Computational Linguistics

6

CSC 2501 / 485
Fall 2018

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,* $\boxed{\textit{noun}_2}$
  Output: *V*-attach <u>or</u> *N*-attach Possibly omitted
- Example:

$\underbrace{\boxed{\textit{examined}}}_{v}$ *the raw* $\underbrace{\boxed{\textit{materials}}}_{n_1}$ $\underbrace{\boxed{\textit{with}}}_{p}$ *the optical* $\underbrace{\boxed{\textit{microscope}}}_{n_2}$

- Does not cover all PP problems.

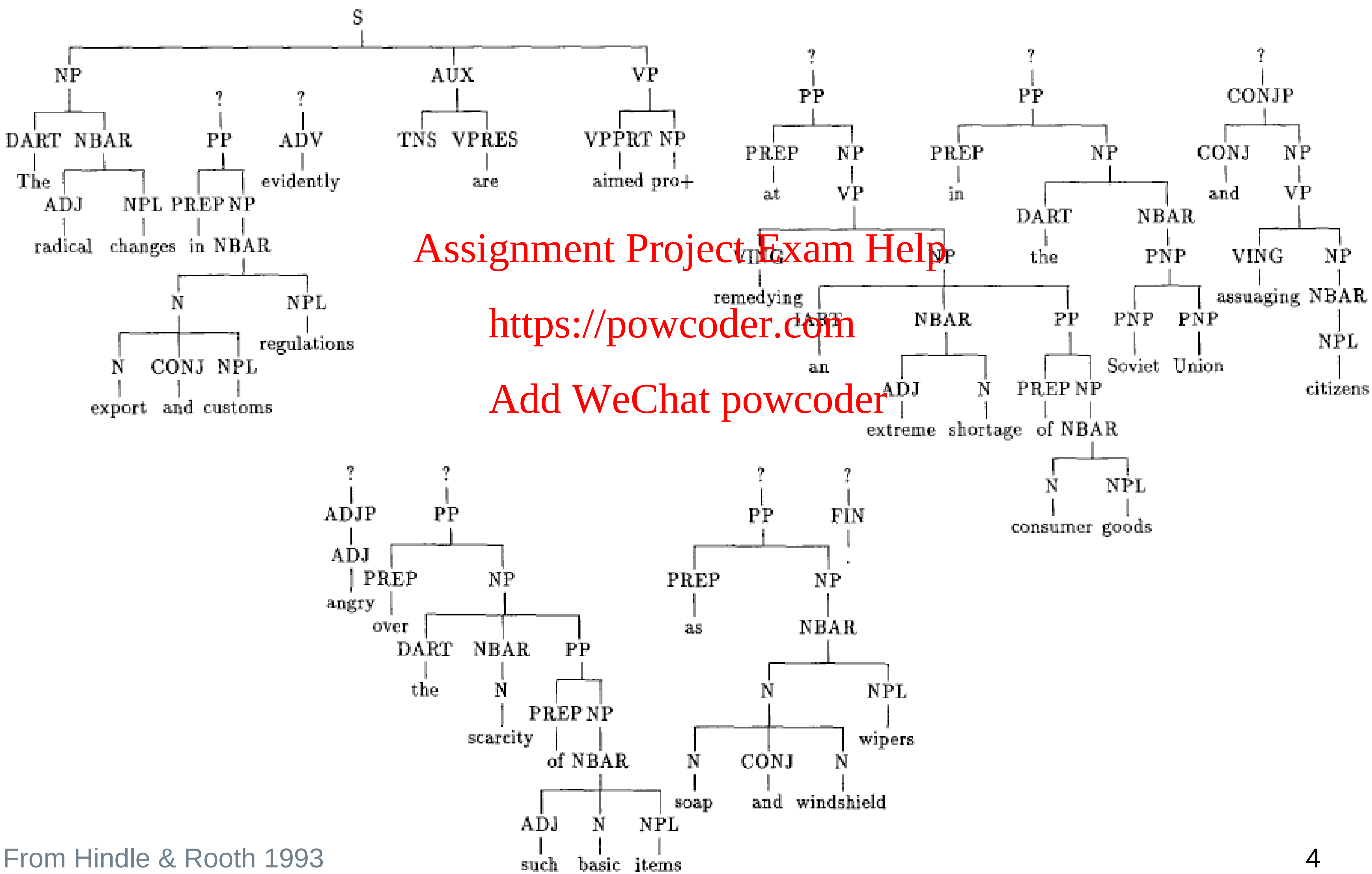
Hindle & Rooth 1993: Input 1

- **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 $\boxed{\approx P(V\text{-attach } p|v)\, P(\textsc{null}|n)}$ $\boxed{\approx P(N\text{-attach } p|n)}$ 2

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*)

$\approx$ P(*V*-attach *into*|*send*) • P(*NULL*|*soldier*)

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

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

$\approx$ P(*N*-attach *into*|*soldier*)

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

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

Unlabelled text [attachments not assigned]

Initial state labeller

Labelled text [attachments assigned, but maybe not correctly]

Learner

Truth [attachments all **correctly** labelled]

Learner uses diffs between truth and labelled text to select new rule, then applies it.

Transformations [ordered list of rules to apply to new data]

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.

Ratnaparkhi 1998: Introduction

- Using large amounts of cheap, noisy data in an unsupervised setting.
- Corpus processing:
  - PoS tagged.
  - "Chunked" using simple regular expressions.
- “Unambiguous” attachment data:
  - Based on errorful heuristics (*cf.* Hindle & Rooth).
- Quantity versus quality of data.

Ratnaparkhi 1998: Outline

The professional conduct of lawyers in other jurisdictions …

Raw text → Tagger → PoS-tagged text

The/DT professional/JJ conduct/NN of/IN lawyers/NNS in/IN other/JJ jurisdictions/NNS …

PoS-tagged text → Chunker → Tagged text with NPs replaced by head nouns

conduct/NN of/IN lawyers/NNS in/IN jurisdictions/NNS …

Tagged text with NPs replaced by head nouns → Extractor → "Unambiguous" triples (n,p,n_2) and (v,p,n_2)

(n = lawyers, p = in, n_2 = jurisdictions) …

"Unambiguous" triples → Morph processor → Final triples with words replaced by base forms

(n = lawyer, p = in, n_2 = jurisdiction) …

Ratnaparkhi 1998: Unambiguous triples

- Extract (n,p,n_2) as "unambiguous" if $p \neq$ *of* and:
  - n is first noun within k words left of p; and
  - no verb occurs within k words left of p; and
  - n_2 is first noun within k words right of p; and
  - no verb occurs between p and n_2.
- Extract (v,p,n_2) as "unambiguous" if $p \neq$ *of* and:
  - v ($\neq$ *be*) is first verb within k words left of p; and
  - no noun intervenes between v and p; and
  - n_2 is first noun within k words right of p; and
  - no verb occurs between p and n_2.
- Why are "unambiguous" data only 69% correct?

Ratnaparkhi 1998: Probabilities

- What we have:
  - Sets of (v,p) and (n,p). [doesn't use n_2]
- What we need:
  -
    argmax$_a$ P(v,n,p,a), where a is either N- or V-attach.

- Notice the probability has all three of v, n, and p, but the extracted data never have both v and n.

Ratnaparkhi 1998: Probabilities

- By the chain rule for probabilities*,
  $P(v,n,p,a) = \boxed{P(v) \bullet P(n|v)} \bullet P(a|v,n) \bullet P(p|a,v,n)$

  ❶ ❷

No influence on argmax_a

- For $a = N$-attach: [analogously for V-attach]

❶ $P(a = N|n,v) \approx \boxed{P(a = N|n)}$ Why?

$= c(\text{extracted } n)/c(\text{all } n)$

How often does this n have an attachment (to any p)?

- But why this specific conditional chain? (We'll see…)

*$P(x_1, \ldots, x_n) = P(x_1) \cdot P(x_2|x_1) \cdot P(x_3|x_2, x_1) \cdot \ldots \cdot P(x_n|x_{n-1}, x_{n-2}, \ldots, x_1)$

Ratnaparkhi 1998: Probabilities

- For $a = N$-attach: [analogously for V-attach]

❷ $P(p|a,v,n) \approx \boxed{P(p|n,a)}$

How often when this n has an attachment is it to this p?

= c(extracted n with p)/c(extracted n)
for this n

OR = [c(extracted n with p) + proportion of all n's with p]/[c(extracted n) + 1] for this n

"add-one" smoothing

Ratnaparkhi 1998: Backoffs

- When a count is zero, *back off* to uniform probabilities:
  - $P(a = N|n) = 0.5$
  - Proportion of all *n*'s with *p*
    = 1 / number of prepositions

[and analogously for *v*].

Why?

Ratnaparkhi 1998: Results 1

- Training: 900K automatically annotated tuples
  Testing: 3K tuples
  Results: 82% accuracy
  (Baseline: 70%)
- Remember: no parse trees
- And it has another feature...

Ratnaparkhi 1998: Results 2

- The *rise num to num* problem:
  - *“profits rose 46% to $52 million”*
  - *num to num* is more frequent than *rise to num*
  - So why is *V*-attach correctly preferred?
- P(*a* = *N*|*rise*,*num*) is lower than for *a* = *V.*
  - Because there are more occurrences of a *p* attached to *rise* than to *num*.
- P(*to*|*a* = *N*,*rise*,*num*) is lower than for *a* = *V.*
  - Because the proportion of all attachments to *num* that are with *to* is lower than the proportion of all attachments to *rise* that are with *to*.

Ratnaparkhi 1998: Discussion

- ***Advantages:*** unsupervised; portable (also Spanish).
- ***Disadvantages:*** very problem specific.
- ***Importance to CL:***
  - Using large amounts of unlabelled data and minimal linguistic tools/knowledge for attachment resolution.
  - Clever formulation of probability to match available info.

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?