Graph-Based Dependency Parsing

Dependency Structures

- A dependency structure is a rooted tree over the words of a sentence
    - Nodes correspond to words
    - Edges represent head-dependent relations between the words

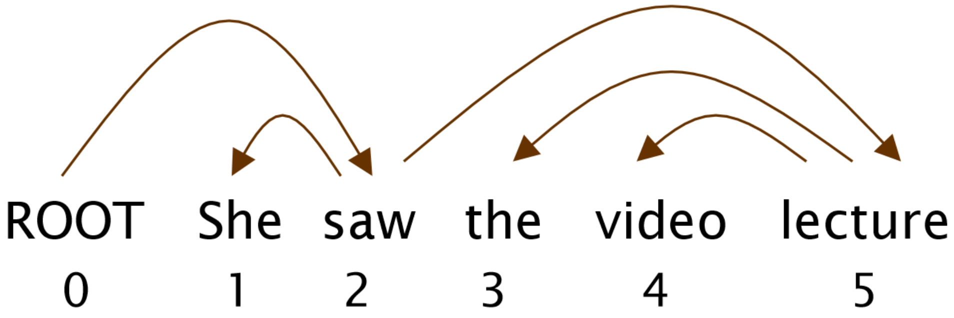
- Verbs are heads of clauses, nouns are heads of noun phrases

- Details vary across dependency schemes

Dependency Parsing

- What are the sources of information for dependency parsing?
  - Bi-lexical affinities
    [issues → the] is plausible, [outstanding → the] is not
  - Dependency distance
    mostly with nearby words
  - Intervening material: dependencies rarely span intervening verbs or punctuation
  - *Valency* – how many dependents on which side are usual for a head?

Dependency Parsing Evaluation



$$ACC = \frac{\#CORRECT\ EDGES}{\#NUMBER\ OF\ WORDS}$$

Gold

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

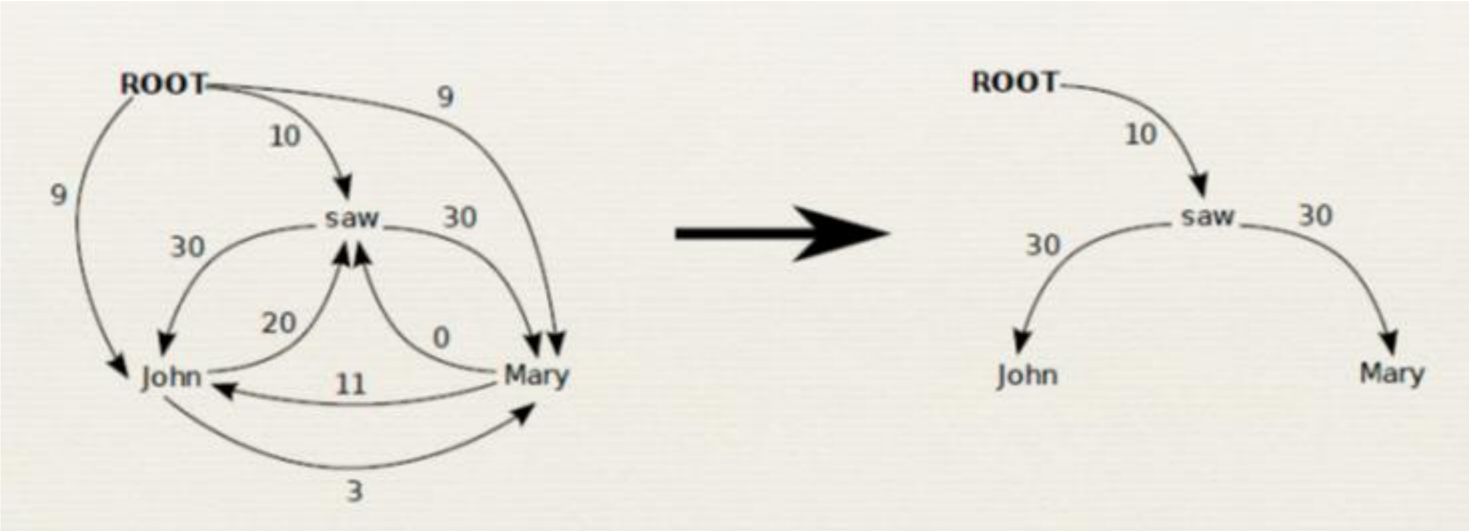
Parsed

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

index head index word edge label

Graph-based Parsing

- Graph-based parsing addresses it as a structured prediction problem
- MST Parser:
  1. Score the arcs independently, based on how likely they are to appear in a parse
  2. Find the maximum directed spanning tree over the resulting weighted graph



Online Large-Margin Training of Dependency Parsers
R. McDonald, K. Crammer, and F. Pereira, *ACL 2005*

MST Parser

Define a scoring function over all possible directed trees over $V = \{w_1, ..., w_n, ROOT\}$ where $ROOT$ is the root of the tree. Let $\Phi : V^2 \times L \to \{0,1\}^d$, where L is the label set (feature values can also be real numbers if needed) be a feature function over possible edges.

Let θ be the weight vector (the parameters of the model):

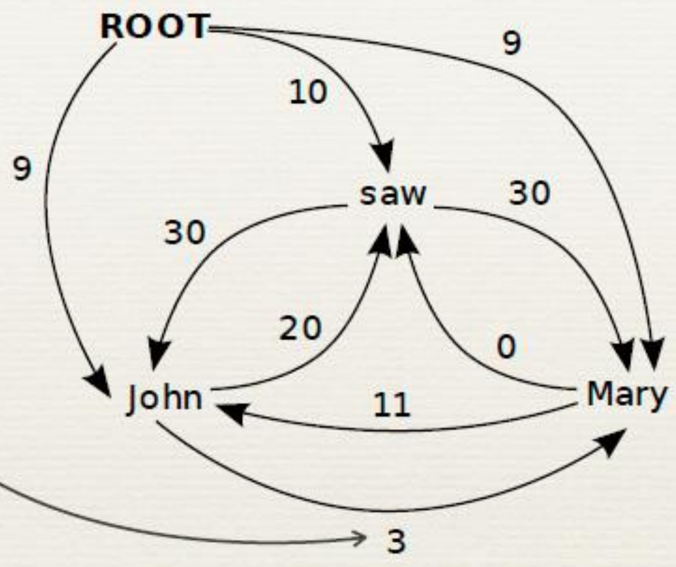
$$score_\theta(v_1, v_2, l) = \theta^t \cdot \Phi(v_1, v_2, l)$$

For a directed tree T define:

$$score_\theta(T) = \sum_{(v_1,v_2,l)\in T} score_\theta(v_1, v_2, l)$$

MST Parser

$$\text{score}(\text{John}\rightarrow\text{Mary}) = \mathbf{w}\cdot\boxed{\mathbf{f}(\text{John}\rightarrow\text{Mary})}$$



Binary Features

- Head-POS=NOUN
- Mod-POS=NOUN
- Head-Word=John
- Mod-Word=Mary
- In-Between=VERB
- Distance=2
- Direction=Right
- etc.

+conjunctions

MST Parser: Inference and Learning

- Note that inference is finding the maximum directed spanning tree
  - We can score each edge based on its features
  - This is done by the Chu-Liu Edmonds algorithm
- It is possible to define this model as log-linear:

$$Pr(T) = \frac{exp(\sum_{(v_1,v_2,l)\in T} \theta^t \cdot \Phi(v_1, v_2, l))}{Z(V,\theta)}$$

- The gradient of the log-likelihood is given by:

$$\frac{\partial LL}{\partial \theta} = \sum_{i=1}^{N} \Big[\sum_{(v_1,v_2,l)\in T_i} \Phi(v_1, v_2, l) - \mathbf{E}_T \big(\sum_{(v_1,v_2,l)\in T} \Phi(v_1, v_2, l) \big) \Big]$$

MST Parser: Inference and Learning

$$\frac{\partial LL}{\partial \theta} = \sum_{i=1}^{N} \Big[\sum_{(v_1,v_2,l) \in T_i} \Phi(v_1, v_2, l) - \mathbf{E}_T \big(\sum_{(v_1,v_2,l) \in T} \Phi(v_1, v_2, l) \big) \Big]$$

- It is possible to compute the second term exactly, but the algorithm is not simple
- The Averaged Perceptron algorithm provides a simple and useful alternative, by replacing the expectation with a maximum →

Structured Prediction Models via the Matrix-Tree Theorem
Koo et al., 2007

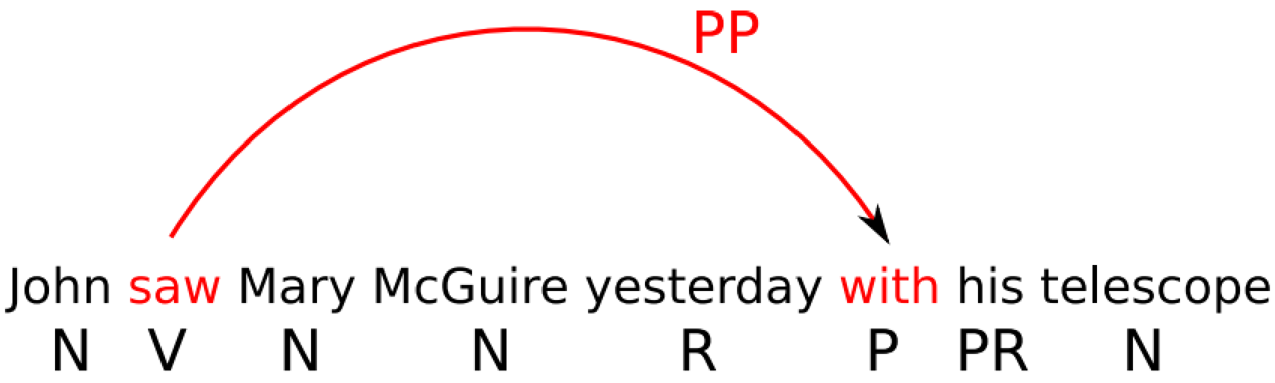
MST Parser: Inference and Learning

1. $\theta^{(0)} \leftarrow 0$
2. **for** $r = 1 \ldots N_{iterations}$
3. **for** $i = 1 \ldots N$
4. $T' \leftarrow \mathrm{arg}max_T \sum_{(v_1,v_2,l)\in T} score_\theta(T)$
5. $\theta^{((r-1)N+i)} \leftarrow \theta^{((r-1)N+i-1)} + \eta \cdot \left(\sum_{(v_1,v_2,l)\in T_i} \Phi(v_1,v_2,l) - \sum_{(v_1,v_2,l)\in T'} \Phi(v_1,v_2,l) \right)$
6. **return** $\frac{1}{N \cdot N_{iterations}} \sum_k \theta^{(k)}$

Learning Rate

Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms
Collins, 2002

Features Used in Graph-based Dep Parsing

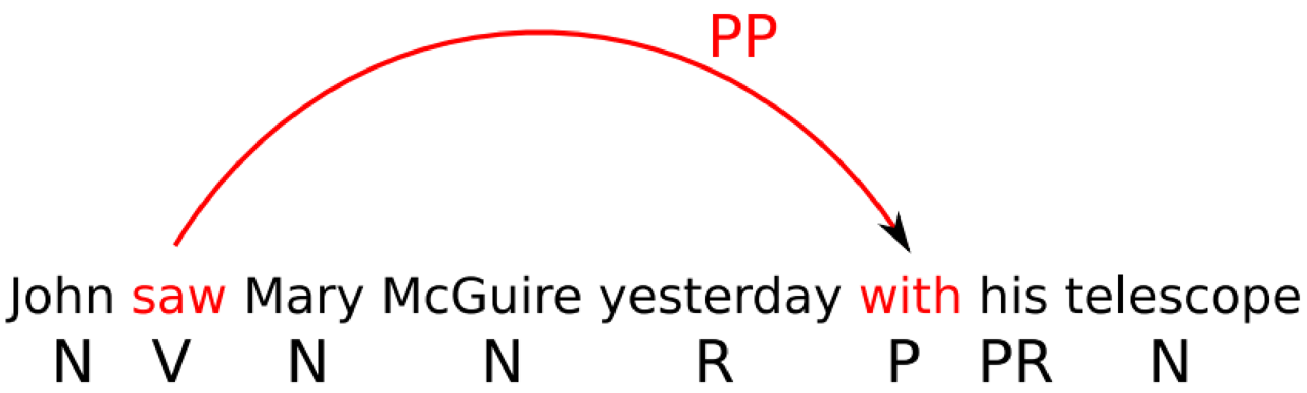


Features from McDonald et al.

- Identities of the words w_i and w_j and the label l_k

head=saw & dependent=with

Features Used in Graph-based Dep Parsing

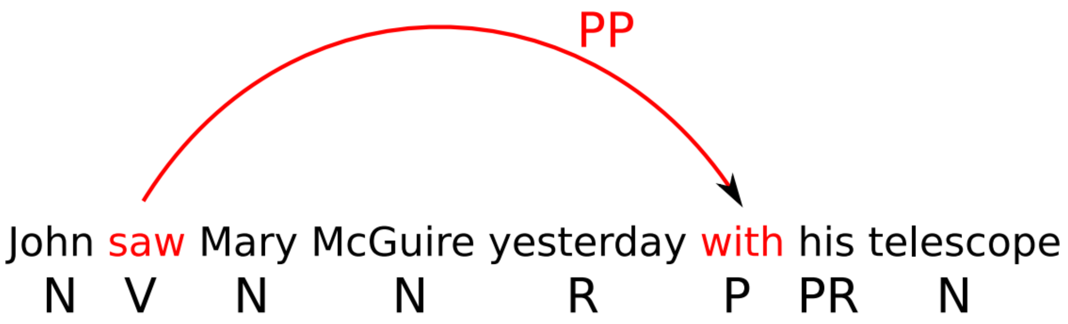


Features from McDonald et al.

- Part-of-speech tags of the words w_i and w_j and the label l_k

head-pos=Verb & dependent-pos=Preposition

Features Used in Graph-based Dep Parsing



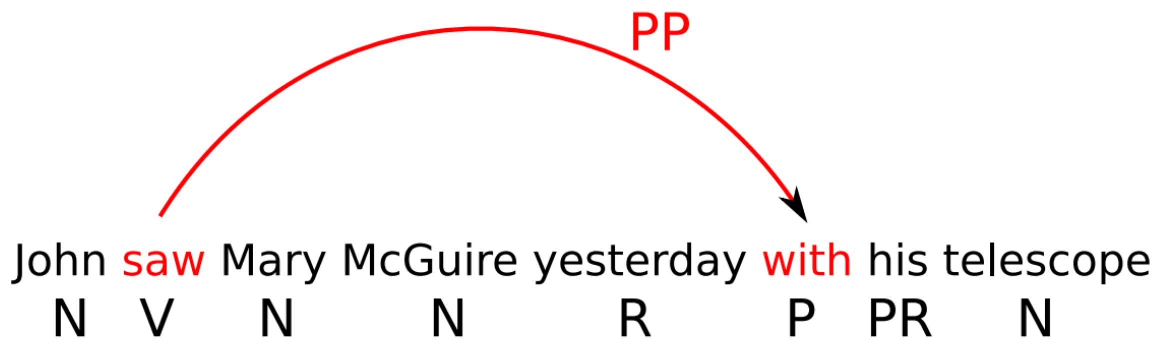
Features from McDonald et al.

- Part-of-speech of words surrounding and between w_i and w_j

inbetween-pos=Noun
inbetween-pos=Adverb
dependent-pos-right=Pronoun
head-pos-left=Noun
...

Again conjoined with the label (omitted from now on for brevity)

Features Used in Graph-based Dep Parsing



Features from McDonald et al.

- Number of words between w_i and w_j, and their orientation

arc-distance=3
arc-direction=right

Features Used in Graph-based Dep Parsing

PP

John saw Mary McGuire yesterday with his telescope

N V N N R P PR N

Label features

arc-label=PP

And Combinations of all these features…

Some Results

- The basic MST parser scores about 88% LAS on English (in domain)
- Recently, using Neural Networks, parsing performance with graph-based and transition-based methods has gone up by a few percents (!)
- Graph-based systems that use higher-order features score a few percents higher as well
  - That is, models who score does not only depend on edges (node pairs), but also on larger sub-sets of words