Lecture 8:
Graph-Based Dependency Parsing

Back to 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

- Dynamic programming:
  - Similar to lexicalized PCFG (which we will discuss)
  - Eisner's (1996) algorithm gives an improved run-time
- Transition-based algorithms:
  - Greedy
  - Left-to-right traversal of the text, each choice is done with a classifiers
- Graph algorithms:
  - Structured prediction

Will not be covered here

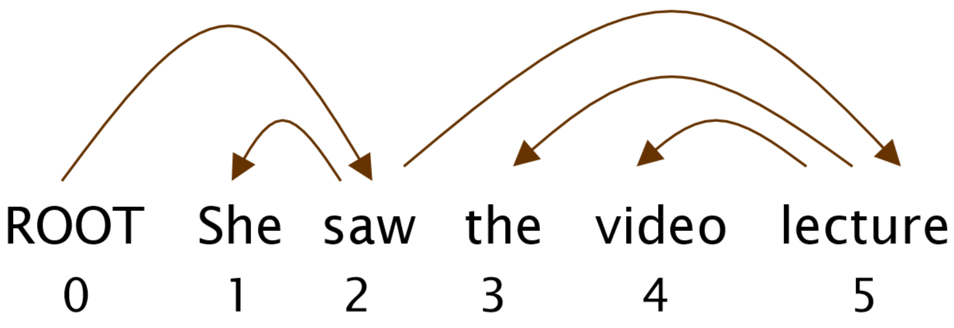
Guest Talk

Cover in This Lesson

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



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

- Accuracy (aka Attachment Score)
- There is Unlabeled Attachment Score and Labeled Attachment Score

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

Projectivity

- **Projectivity:** a dependency graph is projective if for each word *w*, its descendant nodes along with *w* form a contiguous sub-string
- Dependencies derived from constituency trees w/ contiguous phrases are projective
- Dependencies in many languages (certainly in English) tend to be projective
- But: non-projective structures can be found in treebanks

What to do with non-projective Structures?

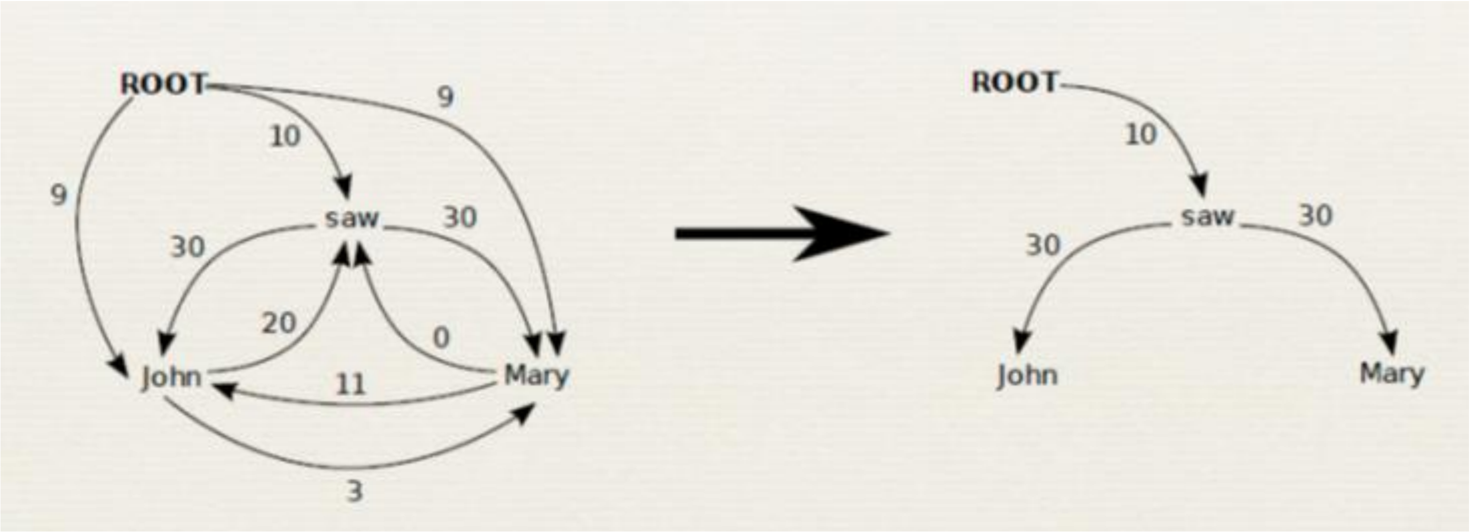
- Some common transition-based and dynamic programming algorithms only yield projective trees
  - Restricts the label space with little performance degradation

- For introducing non-projective edges:
  - Extensions to these models (Daniel will talk about that next lesson)
  - Post-processing

- Some parsing algorithms output are not restricted to projective trees

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 $\Phi : V^2 \times L \rightarrow \{0,1\}^d$, 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):

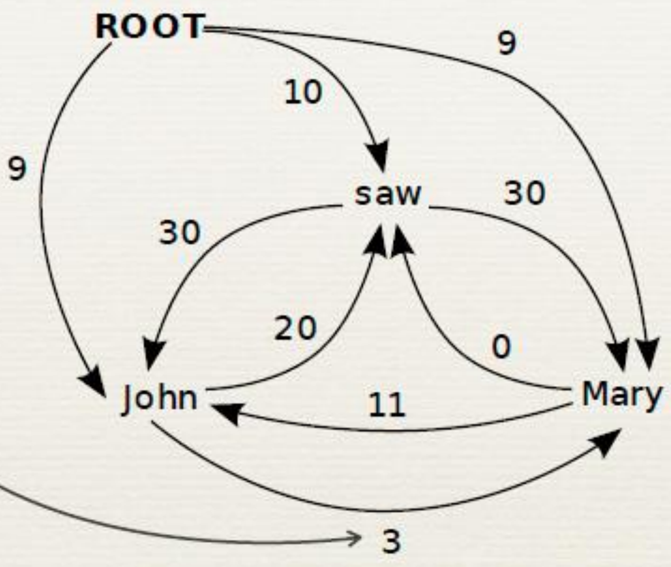
$$score_\theta(v_1, v_2, l) = \theta^t \cdot \Phi(v_1, v_2, l)$$

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

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



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 simply finding the maximum directed spanning tree
  - We can score each edge based on its features and
  - This is done by the Chu-Liu Edmonds algorithm (not necessarily projective)
- 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

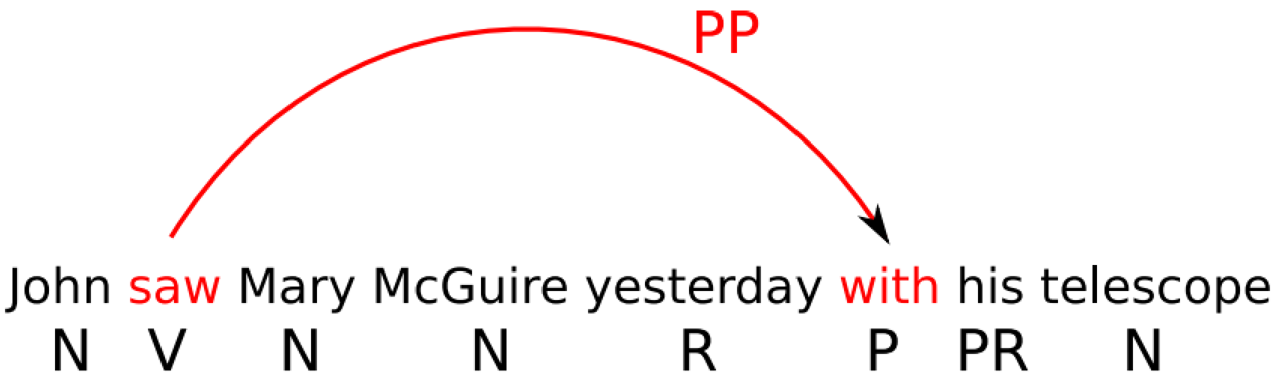
MST Parser: Inference and Learning

1. $\theta^{(0)} \leftarrow 0$
2. **for** $r = 1 \ldots N_{iterations}$
3. **for** $i = 1 \ldots N$
4. $T' \leftarrow argmax_T \sum_{(v_1,v_2,l)\in T} score_\theta(T)$
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

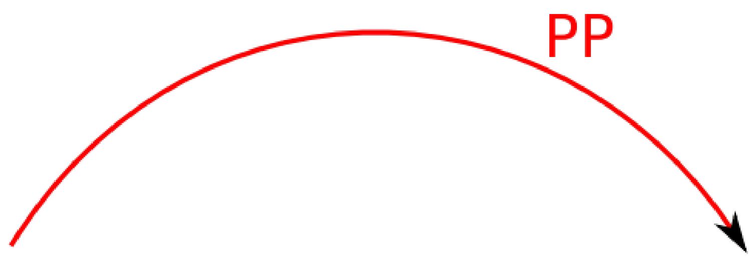


Features from McDonald et al.

- 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



John saw Mary McGuire yesterday with his telescope

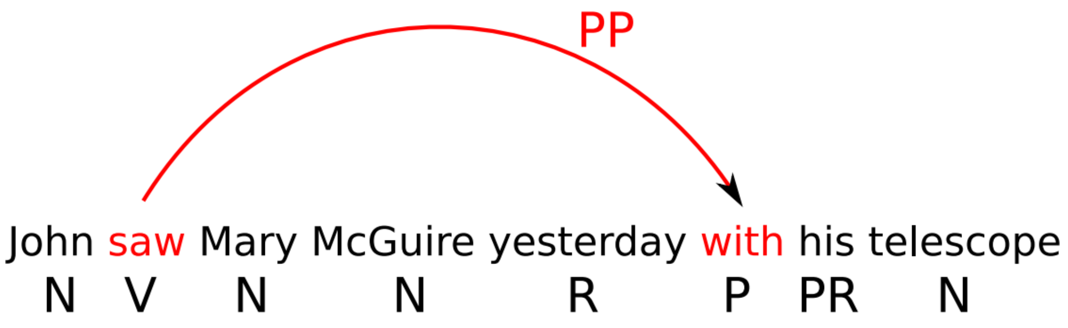
N V N N R P PR N

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



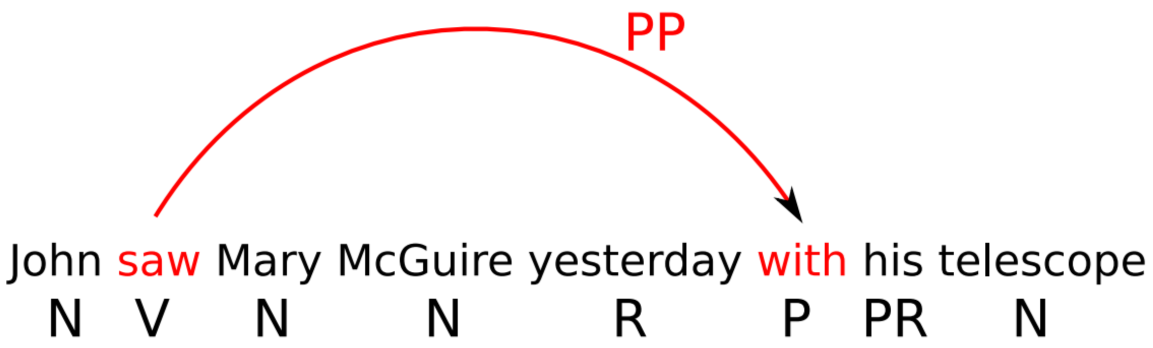
Features from McDonald et al.

- 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

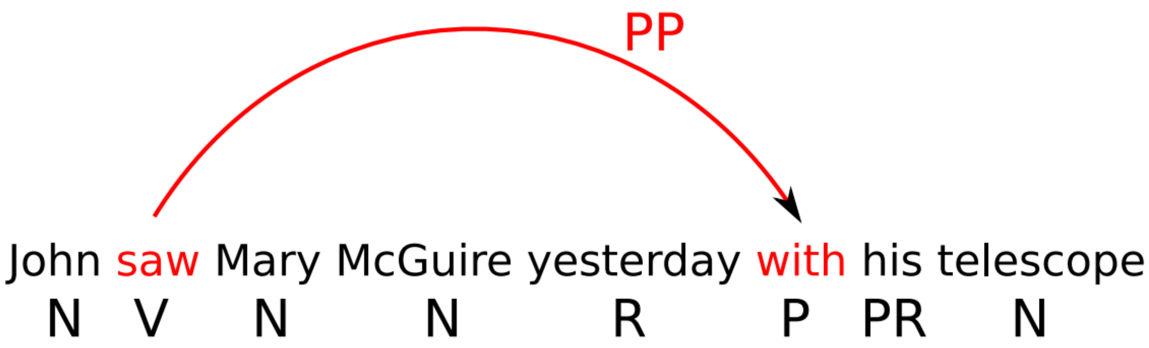


Features from McDonald et al.

- 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



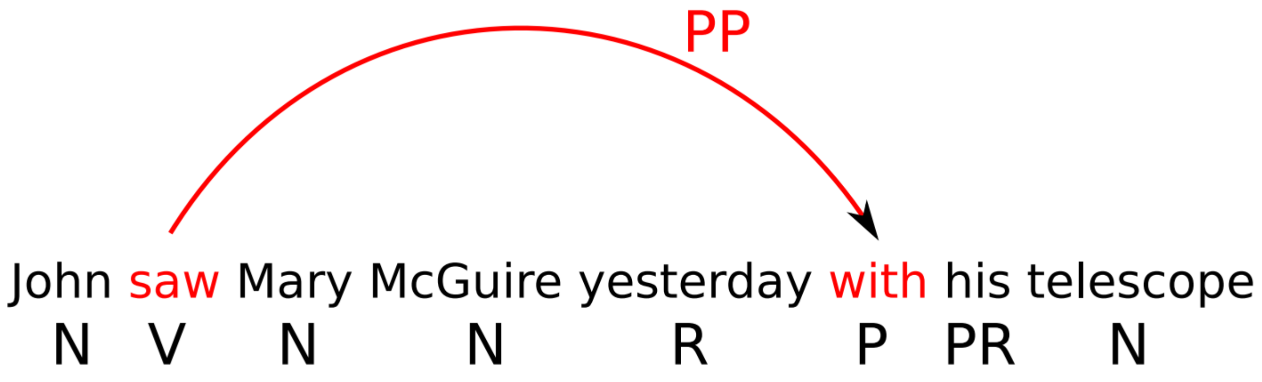
Features from McDonald et al.

- 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



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)
- Comparison: arc-standard transition-based systems less than a point lower
- 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

Results on Multiple Languages with a SOTA Parser

Method	Catalan UAS	Catalan LAS	Chinese UAS	Chinese LAS	Czech UAS	Czech LAS	English UAS	English LAS	German UAS	German LAS	Japanese UAS	Japanese LAS	Spanish UAS	Spanish LAS
Best Shared Task Result	-	87.86	-	79.17	-	80.38	-	89.88	-	87.48	-	92.57	-	87.64
Ballesteros et al. (2015)	90.22	86.42	80.64	76.52	79.87	73.62	90.56	88.01	88.83	86.10	93.47	92.55	90.38	86.59
Zhang and McDonald (2014)	91.41	87.91	82.87	78.57	86.62	80.59	92.69	90.01	89.88	87.38	92.82	91.87	90.82	87.34
Lei et al. (2014)	91.33	87.22	81.67	76.71	88.76	81.77	92.75	90.00	90.81	87.81	**94.04**	91.84	91.16	87.38
Bohnet and Nivre (2012)	92.44	89.60	82.52	78.51	88.82	83.73	92.87	90.60	**91.37**	**89.38**	93.67	92.63	92.24	89.60
Alberti et al. (2015)	92.31	89.17	83.57	79.90	88.45	83.57	92.70	90.56	90.58	88.20	93.99	**93.10**	92.26	89.33
Our Local (B=1)	91.24	88.21	81.29	77.29	85.78	80.63	91.44	89.29	89.12	86.95	93.71	92.85	91.01	88.14
Our Local (B=16)	91.91	88.93	82.22	78.26	86.25	81.28	92.16	90.05	89.53	87.4	93.61	92.74	91.64	88.88
Our Global (B=16)	**92.67**	**89.83**	**84.72**	**80.85**	**88.94**	**84.56**	**93.22**	**91.23**	90.91	89.15	93.65	92.84	**92.62**	**89.95**

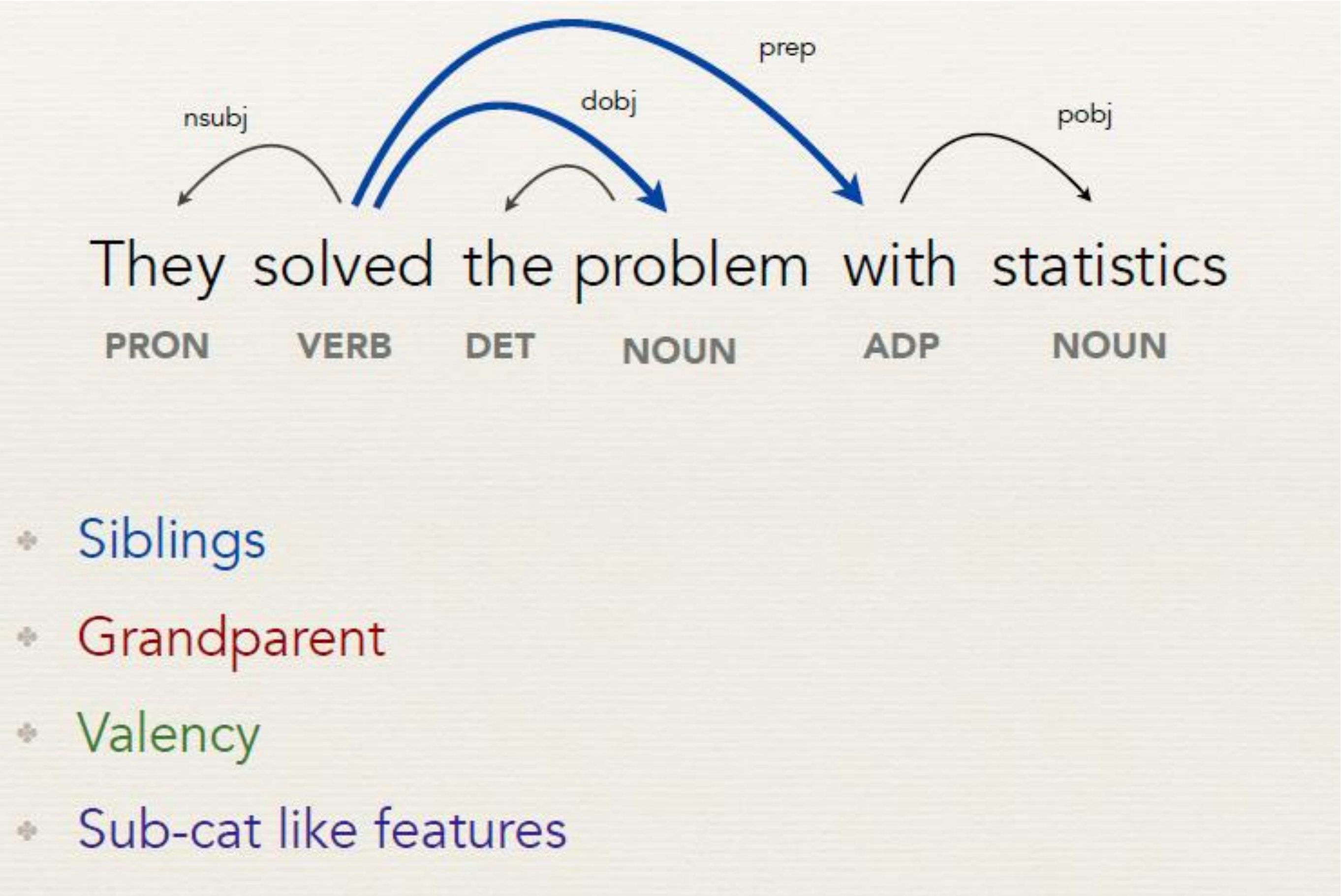
Graph-based parser

Transition-based parsers

Results on the CoNLL 2009 shared task dataset

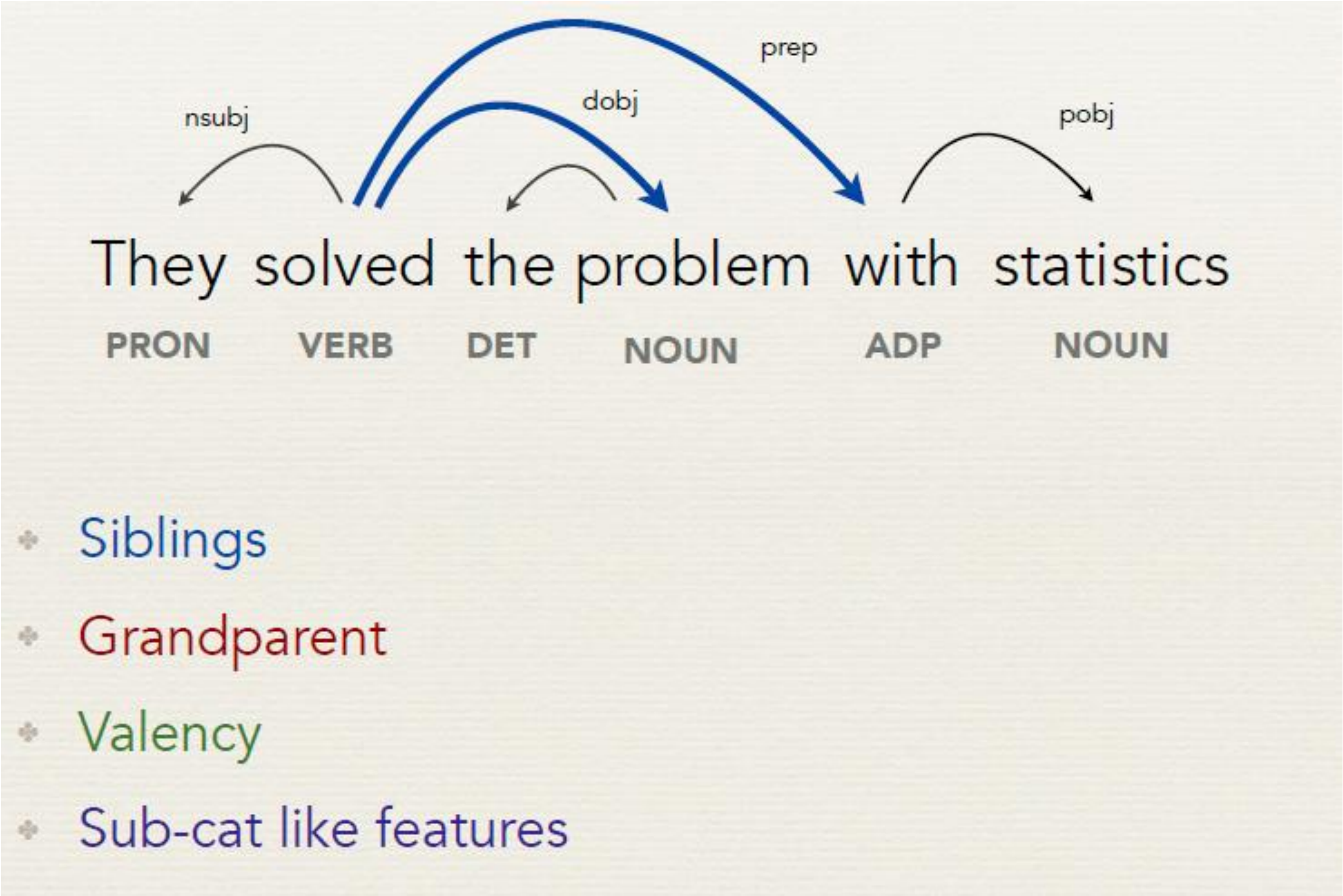
Graph Based Parsing – Higher Order Features

- MST inference works well when the model factorizes over edges
- However, it has been proven useful to add features that involve several edges, which turns inference into intractable



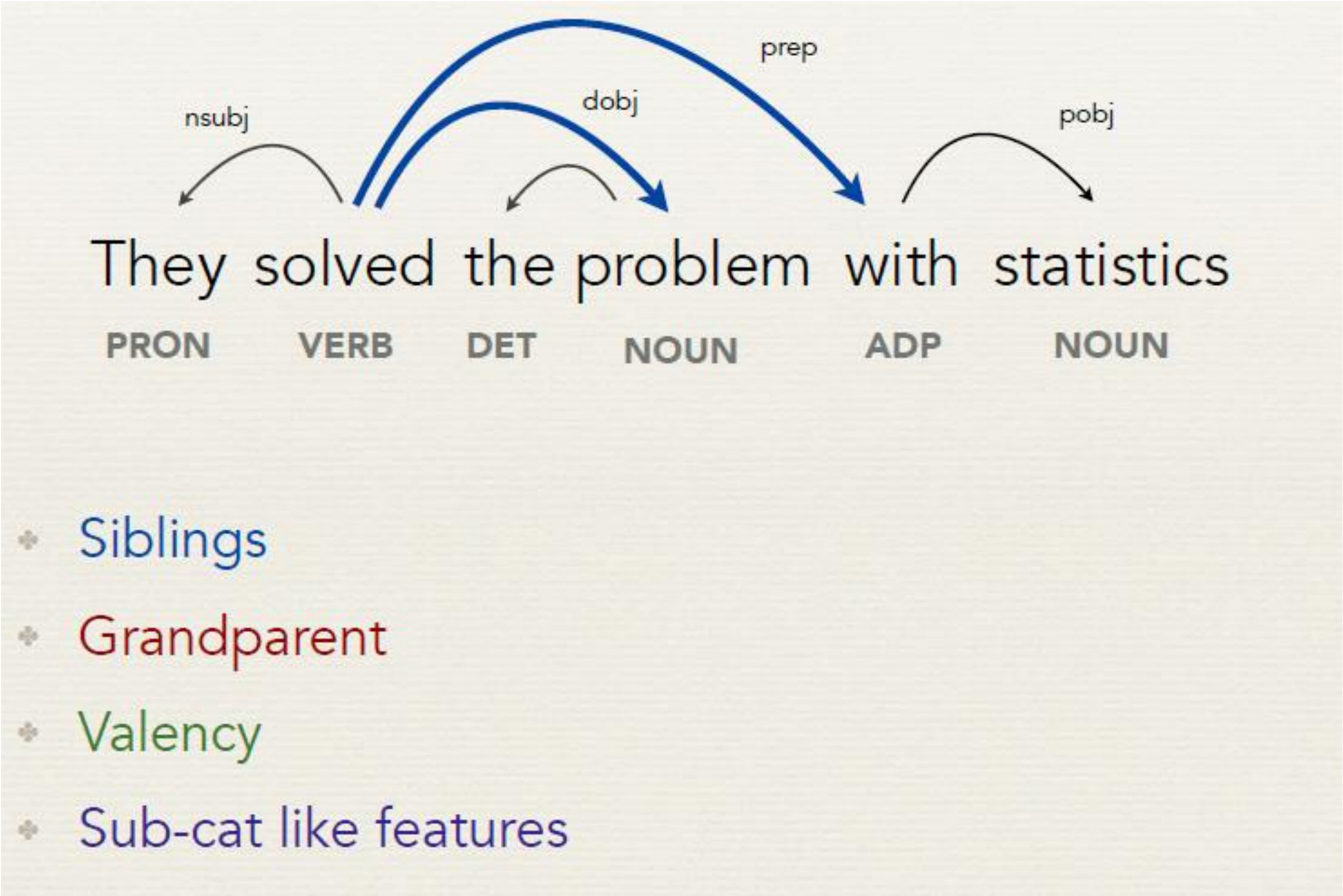
Graph Based Parsing – Higher Order Features

- **Formally,** we assume the score function factorizes over *sets* of edges
- **Siblings:** features defined over two edges that have the same parent
- **Grandparent:** features defined over two edges, where one's child is the parent of the other



Graph Based Parsing – Higher Order Features

- **Valency:** i.e., the number of children a node has. Defined over all edges sharing a parent



Subcategorization Frames

- Verbs tend to appear in specific patterns, in terms of their number of arguments, and their prepositions
  - Subject gave Object$_1$ Obect$_2$
  - Subject gave Obect$_2$ to Object$_1$
  - Object$_1$ was given Obect$_2$
  - It was Object$_2$ that was given to Object$_1$
  - …
- Encoding what sub-categorization frame is associated with each verb can promote subcategorization frames that appeared in the training data and vice-versa
- This feature is defined over all children of the verb and some of their children

Graph-based Parsing – Higher Order Features

- While inference problem becomes intractable when the score function factorizes multiple edges (even pairs of edges), approximate methods exist for computing inference
- Heuristics exist for computing such parses greedily:
    - Yuan Zhang, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2014. Greed is good if randomized: New inference for dependency parsing. In EMNLP
    - Ilan Tchernowitz, Liron Yedidsion and Roi Reichart. 2016. Effective Greedy Inference for Graph-based Non-Projective Dependency Parsing. In EMNLP