Computational Linguistics

CSC 485/2501
Fall 2021

6

6. Statistical resolution of PP attachment ambiguities

Gerald Penn
Department of Computer Science, University of Toronto

Statistical PP attachment methods

- A classification problem.
- Input: *verb, noun*$_1$*, preposition,* *noun*$_2$
  Output: *V*-attach or *N*-attach
  Possibly omitted
- Example:

examined *the raw materials* *with* *the optical microscope*

v n_1 *p* n_2

- Does not cover all PP problems.

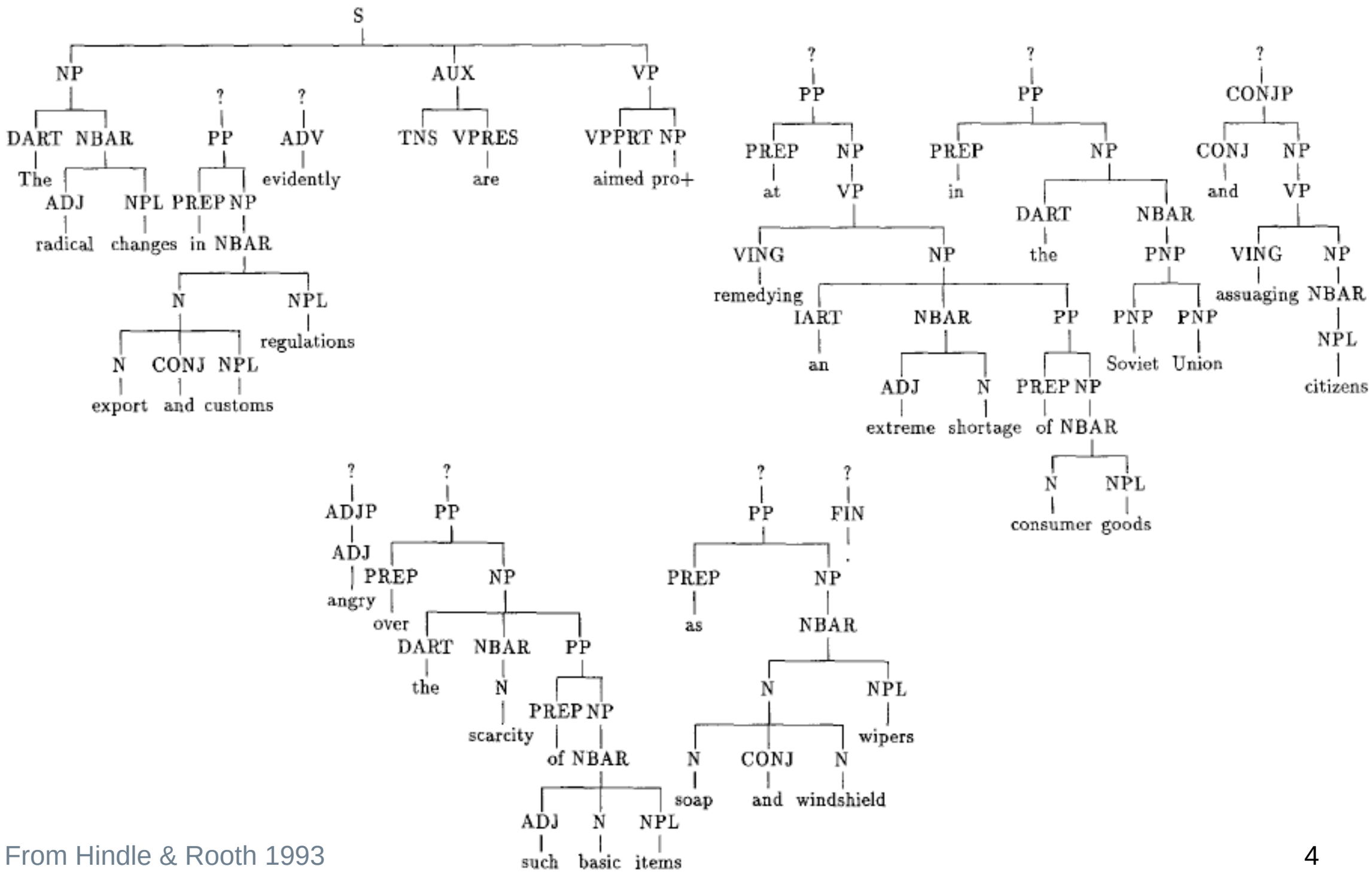
Hindle & Rooth 1993: Input 1

- **Corpus:** *Partially parsed* news text.

—Automatic.
—Many attachment decisions punted.
—A collection of parse fragments for each sentence.

Hindle, Donald and Rooth, Mats. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1), 1993, 103–120.

The radical changes in export and customs regulations evidently are aimed at remedying an extreme shortage of consumer goods in the Soviet Union and assuaging citizens angry over the scarcity of such basic items as soap and windshield wipers.



From Hindle & Rooth 1993

Hindle & Rooth 1993: Input 2

- **Data:** [*v*,*n*,*p*] triples; *v* or *p* may be null; *v* may be –.

The radical changes in export and customs regulations evidently are aimed at remedying an extreme shortage of consumer goods in the Soviet Union and assuaging citizens angry over the scarcity of such basic items as soap and windshield wipers.

v	*n*	*p*
–	change	in
aim	PRO	at
remedy	shortage	of
NULL	good	in
assuage	citizen	*NULL*
NULL	scarcity	of

From Hindle & Rooth 1993

Hindle & Rooth 1993: Algorithm 1

- **Idea:** Compute ***lexical associations*** (LAs) between *p* and each of *v*, *n*.

— Is the *p* more associated with the *v* or with the *n*?

- Learn a way to compute LA for each [*v*,*n*,*p*] triple.
- Use to map from [*v*,*n*,*p*] to {*V*-attach, *N*-attach}.

Hindle & Rooth 1993: Algorithm 2

Method: Bootstrapping.

1. Label unambiguous cases as *N*- or *V*-attach:
   When *v* or *p* is NULL, *n* is pronoun, or *p* is *of*.
2. Iterate (until nothing changes):
   a. Compute ***lexical association*** score for each triple from data labelled so far.
   b. Label the attachment of any new triples whose score is over threshold.
3. Deal with "leftovers" (random assignment).

Test cases: Compute the LA score (or fail).

Hindle & Rooth 1993: Algorithm 3

- ***Lexical association*** score: log-likelihood ratio of verb- and noun-attachment.

$LA(v,n,p) =$

$$\log_2 P(V\text{-attach } p|v,n)/P(N\text{-attach } p|v,n)$$

- Can't get these probabilities directly — data are too sparse.
- So estimate them from the data that we *can* get.

Hindle & Rooth 1993: Algorithm 4

- ***Lexical association*** score: log-likelihood ratio of verb- and noun-attachment.

$LA(v,n,p) =$

$$\log_2 P(V\text{-attach } p|v,n)/P(N\text{-attach } p|v,n)$$

1: $\approx P(V\text{-attach } p|v)\, P(NULL|n)$

2: $\approx P(N\text{-attach } p|n)$

Based on frequency counts c in the labelled data.

What are these probabilities "saying"?

- Why ratio of probabilities? Why log of ratio?

Hindle & Rooth 1993: Example 1

Moscow sent more than 100,000 soldiers into Afghanistan …

Choose between:

V-attach: [VP *send* [NP … *soldier NULL*] [PP *into*…]]
N-attach: [VP *send* [NP … *soldier* [PP *into*…]]…]

Hindle & Rooth 1993: Example 2

1. P(*V*-attach *into*|*send, soldier*)

   ≈ P(*V*-attach *into*|*send*) • P(*NULL*|*soldier*)

$$\frac{c(send, into)}{c(send)} = .049 \qquad \frac{c(soldier, NULL)}{c(soldier)} = .800$$

2. P(*N*-attach *into*|*send, soldier*)

   ≈ P(*N*-attach *into*|*soldier*)

$$\frac{c(soldier, into)}{c(soldier)} = .0007$$

$$LA(send, soldier, into) = \log_2(.049 \times .800/.0007) \approx 5.81$$

Hindle & Rooth 1993: Results

- Training: 223K triples
  Testing: 1K triples
  Results: 80% accuracy
  (Baselines: 66% by noun attachment; 88% by humans.)

Hindle & Rooth 1993: Discussion

- ***Advantages:*** Unsupervised; gives degree of preference.
- ***Disadvantages:*** Needs lots of partially parsed data. Other words don't get a vote.
- ***Importance*** to CL:
  - Use of large amounts of *unlabelled data*, with clever application of *linguistic knowledge*, to learn useful statistics.

Brill & Resnik 1994: Method

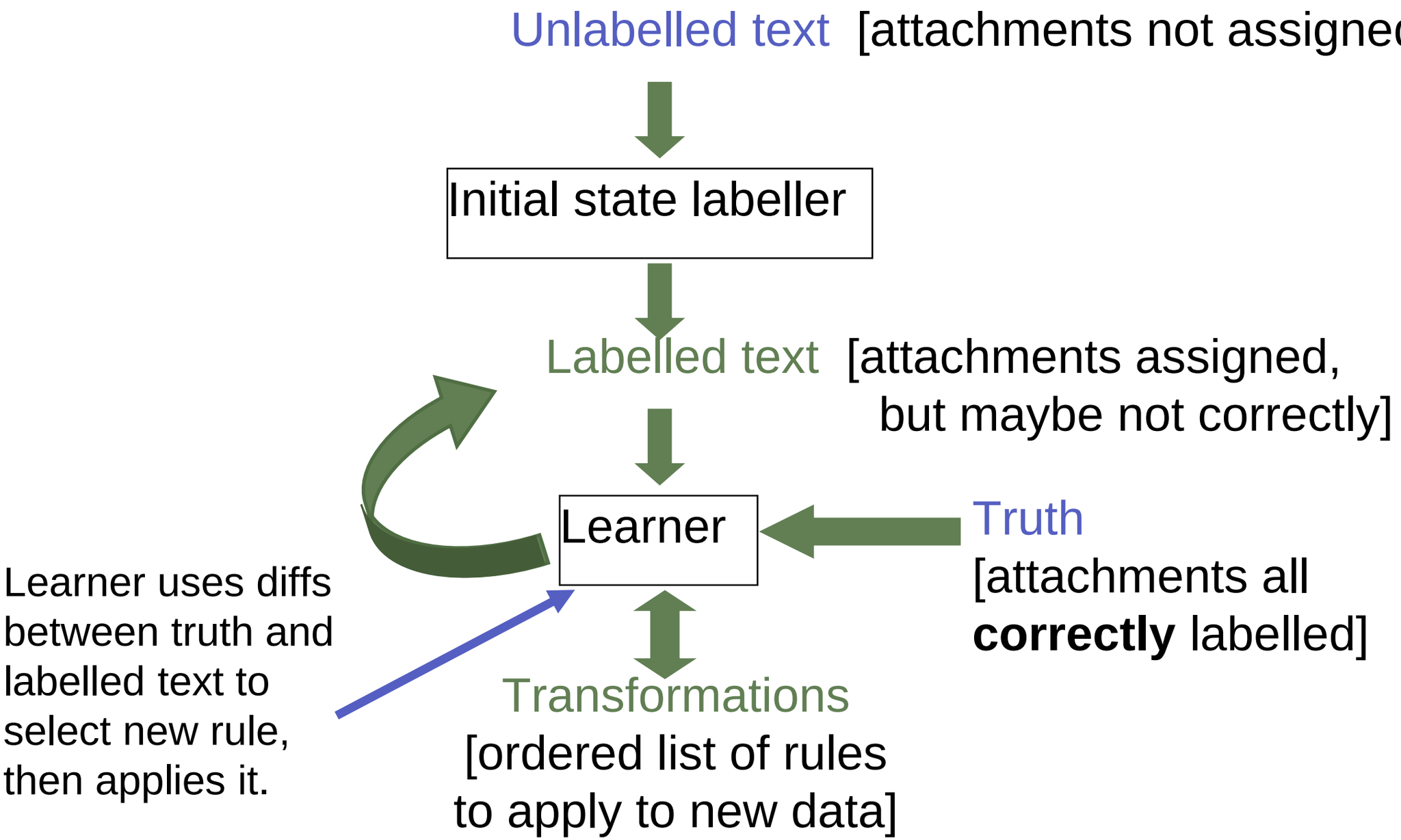
- Corpus-based, *non*-statistical method.
- ***Transformation-based learning:*** Learns sequence of rules to apply to each input item.
- Form of ***transformation rules***:
  - Flip attachment decision (from *V* to N_1 or vice versa) if $\{v,n_1,p,n_2\}$ is w_1 [and $\{v,n_1,p,n_2\}$ is w_2].

A quad: Uses head noun of PP too

Optional conjunct

- All rules apply, in order in which they are learned.

Brill & Resnik 1994: Method



Brill & Resnik 1994: Example

Some rules learned:

Start by assuming N_1 attachment, and then change attachment …

1. from N_1 to V if p is *at*.
2. from N_1 to V if p is *as*.

⋮

6. from N_1 to V if n_2 is *year*.
8. from N_1 to V if p is *in* and n_1 is *amount*.

⋮

15. from N_1 to V if v is *have* and p is *in*.
17. from V to N_1 if p is *of*.

Brill & Resnik 1994: Results

- Training: 12K annotated quads
  Testing: 500 quads
  Results: 80% accuracy
  (Baseline: 64% by noun attachment)

Brill & Resnik 1994: Discussion

- ***Advantages:*** Readable rules (but may be hard); can build in bias in initial annotation; small number of rules.
- ***Disadvantages:*** Supervised; no strength of preference. Very memory-intensive.
- ***Importance to CL:***
  - Successful general method for non-statistical learning from annotated corpus.
  - Based on popular (and relatively easily modified) tagger.

Since then…

- Modestly better methods exist (e.g., Ratnaparkhi 1998; Belinkov et al. 2014) that leverage:
  - large amounts of noisy, unannotated data (most of the partial parses were not being used anyway)
  - early attempts such as Hindle & Rooth 1993, where they are known to be very accurate
  - vector-based language models (neural methods for English?)
- …but the field mostly lost interest when it emerged that parsing decisions could be made with the assistance of language models.

Since then…

- Modestly better methods exist (e.g., Ratnaparkhi 1998; Belinkov et al. 2014).
- …but the field mostly lost interest when it emerged that parsing decisions could be made with the assistance of language models:
  - Far more context taken into account
  - Much better numbers (but lots of easy decisions folded in that inflate these – PP attachment now in high 80s)
  - PP attachment still very important for FWO languages (Do & Rehbein 2020).

Evaluating corpus-based methods 1

Questions to consider in evaluation:

- What are the required resources?
  - How is the corpus annotated?
  - What information is extracted and how?
  - How much data is needed?
- What is the information learned?
  - Statistics or rules?
  - Binary preference or strength of preference?

Evaluating corpus-based methods 2

- What is the size of the test set?
- How good is the performance?
  - Absolute performance?
  - Reduction in error rate relative to a baseline?
  - Measure just the hard cases or all of the cases?