Creating RRG treebanks through semi-automatic conversion of annotated corpora

Tatiana Bladier, Kilian Evang, Laura Kallmeyer,
Robin Möllemann & Rainer Osswald

(RRGbank Team former member: Andreas van Cranenburgh)

RRG Conference Buffalo
19-21 August 2019
University of Buffalo, NY

Overview

Motivation behind RRGbank

Creating RRGbank
- Design of the RRG structures
- Penn Treebank to RRG conversion
- Universal Dependencies to RRG conversion

Evaluation

Conclusions and Future Work

Outline

Motivation behind RRGbank

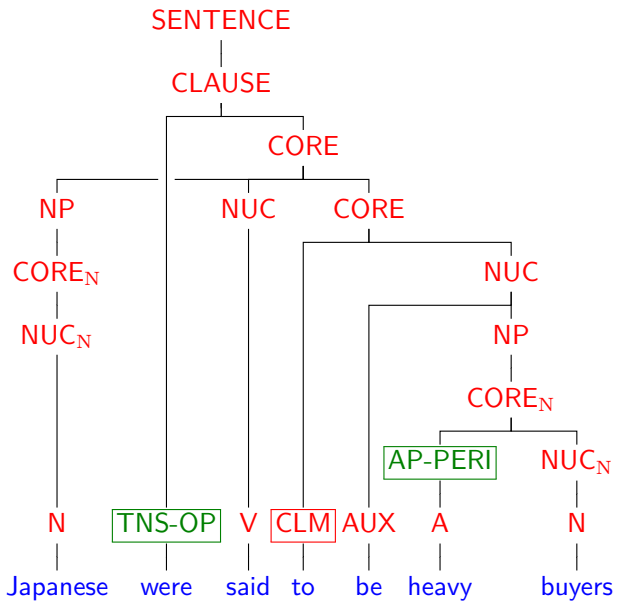
Creating RRGbank
- Design of the RRG structures
- Penn Treebank to RRG conversion
- Universal Dependencies to RRG conversion

Evaluation

Conclusions and Future Work

RRGbank

- ⋆ a large corpus of RRG annotated sentences;
- ⋆ starting point: English
  → 50 000 sentences from the Wall Street Journal;
- ⋆ future work: several languages
  → transformation from Universal Dependencies corpora
  → over 80 languages.



Why RRGbank?

- ⋆ corpus-based investigations for linguistic modeling with RRG,
- ⋆ test corpus for formalization of RRG
  → using tree grammars: Kallmeyer et al. (2013); Kallmeyer (2016); Kallmeyer and Osswald (2017),
- ⋆ test corpus for precision RRG grammars,
- ⋆ training data for supervised data-driven RRG parsing,
- ⋆ new insights into RRG for different languages.

Outline

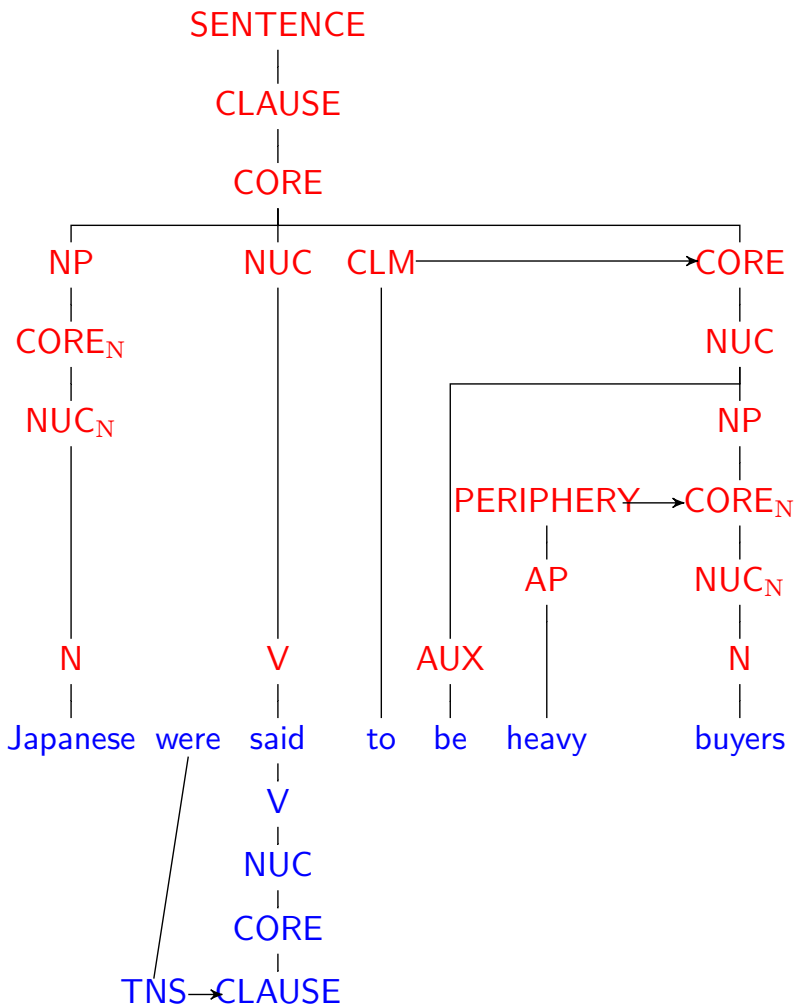
Motivation behind RRGbank

Creating RRGbank
- Design of the RRG structures
- Penn Treebank to RRG conversion
- Universal Dependencies to RRG conversion

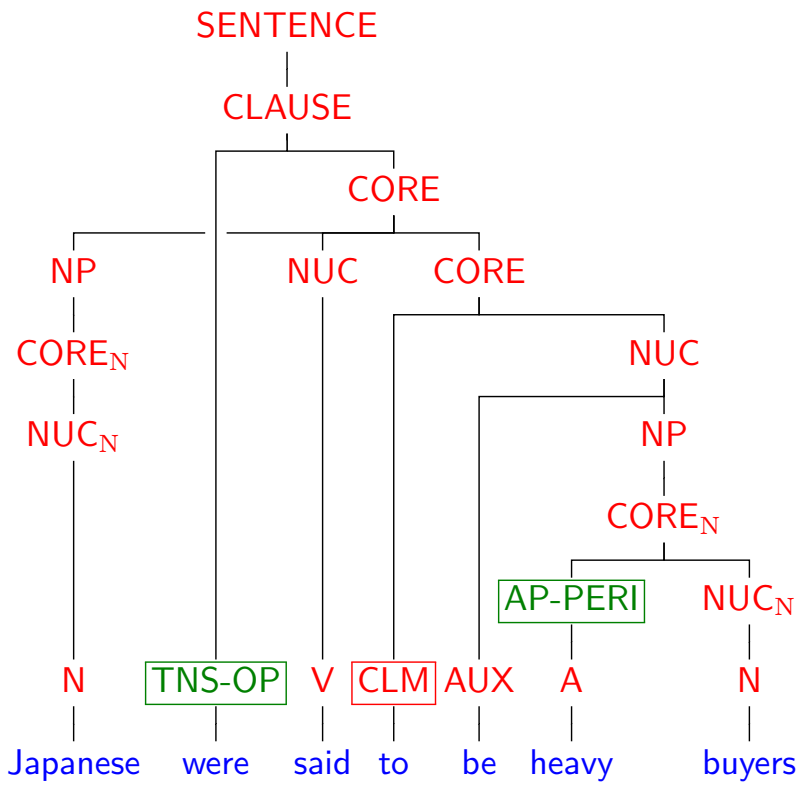
Evaluation

Conclusions and Future Work

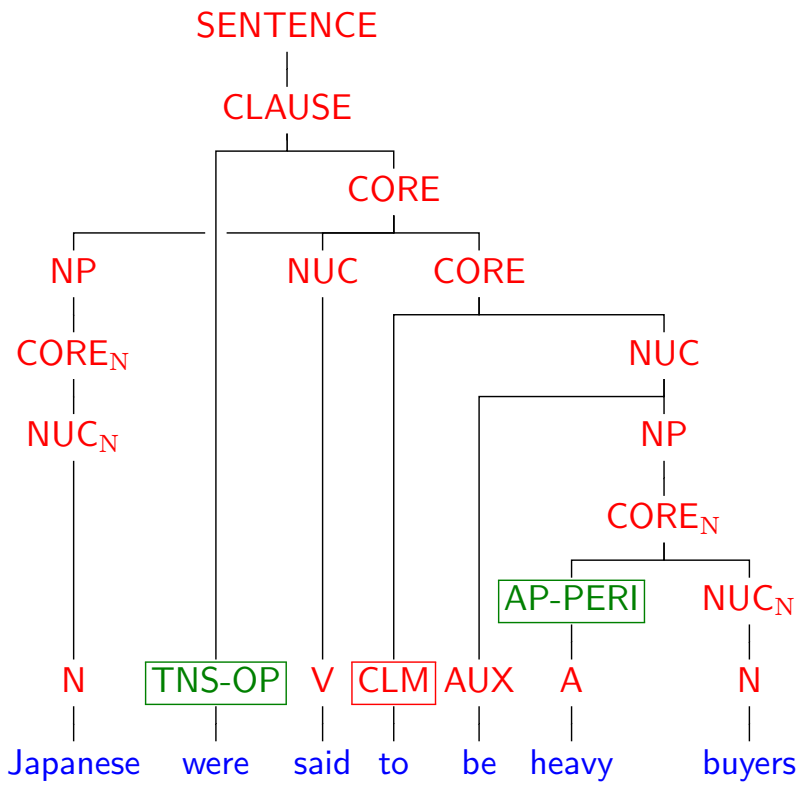
RRG: Textbook Notation



Our Single-Tree Notation

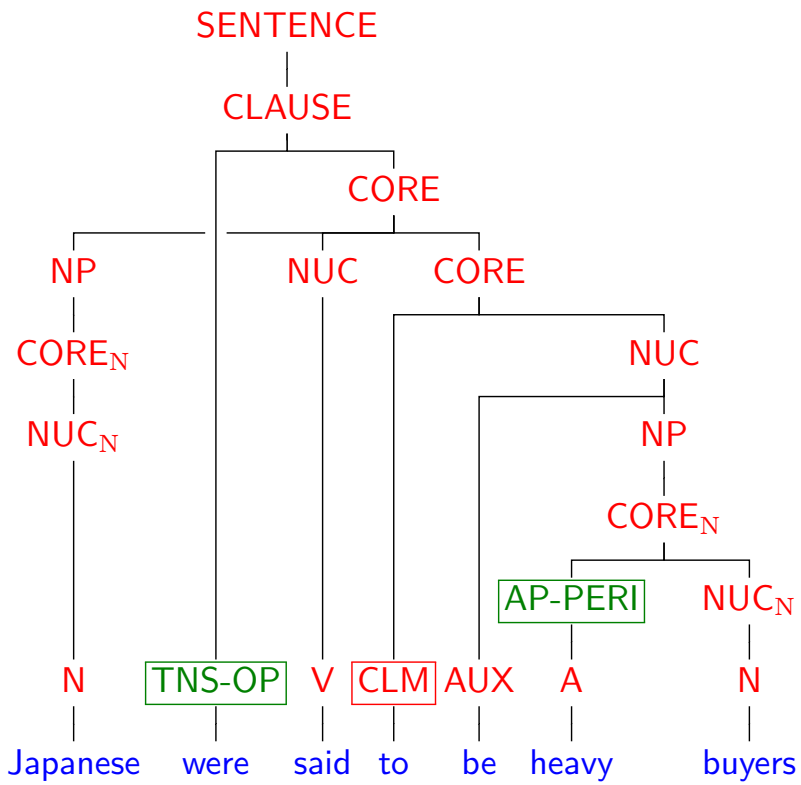


Our Single-Tree Notation



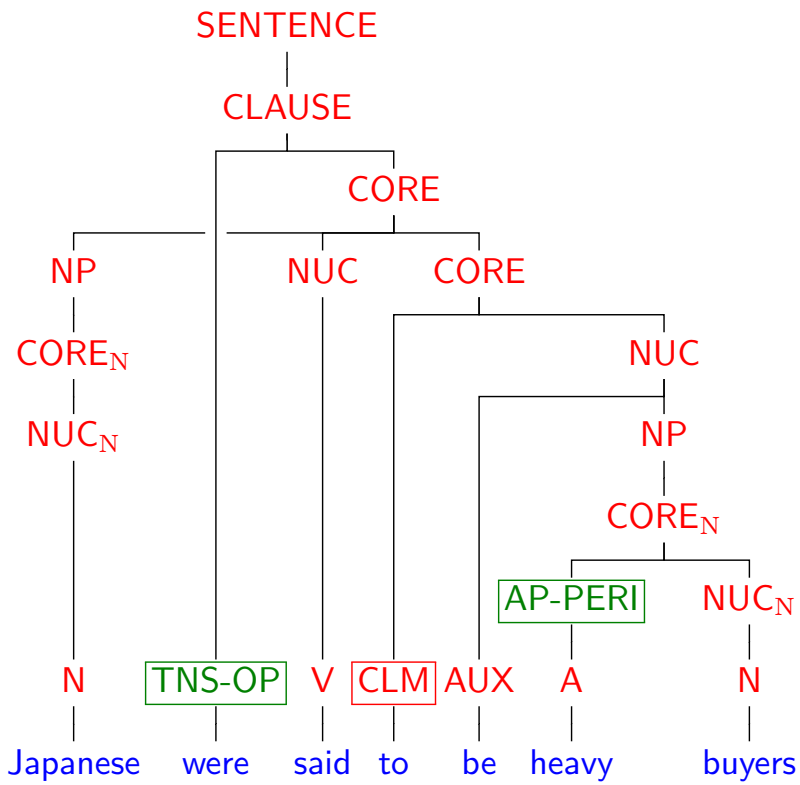
⋆ operators, peripheries and CLM are attached to constituent projection using normal edges,

Our Single-Tree Notation



- ⋆ operators, peripheries and CLM are attached to constituent projection using normal edges,
- ⋆ $\rightarrow$ crossing branches possible,

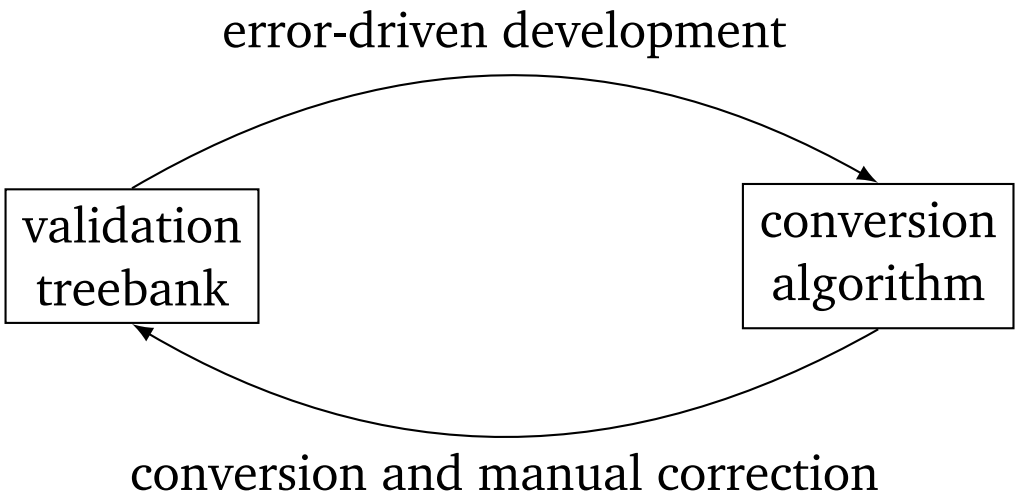
Our Single-Tree Notation



- ⋆ operators, peripheries and CLM are attached to constituent projection using normal edges,
- ⋆ → crossing branches possible,
- ⋆ operators/peripheries are marked by adding -OP/-PERI to label.

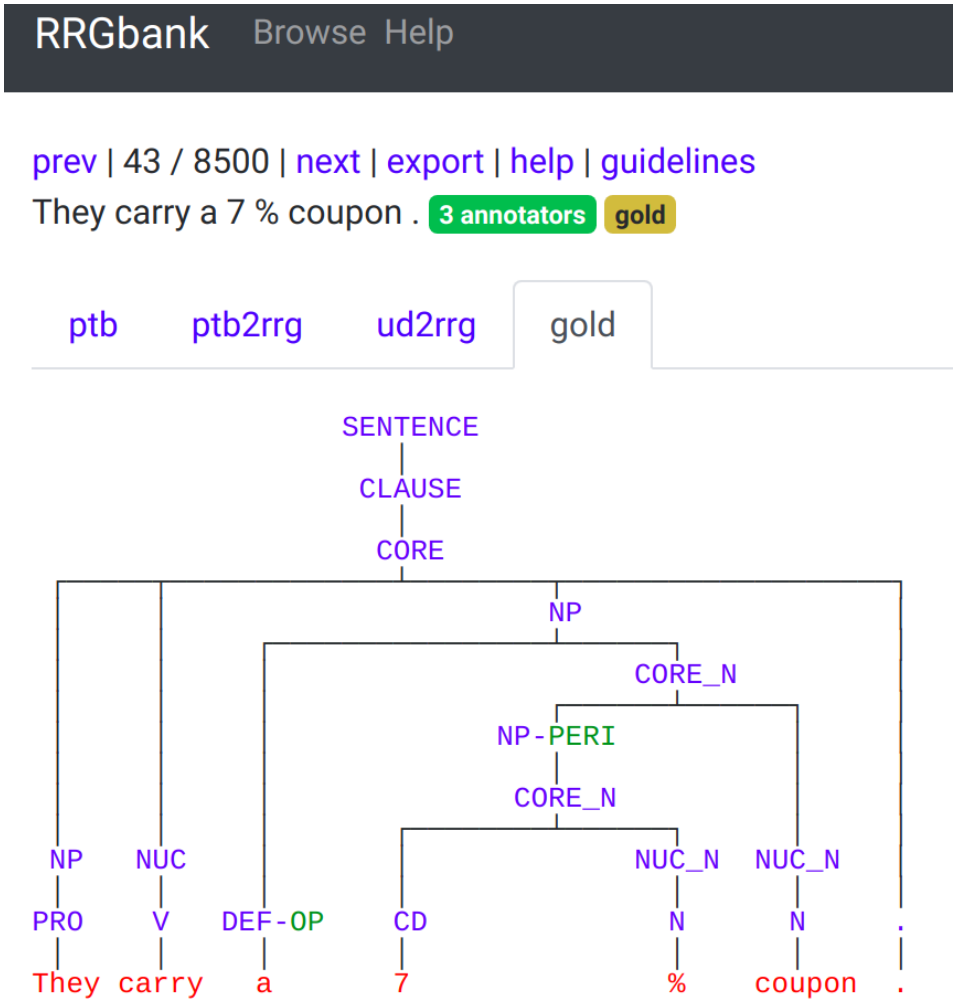
Creating a Validation Treebank

- ⋆ manually check and validate data,
- ⋆ automatic conversion script.



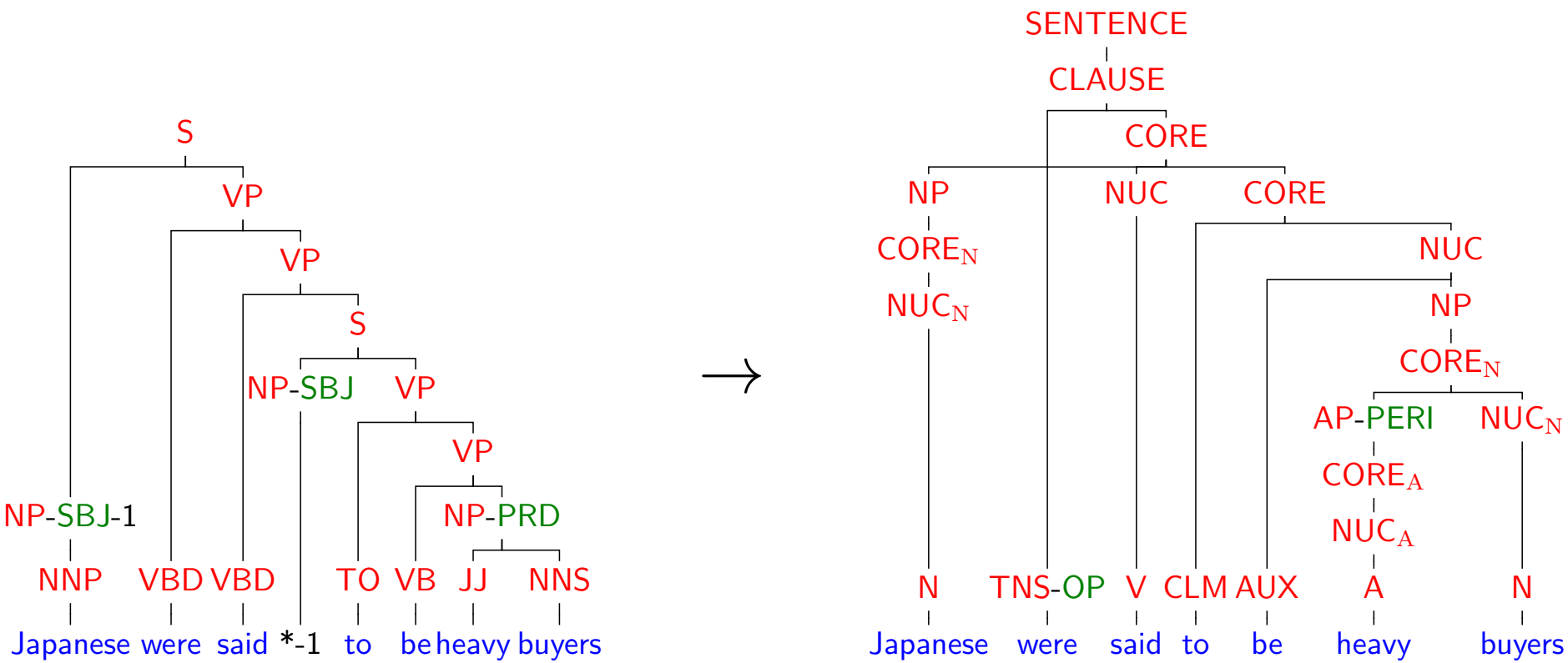
RRG annotation tool: `rrgbank.phil.hhu.de`

Validation Treebank



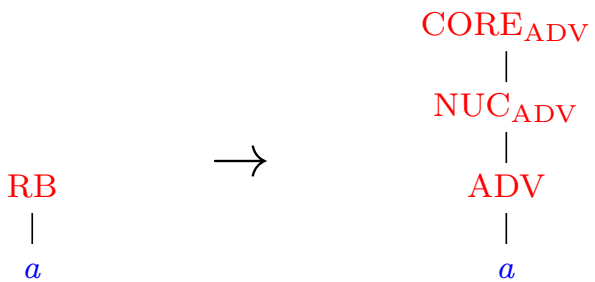
RRG annotation tool: `rrgbank.phil.hhu.de`

Automatic conversion from the Penn Treebank

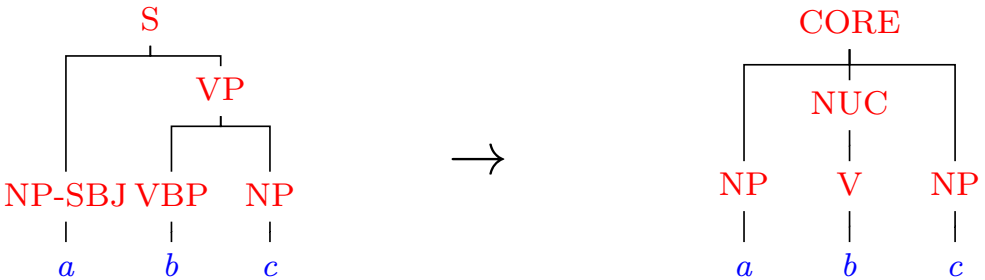


Tree in the PTB and the converted RRG tree

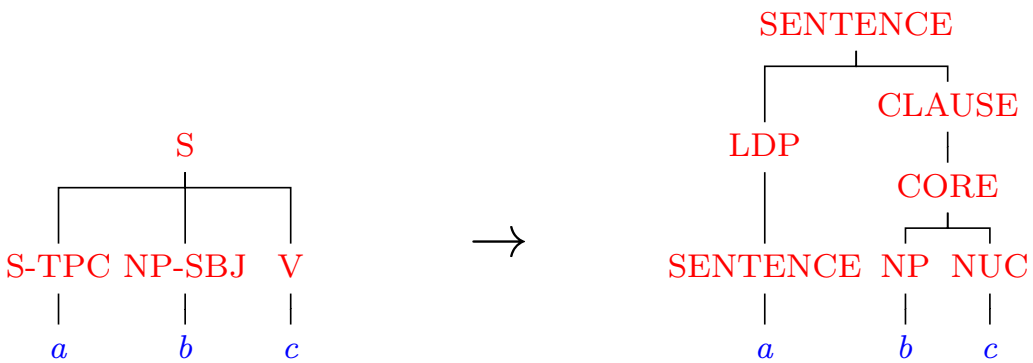
Example Conversion Rule 1/3: Adverb



Example Conversion Rule 2/3: Sentence



Example Conversion Rule 3/3: Topicalization



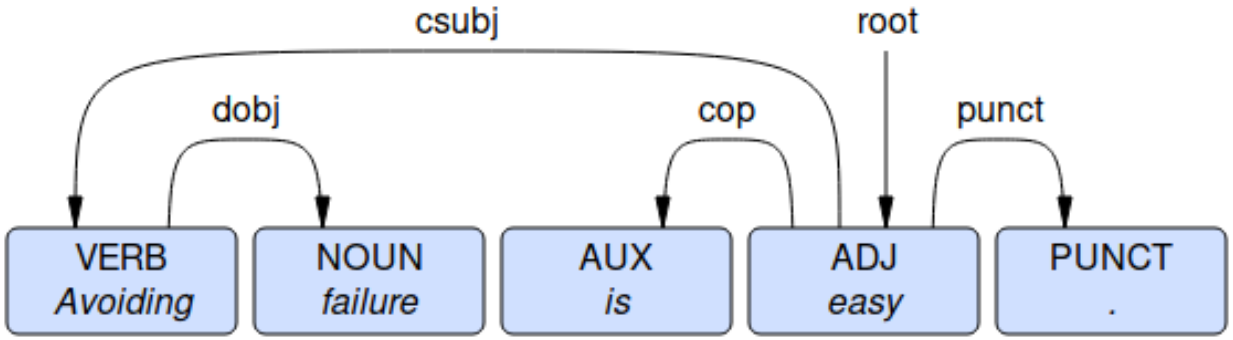
Statistics on RRGbank

- ⋆ 4 annotators,
- ⋆ 395 gold annotated sentences → validated and adjudicated by at least two annotators,
- ⋆ 1090 silver annotated sentences → validated by one annotator,
- ⋆ 8500 sentences $\leq$ 25 tokens,
- ⋆ accuracy: 95.98 (PTB2RRG) and 87.03 (UD2RRG).

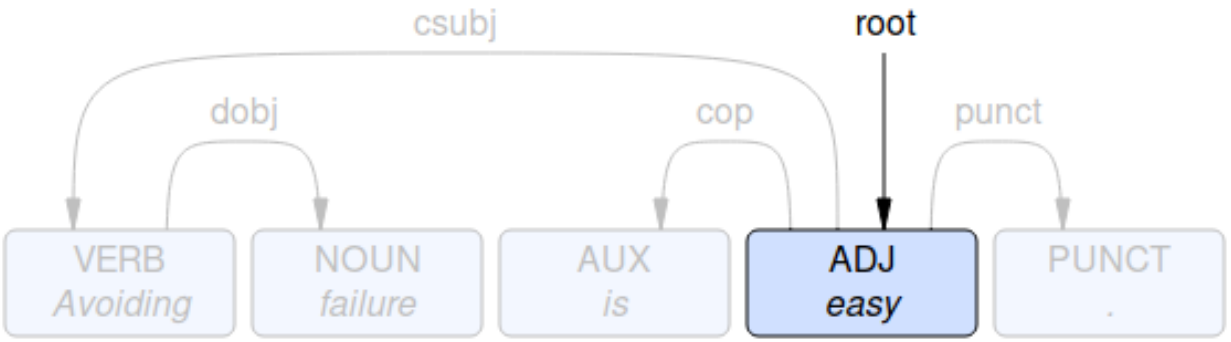
Universal Dependencies to RRG: Automatic Conversion

	`ptb2rrg`	`ud2rrg`
input trees	PTB	UD (converted from PTB with Stanford CoreNLP)
languages	1	83+
algorithm	rewrite rules	complete traversal
treebank-specific information	yes (PTB)	via extensions
accuracy (evalb F1)	**95.98**	**87.03**
coverage (short sent.)	100%	94.9%
converted gold sentences	395 (all)	375 (of 395)

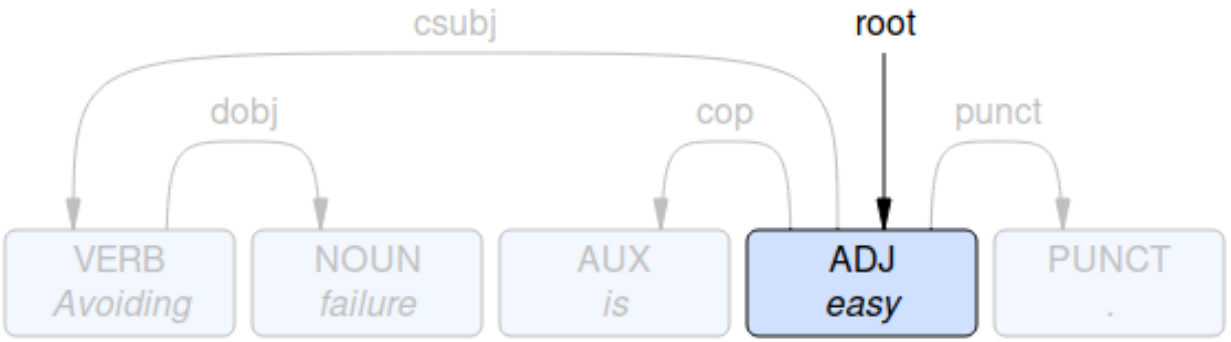
Example PTB-UD to RRG

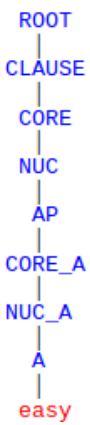


Example PTB-UD to RRG

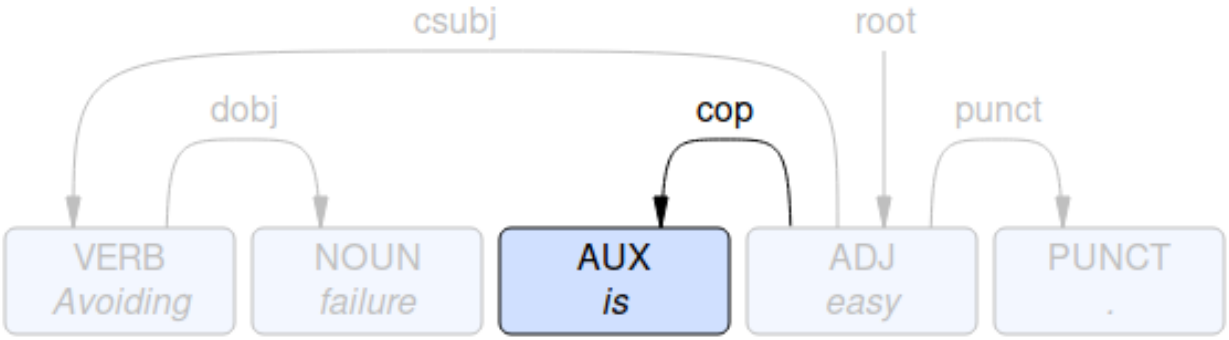


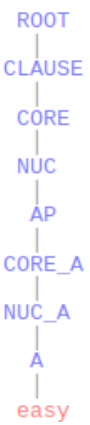
Example PTB-UD to RRG



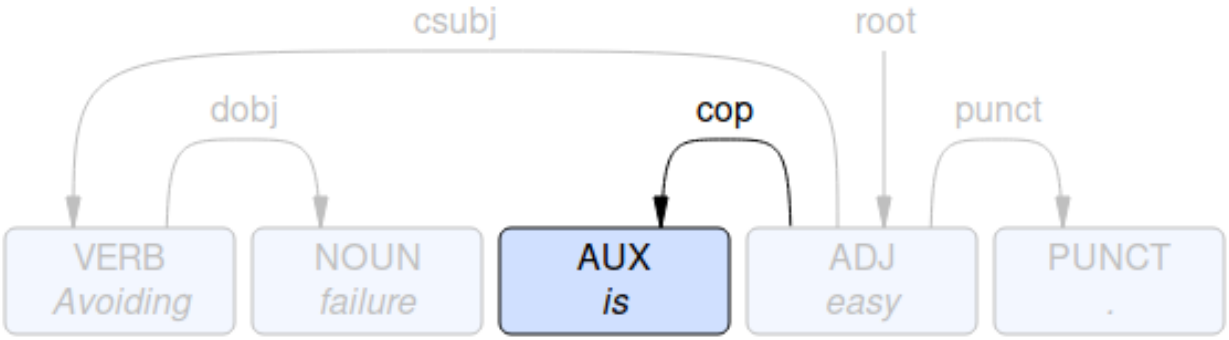


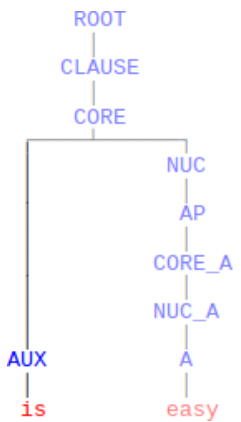
Example PTB-UD to RRG



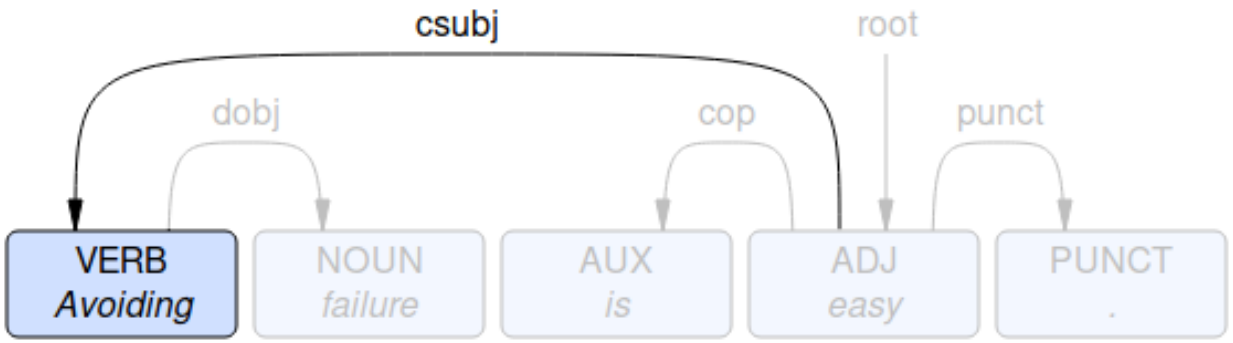


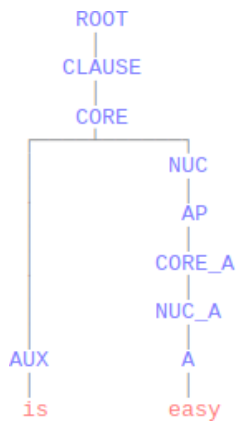
Example PTB-UD to RRG



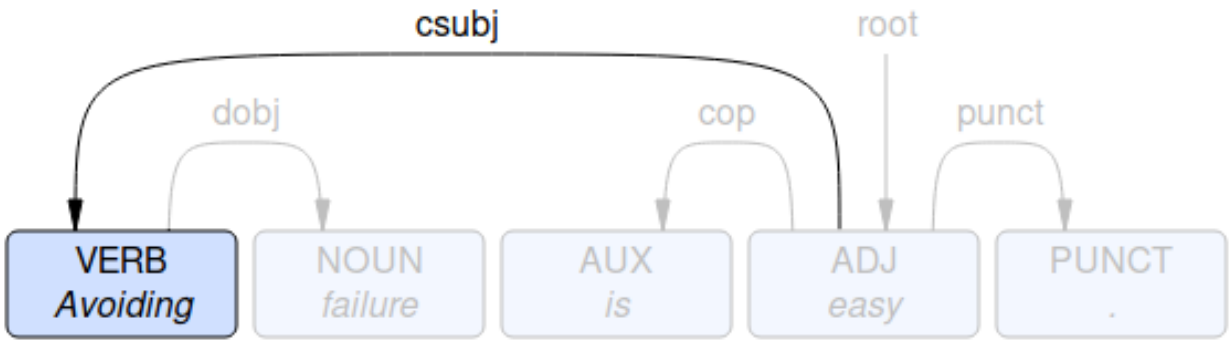


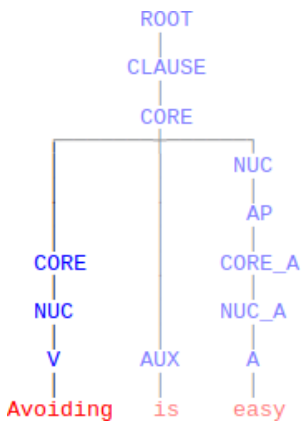
Example PTB-UD to RRG



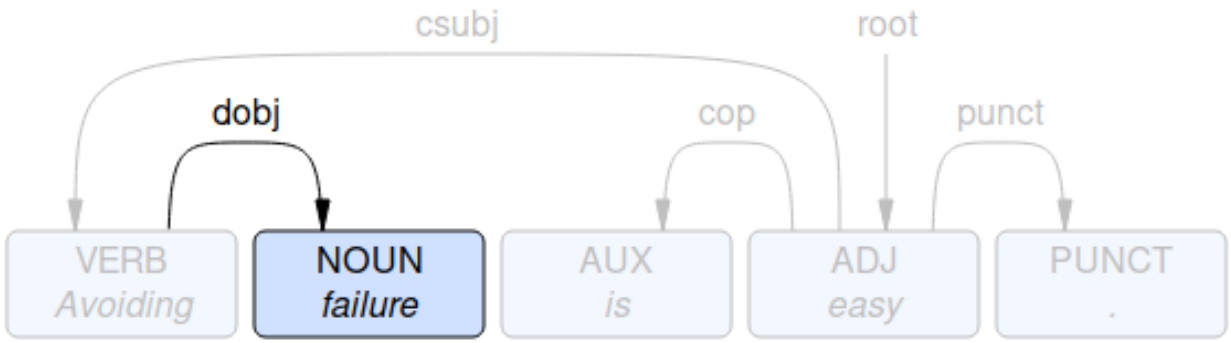


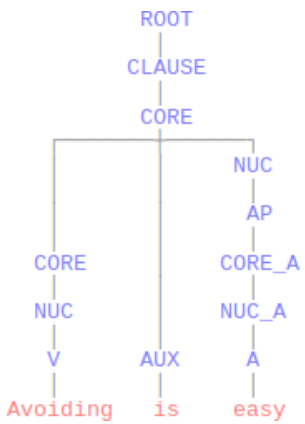
Example PTB-UD to RRG



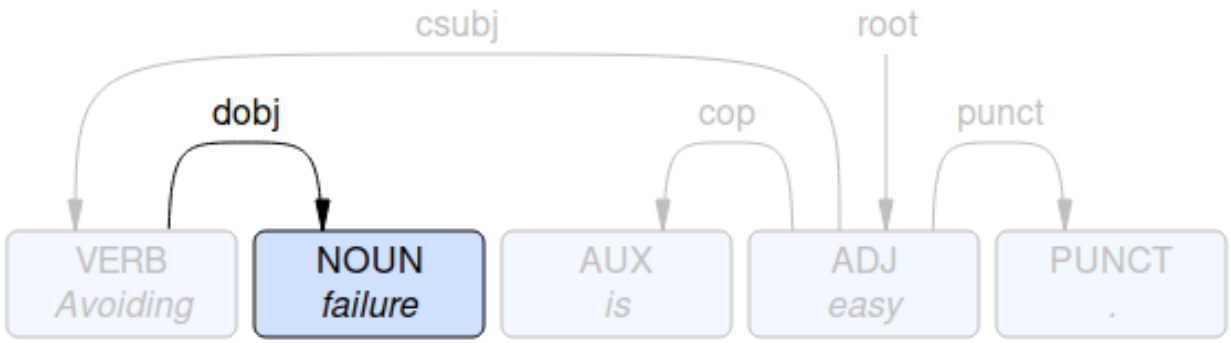


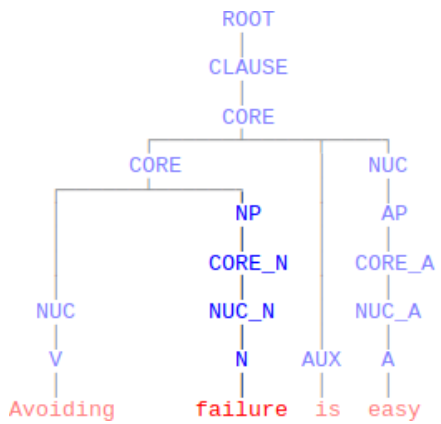
Example PTB-UD to RRG



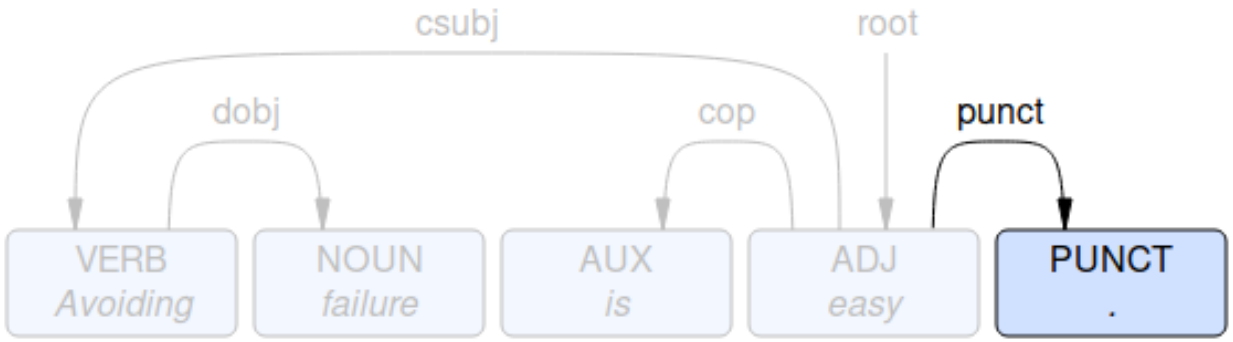


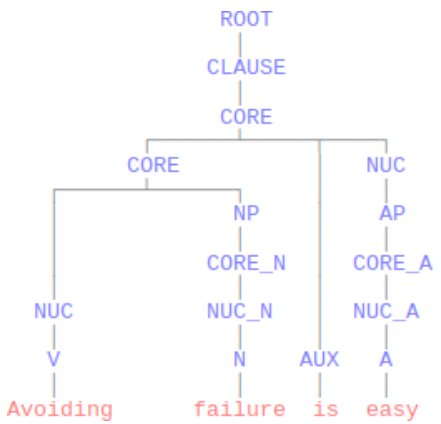
Example PTB-UD to RRG



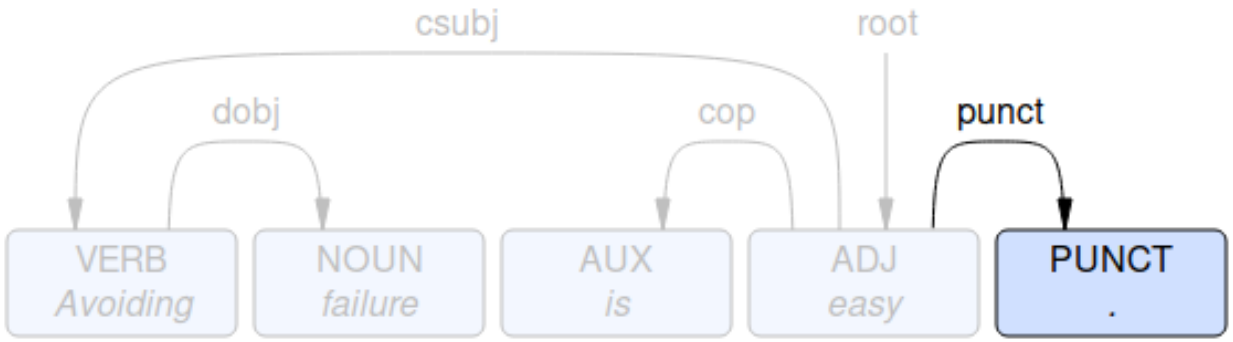


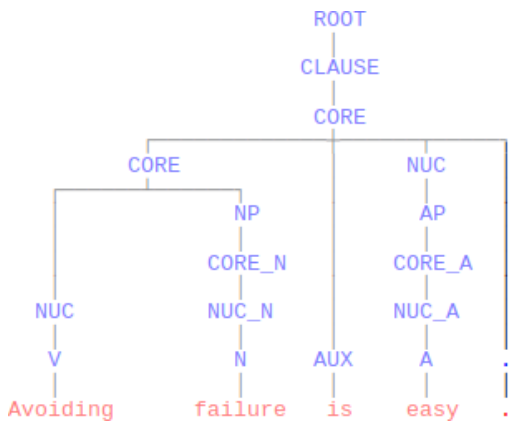
Example PTB-UD to RRG



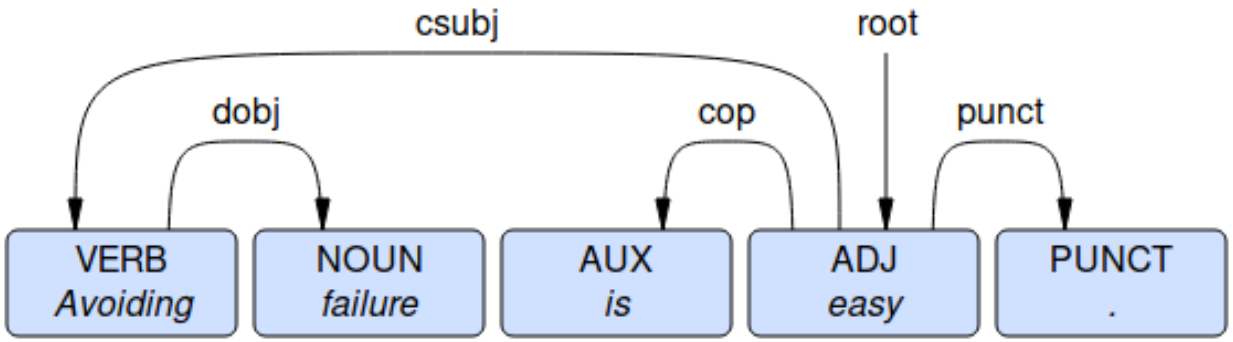


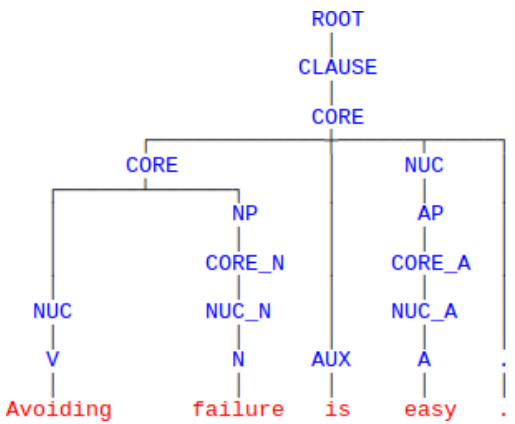
Example PTB-UD to RRG





Example PTB-UD to RRG





Outline

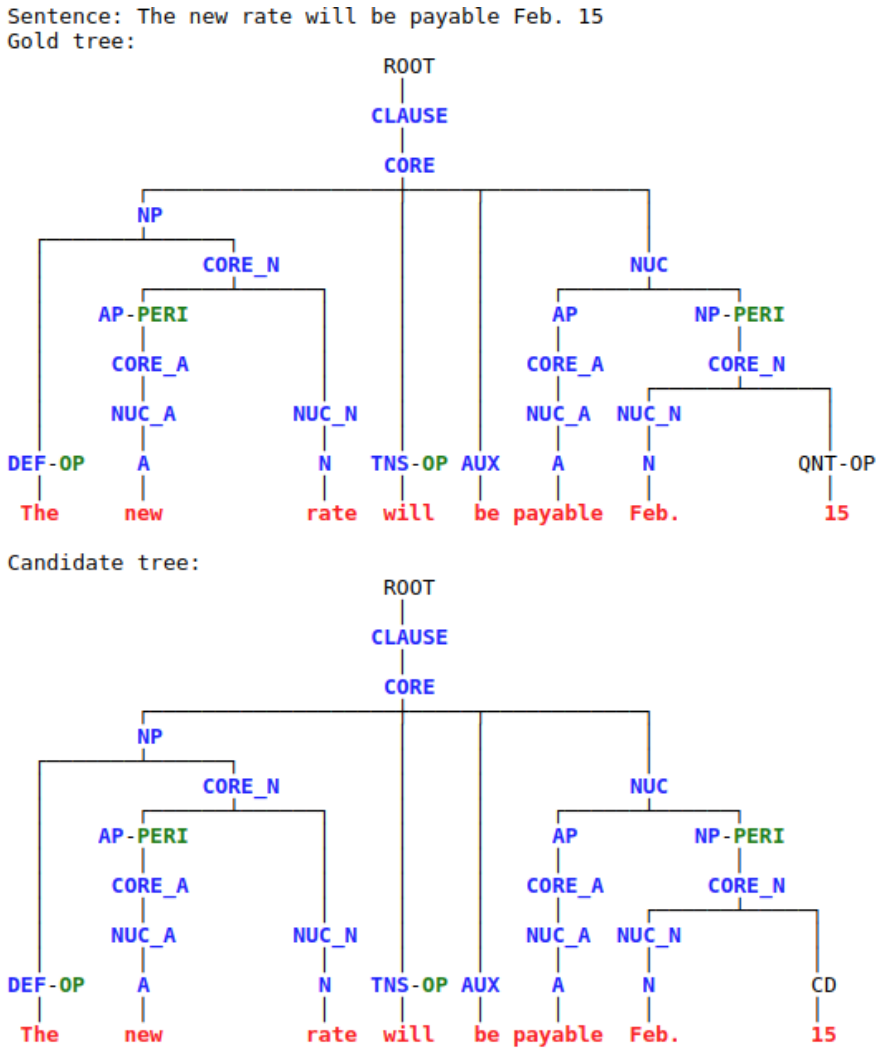
Motivation behind RRGbank

Creating RRGbank
- Design of the RRG structures
- Penn Treebank to RRG conversion
- Universal Dependencies to RRG conversion

Evaluation

Conclusions and Future Work

RRGbank: evaluation



- ⋆ Gold manually validated sentences = 395;
- ⋆ EVALB bracketing scores:
  - ⇒ matching spans,
  - ⇒ matching brackets,
  - ⇒ matching labels.
- ⋆ *zero common bracketings between the PTB and converted RRG trees → full coverage.

RRGbank: evaluation metrics PTB2RRG and UD2RRG

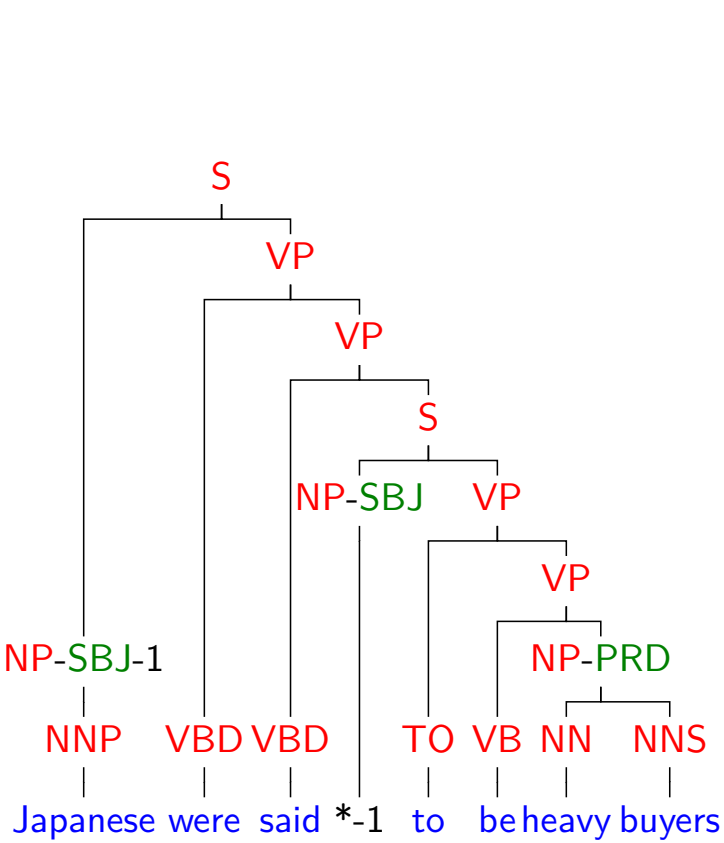
parameter	**ptb2rrg**	**ud2rrg**
number of converted sentences:	395	**375**
longest sentence:	24	24
gold brackets (discontinuous):	5096	5096
candidate brackets (discontinuous):	5076	4807
labeled recall:	95.88	84.54
labeled precision:	96.09	89.66
labeled f-measure:	**95.98**	**87.03**
exactly matched sentences:	66.84	44.30
function tags:	95.16	100.00
POS accuracy:	98.54	96.75

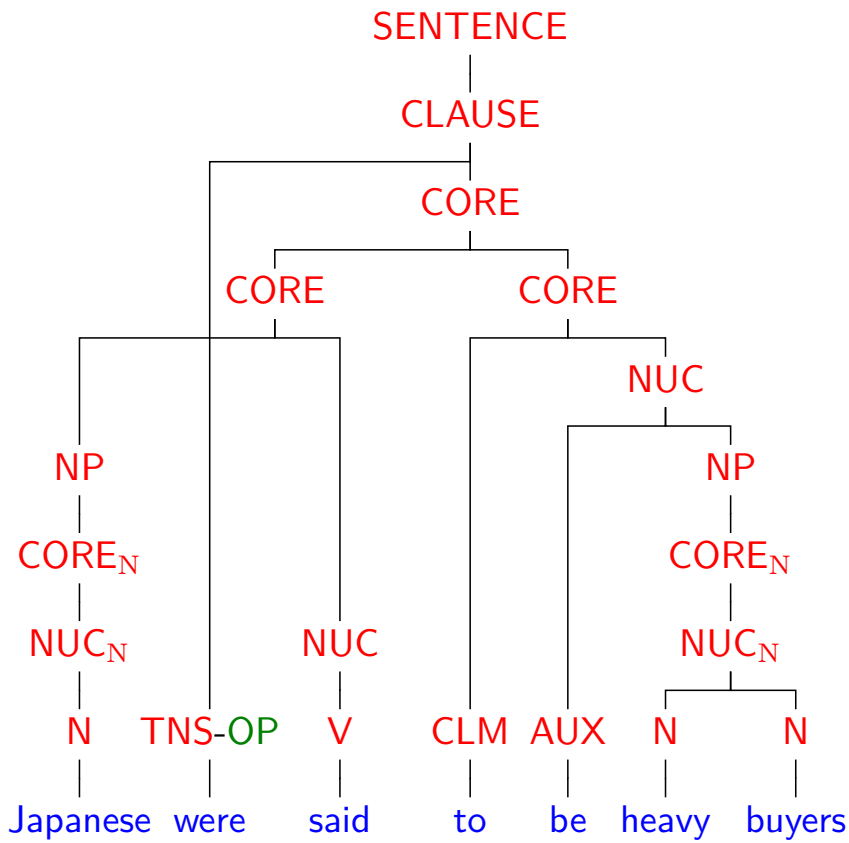
Encountered issues and problematic cases

Three types of problematic cases we encountered during conversion:

- Inconsistencies or errors in Penn Treebank
  $\rightarrow$ lead to inconsistencies in PTB-UD.
- Distinctions made in RRG but not in PTB
  $\rightarrow$ PTB-UD is even more affected.
- Analyses in PTB and PTB-UD which have no direct equivalent in RRG.

Example 1: PTB annotation inconsistencies





Erroneous annotation in PTB

Example 1: Annotation inconsistencies

Example 1: Annotation inconsistencies

- Lexical elements misanalyzed in PTB are manually corrected in the PTB input

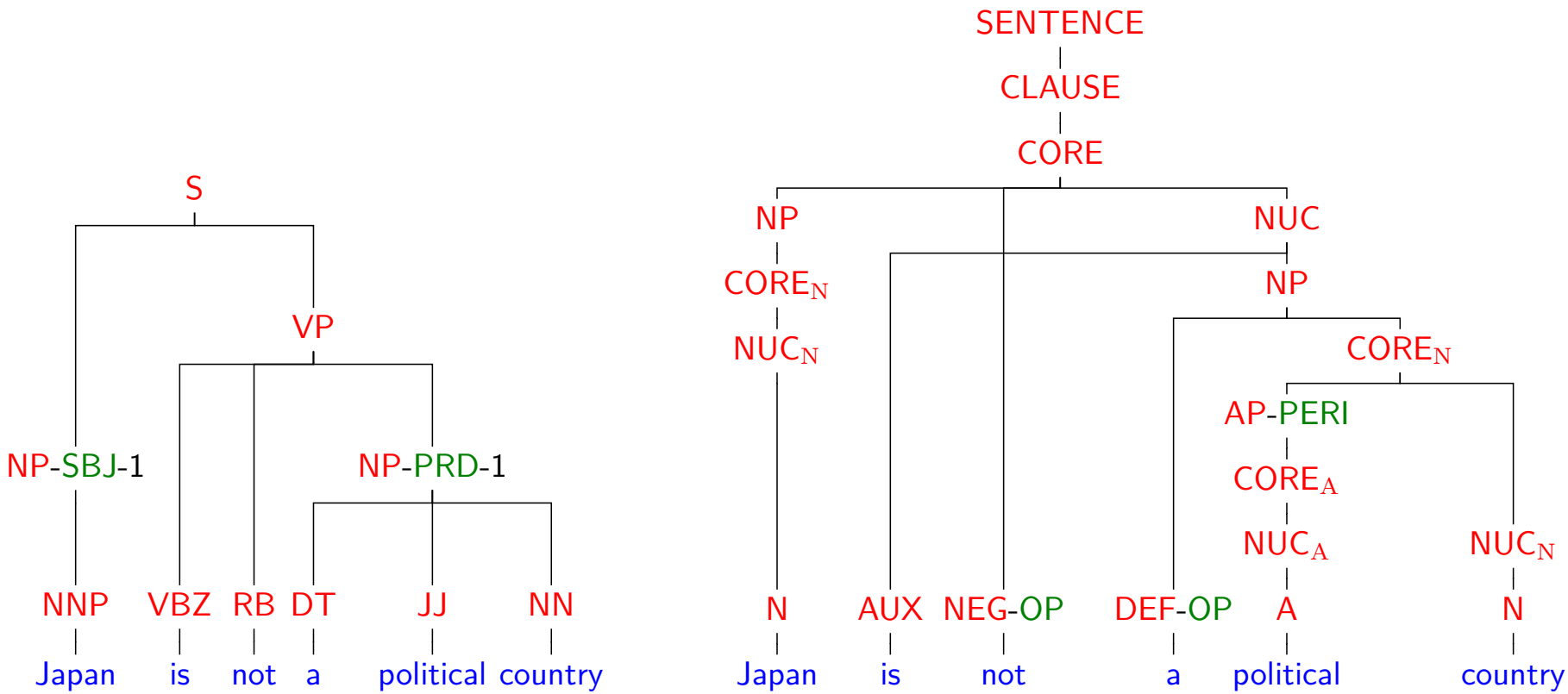
Example 1: Annotation inconsistencies

- Lexical elements misanalyzed in PTB are manually corrected in the PTB input
- However, not all cases are as clear as "heavy" in Example 1

Example 1: Annotation inconsistencies

- Lexical elements misanalyzed in PTB are manually corrected in the PTB input
- However, not all cases are as clear as "heavy" in Example 1
- Some NPs in PTB are not headed by a noun, which could either be an annotation error or a possible conversion

Example 2: Scope of negation



Negation in PTB and RRG

Example 2: Scope of negation

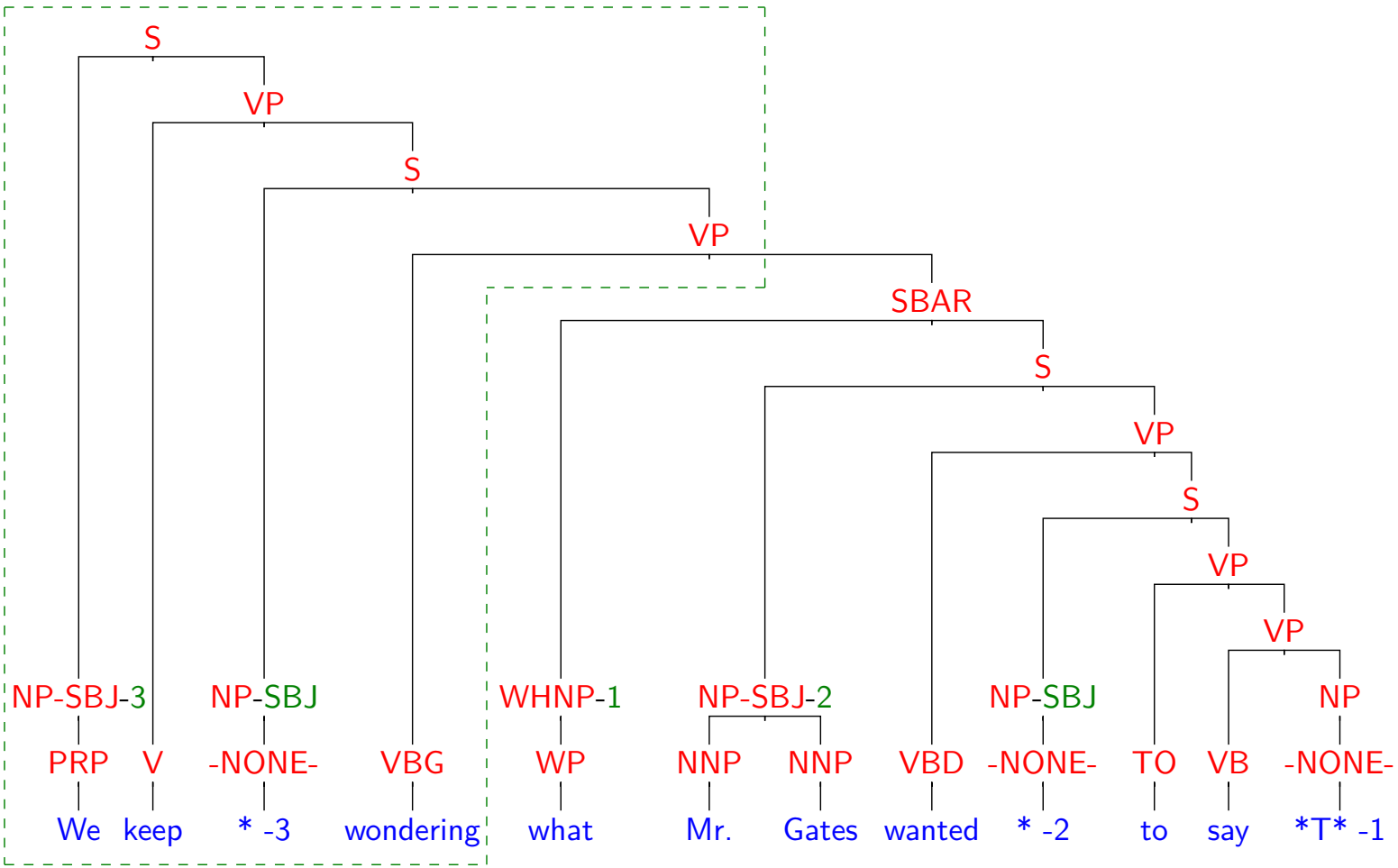
Example 2: Scope of negation

- PTB doesn't differentiate between internal and external negation

Example 2: Scope of negation

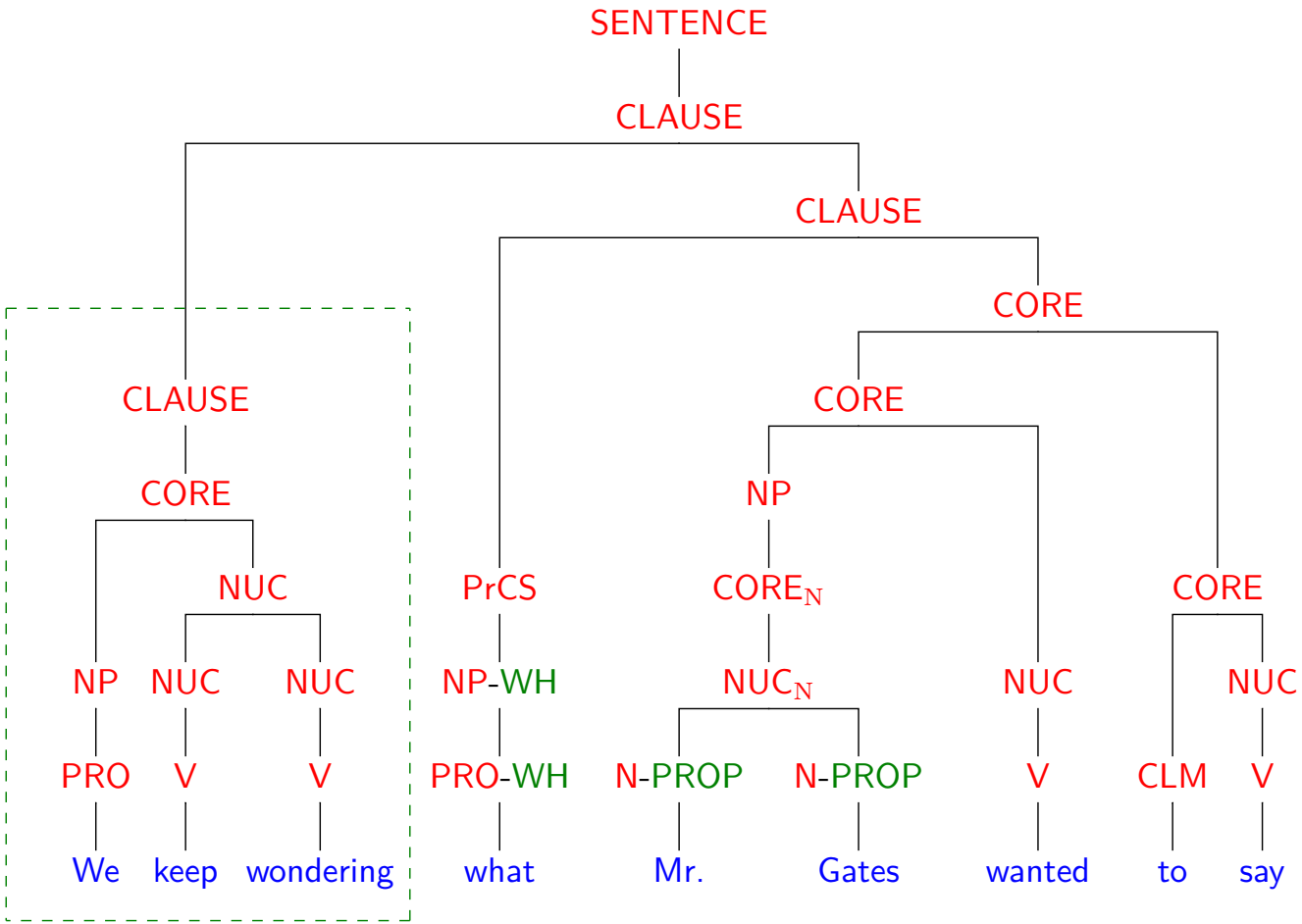
- PTB doesn't differentiate between internal and external negation
- As internal negation is more common, negation is treated as a core-operator

Example 3.1: Different junctures



PTB structure

Example 3.2: Different junctures



PTB-to-RRG structure

Example 3: Different junctures

Example 3: Different junctures

- PTB doesn't differentiate between cosubordination and subordination, barring conversion on a syntactic basis alone.

Example 3: Different junctures

- PTB doesn't differentiate between cosubordination and subordination, barring conversion on a syntactic basis alone.
- Using a lexical approach however, enables consistent conversion of some cosubordinations

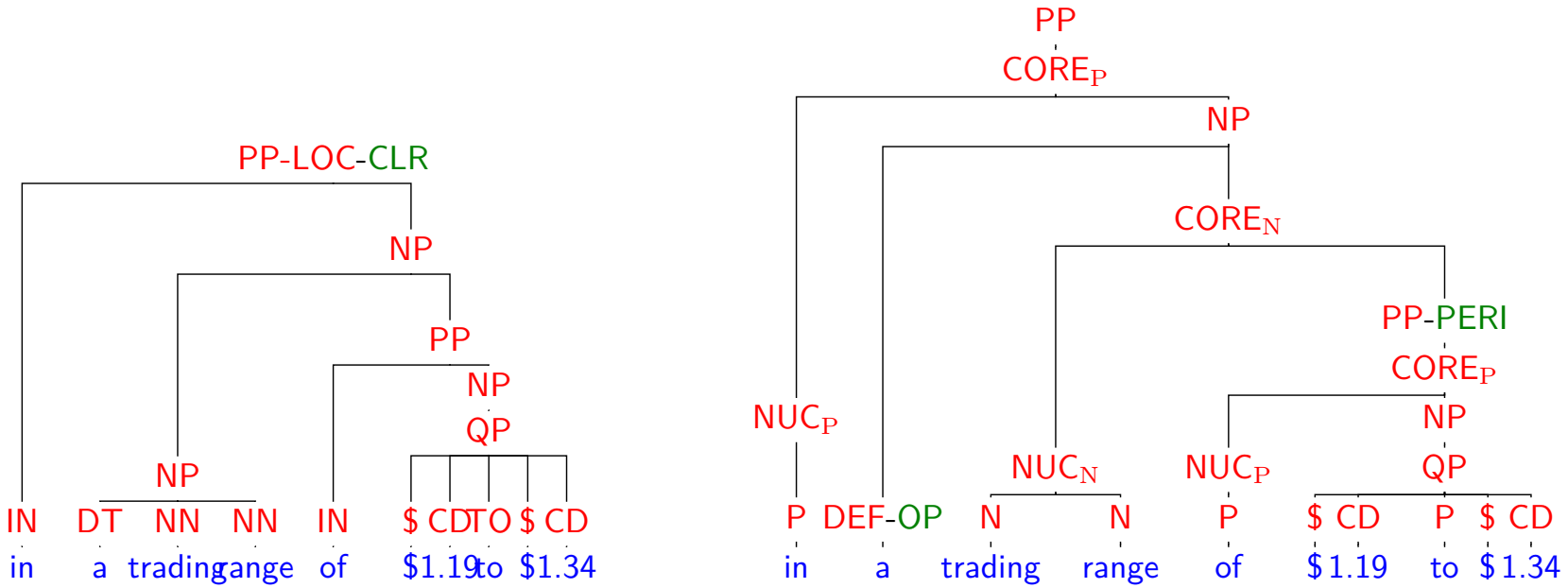
Example 3: Different junctures

- PTB doesn't differentiate between cosubordination and subordination, barring conversion on a syntactic basis alone.
- Using a lexical approach however, enables consistent conversion of some cosubordinations
- When cooccuring with gerunds, certain verbs (like start, keep, or finish) indicate a Phase relation and therefore nuclear cosubordination Van Valin Jr (2005)

Example 3: Different junctures

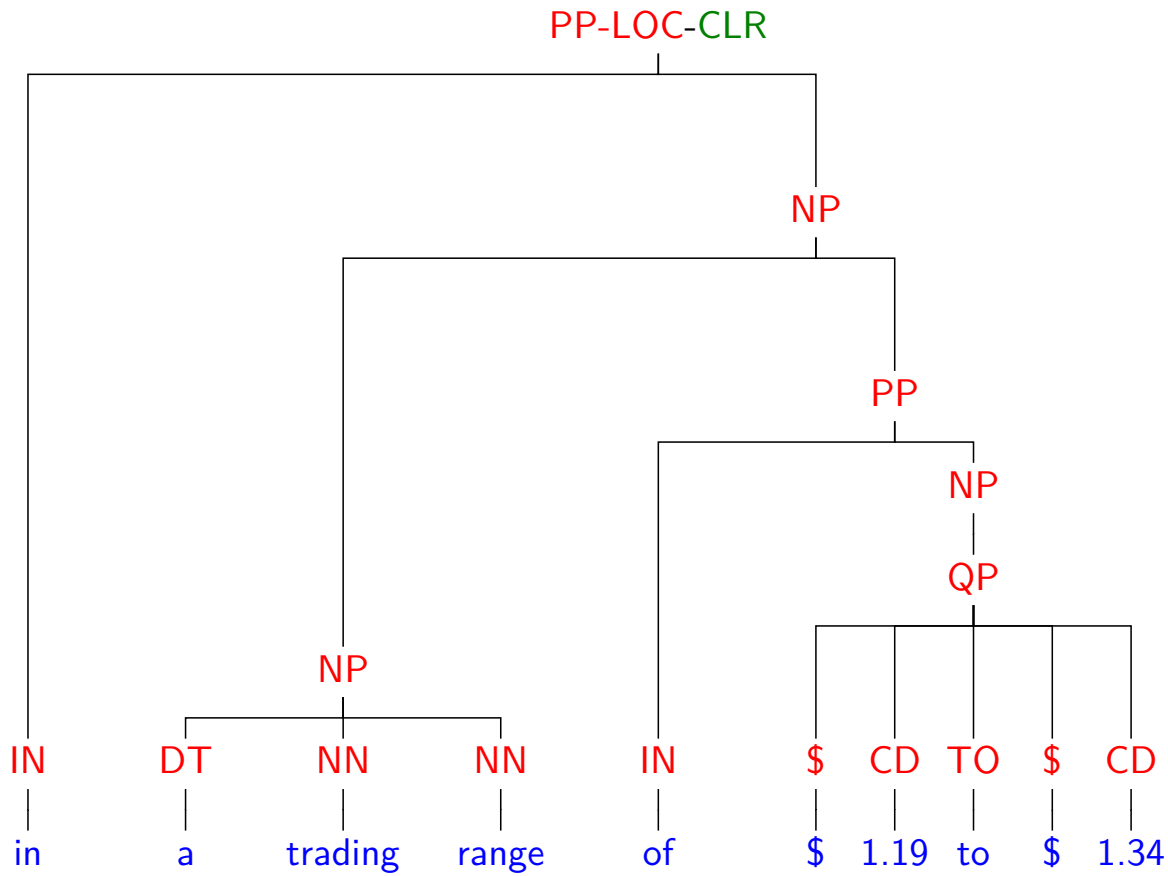
- PTB doesn't differentiate between cosubordination and subordination, barring conversion on a syntactic basis alone.
- Using a lexical approach however, enables consistent conversion of some cosubordinations
- When cooccuring with gerunds, certain verbs (like start, keep, or finish) indicate a Phase relation and therefore nuclear cosubordination Van Valin Jr (2005)
- If necessary, the traces contained in PTB can be used to further restrict the conversion context to avoid false positives

Example 4: Quantifier Phrases



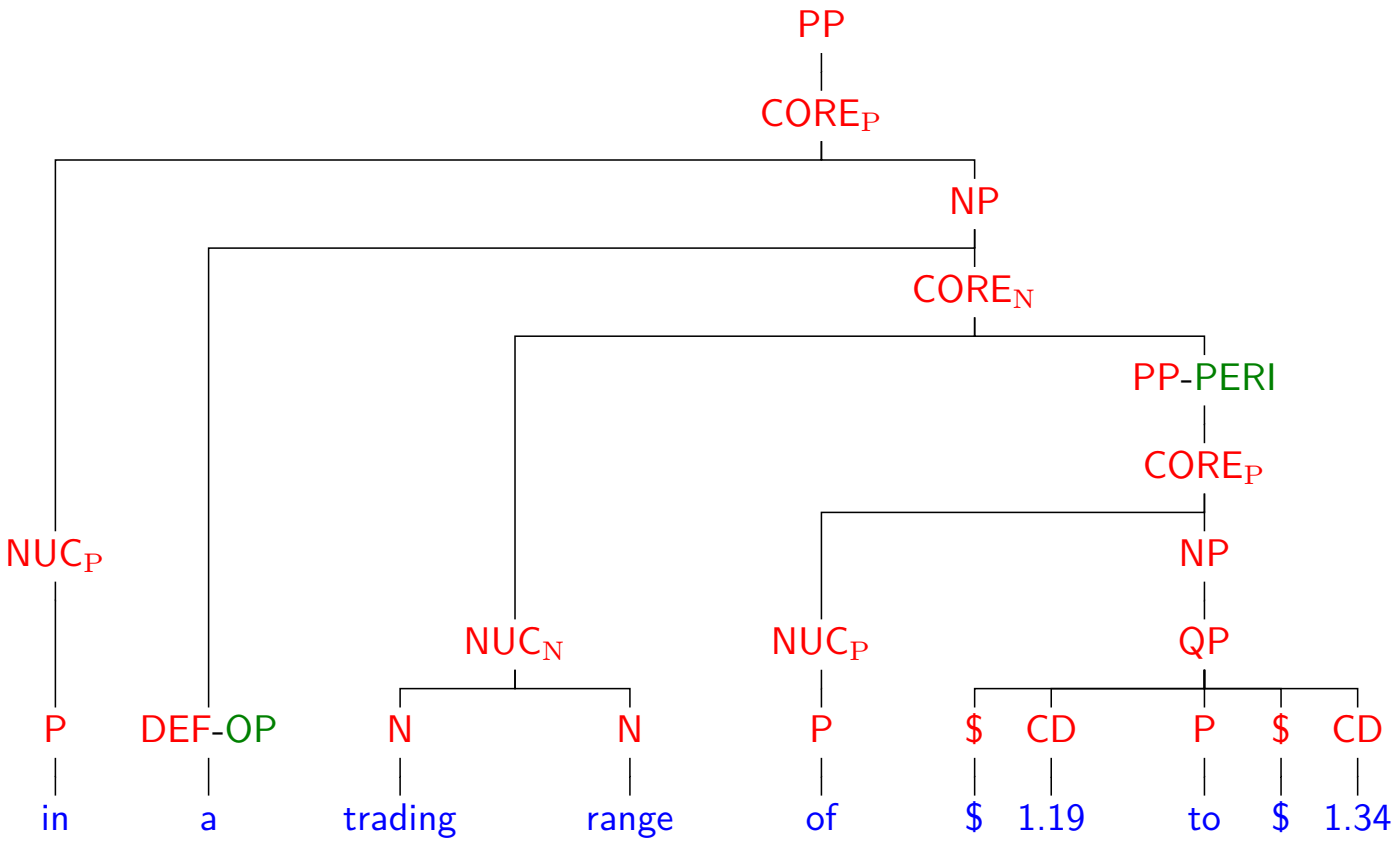
QPs and nounless NPs

Example 4.1: Quantifier Phrases



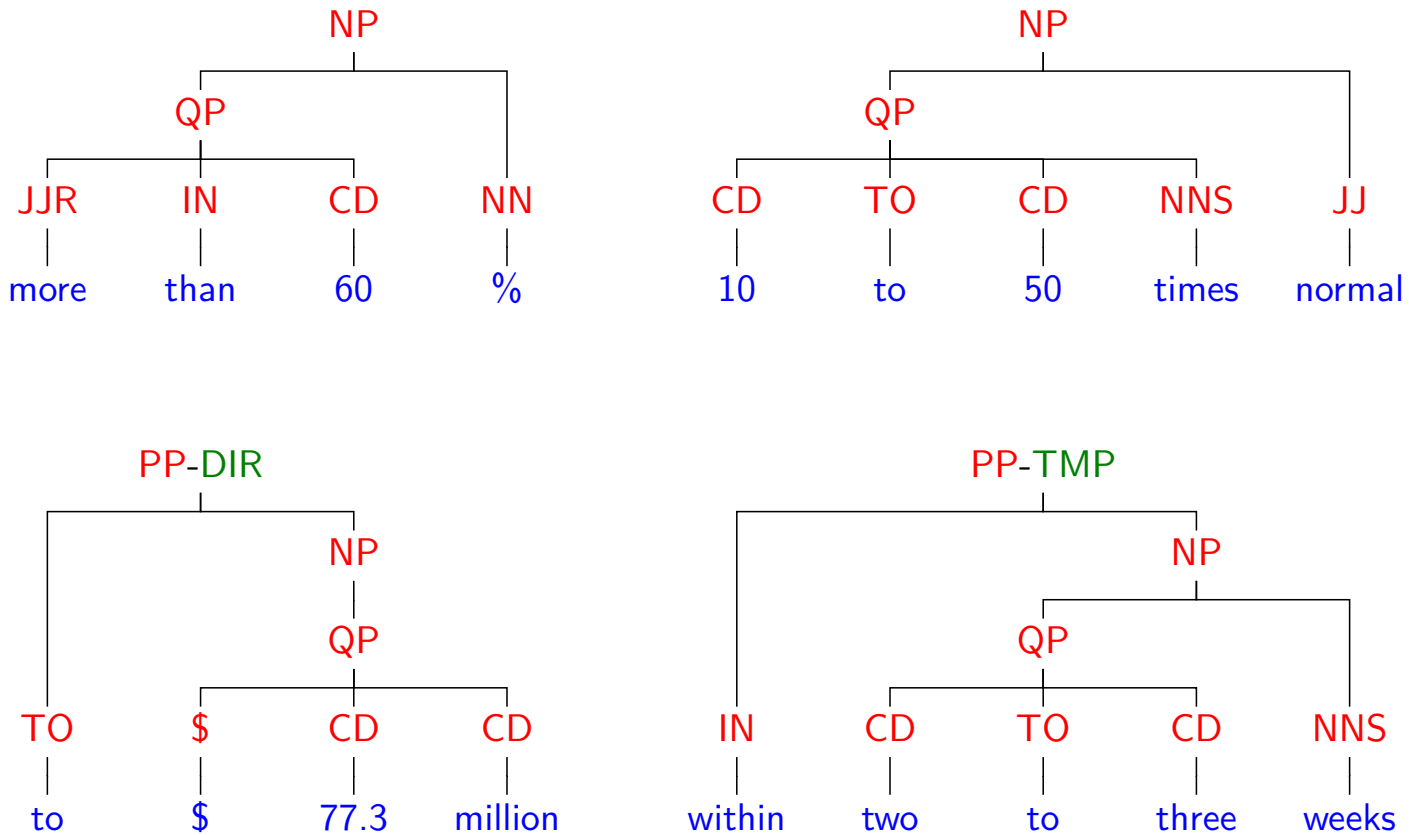
NP headed by a QP in PTB

Example 4.2: Quantifier Phrases



NP headed by a QP retained in RRG

Examples 4.3 - 4.6 Quantifier Phrases



A variety of different QPs

Example 4: Quantifier Phrases

Example 4: Quantifier Phrases

- QPs are inconsistent with regard to both the lexical category and internal position of their heads.

Example 4: Quantifier Phrases

- QPs are inconsistent with regard to both the lexical category and internal position of their heads.
- QPs can function as the only constituent within an NP leaving the NP without a lexical head.

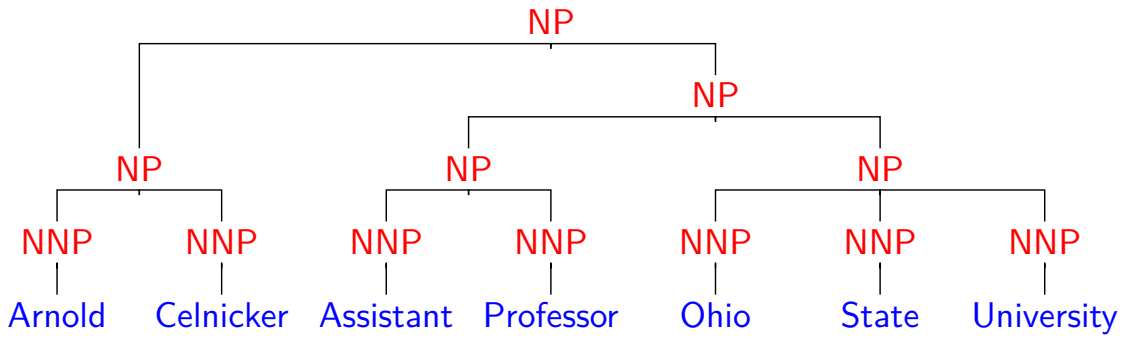
Example 4: Quantifier Phrases

- QPs are inconsistent with regard to both the lexical category and internal position of their heads.
- QPs can function as the only constituent within an NP leaving the NP without a lexical head.
- These properties of QPs have proven highly problematic for automated conversion.

Example 4: Quantifier Phrases

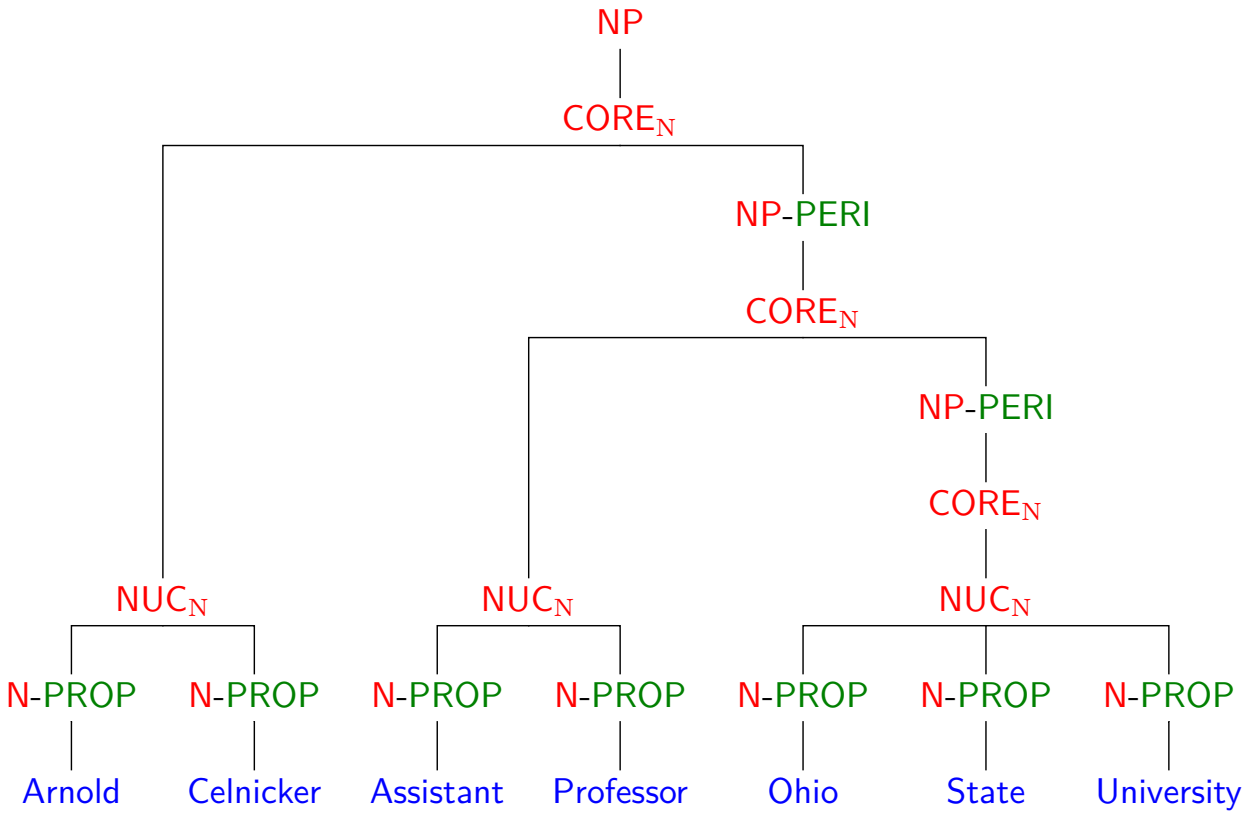
- QPs are inconsistent with regard to both the lexical category and internal position of their heads.
- QPs can function as the only constituent within an NP leaving the NP without a lexical head.
- These properties of QPs have proven highly problematic for automated conversion.
- Until a consistently correct conversion can be achieved, QPs will be retained.

Example 5.1: Complex proper NPs



Complex NP with multiple proper nouns

Example 5.2: Complex proper NPs



Complex NP with multiple proper nouns

Outline

Motivation behind RRGbank

Creating RRGbank
- Design of the RRG structures
- Penn Treebank to RRG conversion
- Universal Dependencies to RRG conversion

Evaluation

Conclusions and Future Work

Conclusions and Future Work

- Further conversion of PTB trees and validation of RRG trees.
- Further formalization of the RRG (Kallmeyer et al., 2013).
- Extraction of fragments ("supertags") of the formalized RRG grammar.
- Statistical parsing with these fragments.

Thank you!

THANK YOU VERY MUCH FOR YOUR ATTENTION!

References I

Kallmeyer, L. (2016). On the mild context-sensitivity of k-Tree Wrapping Grammar. In Foret, A., Morrill, G., Muskens, R., Osswald, R., and Pogodalla, S., editors, *Formal Grammar: 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016, Proceedings*, number 9804 in Lecture Notes in Computer Science, pages 77–93, Berlin. Springer.

Kallmeyer, L. and Osswald, R. (2017). Combining Predicate-Argument Structure and Operator Projection: Clause Structure in Role and Reference Grammar. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 61–70, Umeå, Sweden. Association for Computational Linguistics.

Kallmeyer, L., Osswald, R., and Van Valin, R. D. (2013). Tree wrapping for role and reference grammar. In *Formal Grammar*, pages 175–190. Springer.

Van Valin Jr, R. D. (2005). *Exploring the syntax-semantics interface*. Cambridge University Press.

RRGbank: evaluation metrics PTB2RRG and UD2RRG (2)

Category Statistics
(10 most frequent categories / errors)

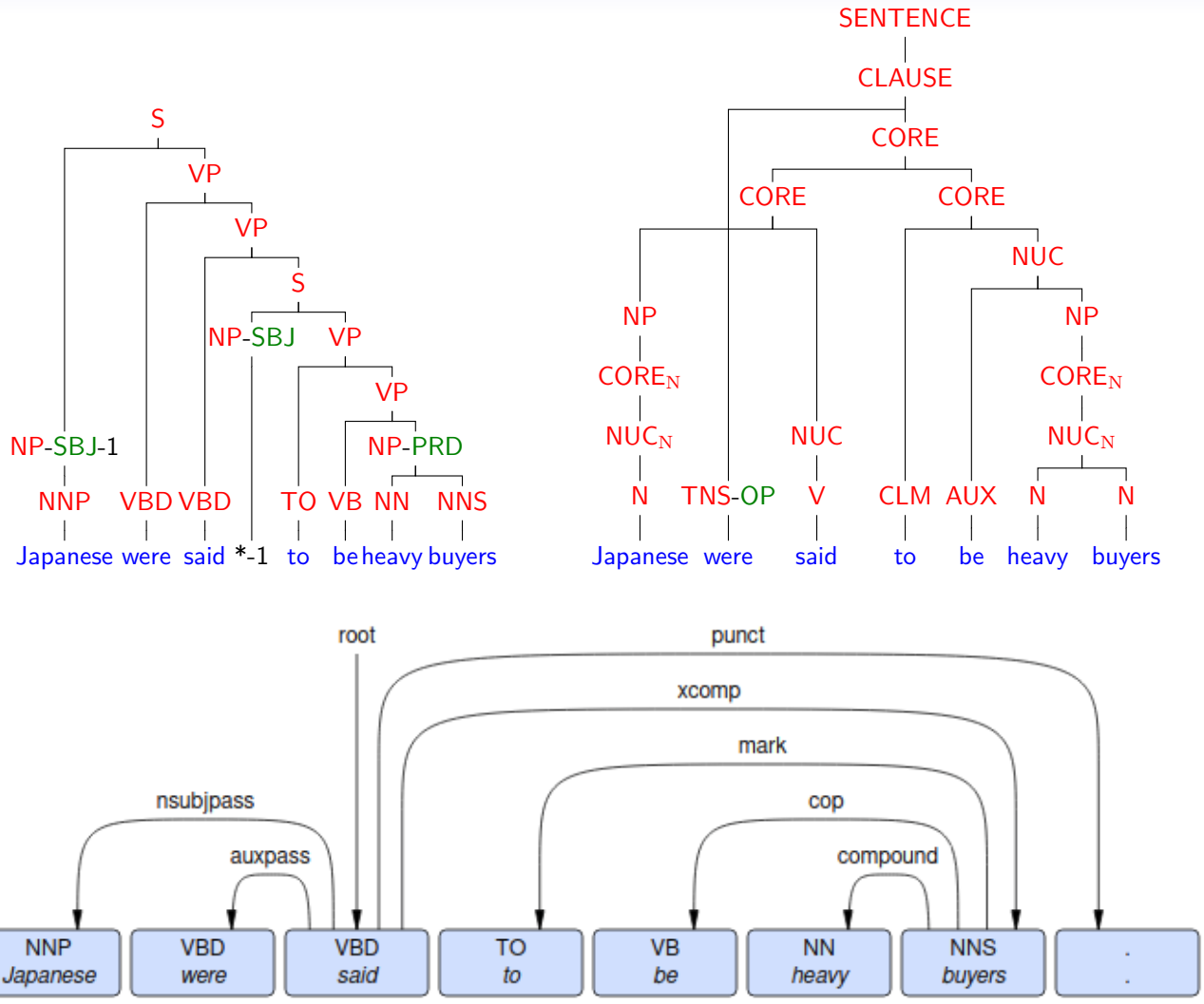
		ptb2rrg			ud2rrg		
label	**% gold**	**recall**	**prec.**	**F1**	**recall**	**prec.**	**F1**
NP	15.89	97.65	99.00	98.32	87.27	95.68	91.28
CORE_N	12.99	96.37	96.67	96.52	89.88	95.05	92.39
NUC_N	12.76	98.62	98.31	98.46	92.62	96.47	94.51
CORE	8.28	88.39	87.15	**87.76**	71.84	72.53	72.18
NUC	8.18	88.49	89.56	89.02	82.73	81.75	82.24
CLAUSE	7.46	99.21	99.21	**99.21**	85.41	88.71	87.03
SENTENCE	6.87	99.71	97.76	98.73	94.59	99.40	96.93
AP	3.71	91.53	91.53	91.53	–	–	–
NUC_A	3.69	98.40	97.37	97.88	93.09	99.43	96.15
CORE_A	3.69	91.49	90.53	91.01	86.70	92.61	89.56

RRGbank: evaluation metrics PTB2RRG and UD2RRG (3)

Function Tag Statistics
(most frequent tags / errors)

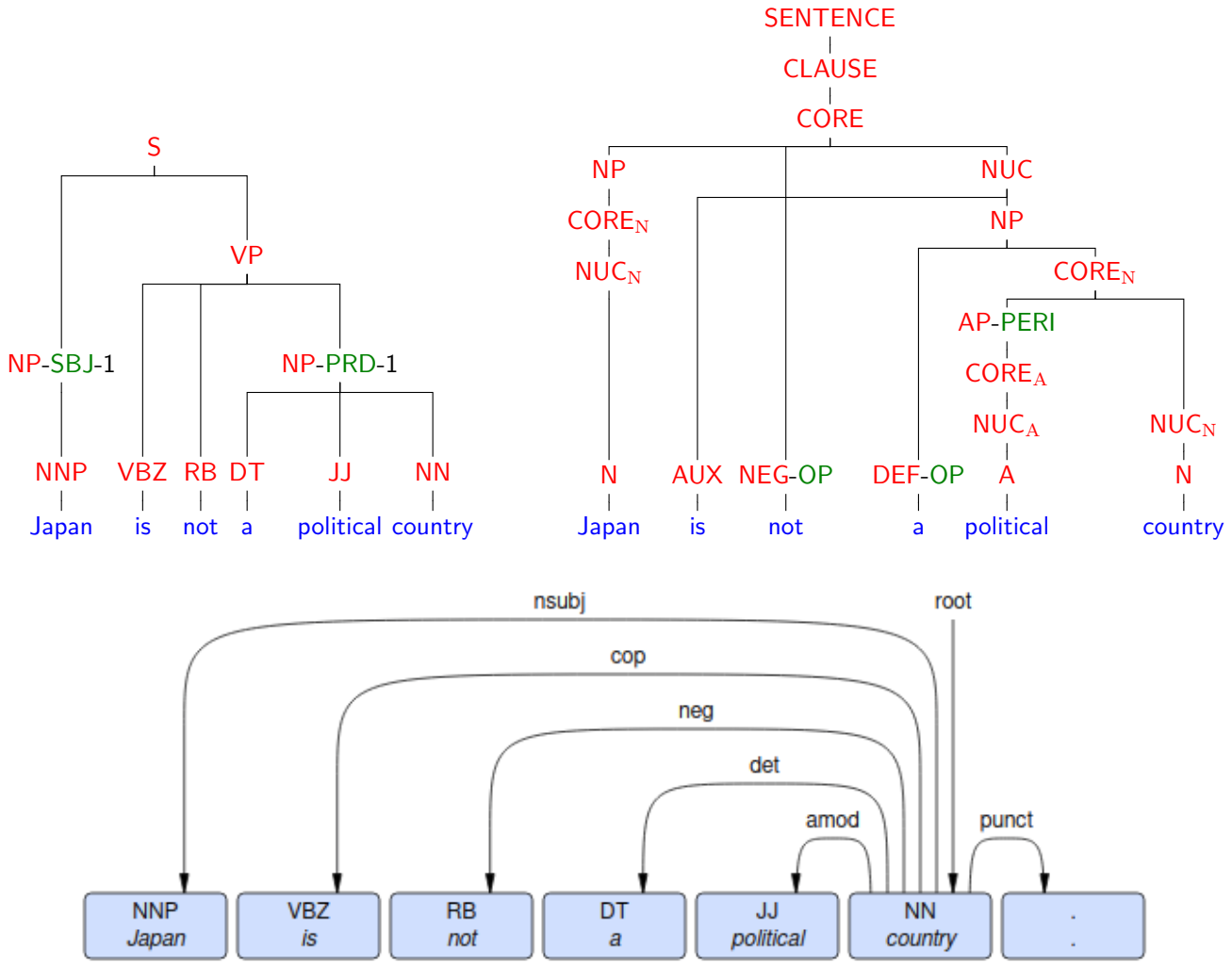
		ptb2rrg			ud2rrg		
func. tag	**% gold**	**recall**	**prec.**	**F1**	**recall**	**prec.**	**F1**
OP	39.12	100.00	100.00	**100.00**	100.00	100.00	100.00
PROP	27.16	99.28	100.00	99.64	100.00	100.00	100.00
PERI	26.96	71.64	96.57	**82.25**	100.00	100.00	100.00
WH	4.12	100.00	95.45	97.67	100.00	100.00	100.00
DEM	1.96	100.00	100.00	100.00	100.00	100.00	100.00

PTB2RRG and UD2RRG: Example 1



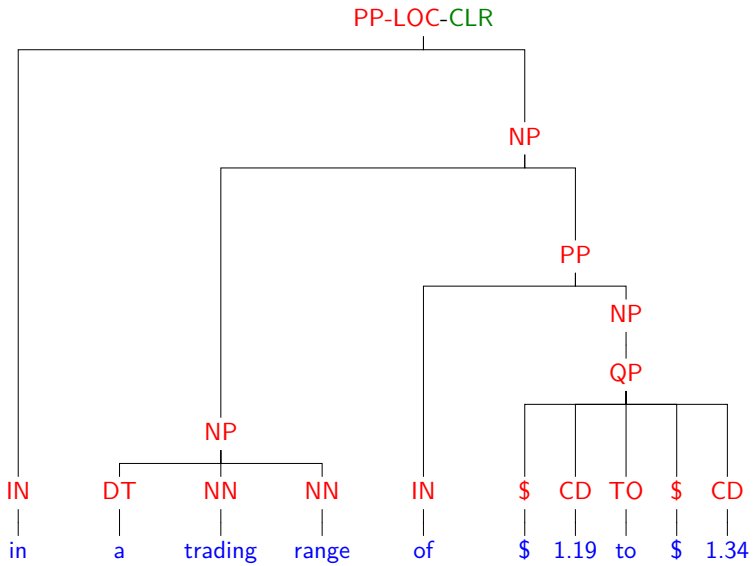
Erroneous annotation in PTB and PTB-UD

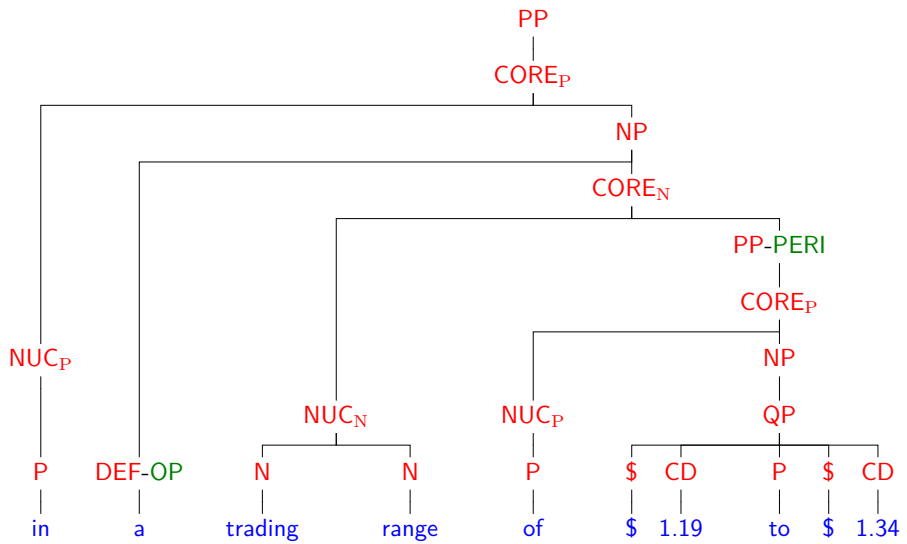
PTB2RRG and UD2RRG: Example 2

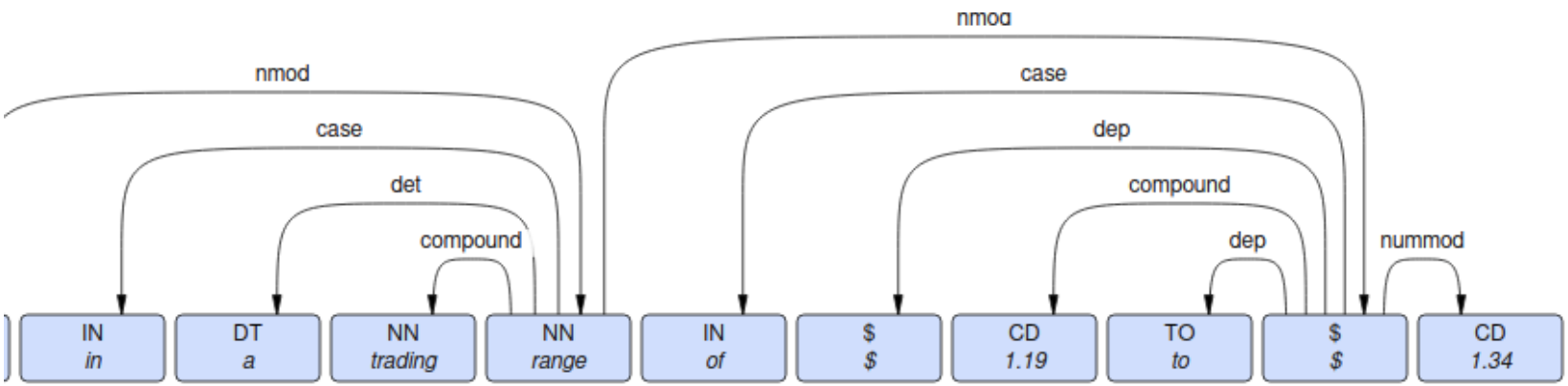


Negation in PTB, PTB-UD, and RRG

PTB2RRG and UD2RRG: Example 3

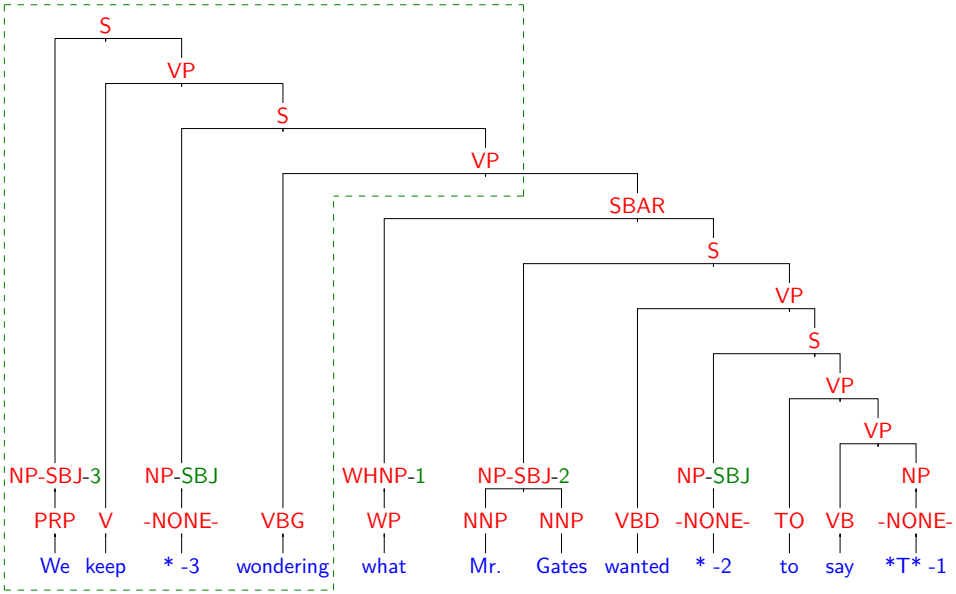


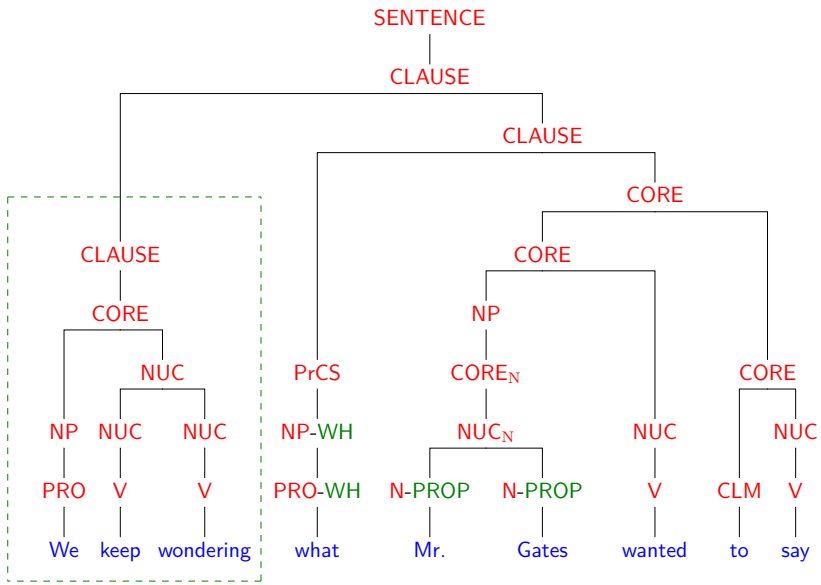


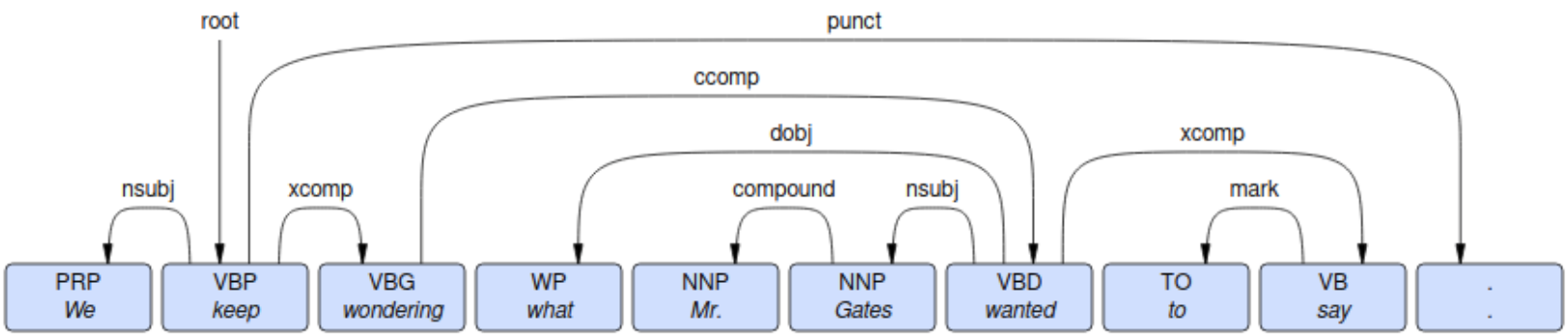


Different junctures

PTB2RRG and UD2RRG: Example 4

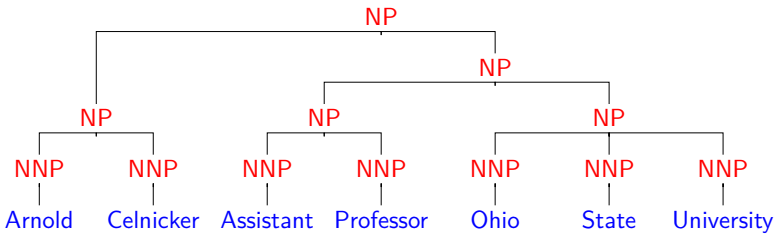


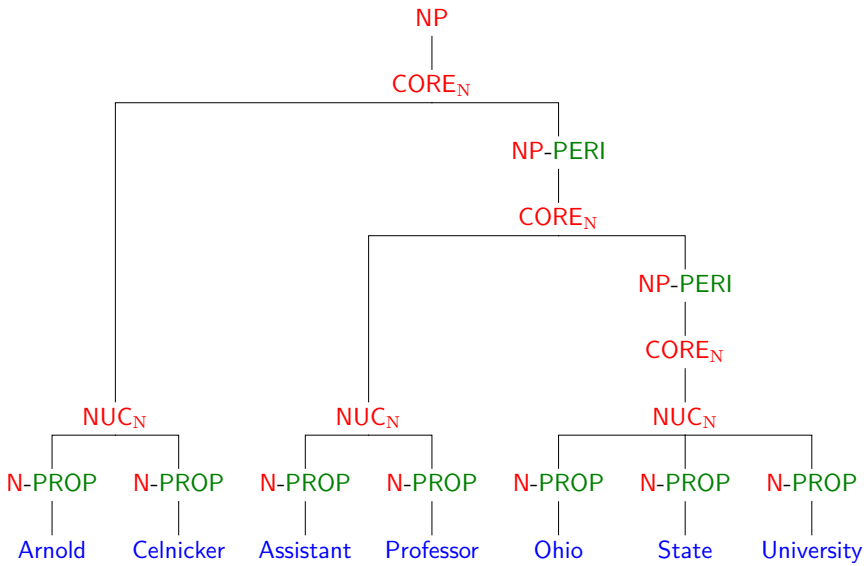


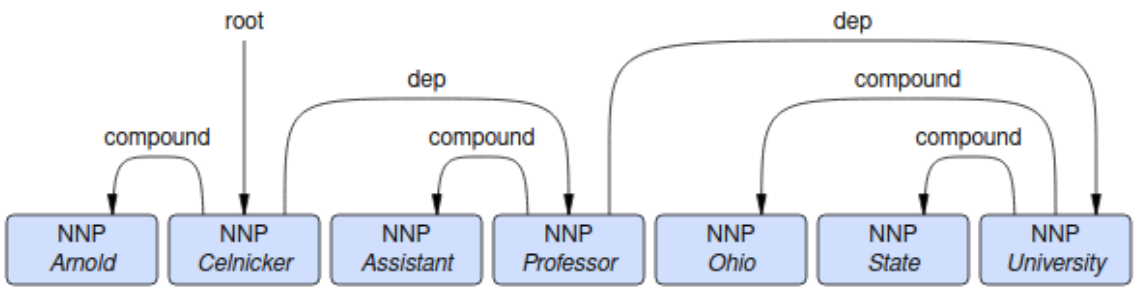


Quantifier Phrases

PTB2RRG and UD2RRG: Example 5







Quantifier Phrases