RRGbank and RRGparbank: Towards a multilingual treebank for Role and Reference Grammar

David Arps, Tatiana Bladier, Kilian Evang, Laura Kallmeyer, Robin Möllemann, Rainer Osswald, Simon Petitjean

HHU MoSy Colloquium
21 January 2020

Heinrich Heine Universität Düsseldorf

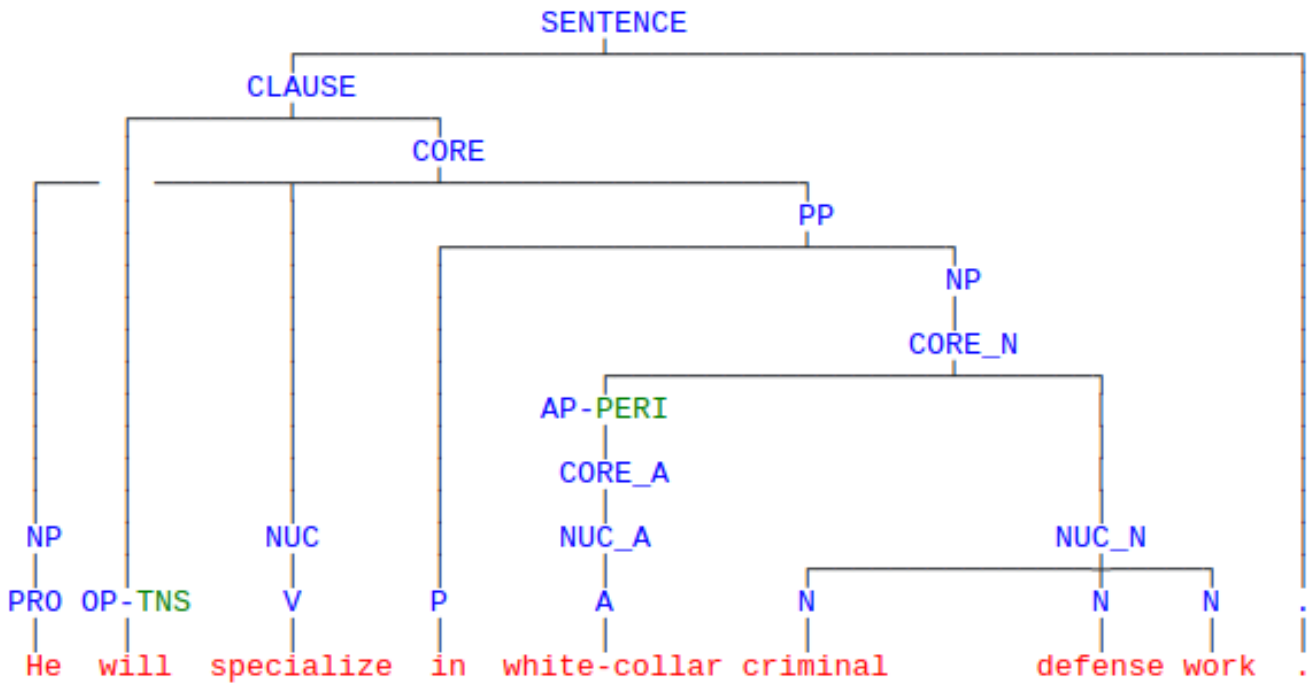
European Research Council

Overview

Outline

RRGbank

- ⋆ RRGbank is a large corpus of RRG annotated sentences (Bladier et al., 2018);
- ⋆ 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,
- ⋆ computational applications.

Outline

RRG: Textbook Notation

- RRG assumes that clauses have a *layered structure*:
  - The *nucleus* specifies the verb/the predication,
  - the *core* layer consists of the nucleus and its arguments,
  - and the *clause* layer contains the core as well as extracted arguments.

RRG: Textbook Notation

- RRG assumes that clauses have a *layered structure*:
  - The *nucleus* specifies the verb/the predication,
  - the *core* layer consists of the nucleus and its arguments,
  - and the *clause* layer contains the core as well as extracted arguments.

- Each of the layers can have a *periphery* for attaching adjuncts.

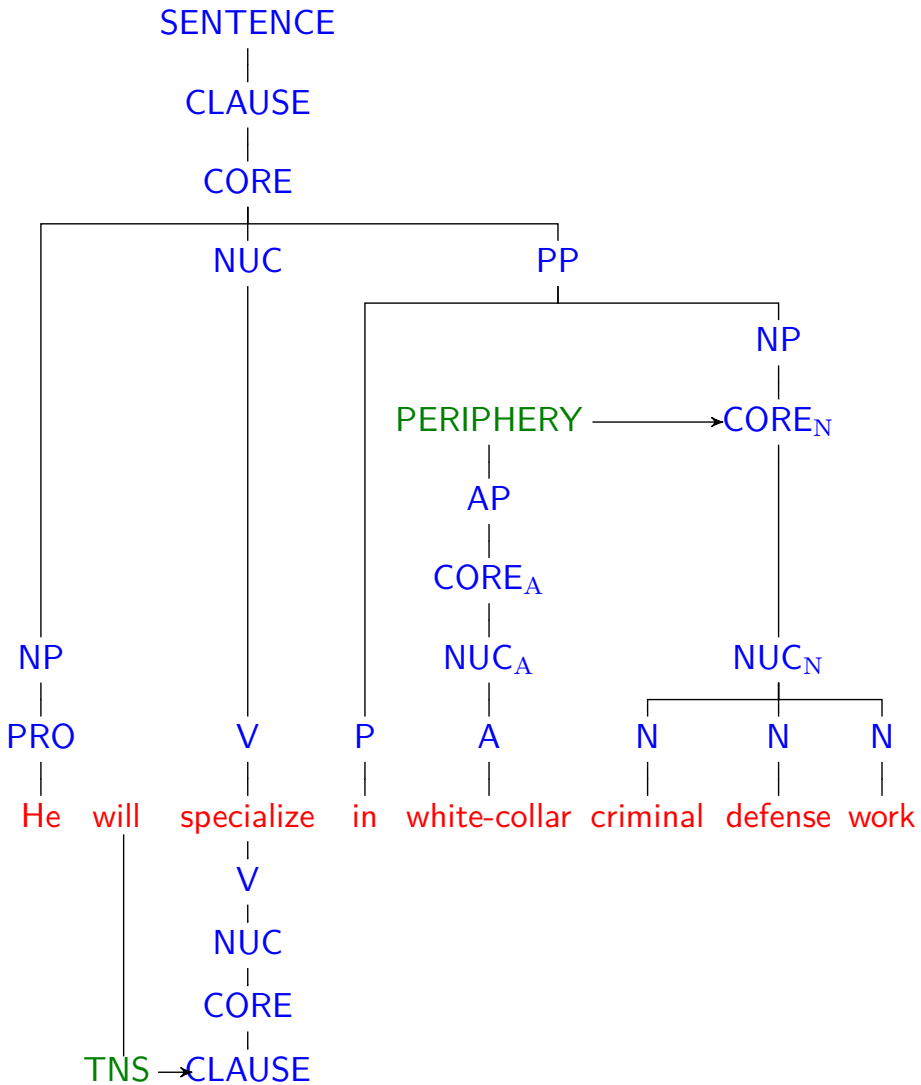
RRG: Textbook Notation

- RRG assumes that clauses have a *layered structure*:
  - The *nucleus* specifies the verb/the predication,
  - the *core* layer consists of the nucleus and its arguments,
  - and the *clause* layer contains the core as well as extracted arguments.
- Each of the layers can have a *periphery* for attaching adjuncts.
- Furthermore, operators (e.g., temporal operators, definiteness operators, modals etc.) are taken to be part of a separate operator projection which is, however, linked to the constituent structure. Each operator scopes over a specific layer.

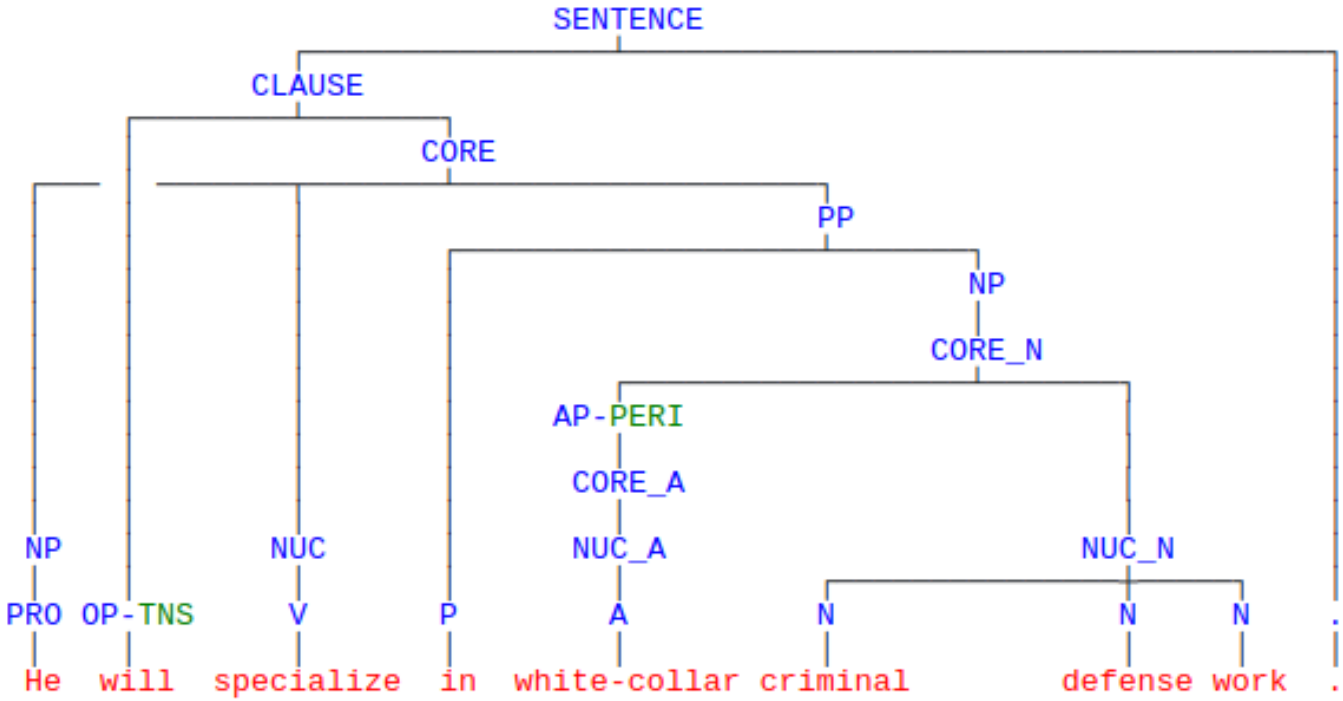
RRG: Textbook Notation

- RRG assumes that clauses have a *layered structure*:
  - The *nucleus* specifies the verb/the predication,
  - the *core* layer consists of the nucleus and its arguments,
  - and the *clause* layer contains the core as well as extracted arguments.
- Each of the layers can have a *periphery* for attaching adjuncts.
- Furthermore, operators (e.g., temporal operators, definiteness operators, modals etc.) are taken to be part of a separate operator projection which is, however, linked to the constituent structure. Each operator scopes over a specific layer.
- Other projections of predicative elements (NPs, APs etc.) also come with layers of NUC, CORE and full phrase.

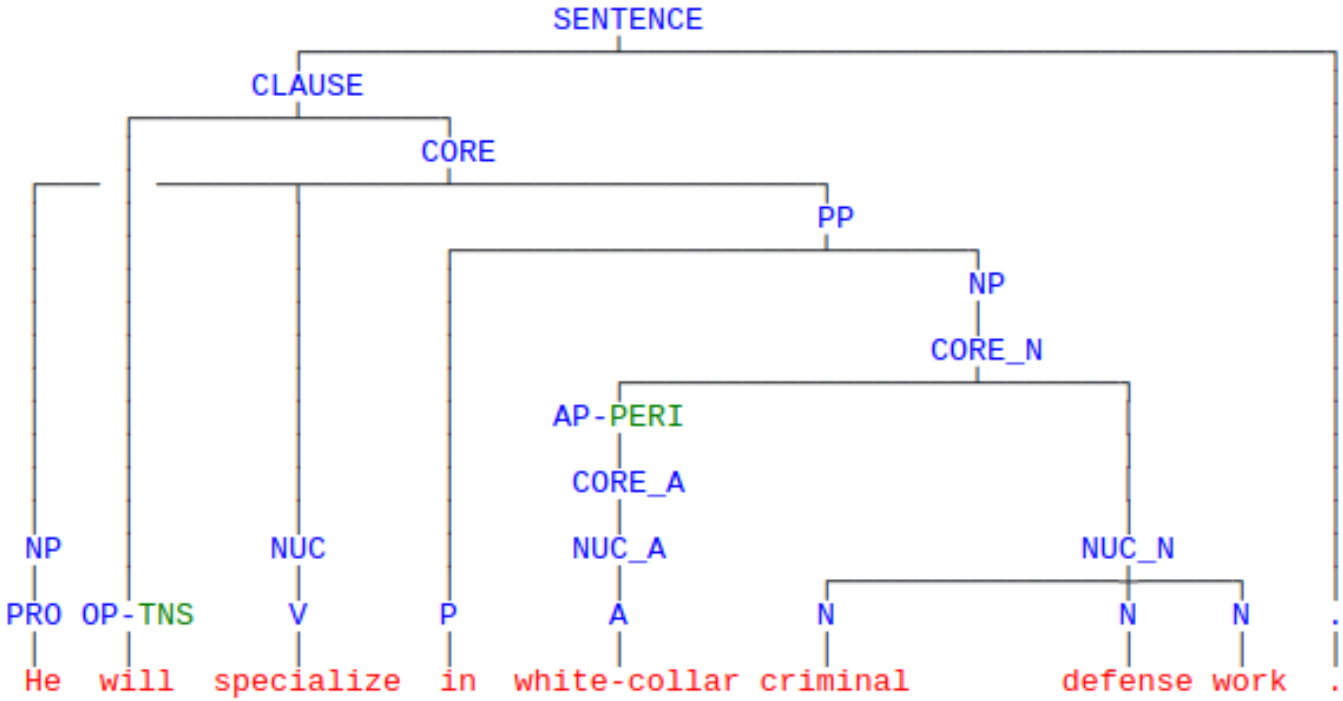
RRG: Textbook Notation



Our Single-Tree Notation

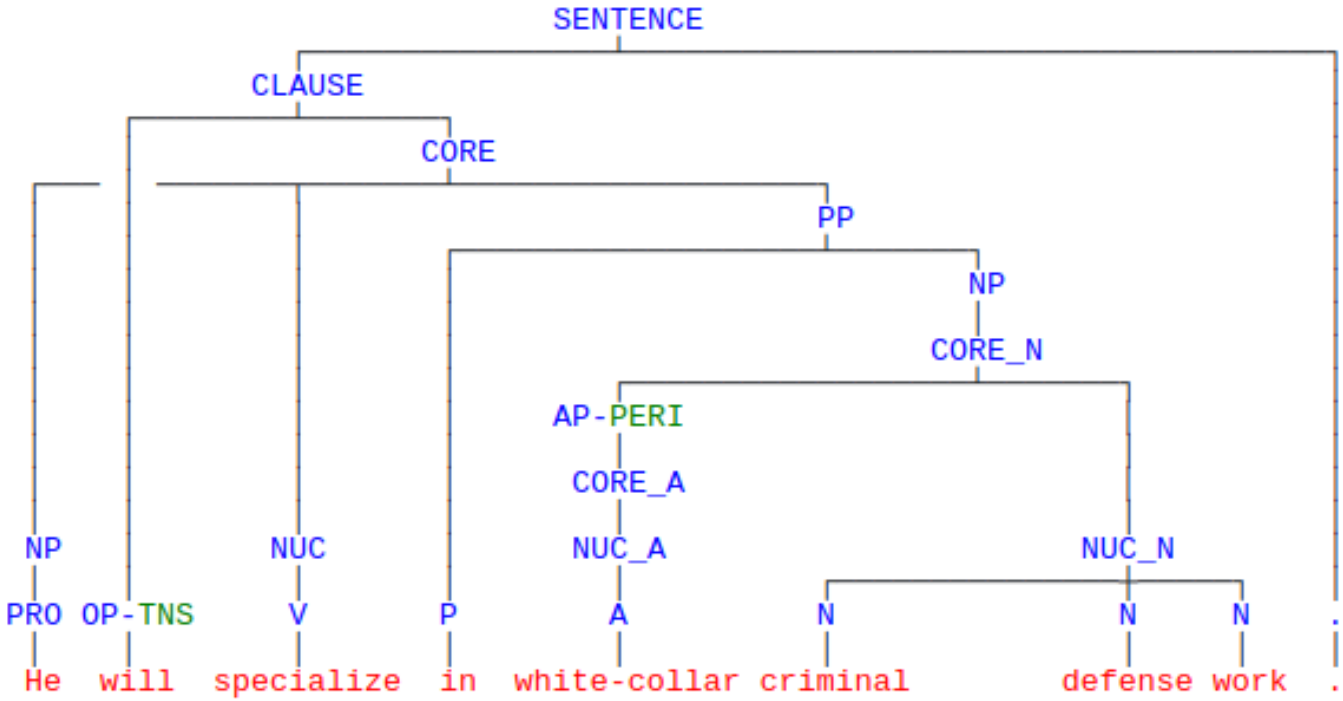


Our Single-Tree Notation



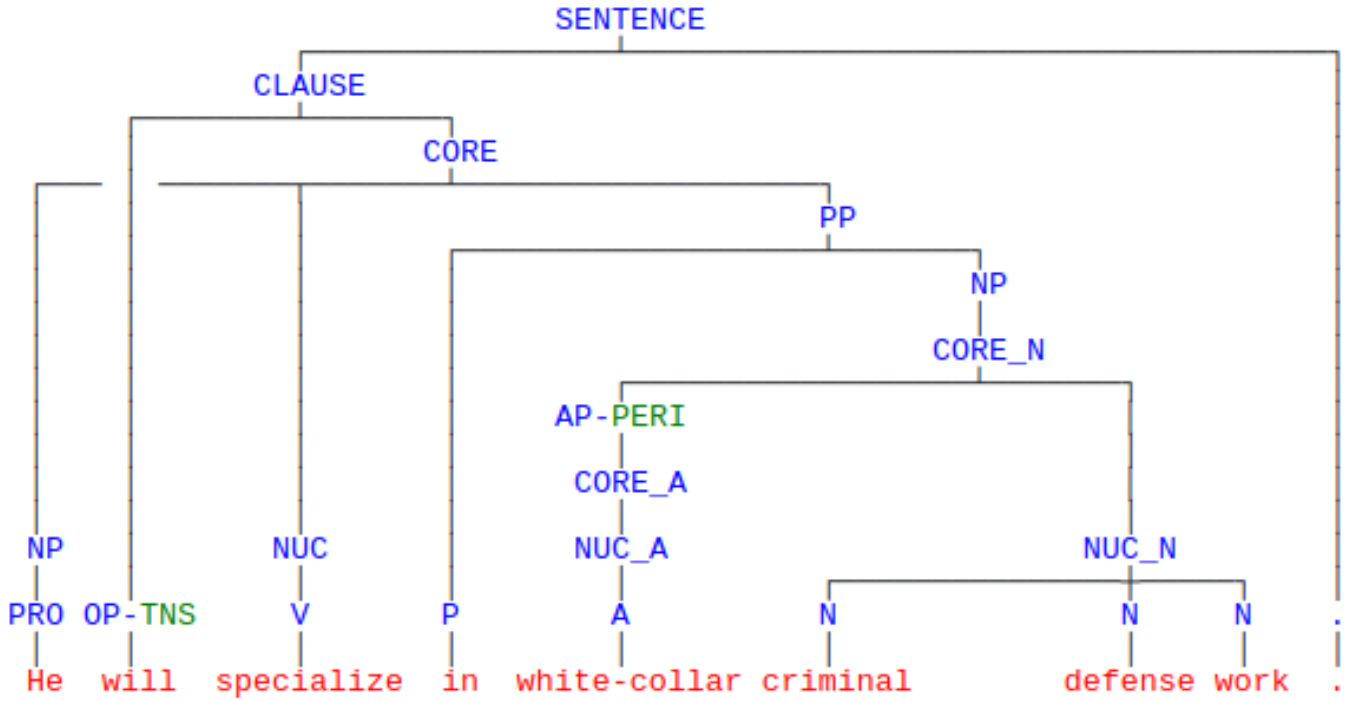
- ⋆ operators, peripheries and CLM are attached to constituent projection where they take scope,

Our Single-Tree Notation



- ⋆ operators, peripheries and CLM are attached to constituent projection where they take scope,
  → crossing branches possible,

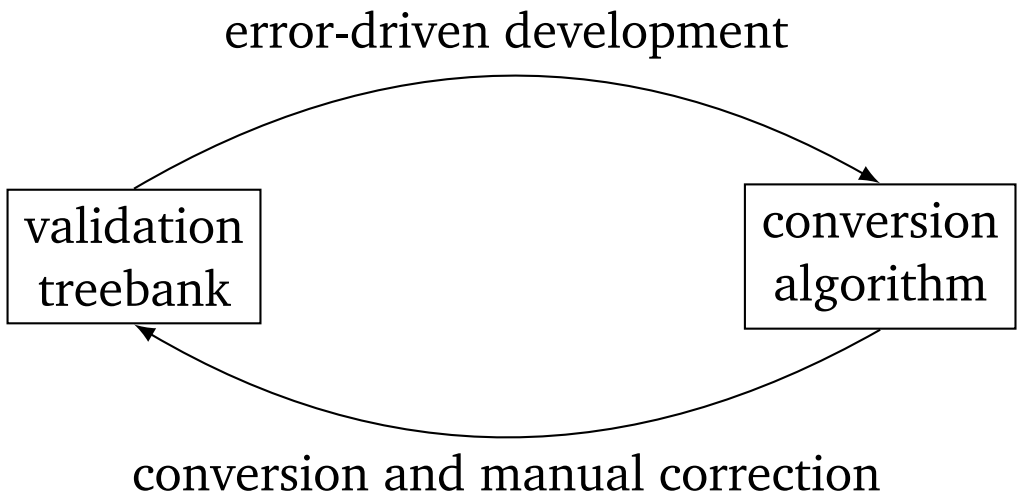
Our Single-Tree Notation



- ⋆ operators, peripheries and CLM are attached to constituent projection where they take scope,
  → crossing branches possible,
- ⋆ operators have category OP and an extension giving their type,
- ⋆ periphery elements have an extension -PERI.

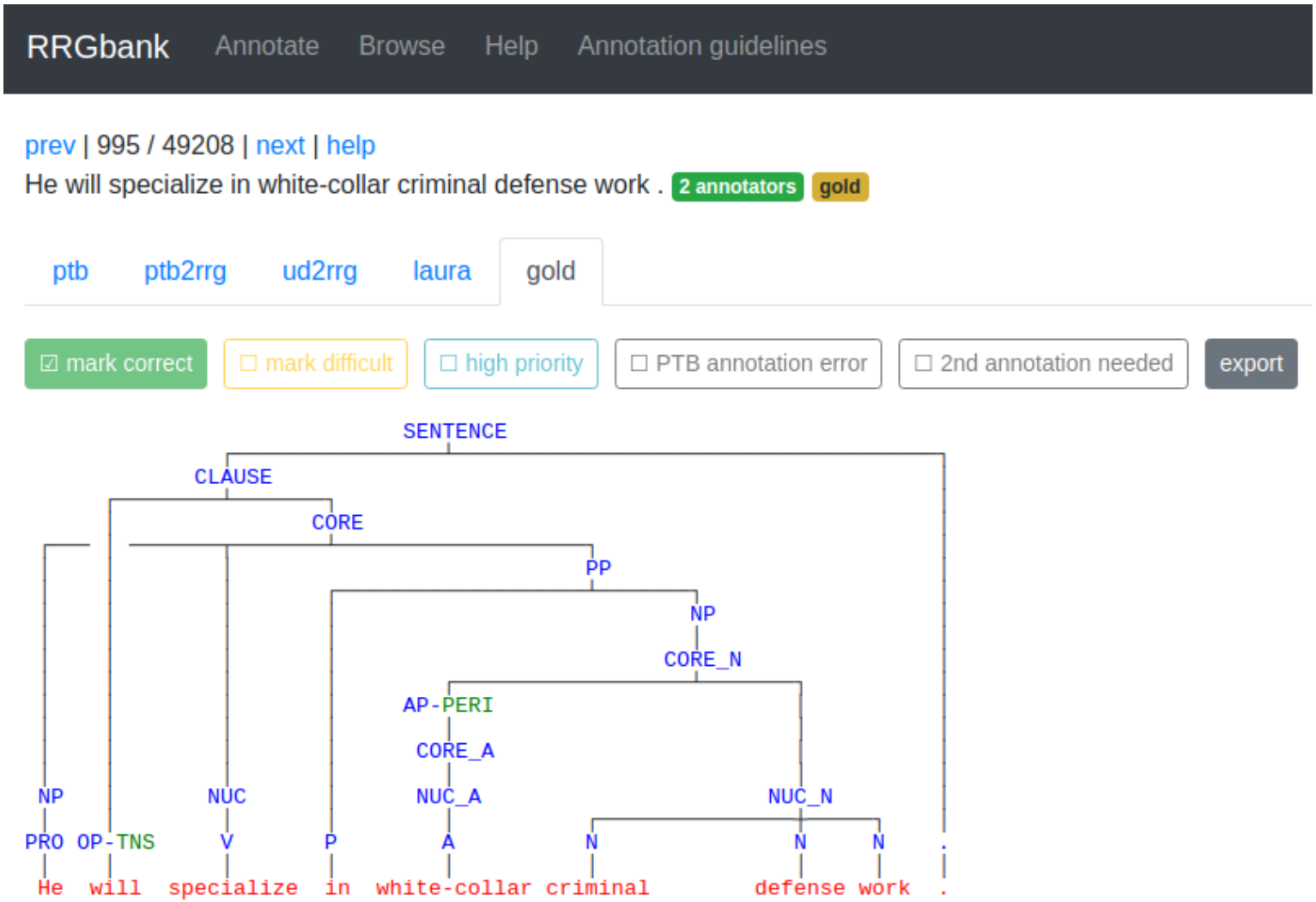
Creating a Validation Treebank

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



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

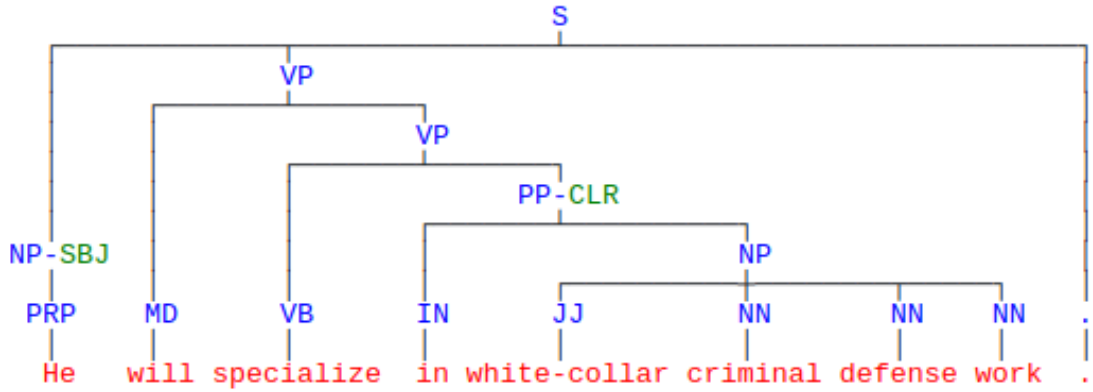
Validation Treebank



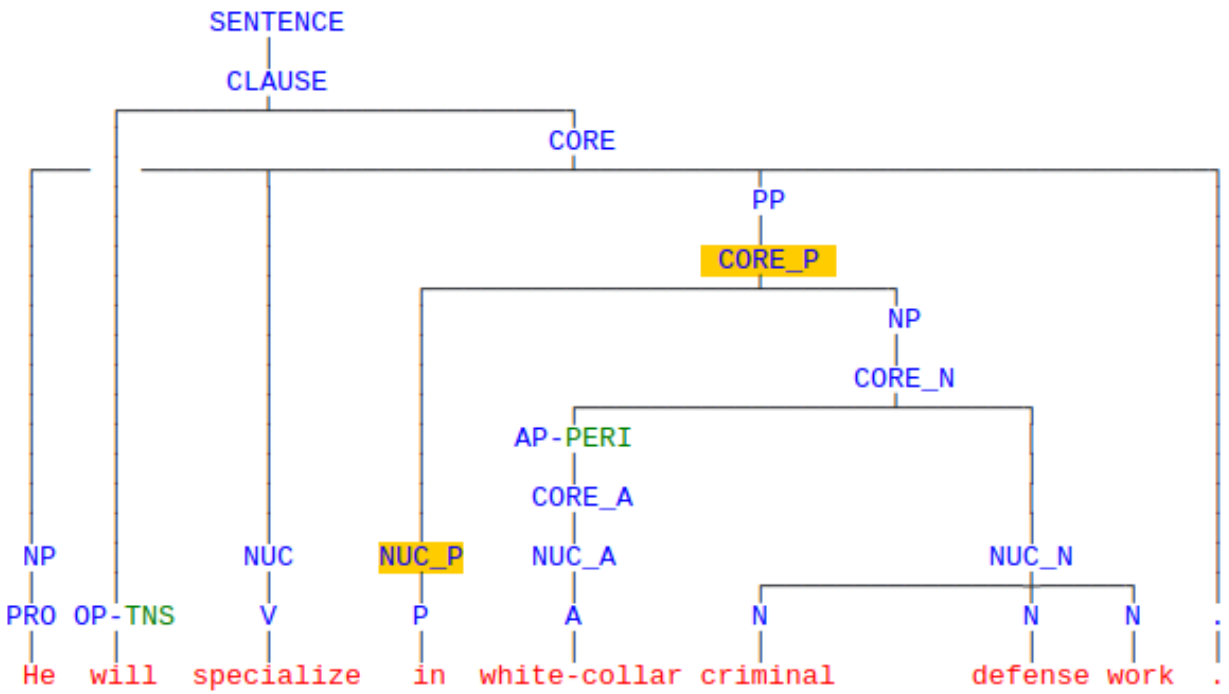
RRG annotation tool: rrgbank.phil.hhu.de

Automatic conversion from the Penn Treebank

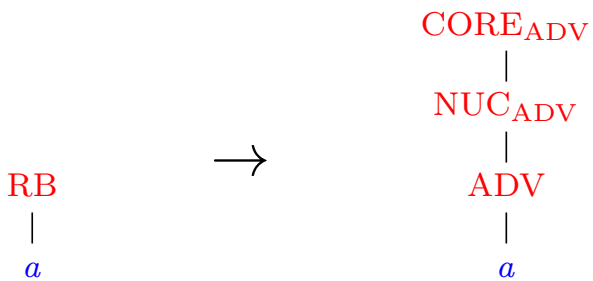
PTB tree:



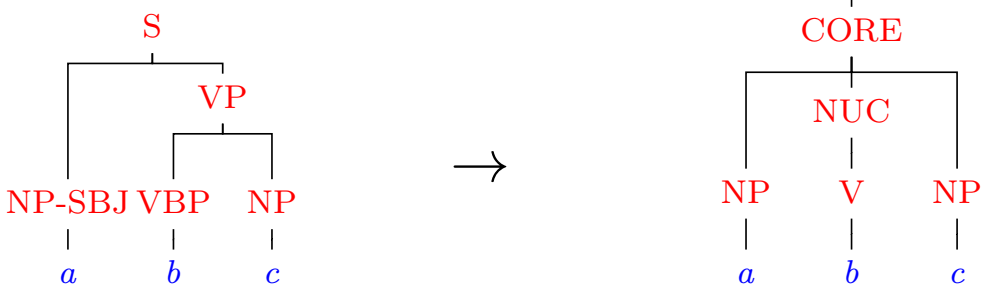
PTB2RRG tree:



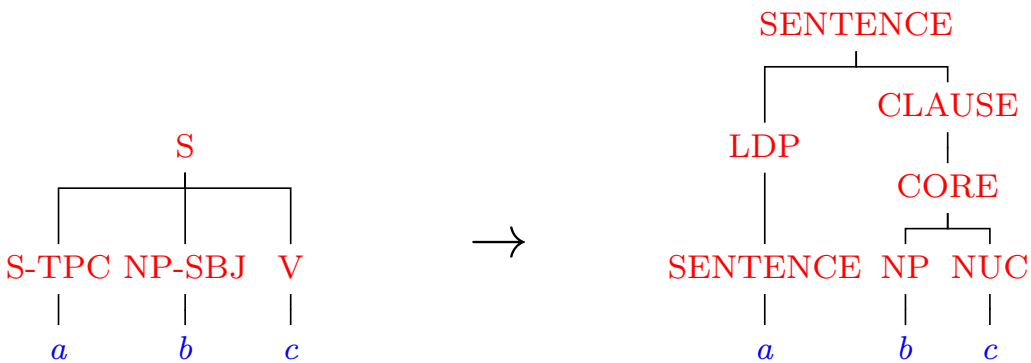
Example Conversion Rule 1/3: Adverb



Example Conversion Rule 2/3: Sentence



Example Conversion Rule 3/3: Topicalization



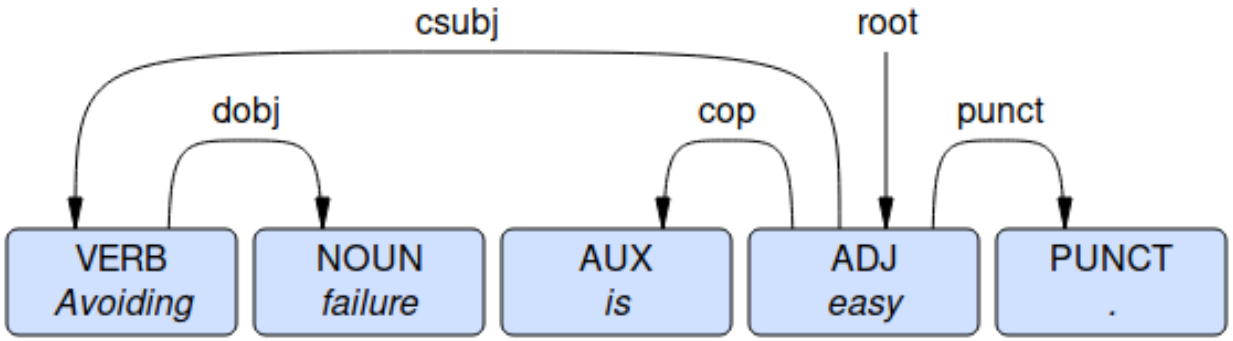
Statistics on RRGbank

- ⋆ 3 active annotators,
- ⋆ 2131 gold annotated sentences → validated and adjudicated by at least two annotators,
- ⋆ 1212 silver annotated sentences → validated by one annotator,
- ⋆ 50.000 sentences from Penn Treebank (PTB).
- ⋆ accuracy: 93.07 (PTB2RRG) and 86.89 (UD2RRG) on the development set.

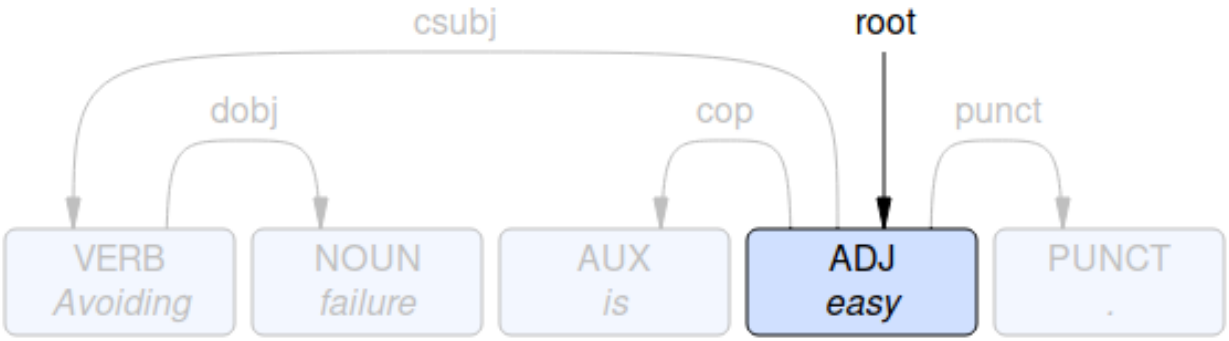
Universal Dependencies to RRG: Automatic Conversion

	`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)	**93.07**	**86.89**
coverage (short sent.)	100%	96.9%
converted gold sentences	2000 (all)	1931 (of 2000)

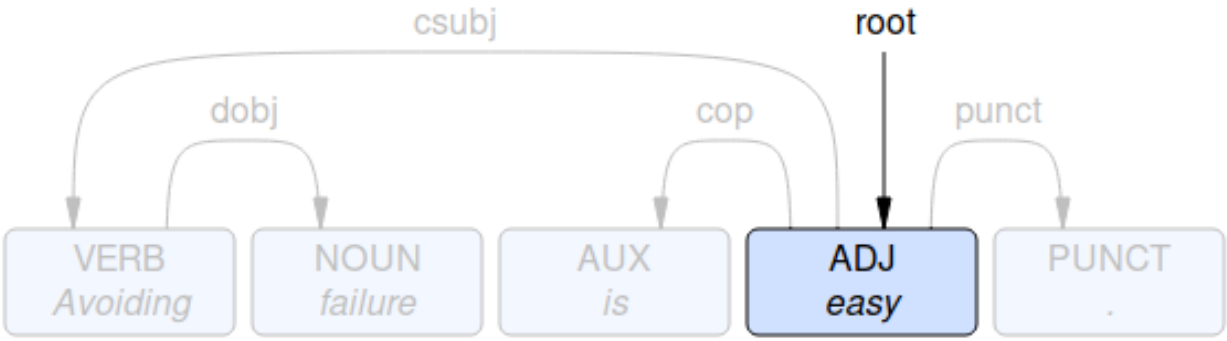
Example PTB-UD to RRG

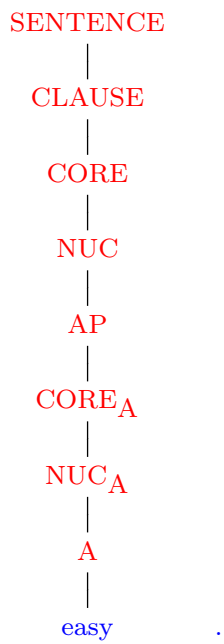


Example PTB-UD to RRG

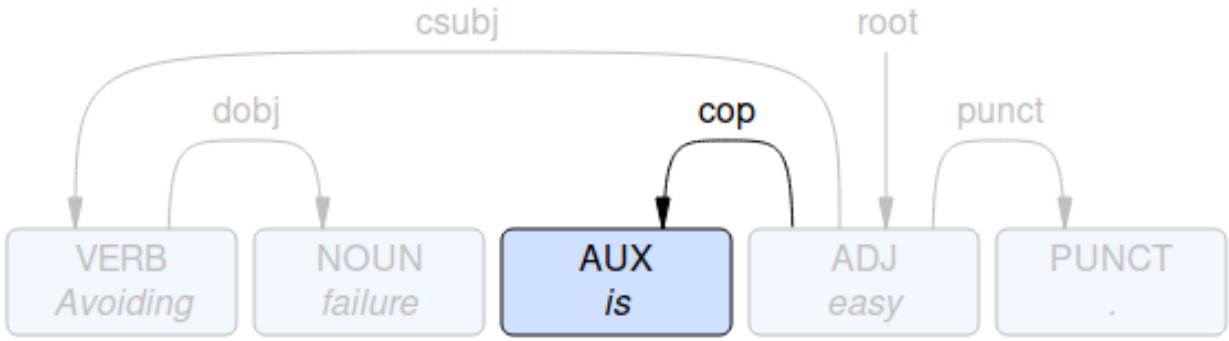


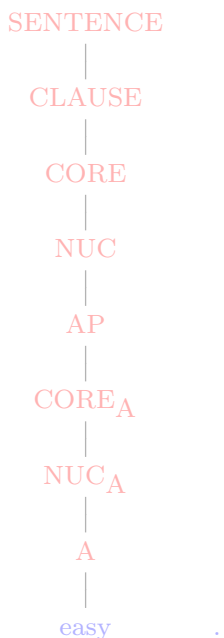
Example PTB-UD to RRG



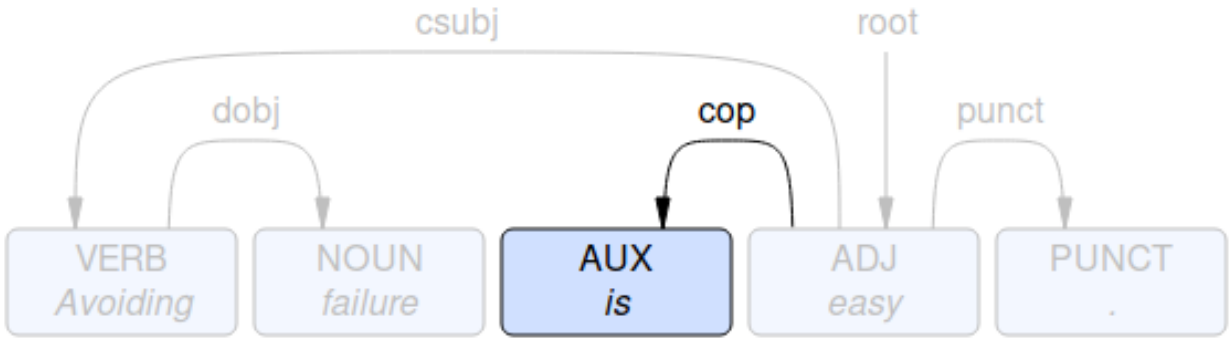


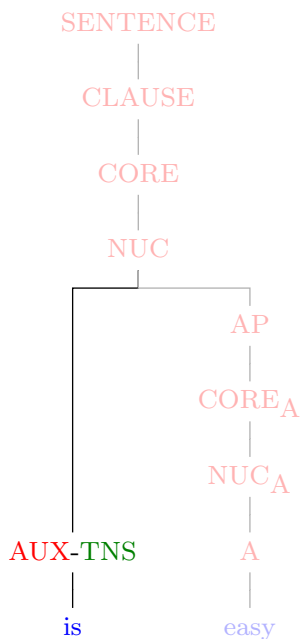
Example PTB-UD to RRG



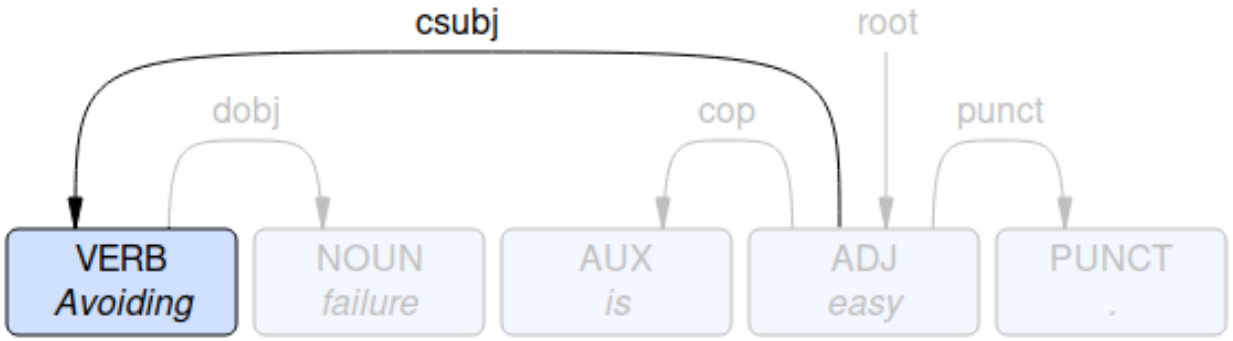


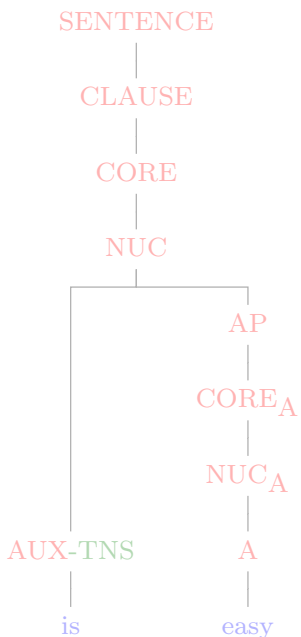
Example PTB-UD to RRG



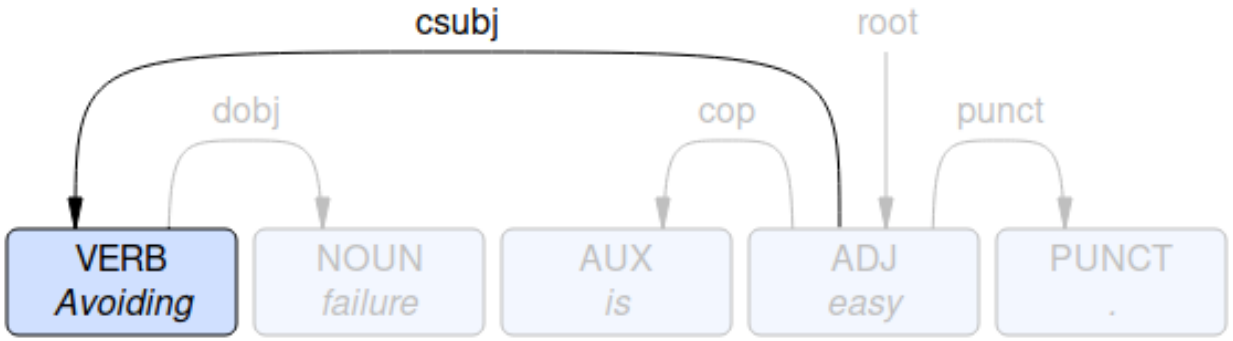


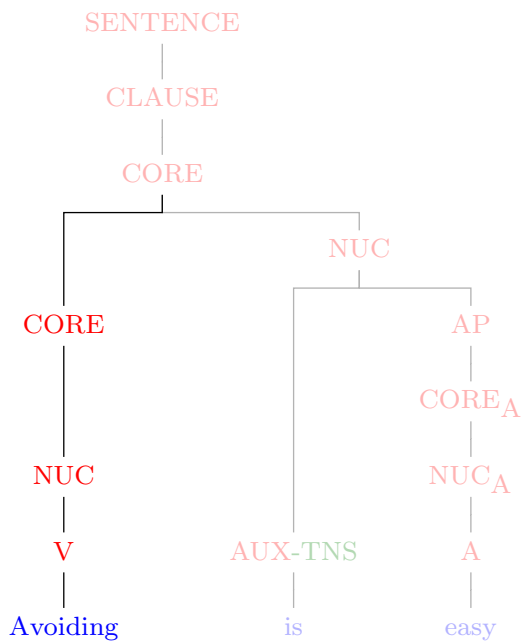
Example PTB-UD to RRG



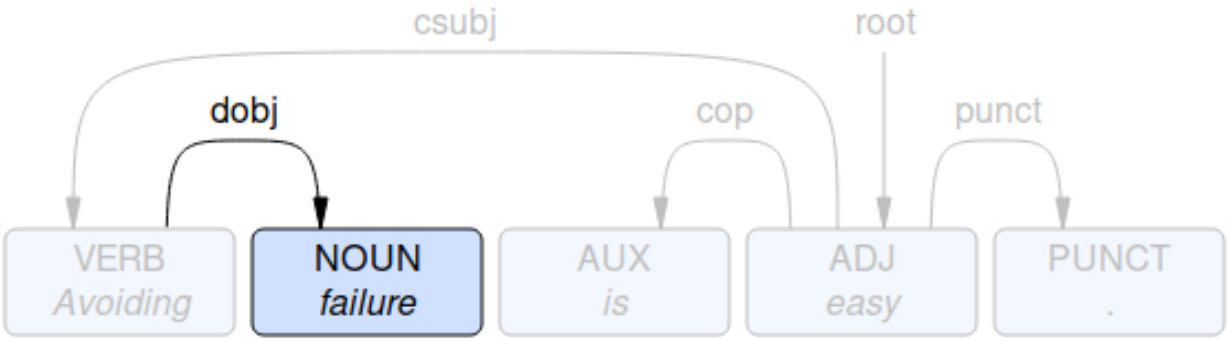


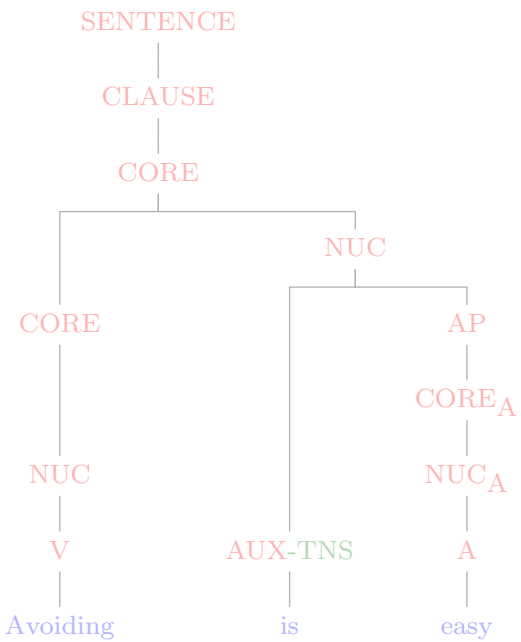
Example PTB-UD to RRG



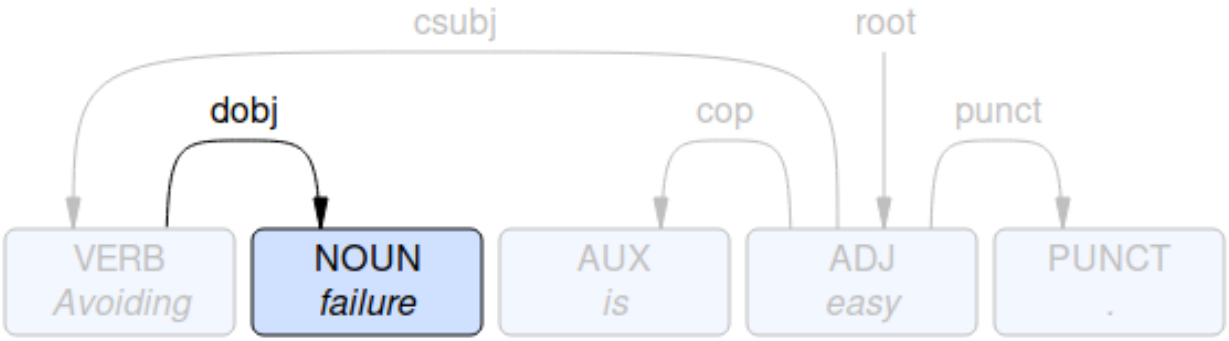


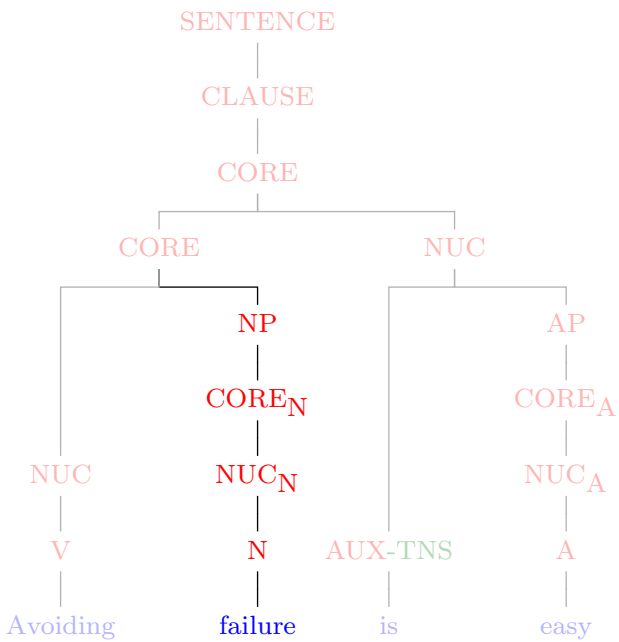
Example PTB-UD to RRG



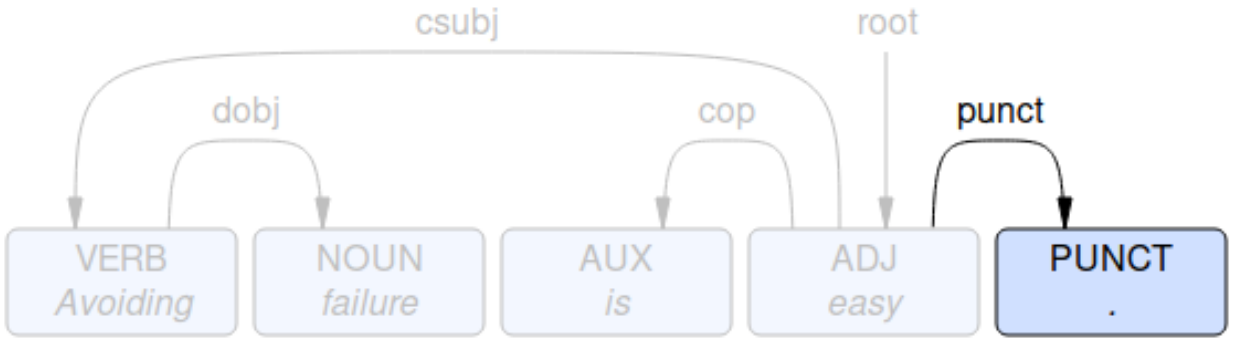


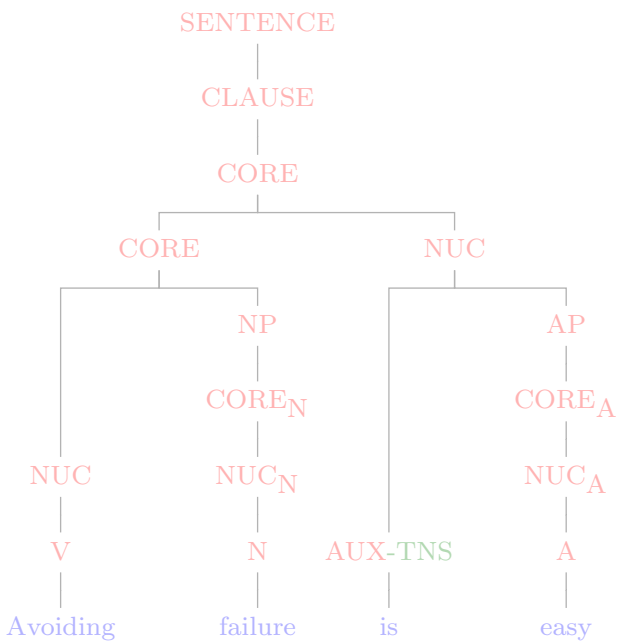
Example PTB-UD to RRG



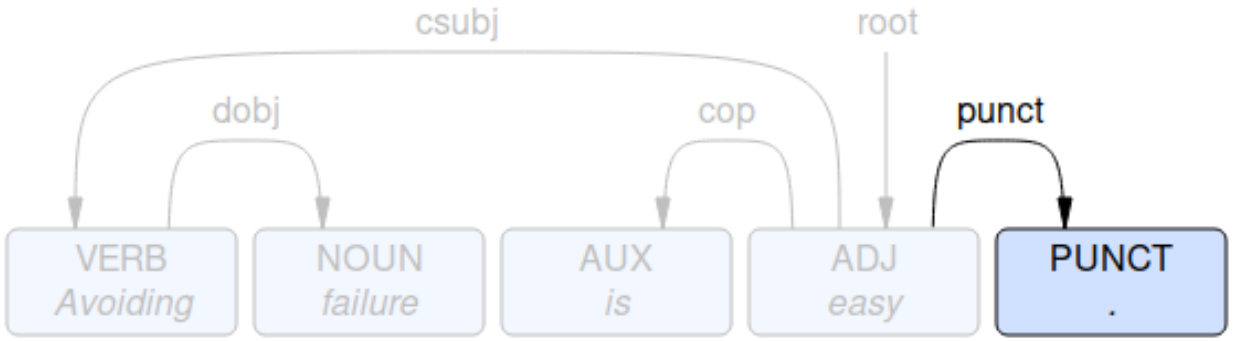


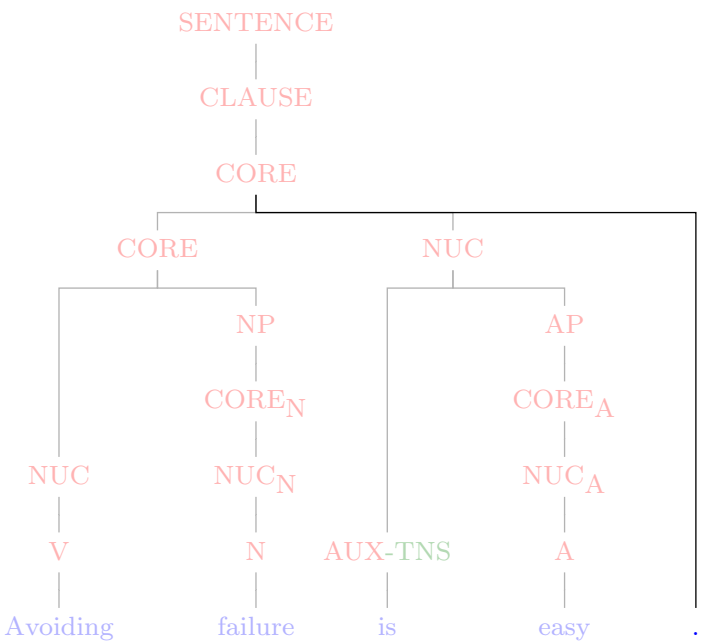
Example PTB-UD to RRG



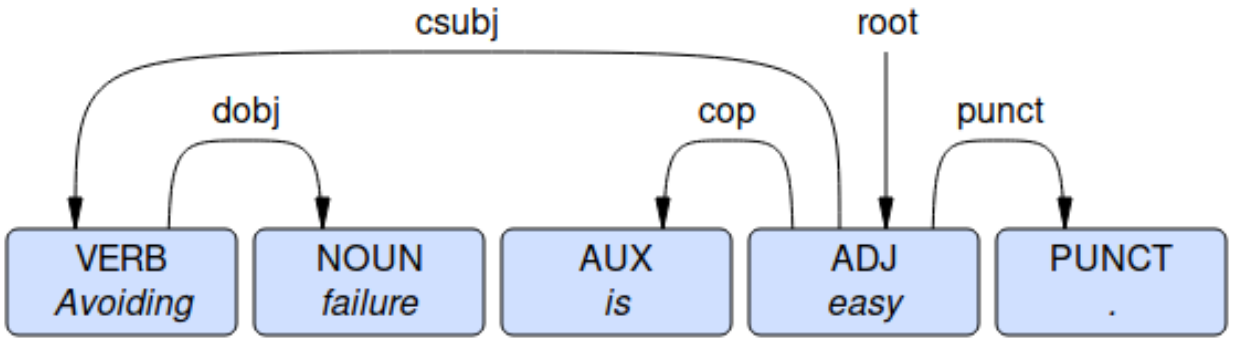


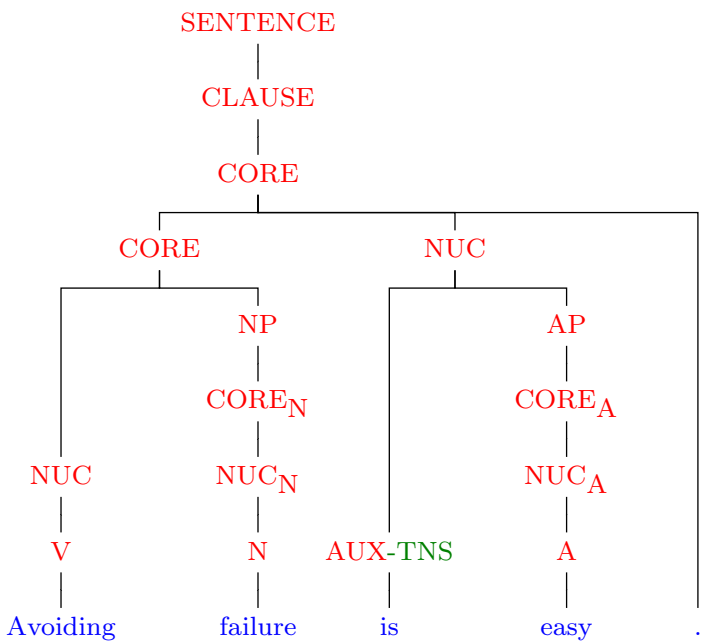
Example PTB-UD to RRG



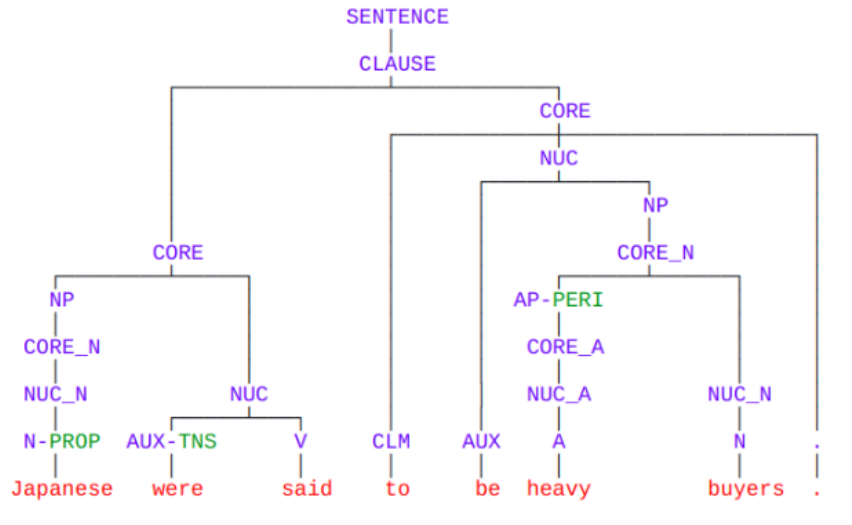


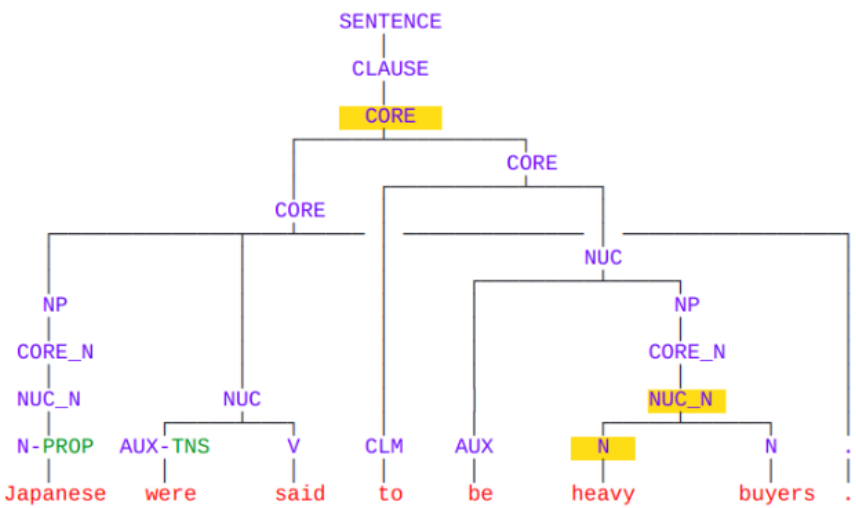
Example PTB-UD to RRG





RRGbank: evaluation





- ⋆ Gold manually validated sentences = 2000;
- ⋆ EVALB bracketing scores:
  - ⇒ matching spans,
  - ⇒ matching brackets,
  - ⇒ matching labels.

RRGbank: evaluation metrics PTB2RRG and UD2RRG

	ptb2rrg	**ud2rrg**
number of converted sentences:	2000	**1931**
longest sentence:	44	44
labeled recall:	92.71	85.20
labeled precision:	93.44	88.65
labeled f-measure:	**93.08**	**86.89**
exactly matched sentences:	48.90	40.45
function tags:	94.50	100.00
POS accuracy:	96.64	96.34

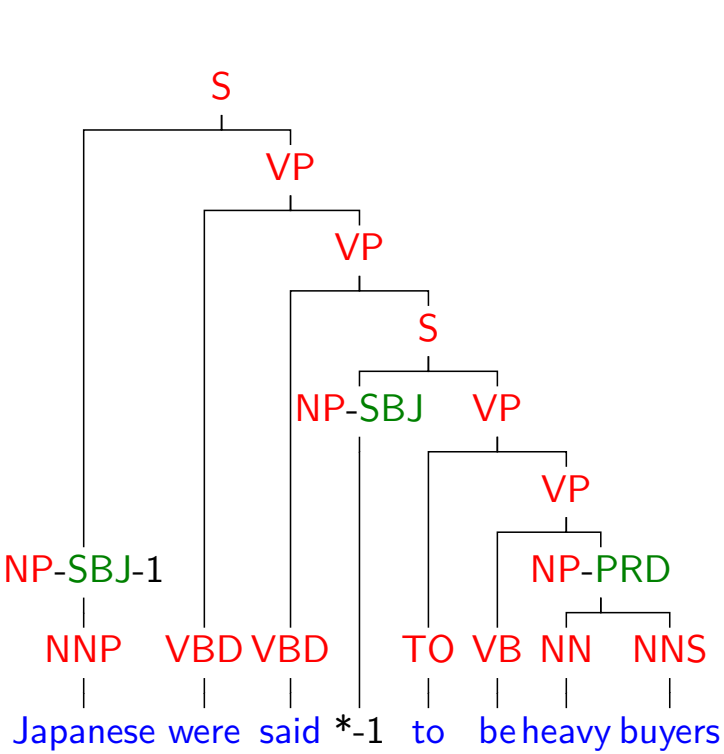
Outline

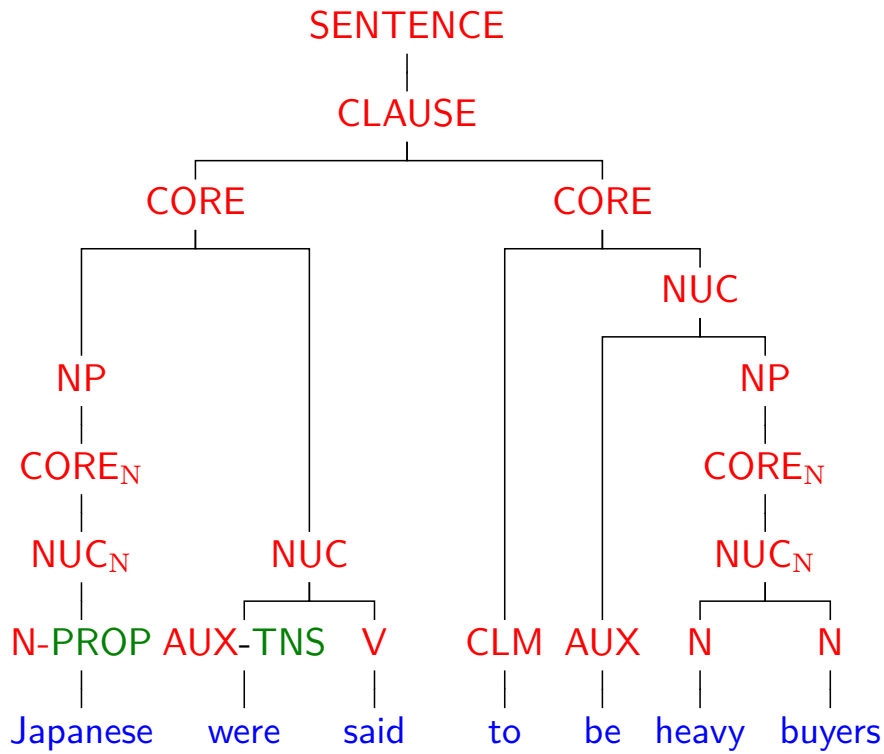
Encountered issues and problematic cases

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

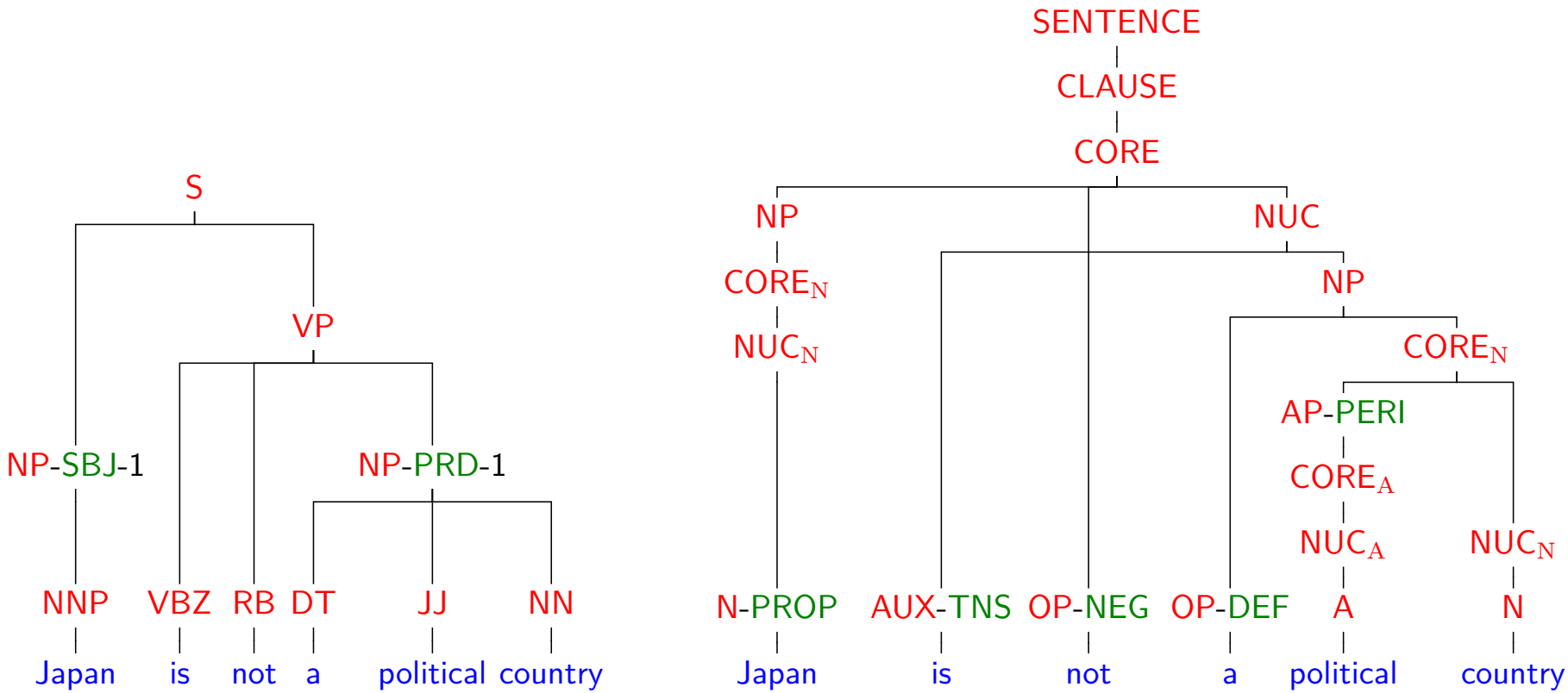
Example 1: Annotation inconsistencies

- 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

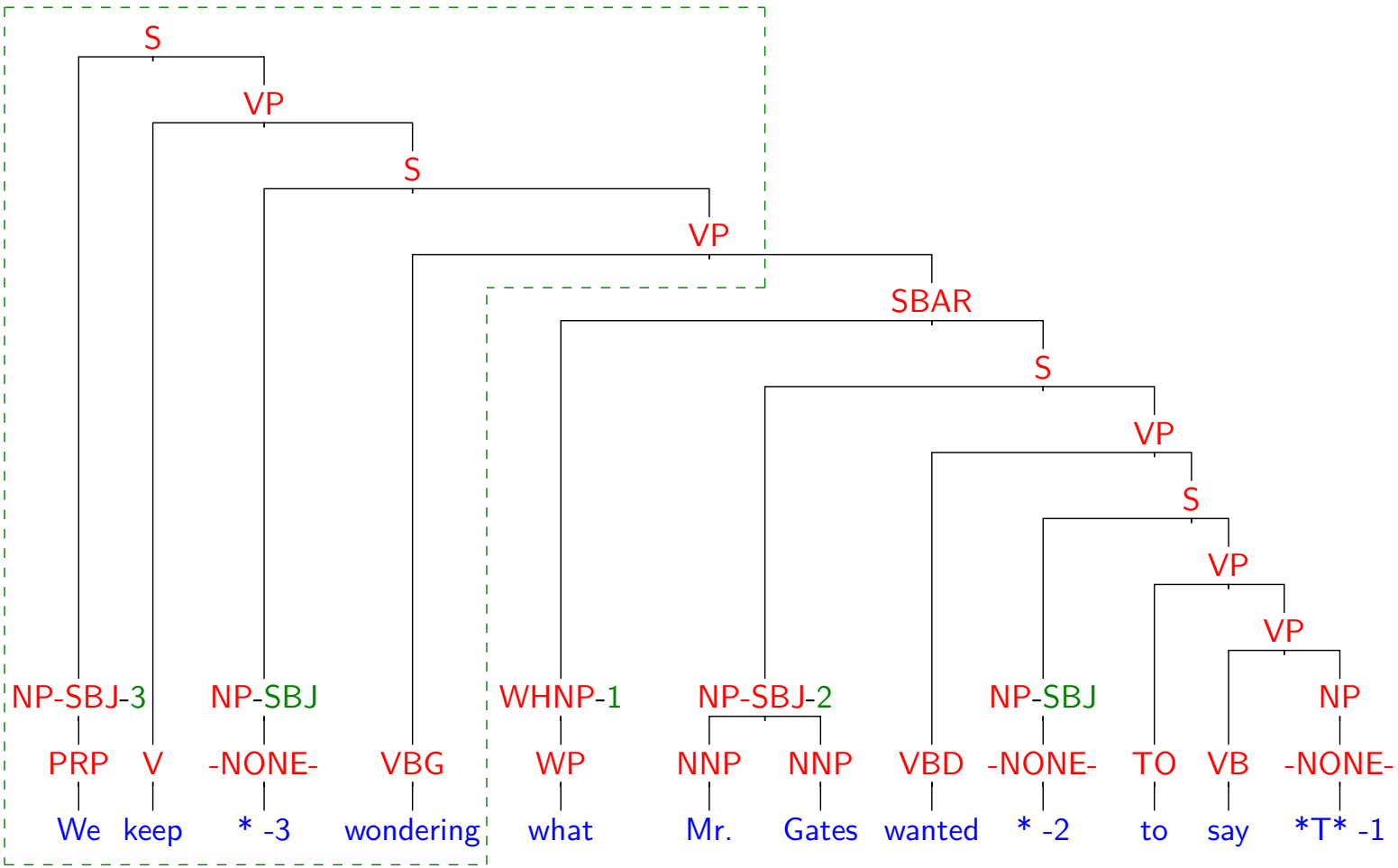
Example 2: Scope of negation

- PTB doesn't differentiate between internal and external negation

Example 2: Scope of negation

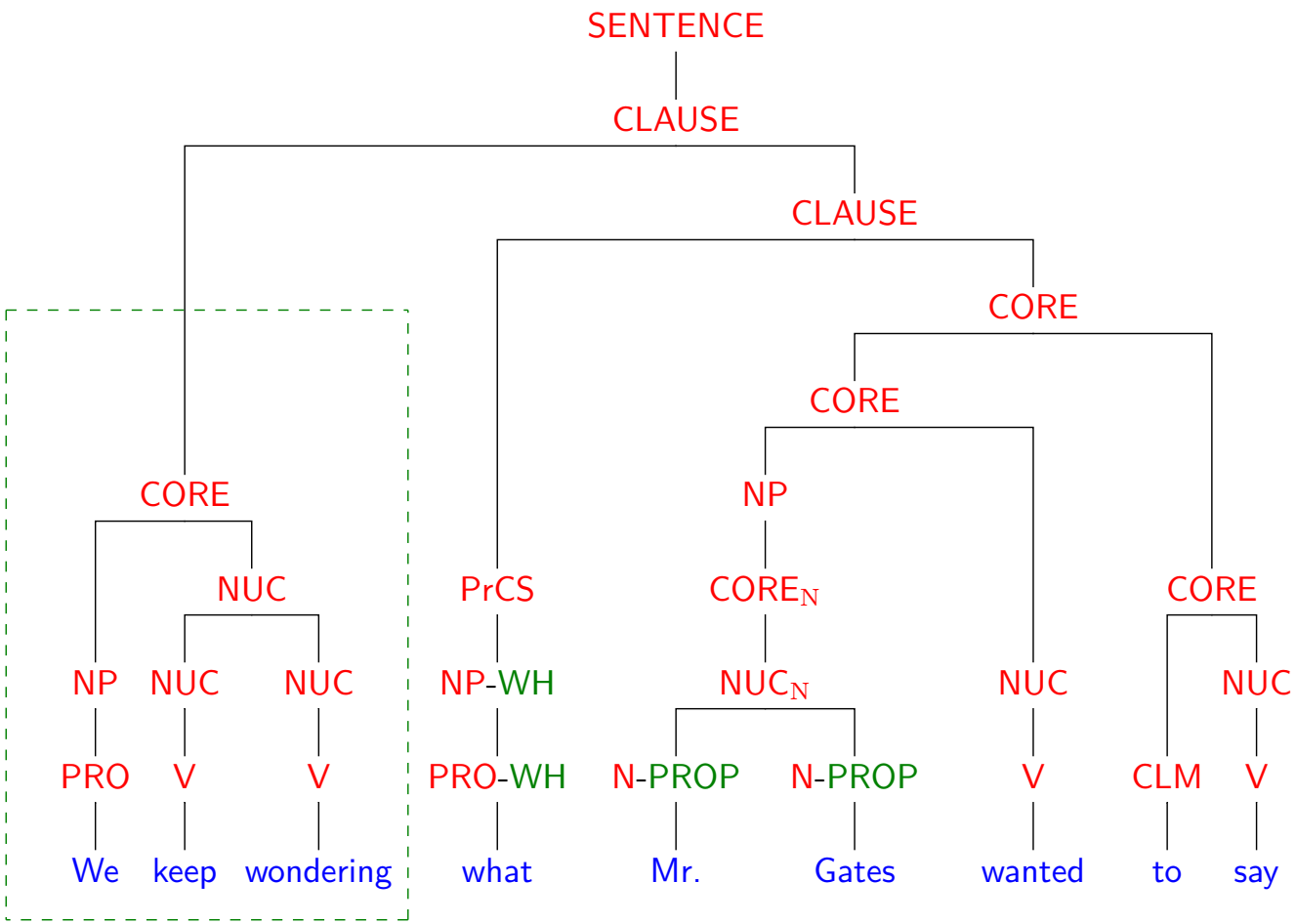
- PTB doesn't differentiate between internal and external negation
- As internal negation is more common, negation is treated as a core-operator

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

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.
- QPs have been retained in the conversion and are placed within the CORE of the phrase they modify.

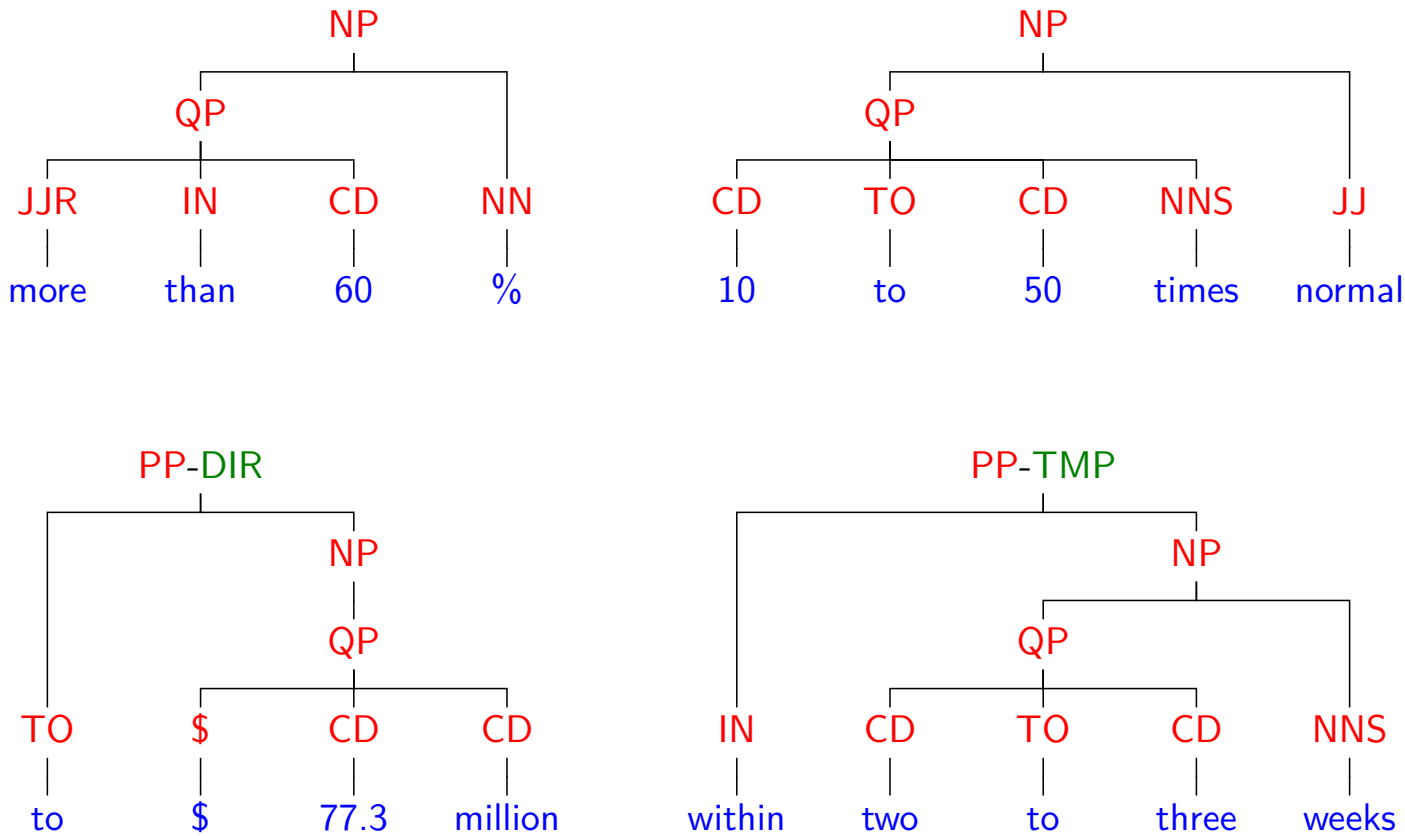
Example 4: Quantifier Phrases

- 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.
- QPs have been retained in the conversion and are placed within the CORE of the phrase they modify.
- Headless NPs receive a nominal head, which is extracted from QPs where necessary

Example 4: Quantifier Phrases

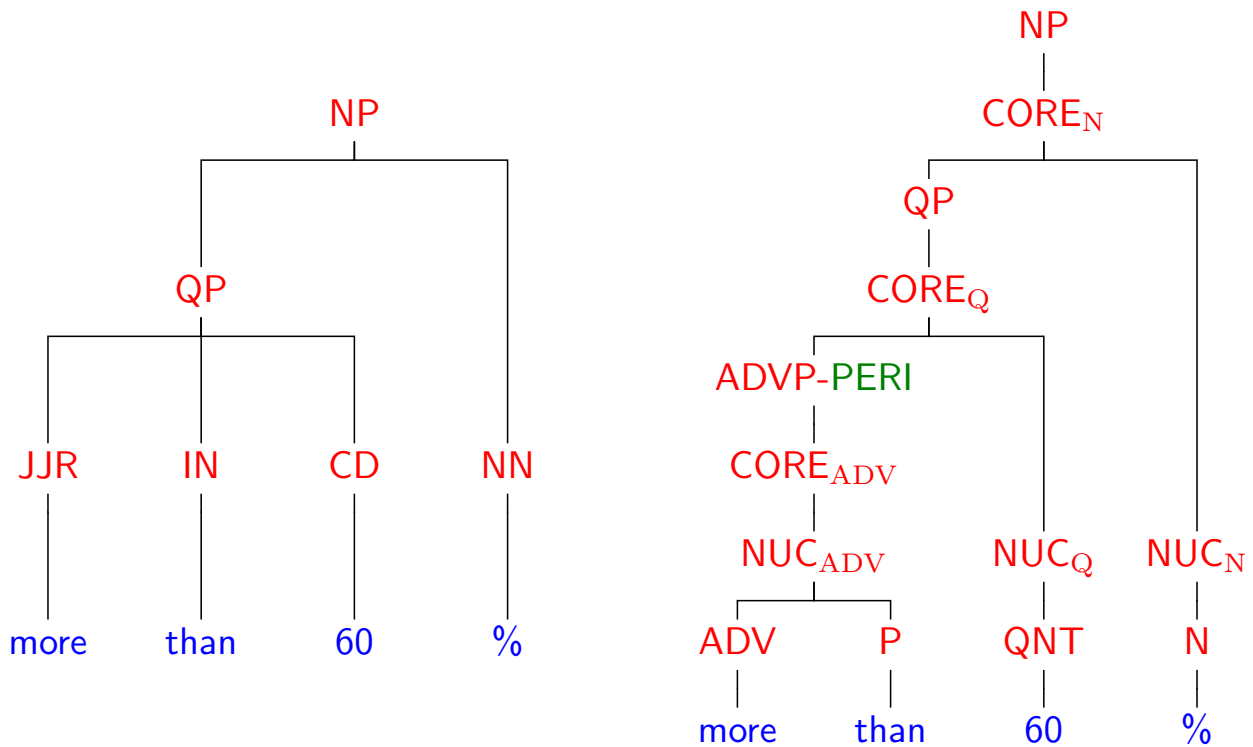
- QPs are inconsistent with regard to both the lexical category and internal position of their heads.
- QPs can function as the only constituent within an NP leaving the NP without a lexical head.
- QPs have been retained in the conversion and are placed within the CORE of the phrase they modify.
- Headless NPs receive a nominal head, which is extracted from QPs where necessary
- Multi word expressions like "more than" are treated as the complex nucleus of a single ADVP modifying the QP

Examples 4.1 - 4.4 Quantifier Phrases



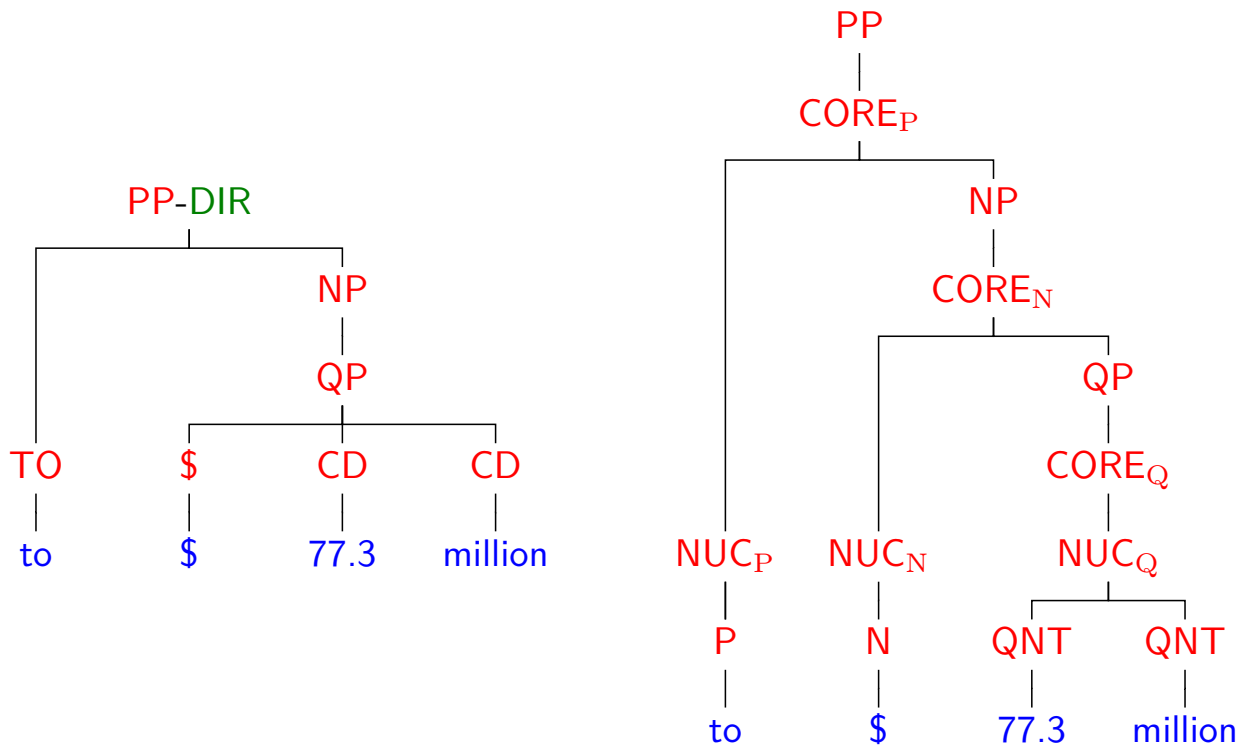
A variety of different QPs in PTB

Example 4.1 Quantifier Phrases



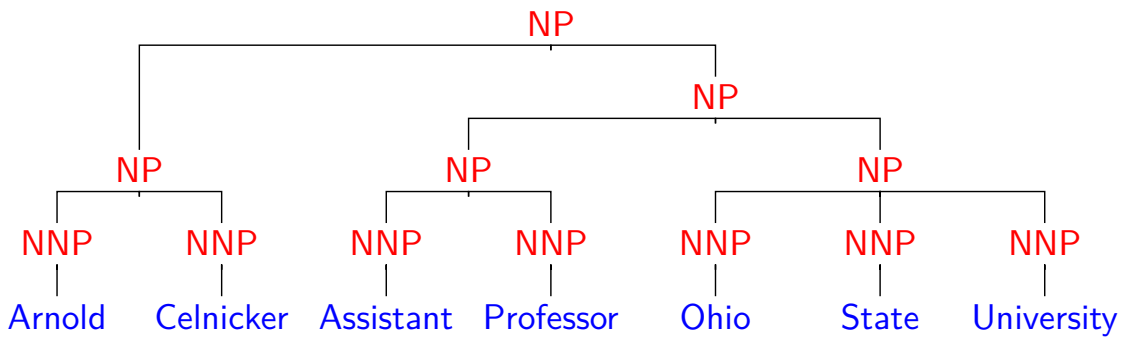
Example 4.1 in PTB and PTB-to-RRG

Example 4.2 Quantifier Phrases



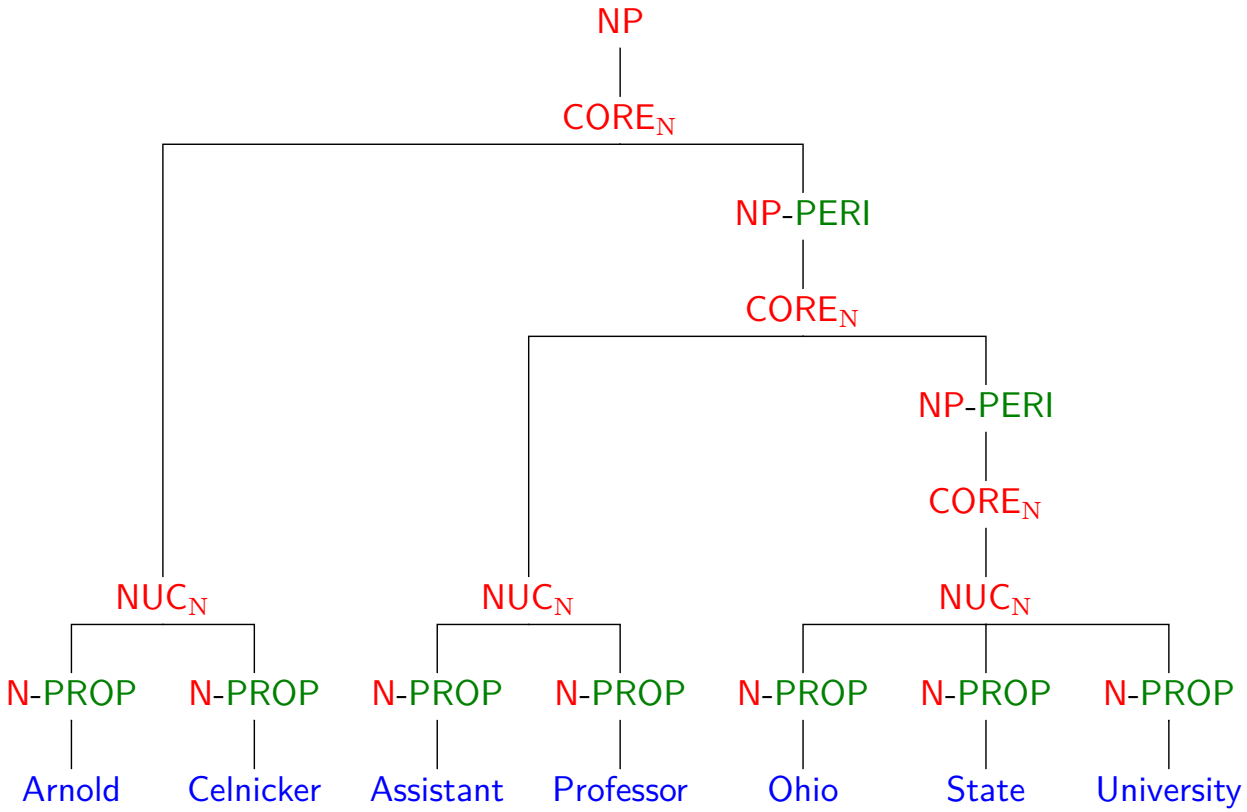
Example 4.2 in PTB and PTB-to-RRG

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

RRGparBank

- Basic idea: Use parallel corpora for RRG annotation
  $\rightarrow$ comparison of language specific constructions
- Current choice: George Orwell's 1984 ($>$ 6500 sentences)

  Available from the MULTEXT-East project page (nl.ijs.si/ME/V4/) in annotated form (of varying degree) in a number of languages: Bulgarian, Czech, English, Estonian, Persian, Hungarian, Macedonian, Polish, Romanian, Slovak, Slovenian, Serbian.

  Current focus on English, Hungarian, Russian, German (old translation from the 1950's).
- Preprocessing by applying available dependency parsers and UD-to-RRG transformation

RRGparBank

Some issues concerning the RRG annotation of German:

- Status of the German **Vorfeld**: Precore or core (or core periphery) or detached or sometimes this and sometimes that?

(1) a. **Maria** hat der Kommilitonin das Syntax-Buch geliehen.
b. **Wer** hat der Kommilitonin das Syntax-Buch geliehen?
c. **Der Kommilitonin** hat Maria das Syntax-Buch geliehen.
d. **Wem** hat Maria das Syntax-Buch geliehen?
e. **Gestern** hat Maria der Kommilitonin das Syntax-Buch geliehen.
f. **Vielleicht** hat Maria der Kommilitonin das Syntax-Buch geliehen.

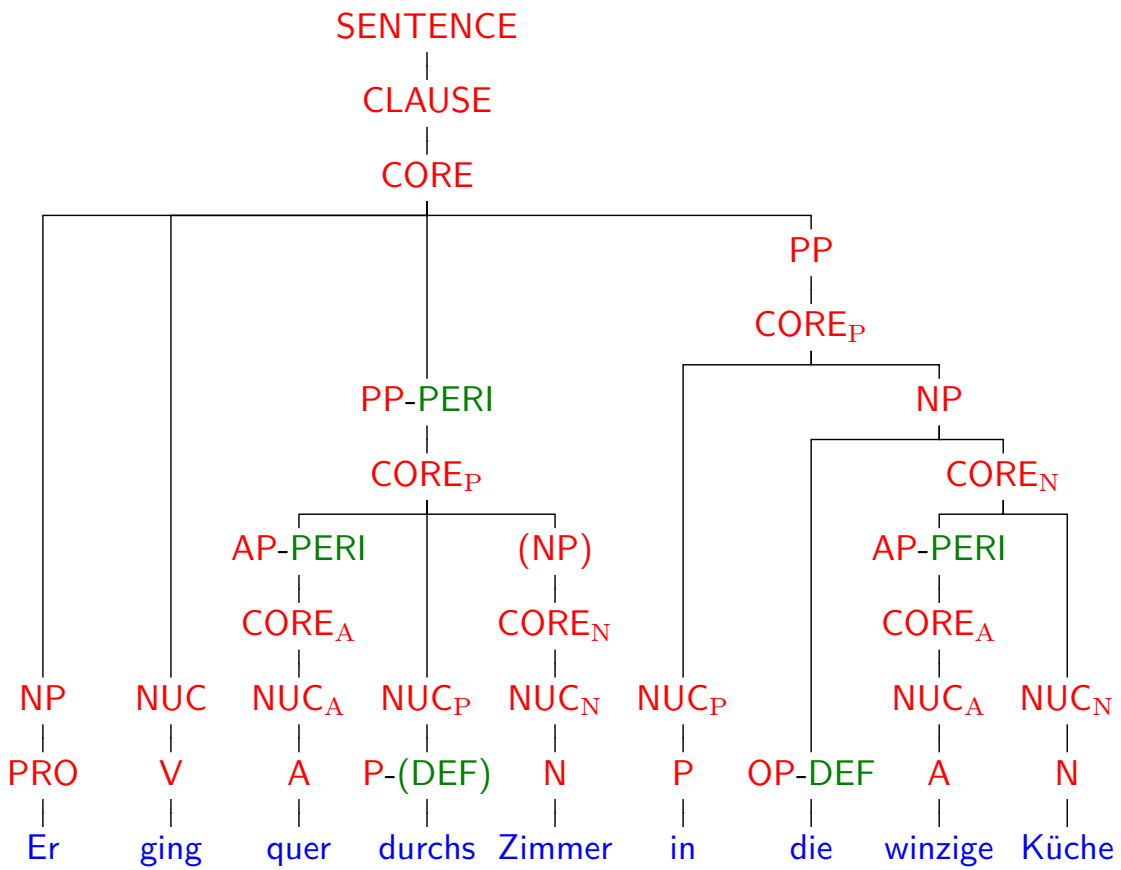
RRGparBank

Some issues concerning the RRG annotation of German:

- Status and scope of definiteness within prepositional phrases headed by a preposition displaying definiteness.
- Is the definite marking on a preposition a clitic, an affix or has it become lexicalized?

(2) a. Er ging quer **durchs** Zimmer.
b. Er wischte **durchs** Schlafzimmer, Esszimmer, Wohnzimmer, ein Gästezimmer und die Küche.
c. Er wischte sowohl **durchs** alte Schlafzimmer als auch das neue Wohnzimmer
d. *Er wischte sowohl **durchs** alte Schlafzimmer als auch neue Wohnzimmer

RRGparBank: "Definite" Prepositions in German



Preposition displaying definiteness marking in German

RRGparBank

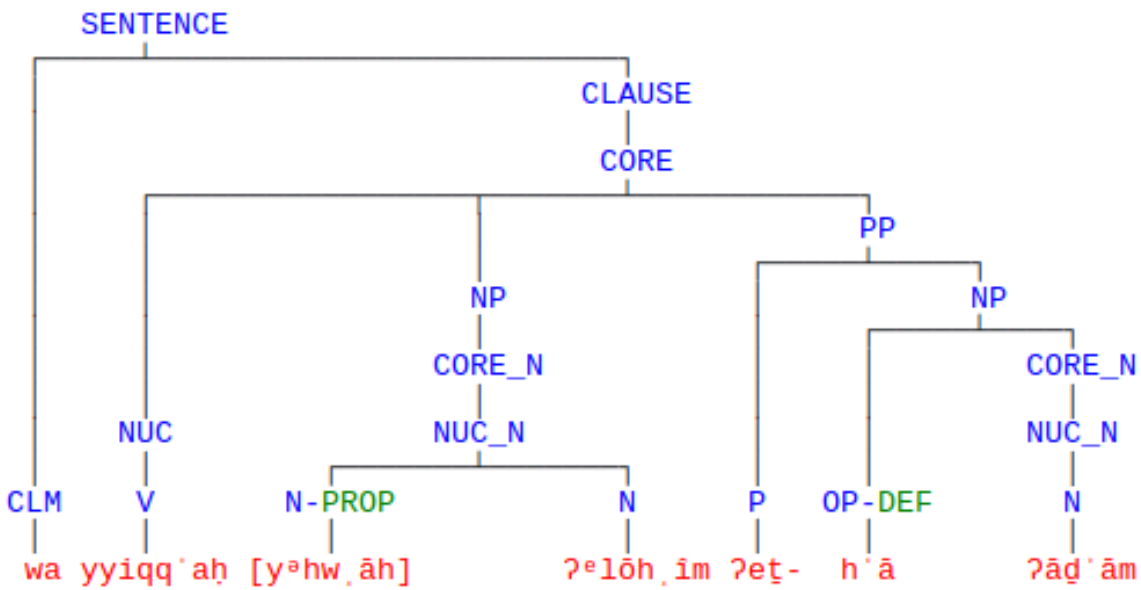
Some issues concerning the RRG annotation of German:

- The analysis of complex verbs, specifically particle verbs
- Differentiating between particles and lexical items in said constructions
- Attachment level of the CLM in (co)subordinated cores

(3) a. Winston **ging** die Treppe **hinauf**
b. Winston **drehte** sich mit einem Ruck **um**
c. Du **nimmst** wahrscheinlich **an**, neue Worte **zu** erfinden

Hebrew Bible Annotation

- ⋆ Collaboration with Nicolai Winther-Nielsen and Christian Canu Højgaard from the Dansk Bibel-Institut
- ⋆ Idea: Annotate sentences of the Hebrew Bible from the BHSA[1] dataset
- ⋆ Input: Constituency trees, in Penn Treebank notation
  → Adapt the ptb2rrg script

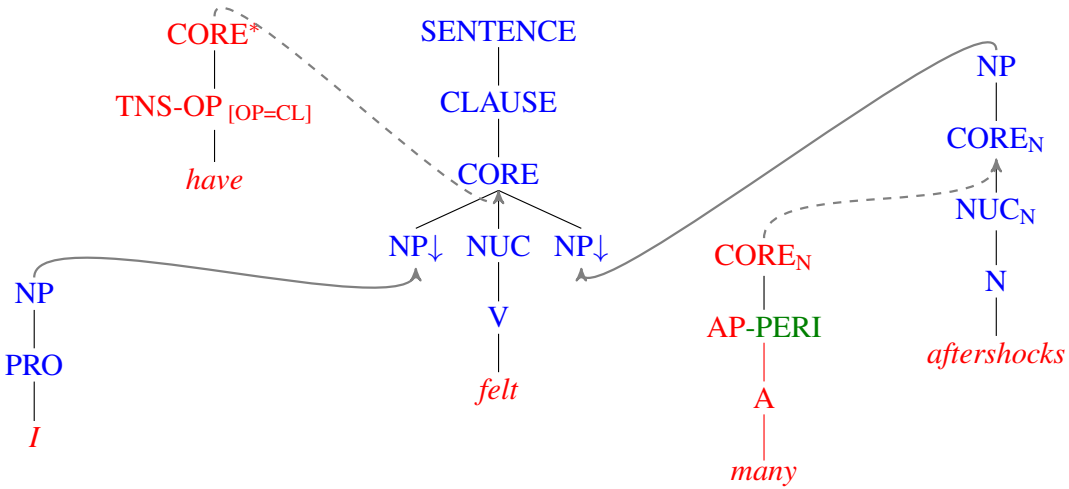


[1] https://etcbc.github.io/bhsa/

Outline

Parsing with RRG Grammars

- ⋆ We follow Kallmeyer et al. (2013) and Osswald and Kallmeyer (2018) in design of the elementary trees in our grammar.

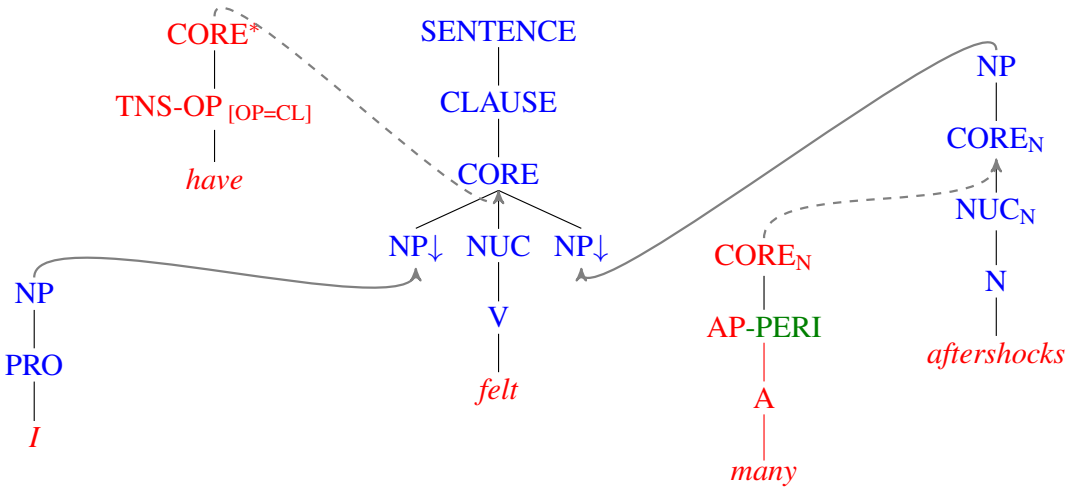


Sentence: *I have felt many aftershocks*,

substitution and sister adjunction.

Parsing with RRG Grammars

- We follow Kallmeyer et al. (2013) and Osswald and Kallmeyer (2018) in design of the elementary trees in our grammar.
- Three tree composition operations:

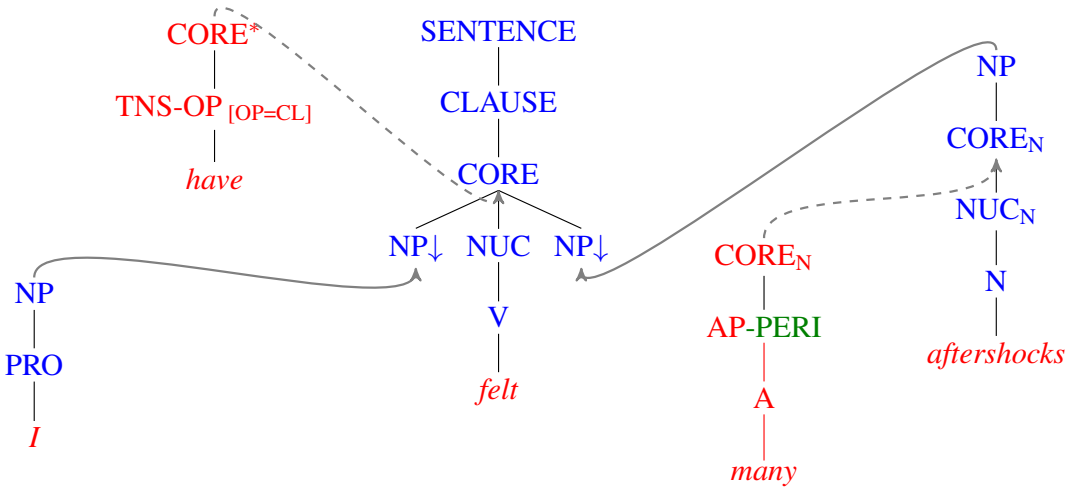


Sentence: *I have felt many aftershocks*,

substitution and sister adjunction.

Parsing with RRG Grammars

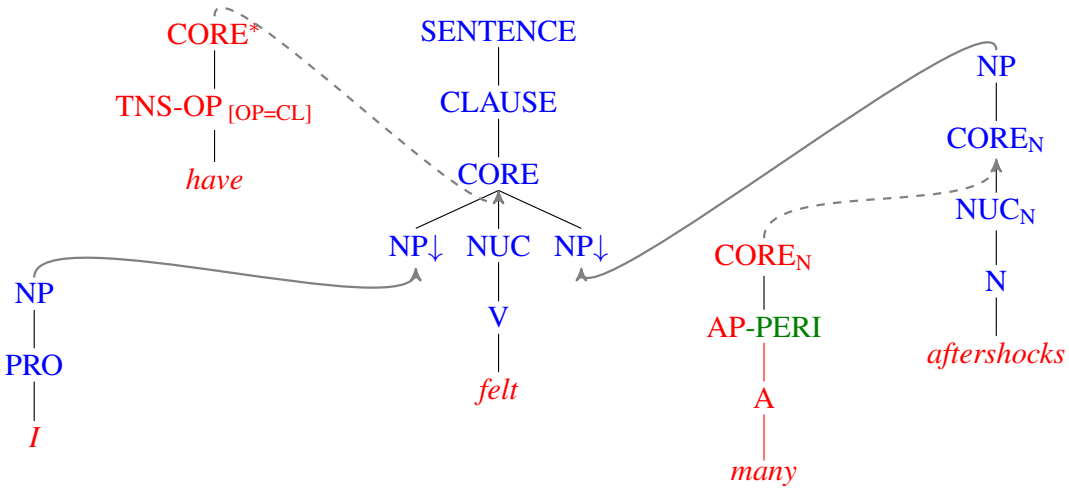
- We follow Kallmeyer et al. (2013) and Osswald and Kallmeyer (2018) in design of the elementary trees in our grammar.
- Three tree composition operations:
  → *substitution*
  (argument slot filling)



Sentence: *I have felt many aftershocks*,
substitution and sister adjunction.

Parsing with RRG Grammars

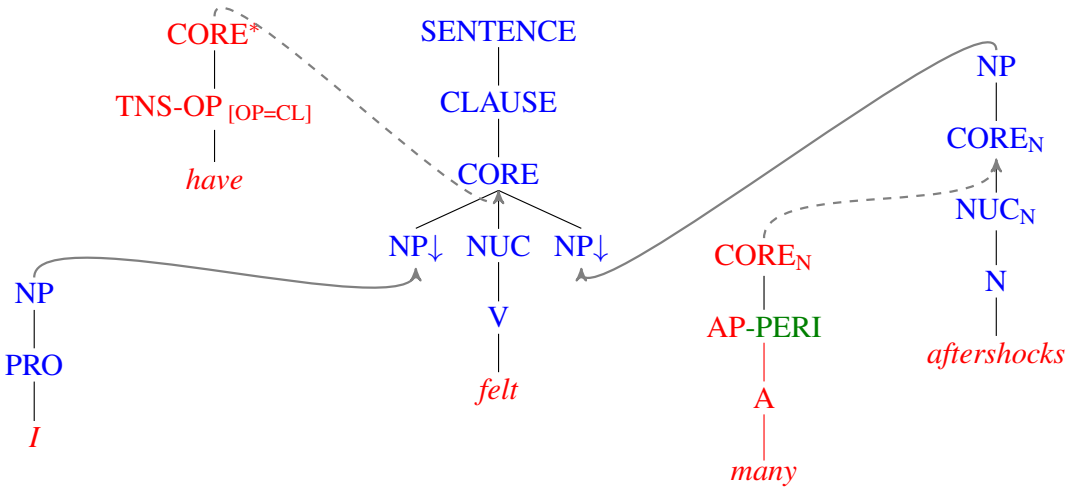
- We follow Kallmeyer et al. (2013) and Osswald and Kallmeyer (2018) in design of the elementary trees in our grammar.
- Three tree composition operations:
  → *substitution* (argument slot filling)
  → *wrapping substitution* (displaced argument slot filling)



Sentence: *I have felt many aftershocks*, substitution and sister adjunction.

Parsing with RRG Grammars

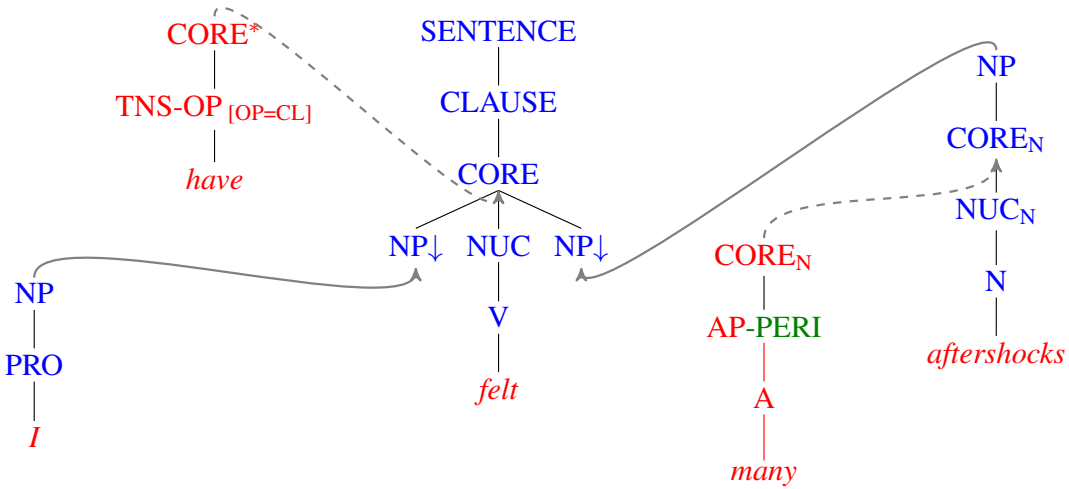
- We follow Kallmeyer et al. (2013) and Osswald and Kallmeyer (2018) in design of the elementary trees in our grammar.
- Three tree composition operations:
  → *substitution*
  (argument slot filling)
  → *wrapping substitution*
  (displaced argument slot filling)
  → *sister adjunction*
  (adding operators and periphery elements);



Sentence: *I have felt many aftershocks*, substitution and sister adjunction.

Parsing with RRG Grammars

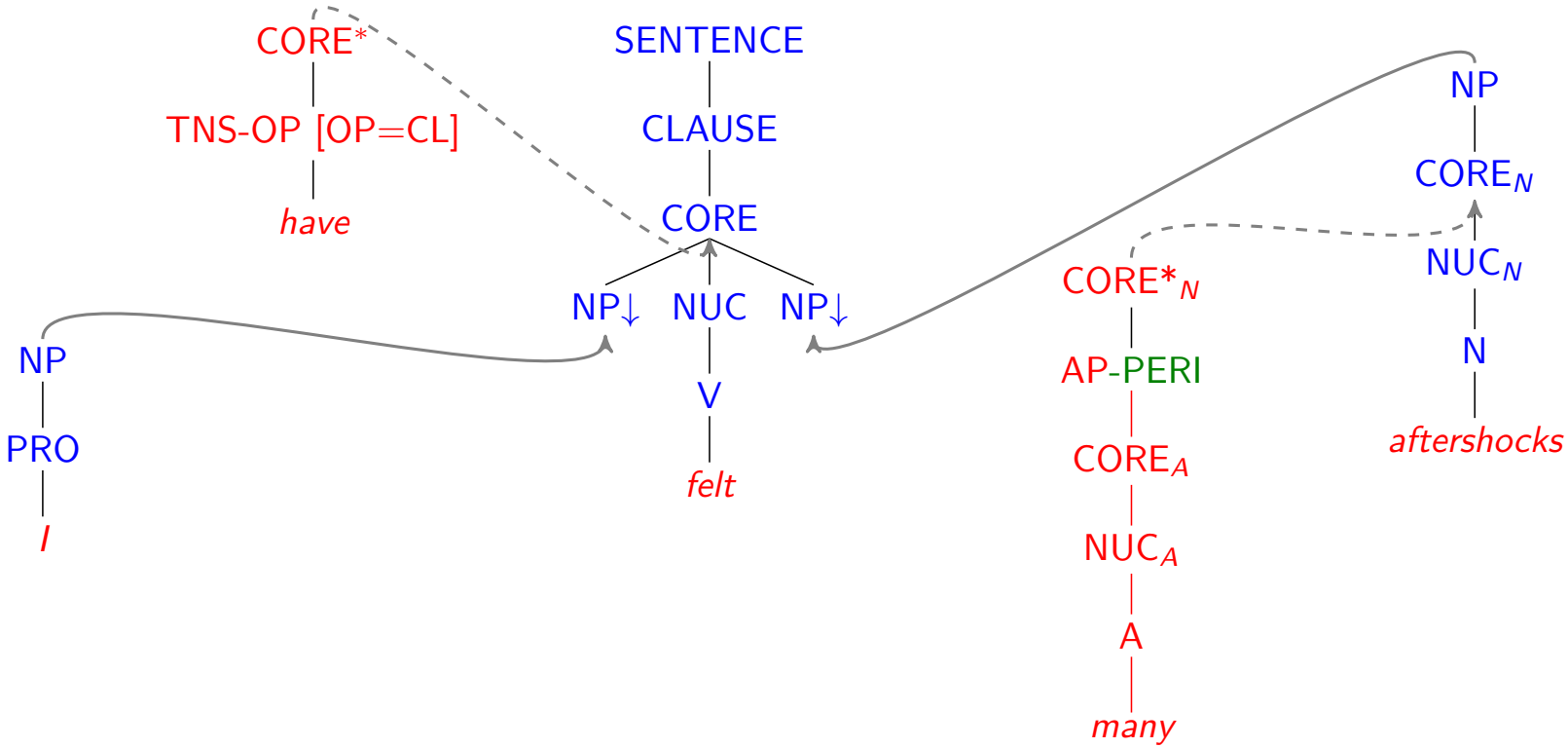
- We follow Kallmeyer et al. (2013) and Osswald and Kallmeyer (2018) in design of the elementary trees in our grammar.
- Three tree composition operations:
  → *substitution* (argument slot filling)
  → *wrapping substitution* (displaced argument slot filling)
  → *sister adjunction* (adding operators and periphery elements);



Sentence: *I have felt many aftershocks*, substitution and sister adjunction.

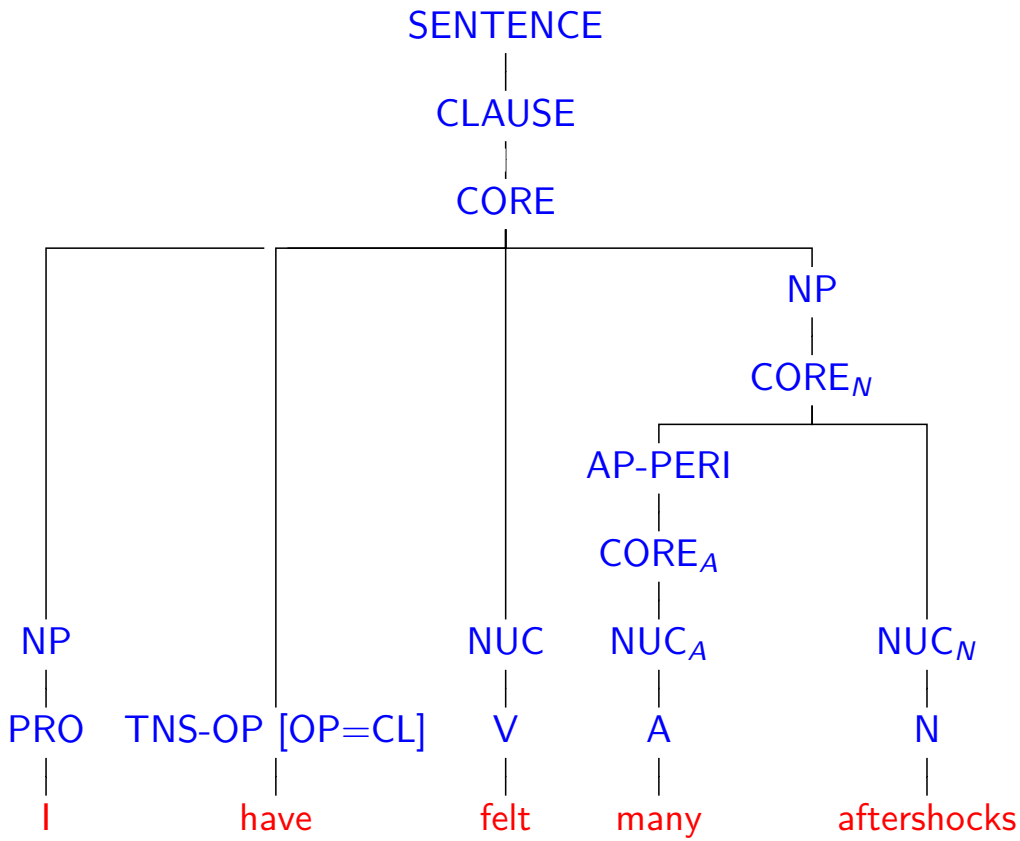
- Such RRG grammars capture long-distance dependencies
  → for example, WH-movement.

Combination operations: Substitution and sister adjunction



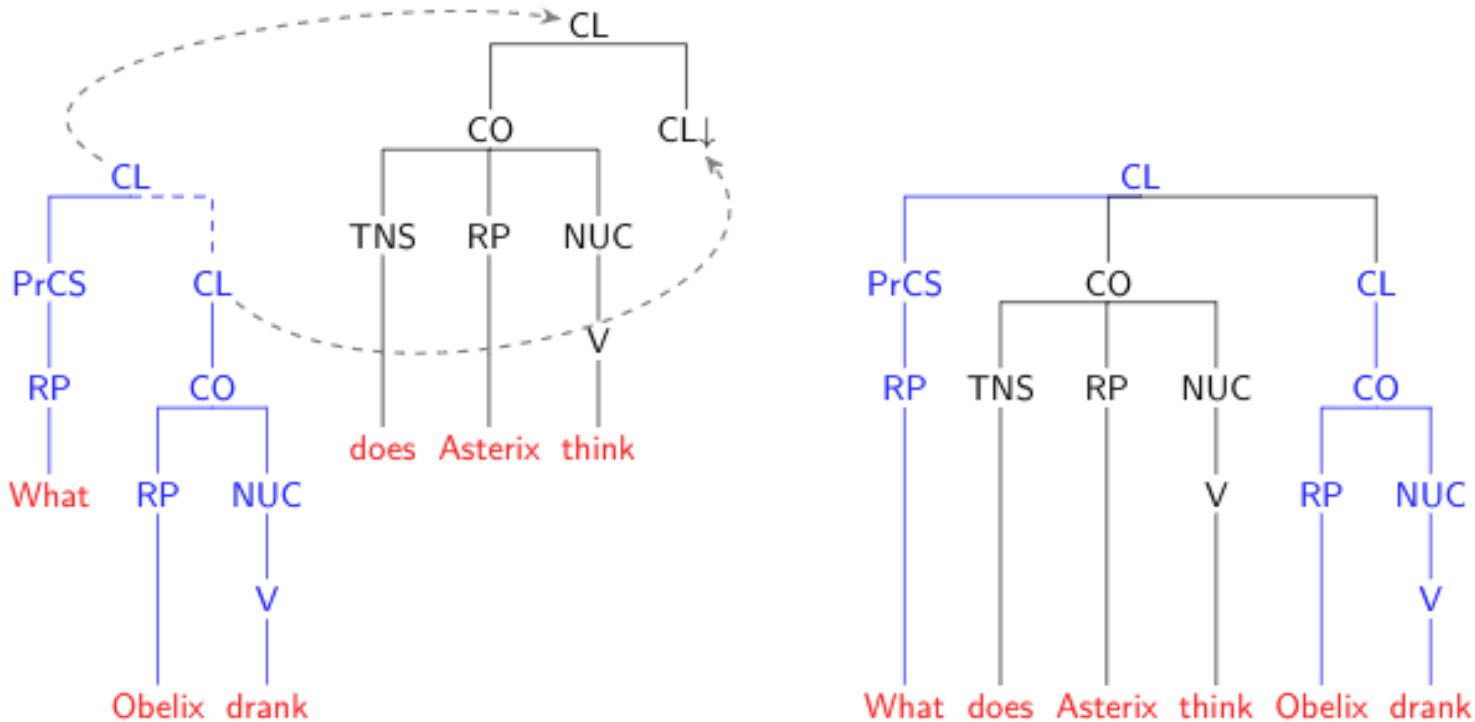
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



Sentence: *I have felt many aftershocks*

Combination operations: Wrapping substitution



Sentence: *What does Asterix think Obelix drank*

Chart-based RRG Parser TuLiPA

- ⋆ input: set of elementary trees and sentences to parse;

Chart-based RRG Parser TuLiPA

- ⋆ input: set of elementary trees and sentences to parse;
- ⋆ output: all derivations that can be derived by combining the elementary trees;

Chart-based RRG Parser TuLiPA

- ⋆ input: set of elementary trees and sentences to parse;
- ⋆ output: all derivations that can be derived by combining the elementary trees;
- ⋆ standard CYK algorithm;

Chart-based RRG Parser TuLiPA

- ⋆ input: set of elementary trees and sentences to parse;
- ⋆ output: all derivations that can be derived by combining the elementary trees;
- ⋆ standard CYK algorithm;
- ⋆ bottom-up, left-to-right traversion of the derived tree;

Chart-based RRG Parser TuLiPA

- ⋆ input: set of elementary trees and sentences to parse;
- ⋆ output: all derivations that can be derived by combining the elementary trees;
- ⋆ standard CYK algorithm;
- ⋆ bottom-up, left-to-right traversion of the derived tree;
- ⋆ software: TuLiPA RRG parser (Arps and Petitjean, 2018) (`https://github.com/spetitjean/TuLiPA-frames`) TuLiPA = Tübingen Linguistic Parsing Architecture;

Chart-based RRG Parser TuLiPA

- ⋆ input: set of elementary trees and sentences to parse;
- ⋆ output: all derivations that can be derived by combining the elementary trees;
- ⋆ standard CYK algorithm;
- ⋆ bottom-up, left-to-right traversion of the derived tree;
- ⋆ software: TuLiPA RRG parser (Arps and Petitjean, 2018) (`https://github.com/spetitjean/TuLiPA-frames`) TuLiPA = Tübingen Linguistic Parsing Architecture;
- ⋆ suitable for hand-crafted precision RRG grammars;

Chart-based RRG Parser TuLiPA

- ⋆ input: set of elementary trees and sentences to parse;
- ⋆ output: all derivations that can be derived by combining the elementary trees;
- ⋆ standard CYK algorithm;
- ⋆ bottom-up, left-to-right traversion of the derived tree;
- ⋆ software: TuLiPA RRG parser (Arps and Petitjean, 2018) (`https://github.com/spetitjean/TuLiPA-frames`) TuLiPA = Tübingen Linguistic Parsing Architecture;
- ⋆ suitable for hand-crafted precision RRG grammars;
- ⋆ suitable for automatically extracted RRG grammars.

Parsing experiments

	Experiment 1	**Experiment 2**
# sentences	395	1480
avg. sentence length	6.1	8.0
token-supertag pairs	1526	6288
avg. number of parses	6.9	1166

- ⋆ The number of possible parses is very high.
- ⋆ A probabilistic parser should help to reduce the number of parses to one most probable
  → e.g. parser ParTAGe by Waszczuk (2017).

Outline

Future Work

- Further conversion of PTB trees and validation of RRG trees.
- ⋆ Further annotation of the German (so far 91 gold, 111 silver sentences), the Hungarian (1 gold sentence;-) and the Russian (no annotations available so far) on "1984" data.
- ⋆ In combination with this: further improvement of UD2RRG for these languages (labeled F1 for German 77.67 on gold).
- ⋆ Data-driven RRG induction:
  - extraction of RRG tree fragments ("supertags") along the lines of our TWG RRG formalization.
  - ⋆ Grammar induction using other formalisms (LCFRS, discontinuous Tree Substitution Grammar).
- ⋆ Statistical parsing with these grammar fragments.

Thank you!

THANK YOU VERY MUCH FOR YOUR ATTENTION!

References I

Arps, D. and Petitjean, S. (2018). A Parser for LTAG and Frame Semantics. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Bladier, T., van Cranenburgh, A., Evang, K., Kallmeyer, L., Möllemann, R., and Osswald, R. (2018). RRGbank: a Role and Reference Grammar Corpus of Syntactic Structures Extracted from the Penn Treebank. In *Proceedings of International Workshop on Treebanks and Linguistic Theory TLT17*, Oslo.

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.

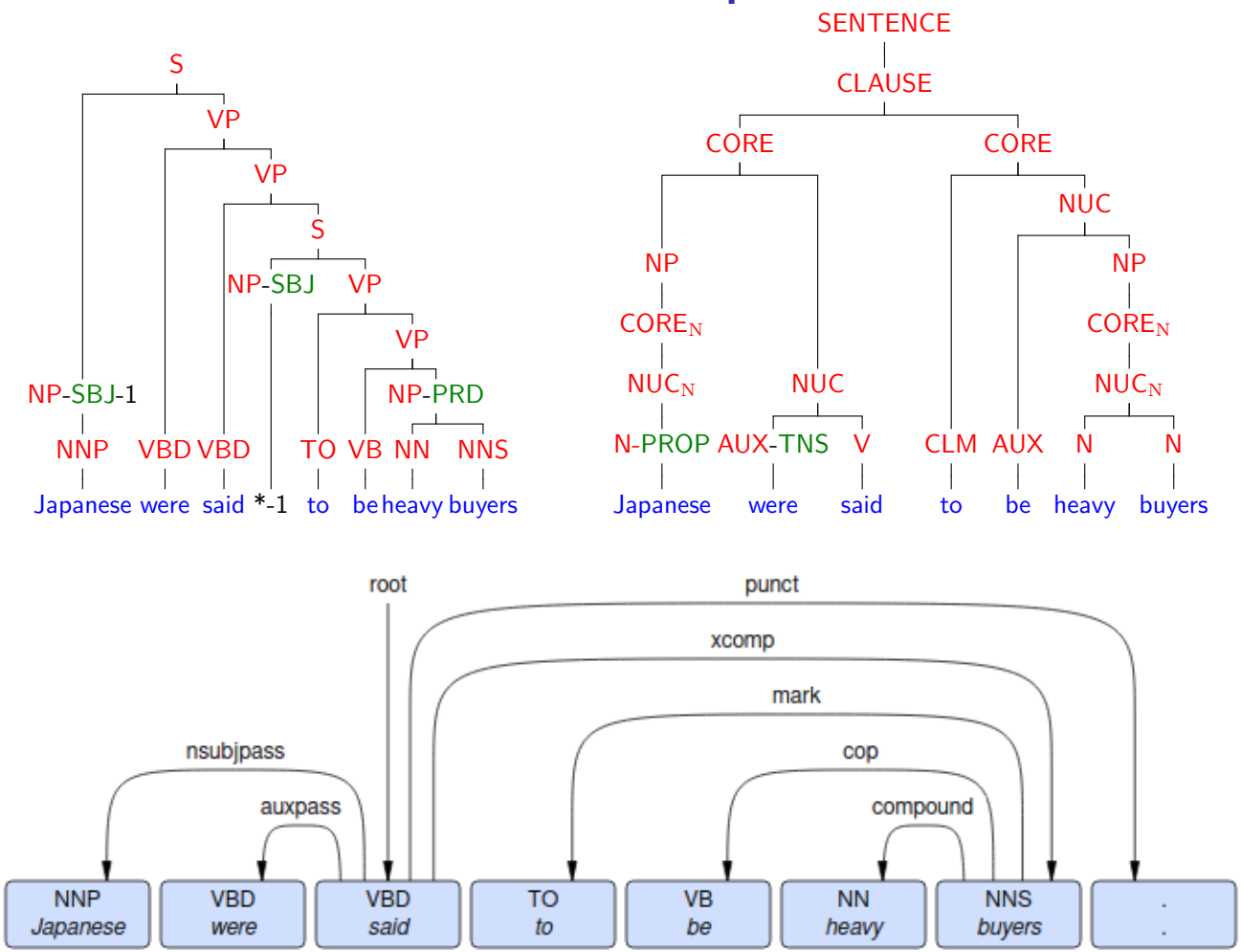
Kallmeyer, L., Osswald, R., and Van Valin, Jr., R. D. (2013). Tree Wrapping for Role and Reference Grammar. In Morrill, G. and Nederhof, M.-J., editors, *Formal Grammar 2012/2013*, volume 8036 of *LNCS*, pages 175–190. Springer.

Osswald, R. and Kallmeyer, L. (2018). Towards a formalization of role and reference grammar. In Kailuweit, R., Künkel, L., and Staudinger, E., editors, *Applying and Expanding Role and Reference Grammar.*, pages 355–378. Albert-Ludwigs-Universität, Universitätsbibliothek. [NIHIN studies], Freiburg.

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

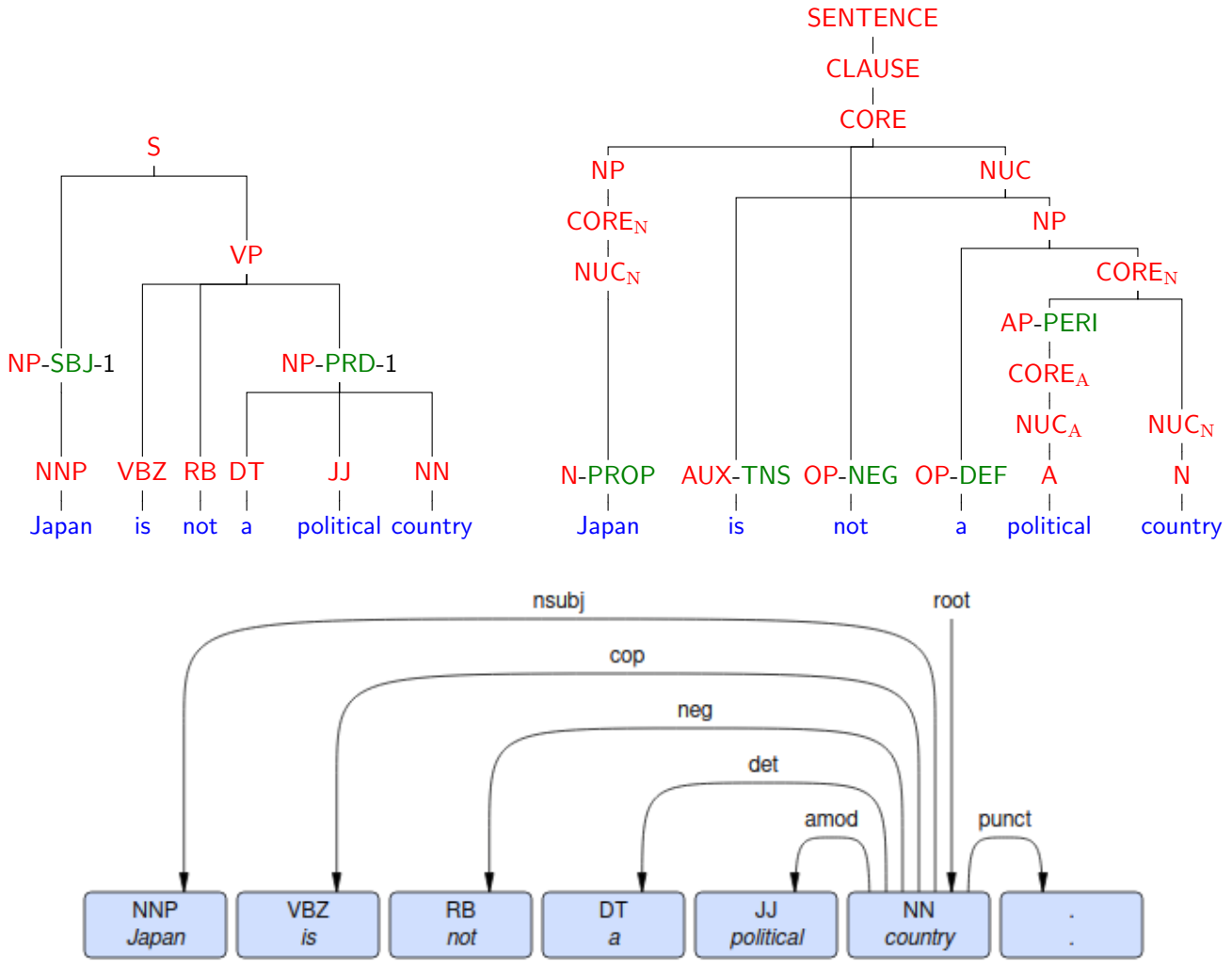
Waszczuk, J. (2017). *Leveraging MWEs in practical TAG parsing: towards the best of the two worlds*. PhD thesis.

PTB2RRG and UD2RRG: Example 1



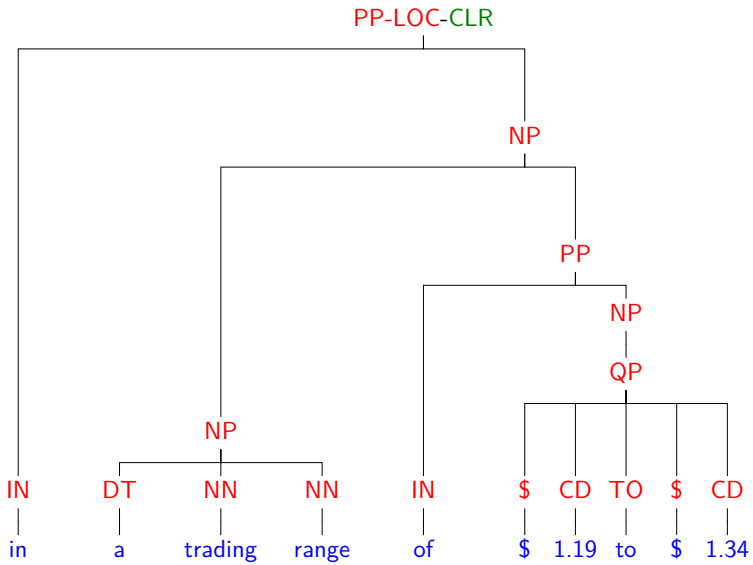
Erroneous annotation in PTB and PTB-UD

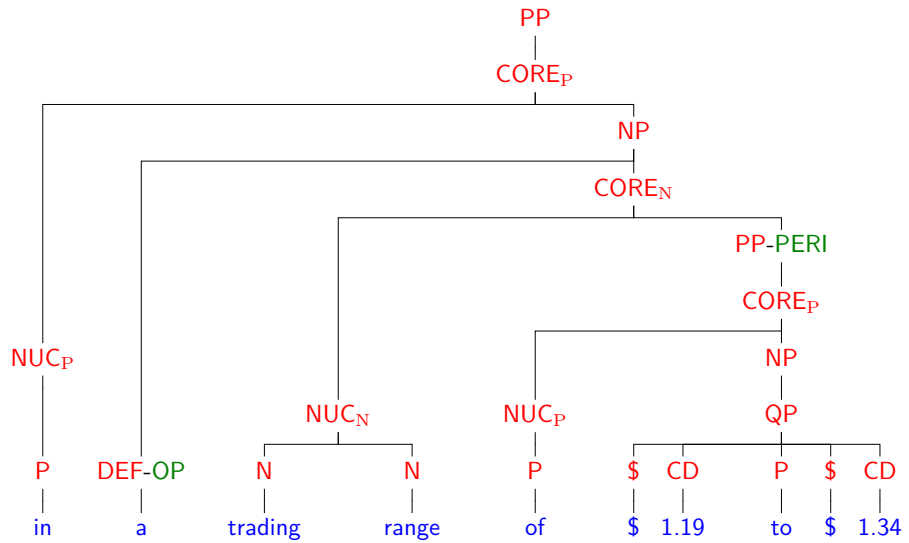
PTB2RRG and UD2RRG: Example 2

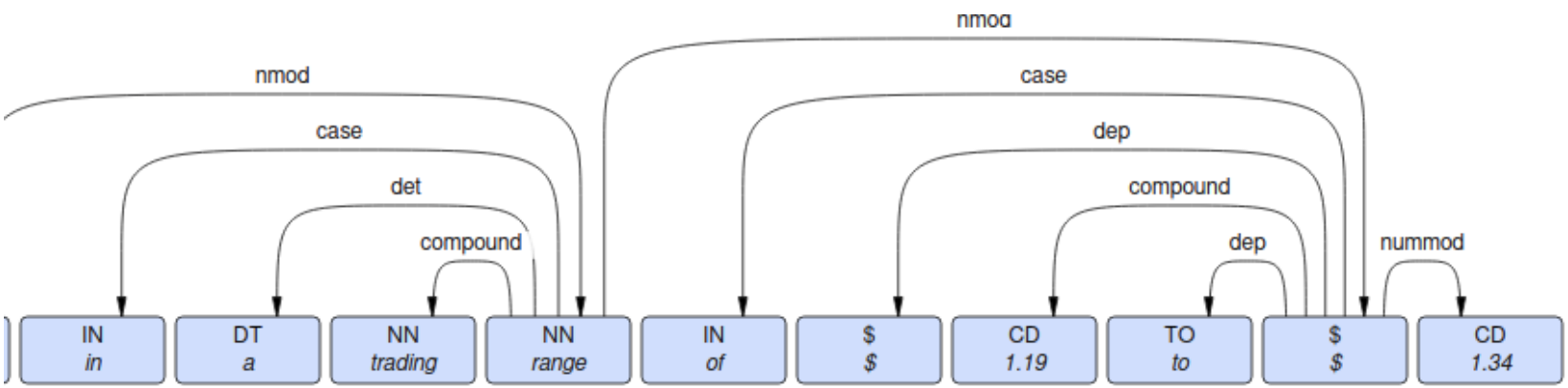


Negation in PTB, PTB-UD, and RRG

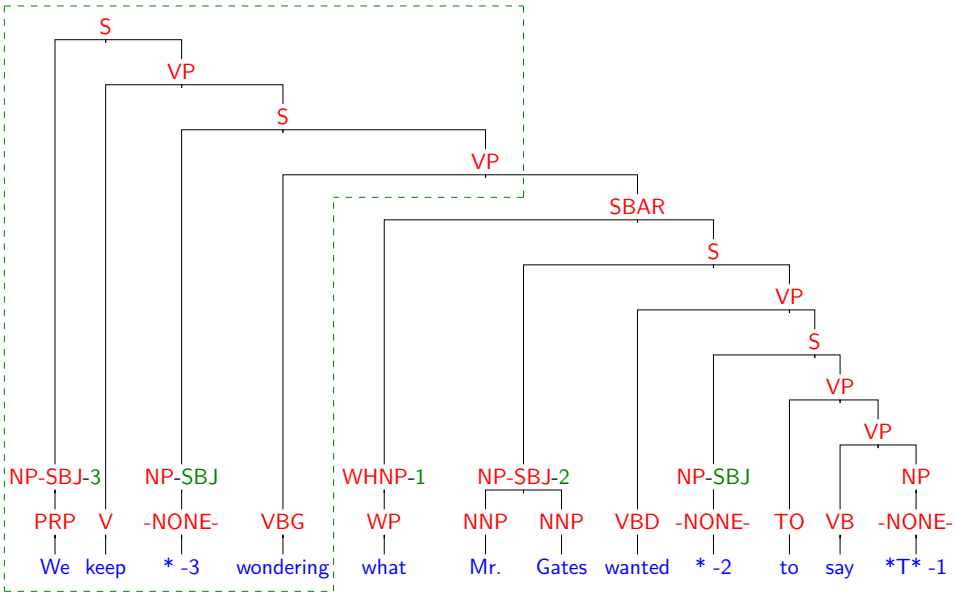
PTB2RRG and UD2RRG: Example 3

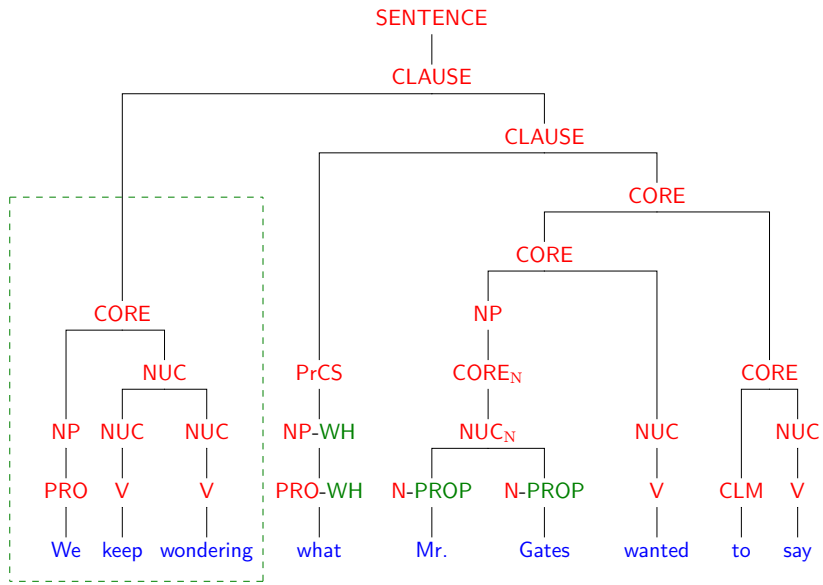


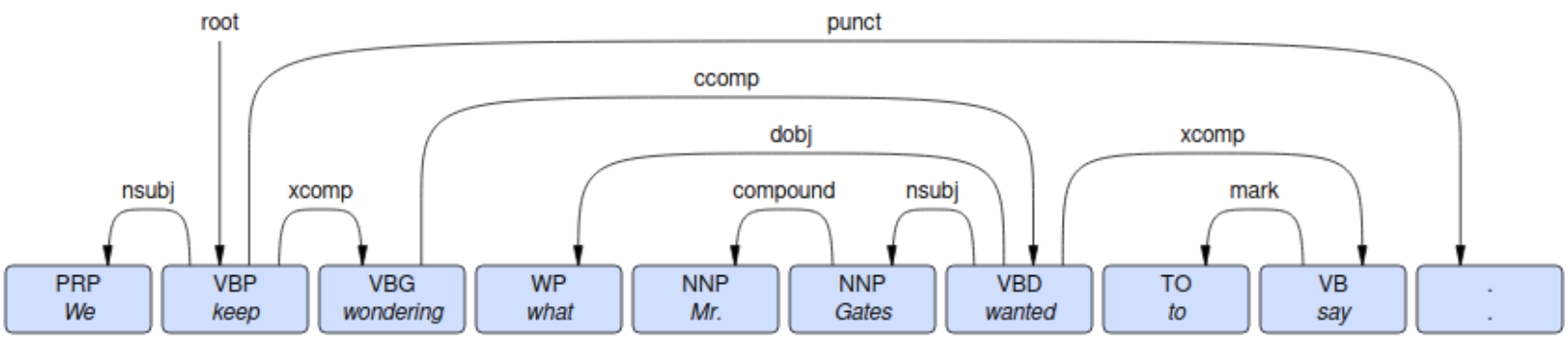




PTB2RRG and UD2RRG: Example 4

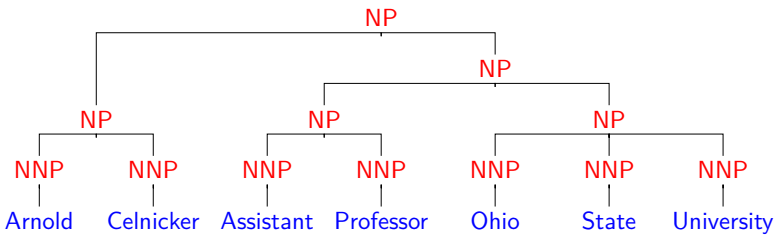


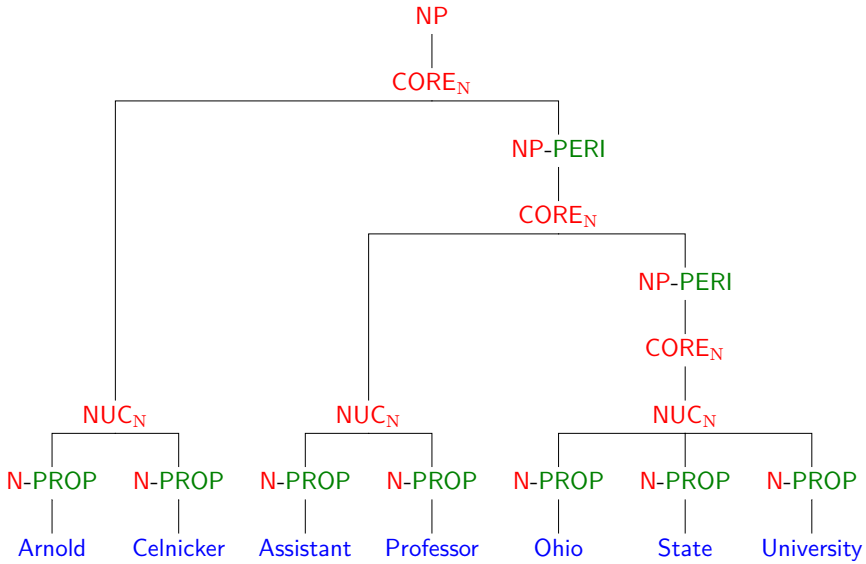


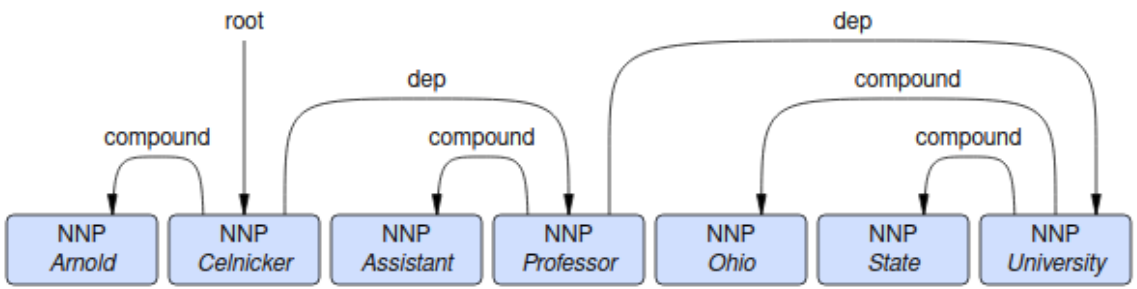


Different junctures

PTB2RRG and UD2RRG: Example 5







Complex NPs with proper nouns