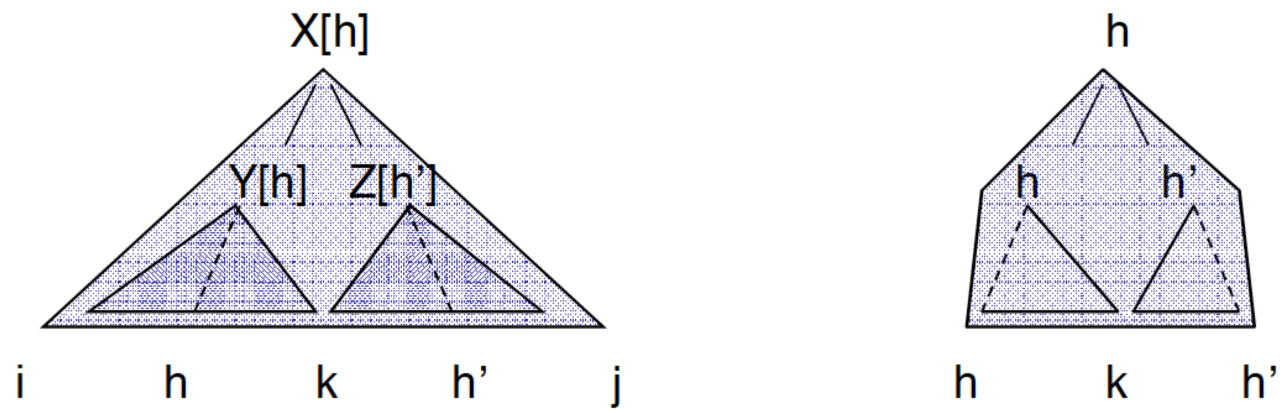
Dependency Parsing

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*.

[Jurafsky-Martin-SLP3]

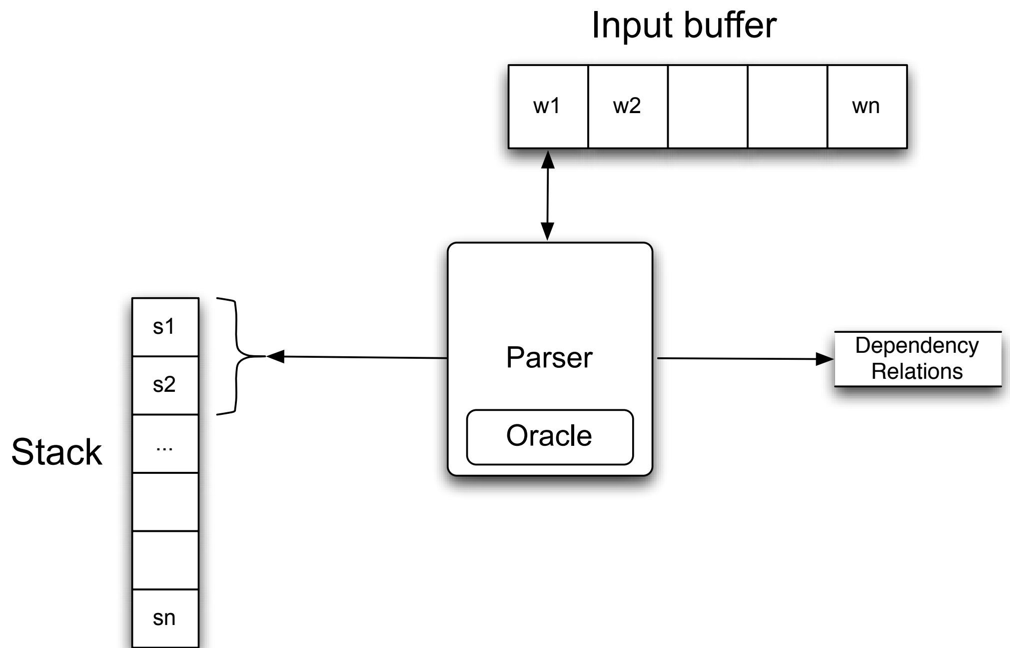
Dependency Parsing

- Pure (projective, 1st order) dependency parsing is only cubic [Eisner, 1996]



- Non-projective dependency parsing useful for Czech & other languages – MST algorithms [McDonald et al., 2005]

Transition-based Dependency Parsing



Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book $\rightarrow$ me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning $\leftarrow$ flight)
7	[root, book, the, flight]	[]	LEFTARC	(the $\leftarrow$ flight)
8	[root, book, flight]	[]	RIGHTARC	(book $\rightarrow$ flight)
9	[root, book]	[]	RIGHTARC	(root $\rightarrow$ book)
10	[root]	[]	Done	

Parsing: Other Models and Methods

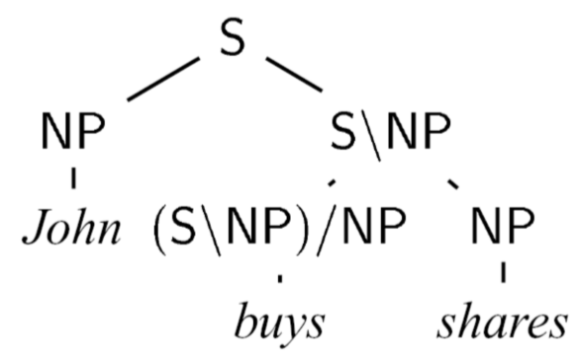
- Combinatory Categorial Grammar [Steedman, 1996, 2000; Clark and Curran, 2004]
- Transition-based Dependency Parsing [Yamada and Matsumoto, 2003; Nivre, 2003]
- Tree-Insertion Grammar, DOP [Schabes and Waters, 1995; Hwa, 1998; Scha, 1990; Bod, 1993; Goodman, 1996; Bansal and Klein, 2010]
- Tree-Adjoining Grammar [Resnik, 1992; Joshi and Schabes, 1998; Chiang, 2000]
- Shift-Reduce Parser [Nivre and Scholz, 2004; Sagae and Lavie, 2005]
- Other: Reranking, A*, K-Best, Self-training, Co-training, System Combination, Cross-lingual Transfer [Sarkar, 2001; Steedman et al., 2003; Charniak and Johnson, 2005; Hwa et al., 2005; Huang and Chiang, 2005; McClosky et al., 2006; Fossum and Knight, 2009; Pauls and Klein, 2009; McDonald et al., 2011]

- Other Demos: http://svn.ask.it.usyd.edu.au/trac/candc/wiki/Demo, http://4.easy-ccg.appspot.com/

CCG Parsing

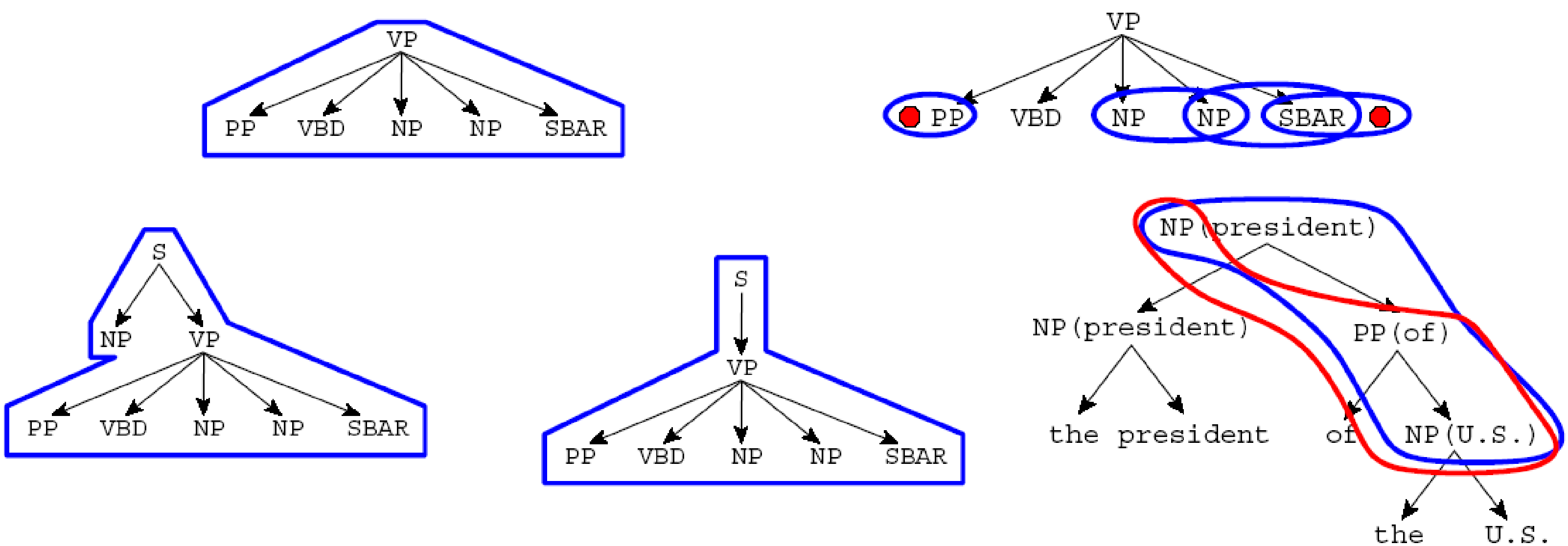
- Combinatory Categorial Grammars:
- Each category encodes an argument sequence (fwd/bwd slashes specify argument order/direction)
- Closely related to lambda calculus
- Captures both syntactic and semantic info
- Naturally allows meaning representation and semantic parsing (next week!)

$$\mathit{John} \vdash \mathsf{NP}$$
$$\mathit{shares} \vdash \mathsf{NP}$$
$$\mathit{buys} \vdash (\mathsf{S}\backslash\mathsf{NP})/\mathsf{NP}$$
$$\mathit{sleeps} \vdash \mathsf{S}\backslash\mathsf{NP}$$
$$\mathit{well} \vdash (\mathsf{S}\backslash\mathsf{NP})\backslash(\mathsf{S}\backslash\mathsf{NP})$$



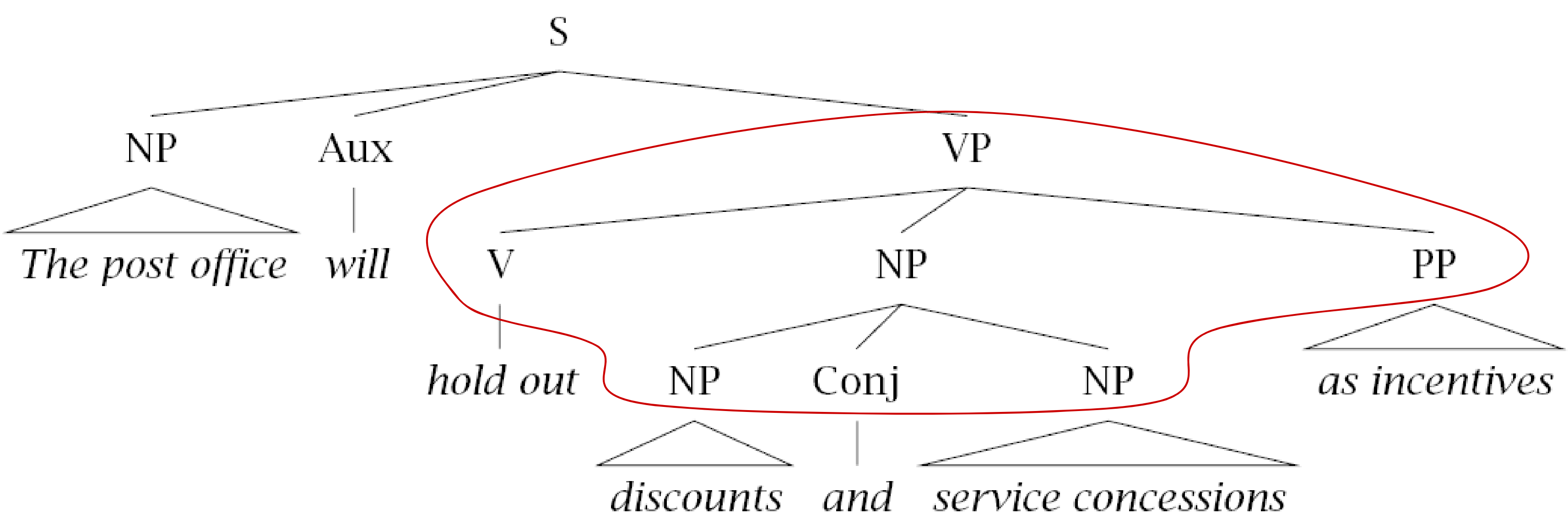
Parser Reranking

- Can first get the *k*-best list of parses based on parser probability
- Then we can fire features on full tree (as opposed to local features in the parser's dynamic program)
- Can fire non-local, global features like tree depth, width, right-branching vs left-branching, etc.
- See [Charniak and Johnson, 2005] for feature list.

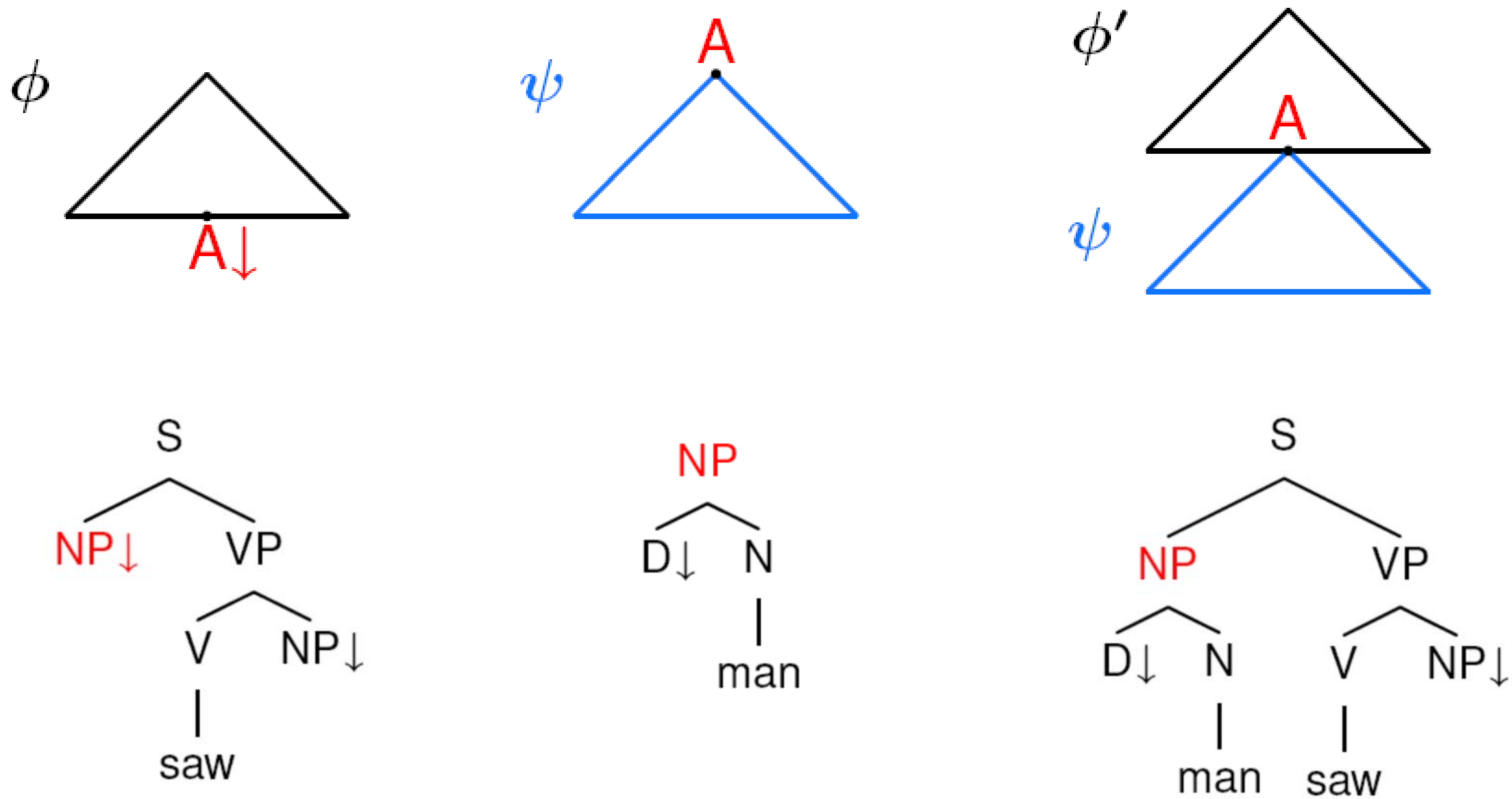


Data Oriented Parsing (TIGs)

- DOP is formally a Tree-Insertion Grammar, i.e., we can rewrite a large subtree in a single step
- Hence, this brings in derivational ambiguity



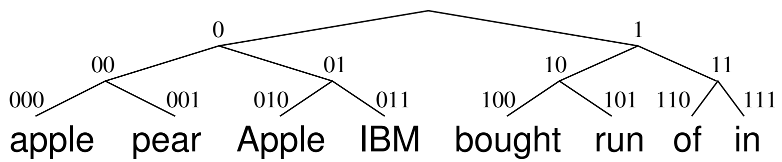
Data Oriented Parsing (TIGs)



Neural Models for Parsing

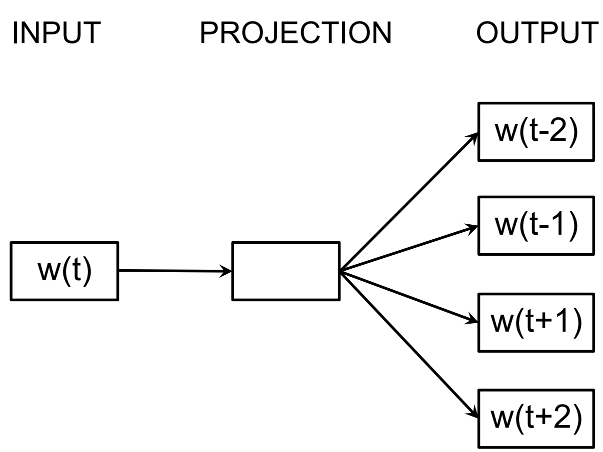
Word Embeddings for Parsing

- Discrete or continuous, trained on large amounts of context
- BROWN (Brown et al., 1992):



apple → 000
pear → 001
Apple → 010

- SKIPGRAM (Mikolov et al., 2013):



apple → [0.65 0.15 -0.21 0.15 0.70 -0.90]
pear → [0.51 0.05 -0.32 0.20 0.80 -0.95]
Apple → [0.11 0.33 0.51 -0.05 -0.41 0.50]

[Koo et al., 2008; Bansal et al., 2014]

Word Embeddings for Parsing

- Condition on dependency context instead of linear, then convert each dependency to a tuple:

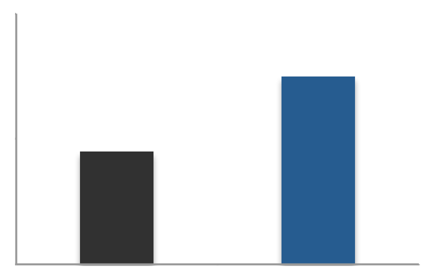
dep label grandparent parent child dep label

$$\left[PMOD_{<L>} \quad regulation_{<G>} \quad of \quad safety \quad PMOD_{<L>} \right]$$

[Mr., Mrs., Ms., Prof., III, Jr., Dr.]
[Jeffrey, William, Dan, Robert, Stephen, Peter, John, Richard, ...]
[Portugal, Iran, Cuba, Ecuador, Greece, Thailand, Indonesia, ...]

[his, your, her, its, their, my, our]
[Your, Our, Its, My, His, Their, Her]
[truly, wildly, politically, financially, completely, potentially, ...]

- 10% rel. error reduction over 90-92% parsers



[Bansal et al., 2014]

Neural Dependency Parser

[Chen and Manning, 2014; CS224n]

Softmax probabilities

Output layer y

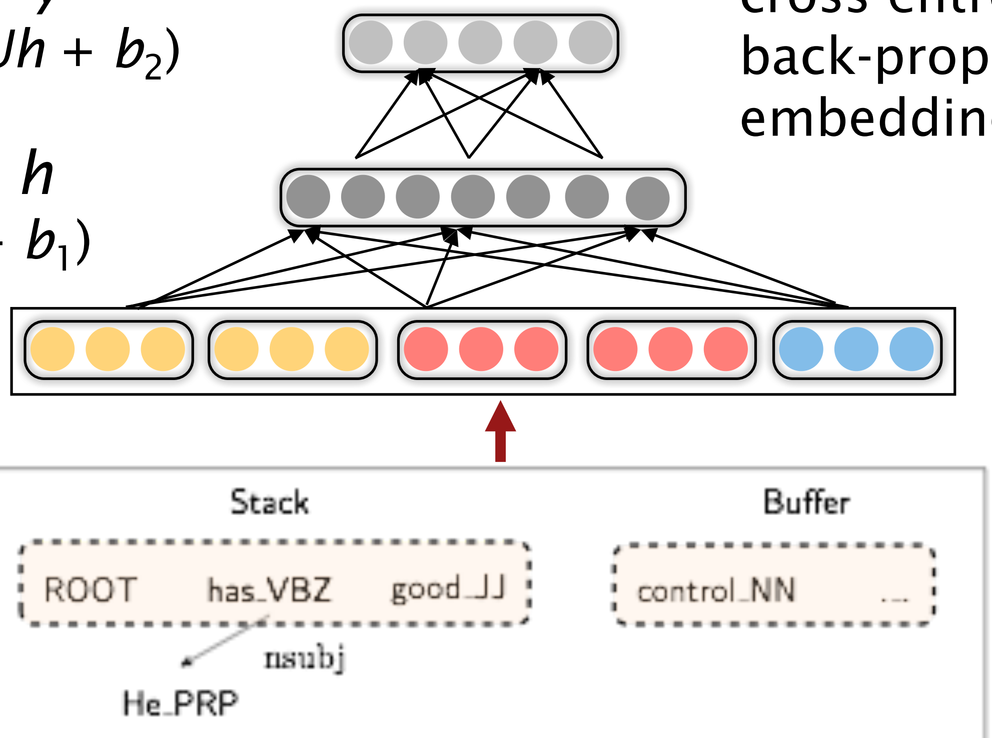
$y = \text{softmax}(Uh + b_2)$

Hidden layer h

$h = \text{ReLU}(Wx + b_1)$

Input layer x

lookup + concat



cross-entropy error will be back-propagated to the embeddings.

Neural Dependency Parser

[CS224n]

Further developments in transition-based neural dependency parsing

This work was further developed and improved by others, including in particular at Google

- Bigger, deeper networks with better tuned hyperparameters
- Beam search
- Global, conditional random field (CRF)-style inference over the decision sequence

Leading to SyntaxNet and the Parsey McParseFace model

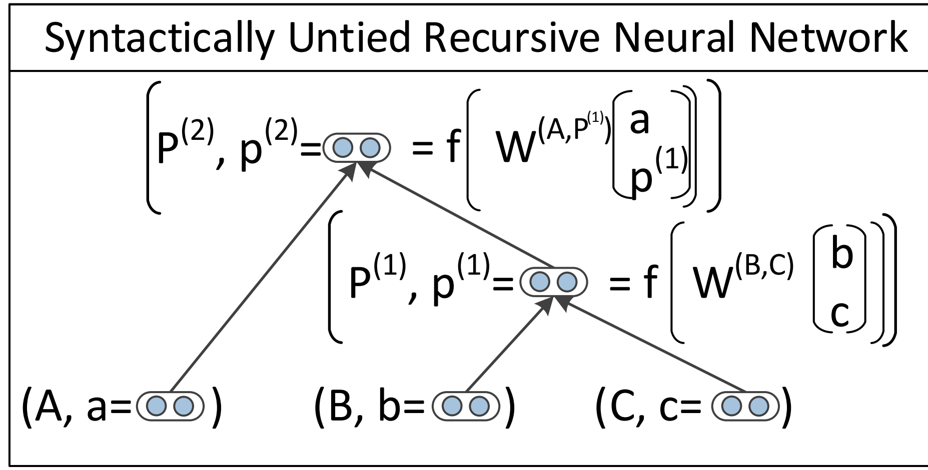
https://research.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html

Method	UAS	LAS (PTB WSJ SD 3.3
Chen & Manning 2014	92.0	89.7
Weiss et al. 2015	93.99	92.05
Andor et al. 2016	94.61	92.79

Neural Constituent Parser

[Socher et al., 2013; CS224n]

- Compositional Vector Grammar (CVG)



$$p^{(1)} = f\left(W^{(B,C)}\begin{bmatrix} b \\ c \end{bmatrix}\right)$$

$$s\left(p^{(1)}\right) = \left(v^{(B,C)}\right)^T p^{(1)} + \log P(P_1 \to B \ \ C)$$

$$P((P_1, p_1) \to (B, b)(C, c))$$
$$= P(p_1 \to b \ \ c | P_1 \to B \ \ C) P(P_1 \to B \ \ C)$$

$$p^{(2)} = f\left(W^{(A,P_1)}\begin{bmatrix} a \\ p^{(1)} \end{bmatrix}\right)$$

$$s\left(p^{(2)}\right) = \left(v^{(A,P_1)}\right)^T p^{(2)} + \log P(P_2 \to A \ \ P_1)$$

Goodness of a tree is measured in terms of its score and the CVG score of a complete tree is the sum of the scores at each node →

$$s(\texttt{CVG}(\theta, x, \hat{y})) = \sum_{d \in N(\hat{y})} s\left(p^d\right)$$