Contents

(Collins, 1996) is available at
http://www.ai.mit.edu/people/mcollins/papers/acl9629.ps or
http://citeseer.nj.nec.com/collins96new.html

1 Preliminaries

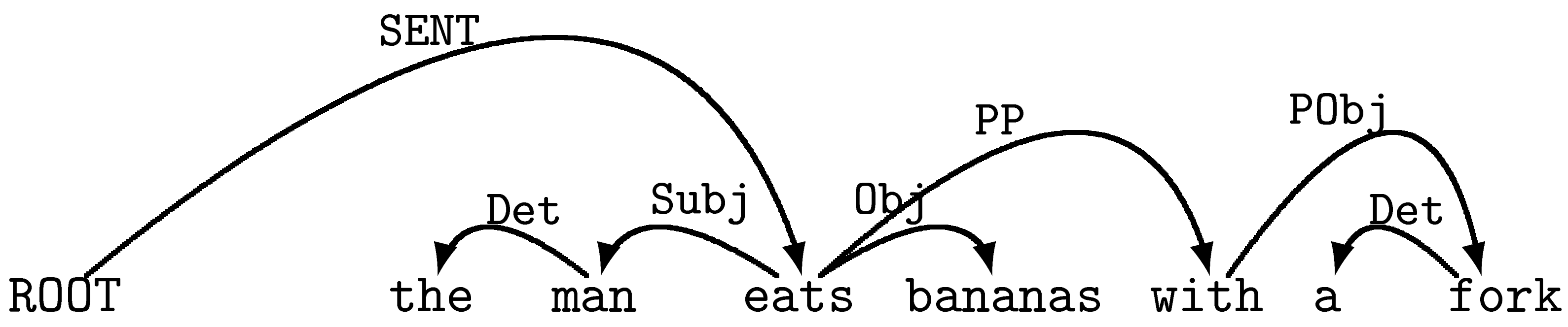
- non-lexicalized PCFGs are not enough
- flat CFG rules, especially in Treebank
- Lexicalized PCFG models such as SPATTER are complex

2 Extracting Dependencies from Treebanks

2.1 What is Dependency?

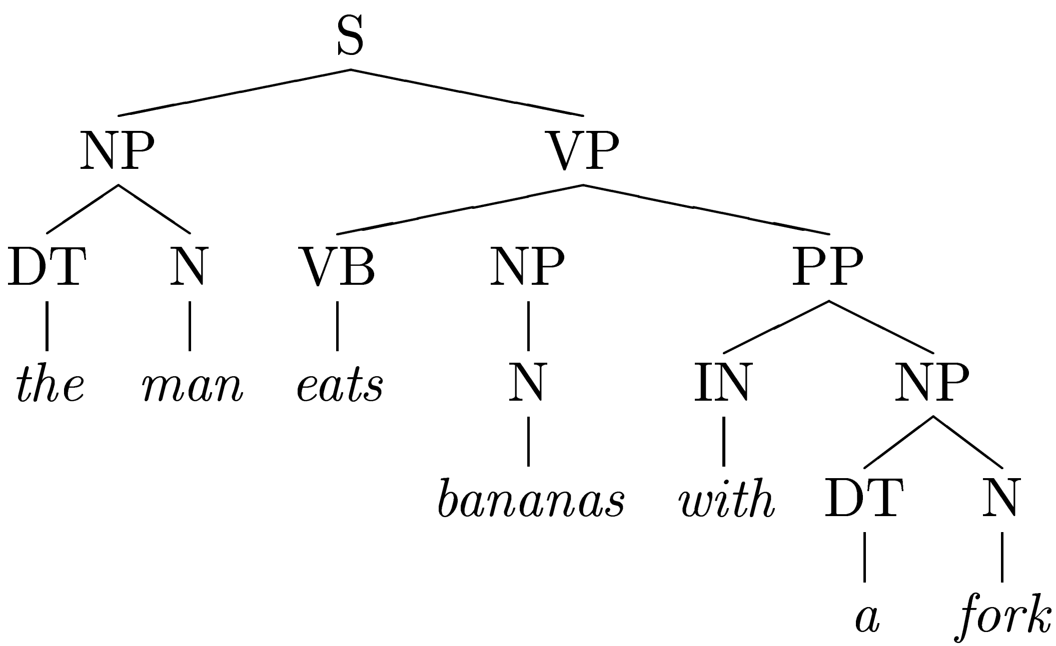
Dependency Grammar focuses on the dependencies between words

An example of a dependency structure:



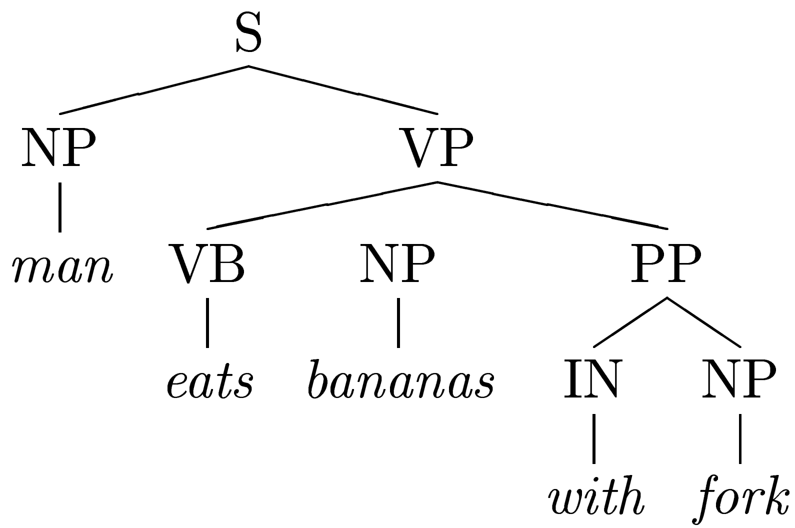
2.2 Sentences in the Treebank representation

The same sentence in Treebank representation:



2.3 Mapping Treebank trees to Dependencies

1. Use the heads of base NPs (base NP=unnested NPs)

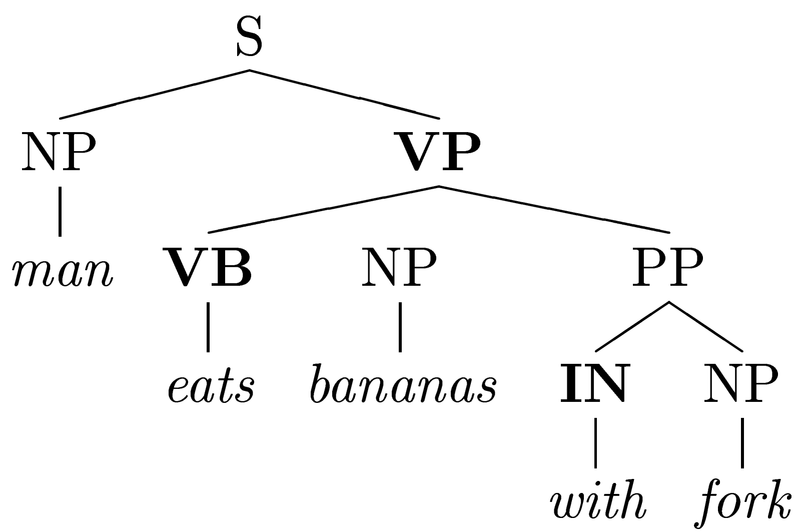


Mapping Treebank Trees to Dependencies - ctd.

2. Establish **head** for each CFG rewrite rule, e.g.

   S $\longrightarrow$ NP **VP**

   PP $\longrightarrow$ **IN** NP



Mapping Treebank Trees to Dependencies - ctd.

3. Dependency = $arrow\_from$ each dep. to its head with type t:
   t = $\langle Dependent, MotherNode, Head\rangle$ if head is to the right OR
   t = $\langle Head, MotherNode, Dependent\rangle$ if head is to the left

$\langle NP, S, VP\rangle$

`man eats` $\quad arrow\_from(loc_{man}) = (loc_{eats}, \langle NP, S, VP\rangle)$

$\langle VB, VP, NP\rangle$

`eats banana`

$\langle VB, VP, PP\rangle$

`eats with`

$\langle IN, PP, NP\rangle$

`with fork`

2.4 The advantage

Breaking up CFG rules into individual dependencies - less sparse data, more valuable information

VP ⟶ V NP (“gives the money”)
VP ⟶ V NP NP (“gives them all his money”)
VP ⟶ V NP PP (“gives his money to the poor”)
VP ⟶ V PP (“gives to the poor”)

cf. NP ⟶ DT $ CD NN (”the $ 200 hat”)

3 The Statistical Model

P[$Tree$|$Sent$]

P[B|$Sent$] P[D|$Sent$]

B=$Base\ NP\ Model$ D=$Dependency\ Model$

- Base NP Model: Use head of NPs only, chunking techniques
- Dependency Model: The core of Collins' paper

$$P(D|Sent, B) = \prod_{j=1}^{m} P(arrow\_from(j)|Sent, B) \tag{1}$$

Statistical Model - ctd.

Calculate MLE probabilities for relation R from training corpus (relative frequencies):
COUNT = Number of occurrences in same sentence

$$P(R|\langle depword, deptag\rangle \wedge \langle headword, headtag\rangle) =$$

$$\frac{COUNT(R \wedge \langle depword, deptag\rangle \wedge \langle headword, headtag\rangle)}{COUNT(\langle depword, deptag\rangle \wedge \langle headword, headtag\rangle)} \tag{2}$$

At parsing, the expected probability for a current word w_j to have a dependency of type R_j to some head h_j, i.e.
$arrow\_from(w_j) = (h_j, R_j)$, is in direct correlation to the MLE probability $P(R_j|\langle w_j, wtag_j\rangle \wedge \langle h_j, htag_j\rangle)$

Statistical Model - ctd.

The best dependency-model parse maximizes over the product of all the dependencies thus possible in the current sentence.

$$argmax_T P(D|Sent) = \prod_{j=1}^{m} P(R_j | \langle w_j, wtag_j \rangle \wedge \langle h_j, htag_j \rangle) \qquad (3)$$

Dependency Probability for current word w_j is in *direct relation* to MLE probability. For maximizing, the denominator does not matter.

4 Insufficiencies of the Core Model

- The only boundary for dependencies is the sentence
- Projective dependencies are not preferred over unbounded dependencies
- Sparse data problems
- Independence assumptions: no probability relations across single dependencies

4.1 Only dependency boundary is the sentence

Longer and shorter distance dependencies have equal weights. Collins thus introduces the distance measure heuristics

- Distance
- Punctuation
- Intervening verbs
- Adjacency

NB: These heuristics are non-linguistic “hacks”

4.2 Projective dependencies are not preferred over unbounded dependencies

Adjacency is an incomplete form of the Projectivity Constraint (Adjacency of higher nodes).

In the base-NP model this insufficiency is less serious. 74.2% of all WSJ dependencies are adjacent (distance=1).

This insufficiency was only corrected in (Collins, 1997)

4.3 Sparse data problems

At parsing, often no $\langle w_j, wtag_j\rangle \wedge \langle h_j, htag_j\rangle$ - pairs exist. Collins thus backs off to tags only:

$$COUNT(\langle w_j, wtag_j\rangle \wedge \langle h_j, htag_j\rangle)$$
$$> COUNT(\langle w_j, wtag_j\rangle \wedge \langle htag_j\rangle)$$
$$= COUNT(\langle wtag_j\rangle \wedge \langle h_j, htag_j\rangle)$$
$$> COUNT(\langle wtag_j\rangle \wedge \langle htag_j\rangle)$$

4.4 Independence assumptions: no probability relations across several dependencies

Some syntactic relations span several dependencies. E.g. in PP-attachment, the relation $\langle verb/noun, prep, PP - headnoun\rangle$ is used by current approaches for resolving PP-Attachment.

5 Conclusions

- The system is simple but performs very well
- Breaks up flat & sparse CFG rules into individual dependencies
- Importance of lexicalized data
- Many non-linguistic heuristics

References

Collins, Michael. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 184–191, Philadelphia.

Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. of the 35th Annual Meeting of the ACL*, pages 16–23, Madrid, Spain.