Detecting Errors in Corpus Annotation

Exploring Endocentricity

(Dickinson & Meurers 2005)

Detmar Meurers
University of Tübingen

CLARA Thematic Training Course on Methods and Technologies
for Consolidating and Harmonising Treebank Annotation
UFAL, Charles University, Prague
December 13–16, 2010

A general linguistic insight: Endocentricity

Most natural language expressions analyzed as endocentric: a category projects to a phrase of the same category (e.g., X-bar Schema, Jackendoff 1977)

- Generally speaking, the category of the mother is constrained by the categories of the daughters.

Idea: Combine the two strands—variation detection and the insight behind endocentricity:

- To detect errors, search for variation in mother categories dominating the same daughters.

Our approach

The basic procedure

We implement this idea as follows:

1. Extract all local trees from treebank and index them by the daughters lists.
2. For each daughters list, determine the set of immediately dominating mothers in the corpus, the **immediate dominance set (ID set)**.
3. If the ID set has more than one element, the daughters list shows **ID variation**, indicating a potential error.

Our approach

An example

Example from the Wall Street Journal (WSJ) corpus, as part of the Penn Treebank 3 (Marcus et al. 1993)

- Daughters list: *ADVP VBN NP*
  (adverbial phrase, past participle, noun phrase)
- ID set: *VP* (165 times) and *PP* (2 times)
  (verb phrase and prepositional phrase)

⇒ *VP* is correct mother label, *PP* is incorrect
(i.e., *VBN* can project to *VP*, but not to *PP*)

A case study

This procedure returns 844 daughters lists with ID variation for the WSJ corpus as annotated in the Penn Treebank 3.

- Sampled and inspected 100 daughters lists/ID sets:
  - 74 pointed to at least one error
  - 24 correct ambiguities
  - 2 unclear

  We count a daughters list as erroneous if for at least one of the mothers in the ID set, every occurrence of the daughters with that mother is incorrect.
- For all 844 daughters lists, we can estimate that
  - 625 point to at least one error (95% CI: 552–697)

Detected erroneous rules

For the 100 sampled daughters lists:

- each ID set contains on average 2.91 mother categories i.e., a total of 291 distinct rules (local tree types)
- The 74 erroneous cases point to 127 erroneous rules.
  - e.g.: daughters list *IN NP* has nine mothers in ID set:
    - 3 correct (*PP*, *FRAG*, *X*)
    - 6 erroneous (*ADJP*, *ADVP*, *NP*, *SBAR*, *VP*, *WHPP*)
- The 127 rules occur 847 times (local tree tokens) in WSJ.

For the full set of 844 daughters lists, there are 2201 rules, for which we can estimate that

- 961 are erroneous rules (95% CI: 834–1087 rules), i.e, about 5.5% of all rule types (17,346) in the WSJ.

Error examples

Three main kinds of errors in the 74 erroneous cases.

- **Bracketing error (13):** *runs*, *up*, and *high commission costs* should all be sisters

(1) *Frequent trading runs [$_{VP}$ up [$_{NP}$ high commission costs]]*

- **Mother label error (41):** *past it* should be a PP

(2) *Turkey in any event is long [$_{VP}$ past/IN [$_{NP}$ it]] .*

- **Daughter label error (38):** *like* should be VB

(3) *Mr. Friend's client [. . .] didn't [$_{VP}$ like/IN [$_{NP}$ the way 0 defense attorney Tom Alexander acted during the legal proceedings *T*]] .*

From ID variation to error detection

- We have established that ID variation is useful for finding incorrectly annotated local trees.
- To make this practically useful, we want to define a heuristic for *automatically* detecting
  - which of the elements in the ID set of a given daughters list are errors and which aren't.
- What information will be useful/necessary for this?

Frequency-based error detection heuristics

Absolute frequency

Remove all rules in the ID sets which occur only once

- Based on idea that pruning low-frequency rules in parsing will not degrade performance (Gaizauskas 1995; Charniak 1996; Cardie & Pierce 1998)
- Results:

	Precision	Recall
Types	74.75% (74/99)	58.27% (74/127)
Tokens	74.75% (74/99)	8.74% (74/847)

  - Fairly high precision
  - Very low token recall

Frequency-based error detection heuristic

Relative frequency

Remove rules which occur less than 10% of the time within their ID sets.

- Results:

	Precision	Recall
Types	60.47% (78/129)	61.42% (78/127)
Tokens	9.20% (499/5424)	58.91% (499/847)

  - Fairly high recall
  - Very low token precision

Adding an ambiguity measure

Idea:

- certain pairs of mother categories are likely to occur as alternatives, regardless of their frequency.

Example:

- NP vs. NX
  - NP labels noun phrases
  - NX is used for noun phrases which share a modifier with another noun phrase
- 114 of the 844 ID sets include both NP and NX as mothers (the second most-common variation)
- e.g.: *NX* → *VBG NN* occurs only three times, but in variation with *NP* as mother, and both are correct.

A combined heuristic: frequency + ambiguity

Procedure

1. Start with relative frequency heuristic:
   - Mark as errors all ID set elements whose token occurrences are less than 10% of occurrences in ID set.
2. Restrict set of potential errors by eliminating all ID set categories deemed ambiguous:
   - Eliminate all ID set categories which, when paired with the most frequent category in the ID set, are among the top five variations in the corpus.

A combined heuristic: frequency + ambiguity

Results

	Precision	Recall
Types	73.03% (65/89)	51.18% (65/127)
Tokens	65.59% (364/555)	42.98% (364/847)

- Much better token recall
- Precision still quite high

⇒ Results are encouraging enough to try to measure the impact of removing all rules detected by this method

Impact of automatically removing errors

Setup

We tested the impact of erroneous rules in training data on PCFG parsing, using LoPar (Schmid 2000) (unlexicalized, non-headed version).

- Left-corner parser which allows for easy manipulation of the set of grammar rules.
- Used sections 2-21 of WSJ to train, section 23 to test.
- Training data, rules used:
  - All (15,246 rules): All grammar rules from the treebank without modification.
  - Reduced (14,798 rules): Grammar rules after removing rules flagged by combined frequency/ambiguity heuristic.

Impact of automatically removing errors

Results

- Standard PARSEVAL measures of bracketing precision, recall, and F-measure, for labeled evaluation:

	Precision	Recall	$F_{\beta=1}$
All	70.39%	67.31%	68.82%
Reduced	71.48%	68.40%	69.91%

- Changes significant at $\alpha = 0.001$ (using stratified shuffling)

Conclusion:

- Presence of erroneous rules in a grammar induced from a treebank is harmful for parsing precision and recall
- Targeting and eliminating erroneous rules can improve parser performance

Summary and Outlook

Summary:

- Introduced effective new way of detecting treebank errors
  - combines variation detection with endocentricity insight
- Demonstrated that removing erroneous training data detected by method improves PCFG performance

Outlook:

- Continue exploration of heuristics to improve precision/recall of errors
- Determine what exactly causes the improvement for PCFG parser
- Perform dependency-based evaluation measures
- Test methods on other treebanks with different annotation schemes

Endocentricity and Real Life Treebanks

Some treebank annotation guidelines violate endocentricity, e.g., WSJ guidelines for proper nouns (NNP, NNPS)

- Rule for POS annotation (Santorini 1990, p. 32): capitalized words which appear in a title tagged NNP

  (4) *A/NNP Tale/NNP of/IN Two/NNP Cities/NNP*

- Rule for syntactic annotation (Bies et al. 1995, p. 207): titles specified to be annotated like running text

  (5) *[$_{S-TTL}$ [$_{NP-SBJ}$ *] [$_{VP}$ Driving [$_{NP}$ Miss Daisy]]]*

⇒ WSJ includes VPs headed by NNP (VP → NNP PP):

(6) *[$_{NP-TTL-PRD}$ [$_{S}$ [$_{NP-SBJ}$ *]*
[$_{VP}$ Saved/NNP [$_{PP}$ By/NNP [The/NNP Bell/NNP]]]]

Frequency-based error detection heuristic

An example for relative frequency

Remove rules which occur less than 10% of the time within their ID sets

- e.g.: daughters list *NNP CC NNP NNP*
  - appears 86 times
  - with *UCP* as mother only twice (2.33% of 86)

  ⇒ *UCP* can be removed

Frequency-based error detection heuristics

Insufficiency of absolute frequency

Two main reasons this predictor is insufficient:

- Frequently-occurring rules which are incorrect
  - e.g.: NP → VBG appears 177 times, despite being wrong
- Infrequently-occurring rules which are correct
  - Of the 99 rules in our set which occur once, a full 25 of them are correct
  - e.g.: S → NP S occurs once, but is corect

Frequency-based error detection heuristic

Insufficiency of relative frequency

Problems for relative frequency heuristic:

- Again, infrequently-occurring rules which are correct
  - e.g.: *NX* → *NNP CC NNP NNP* is correct, despite occurring only once out of 86 total token occurrences in ID set
- Very frequent rules are too dominant:
  - Despite appearing 102 times, *NX* → *JJ NN* is under 10% threshold (*NP* appears 5972 times as mother)

⇒ Frequency-based heuristic is insufficient by itself

Exemplifying combined heuristic

Why combining frequency + ambiguity measure works

Sort out highly frequent rules based on something other than frequency

- With *JJ NN*, mother *ADJP* occurs 25 times as a mother but less than 10% of the time within the variation
  - Incorrectly flagged as an error by relative frequency heuristic alone
  - The pairing *ADJP–NP* is most frequent ambiguity, so rule is correctly not flagged as error by combined heuristic
- With *IN NP*, mother *ADVP* occurs 170 times
  - Pairing *ADVP–PP* not one of the five most frequent, so correctly flagged as an error

Full LoPar results

	Precision		Recall		$F_{\beta=1}$	
	Lab.	Unl.	Lab.	Unl.	Lab.	Unl.
All	70.39%	74.73%	67.31%	71.46%	68.82%	73.06%
Reduced	71.48%	75.68%	68.40%	72.42%	69.91%	74.01%

References

Bies, A., M. Ferguson, K. Katz & R. MacIntyre (1995). *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. University of Pennsylvania. ftp://ftp.cis.upenn.edu/pub/treebank/doc/manual/root.ps.gz.

Cardie, C. & D. Pierce (1998). Error-driven pruning of Treebank grammars for base noun phrase identification. In *Proceedings of the 17th international conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, pp. 218–224.

Charniak, E. (1996). Tree-Bank Grammars. In *AAAI/IAAI, Vol. 2*. pp. 1031–1036. URL citeseer.nj.nec.com/charniak96treebank.html.

Dickinson, M. & W. D. Meurers (2005). Prune Diseased Branches to Get Healthy Trees! How to Find Erroneous Local Trees in a Treebank and Why It Matters. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*. Barcelona, Spain. URL http://ling.osu.edu/~dm/papers/dickinson-meurers-tlt05.html.

Gaizauskas, R. (1995). *Investigations into the grammar underlying the Penn Treebank II*. Tech. Rep. Research Memorandum CS-95-25, University of Sheffield. URL citeseer.ist.psu.edu/111349.html.

Jackendoff, R. (1977). *X-bar Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT Press.

Marcus, M., B. Santorini & M. A. Marcinkiewicz (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2), 313–330. ftp://ftp.cis.upenn.edu/pub/treebank/doc/cl93.ps.gz.

Santorini, B. (1990). Part-Of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision, 2nd printing). Ms., UPenn.

Schmid, H. (2000). Parsing by Successive Approximation. In H. Bunt & A. Nijholt (eds.), *Advances in Probabilistic and other Parsing Technologies*, Dordrecht: Kluwer Academic Publishers, vol. 16 of *Text, Speech and Language Technology*, pp. 243–261.