Chart-based RRG parsing using an automatically extracted RRG grammar with features

David Arps & Tatiana Bladier & Laura Kallmeyer

International Conference on Role and Reference Grammar
19 August 2019
University at Buffalo, The State University of New York

Overview

Design of RRG Grammars

Automatic RRG Grammar Extraction

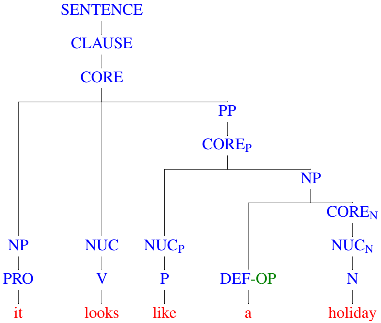
Parsing experiments

Issues

Summary & Outlook

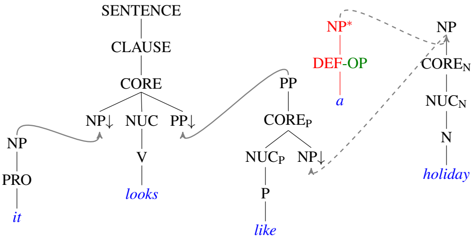
Introduction

1. RRG annotated treebank: RRGBank [2],
2. extract RRG elementary trees automatically,
3. use these elementary trees for RRG parsing.



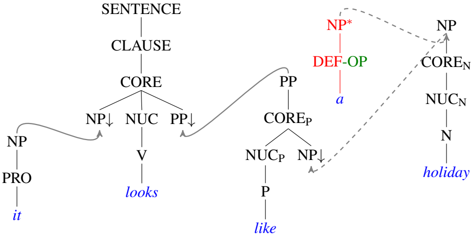
Introduction

1. RRG annotated treebank: RRGBank [2],
2. extract RRG elementary trees automatically,
3. use these elementary trees for RRG parsing.



Introduction

1. RRG annotated treebank: RRGBank [2],
2. extract RRG elementary trees automatically,
3. use these elementary trees for RRG parsing.



Outline

Design of RRG Grammars

Automatic RRG Grammar Extraction

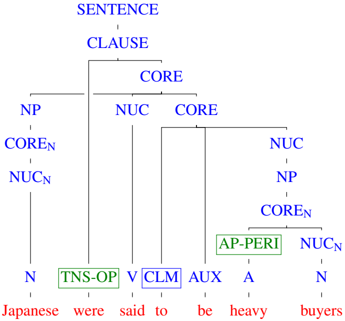
Parsing experiments

Issues

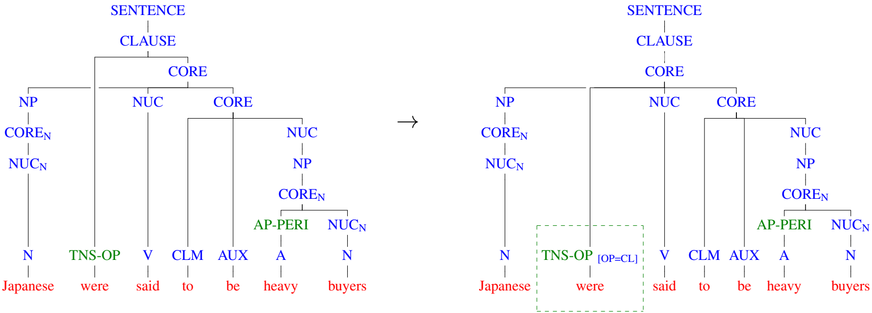
Summary & Outlook

RRGbank

- ⋆ Corpus of RRG annotated sentences [2]
  → automatically converted from Penn Treebank,
  → manually checked and validated;
- ⋆ 395 gold sentences, 1090 silver annotated sentences;
- ⋆ RRGbank and RRG annotation tool: rrgbank.phil.hhu.de.

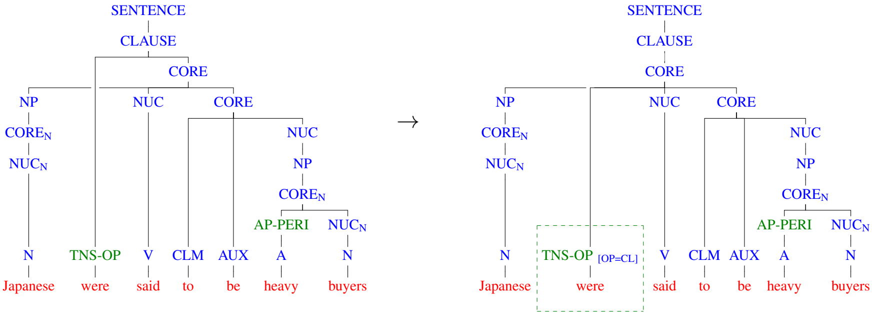


Removing crossing branches



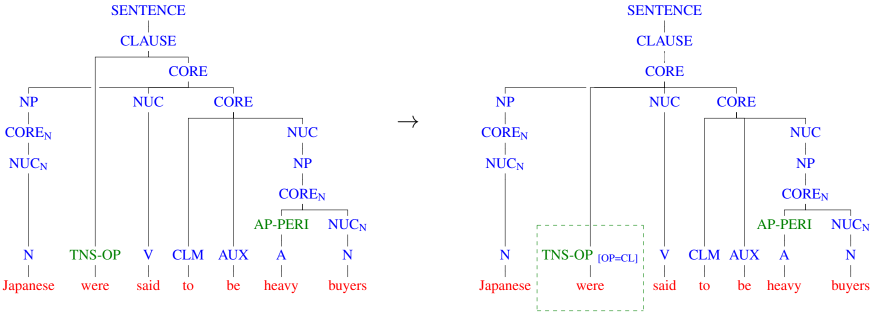
⋆ we transform the RRG structures to remove crossing branches,

Removing crossing branches



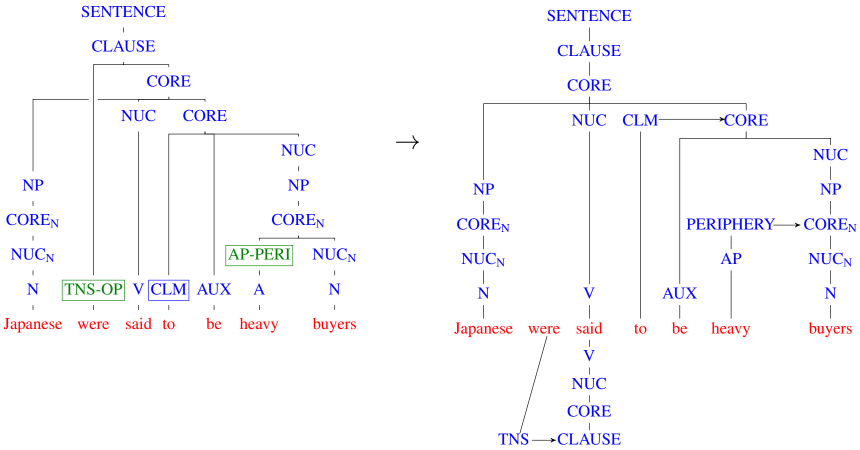
- ⋆ we transform the RRG structures to remove crossing branches,
- ⋆ we mark the original position of the node with [OP=CL],

Removing crossing branches



- ⋆ we transform the RRG structures to remove crossing branches,
- ⋆ we mark the original position of the node with [OP=CL],
- ⋆ original tree structure is easily recovered.

Operator projection and periphery can be recovered



Elementary trees in RRG Grammars

⋆ We follow Kallmeyer et al. (2013) and Osswald & Kallmeyer (2018) [3, 4] in design of the elementary trees in our grammar.

Elementary trees in RRG Grammars

- ⋆ We follow Kallmeyer et al. (2013) and Osswald & Kallmeyer (2018) [3, 4] in design of the elementary trees in our grammar.
- ⋆ Three tree composition operations:

Elementary trees in RRG Grammars

- ⋆ We follow Kallmeyer et al. (2013) and Osswald & Kallmeyer (2018) [3, 4] in design of the elementary trees in our grammar.

- ⋆ Three tree composition operations:
  → *substitution* (argument slot filling)

Elementary trees in RRG Grammars

- ⋆ We follow Kallmeyer et al. (2013) and Osswald & Kallmeyer (2018) [3, 4] in design of the elementary trees in our grammar.

- ⋆ Three tree composition operations:
  → *substitution* (argument slot filling)
  → *wrapping substitution* (displaced argument slot filling)

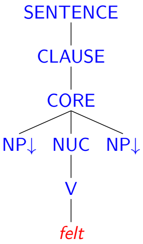
Elementary trees in RRG Grammars

- ⋆ We follow Kallmeyer et al. (2013) and Osswald & Kallmeyer (2018) [3, 4] 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);

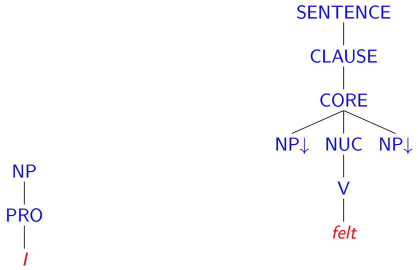
- ⋆ Such RRG grammars capture long-distance dependencies
  → for example, WH-movement.

Combination operations: Substitution and sister adjunction



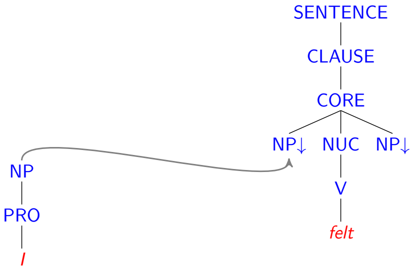
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



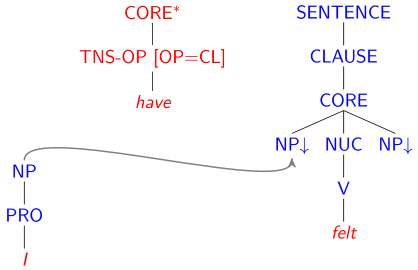
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



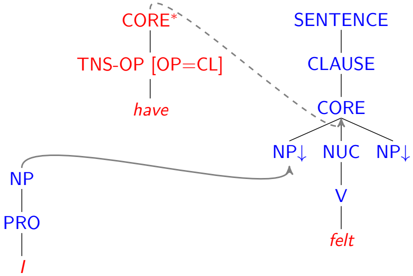
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



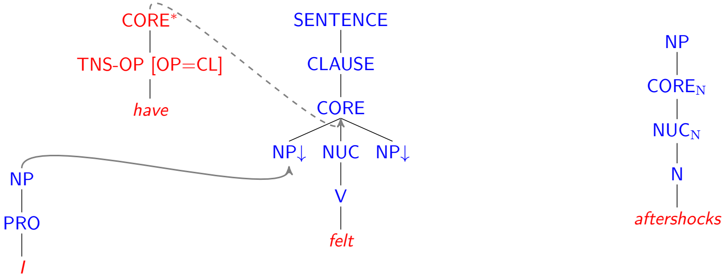
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



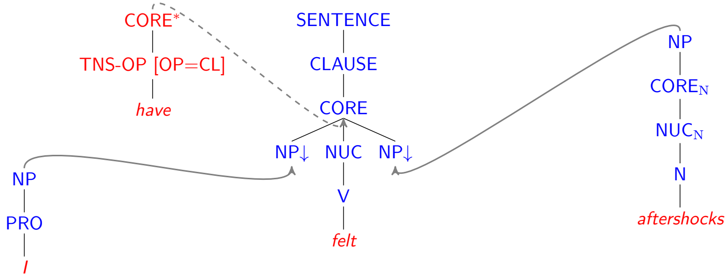
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



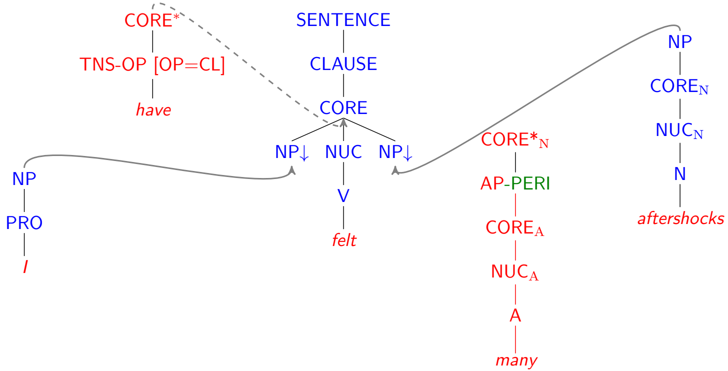
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



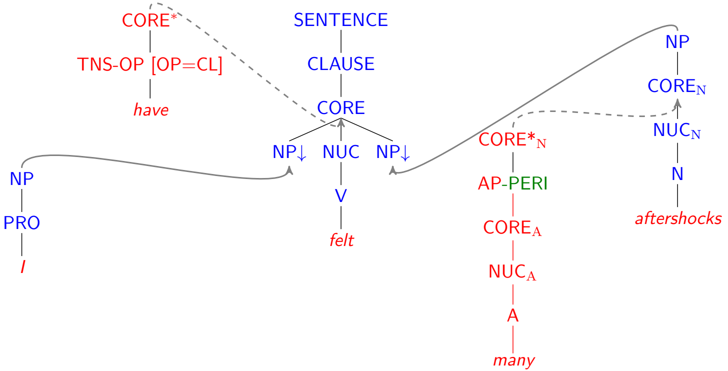
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



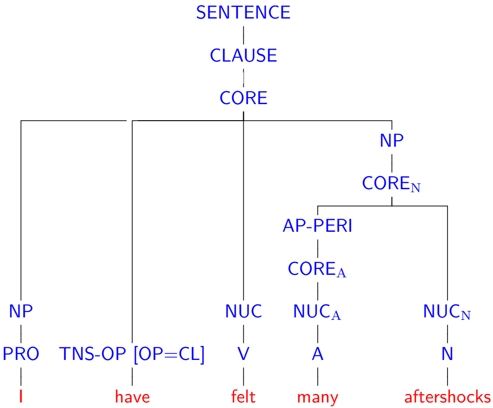
Sentence: *I have felt many aftershocks*

Combination operations: Substitution and sister adjunction



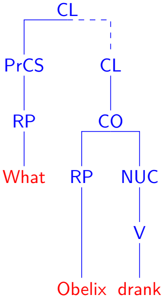
Sentence: *I have felt many aftershocks*

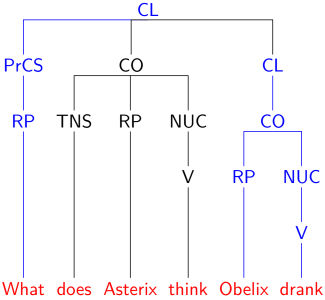
Combination operations: Substitution and sister adjunction



Sentence: *I have felt many aftershocks*

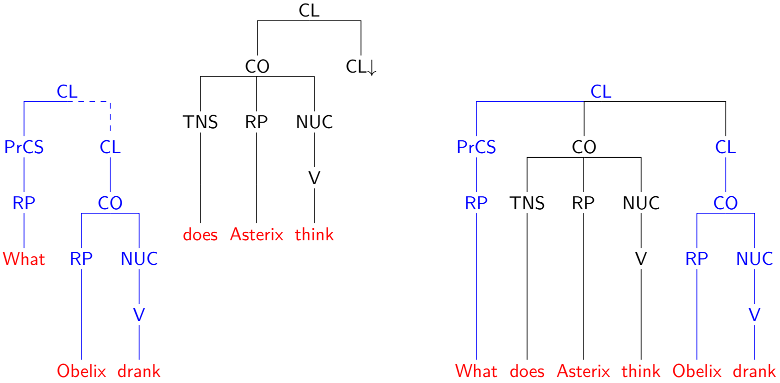
Combination operations: Wrapping substitution





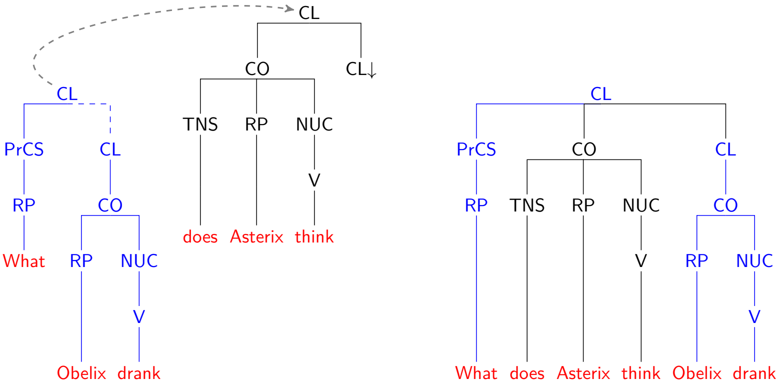
Sentence: *What does Asterix think Obelix drank*

Combination operations: Wrapping substitution



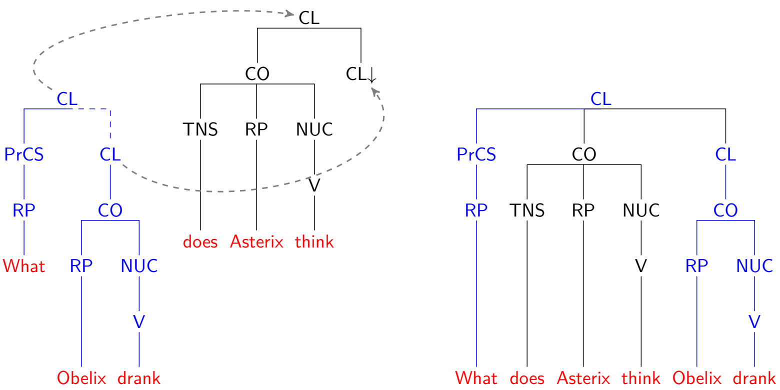
Sentence: *What does Asterix think Obelix drank*

Combination operations: Wrapping substitution



Sentence: *What does Asterix think Obelix drank*

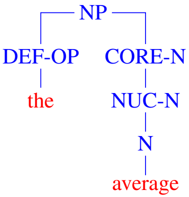
Combination operations: Wrapping substitution



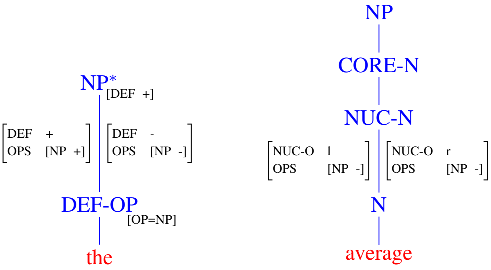
Sentence: *What does Asterix think Obelix drank*

Elementary trees with features

RRGbank

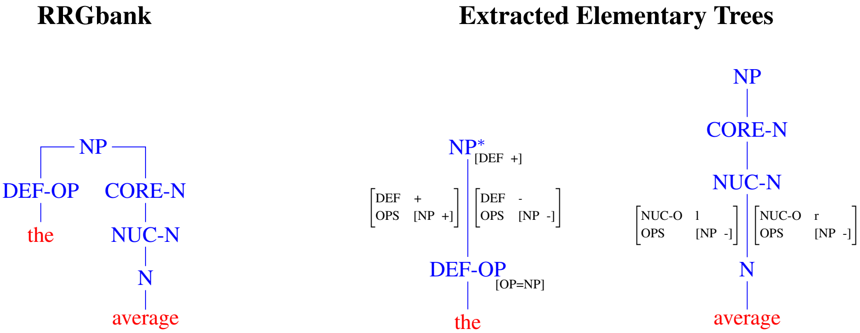


Extracted Elementary Trees



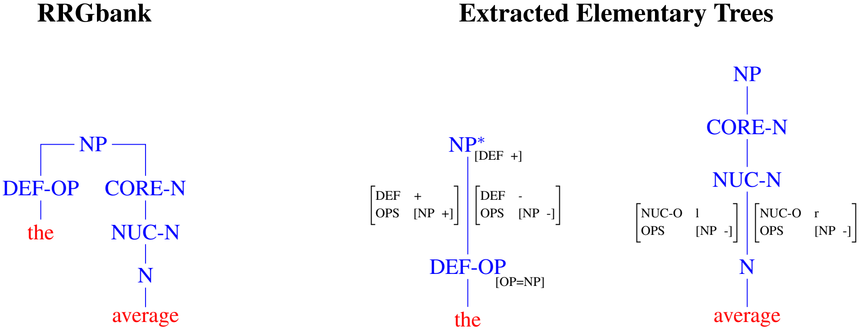
- ⋆ elementary trees are enhanced with features

Elementary trees with features



- ⋆ elementary trees are enhanced with features
  → edge features

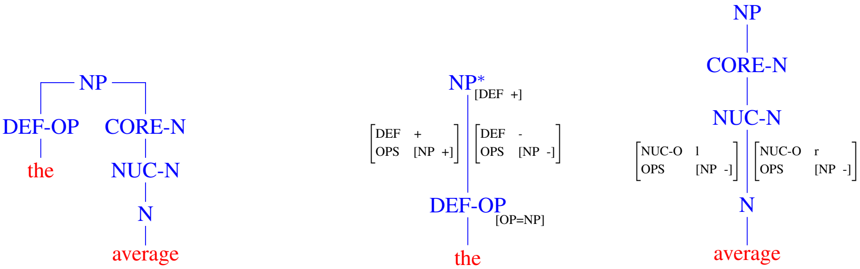
Elementary trees with features



- ⋆ elementary trees are enhanced with features
  - → edge features
  - → node features

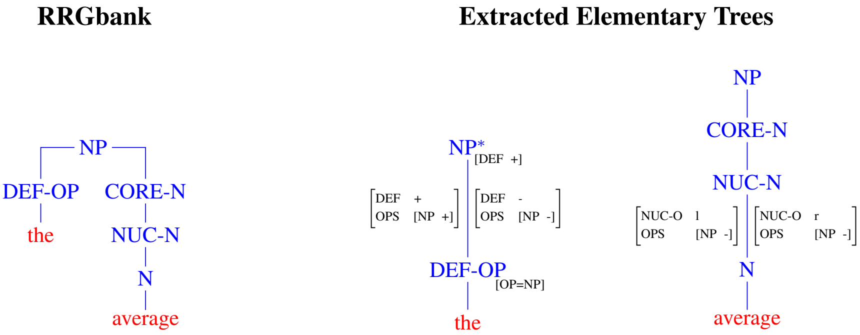
Edge features

RRGbank **Extracted Elementary Trees**



Left and right edge feature structures:

Edge features

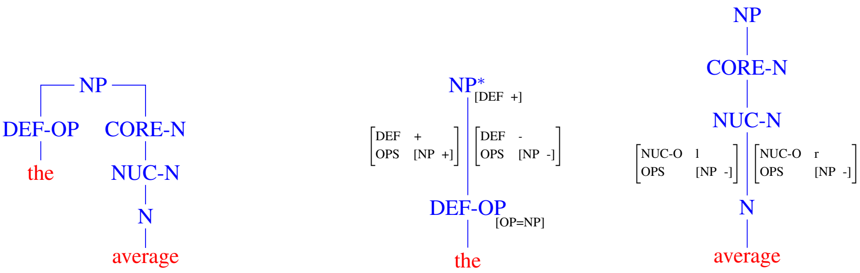


Left and right edge feature structures:

- ⋆ unify adjacent structures in the derived tree

Edge features

RRGbank **Extracted Elementary Trees**

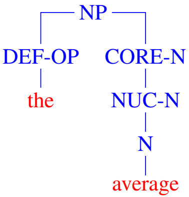


Left and right edge feature structures:

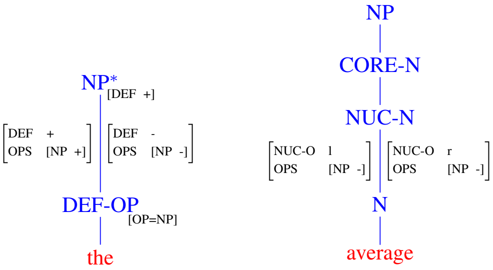
- ⋆ unify adjacent structures in the derived tree
- ⋆ model ordering constraints

Edge features

RRGbank



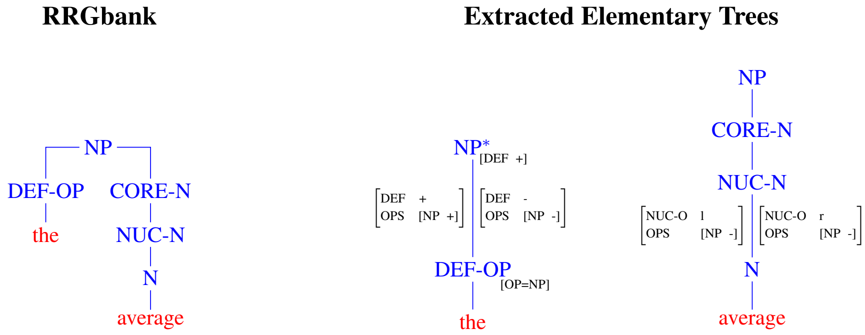
Extracted Elementary Trees



Left and right edge feature structures:

- ⋆ unify adjacent structures in the derived tree
- ⋆ model ordering constraints
- ⋆ percolate upwards until phrasal nodes

Node features



⋆ **One feature structure per node:**

- unify during tree composition
- store syntactic or syn-sem interface information.

⋆ Unification successful → accept parse tree

Outline

Design of RRG Grammars

Automatic RRG Grammar Extraction

Parsing experiments

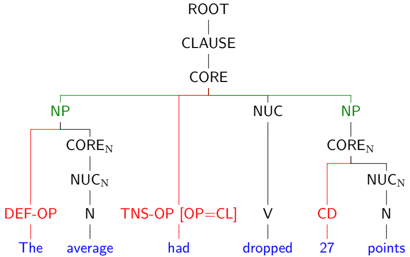
Issues

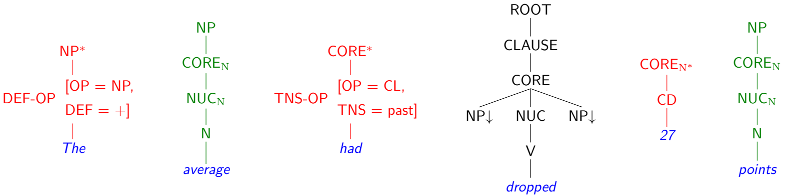
Summary & Outlook

RRG Grammar extraction algorithm (1)

- ⋆ Elementary tree extraction inspired by Xia [6] algorithm for induction of Tree-Adjoining Grammars.
- ⋆ Top-down extraction of elementary trees.
- ⋆ Heuristics from head-modifier percolation tables.
- ⋆ We use RRG structures from RRGbank for automatic grammar induction
  → `rrgbank.phil.hhu.de`.

RRG Grammar extraction algorithm (2)





Outline

Design of RRG Grammars

Automatic RRG Grammar Extraction

Parsing experiments

Issues

Summary & Outlook

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 [1]
  (`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 [1]
  (`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 [1] (`https://github.com/spetitjean/TuLiPA-frames`) TuLiPA = Tübingen Linguistic Parsing Architecture;
- ⋆ suitable for hand-crafted precision RRG grammars;
- ⋆ suitable for automatically extracted RRG grammars.

Extracted RRG grammar

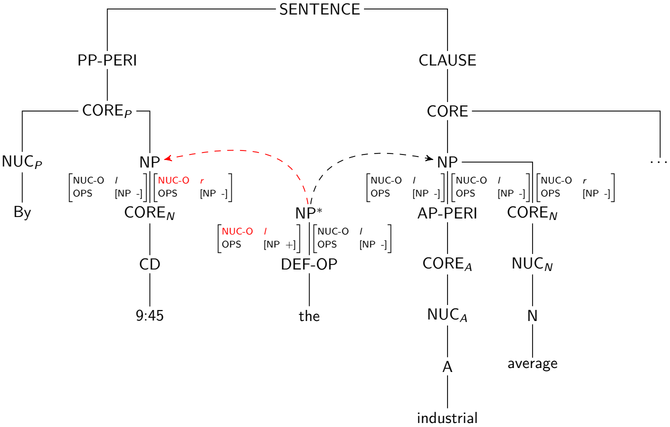
- removed punctuation
- exhaustive parsing (i.e. not probabilistic, overgenerating a lot)
- 2 versions:
  1. no features
  2. edge features for operators model adjunction constraints
- do feature structures eliminate parse trees that contradict linguistic intuitions?

Parsing experiments

	Gold Grammar		Silver Grammar	
	w feats	w/o feats	w feats	w/o feats
Sentences	395		1480	
avg. sentence length	6.1		8.0	
token-supertag pairs	1526	1497	6288	6044
avg. number of parses	6.9	12.7	1166	2939
savings	45.1%		39.7%	

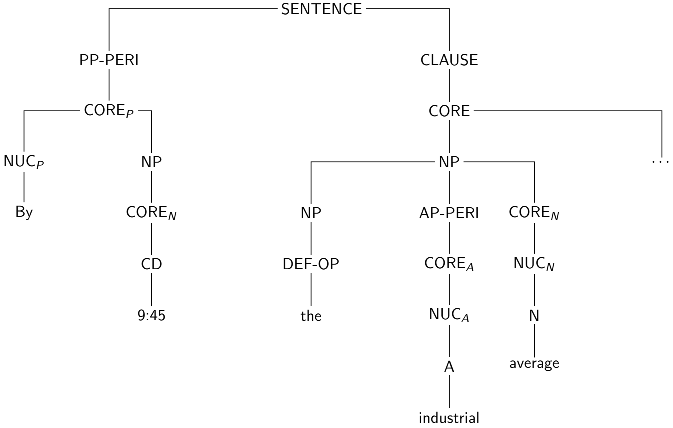
features decrease number of results by ≈ 45%

Features restrict adjunction of operators and periphery (1)



Sentence: *By 9:45, the industrial average had dropped 27 points.*

Features restrict adjunction of operators and periphery (2)



Sentence: *By 9:45, the industrial average had dropped 27 points.*

Outline

Design of RRG Grammars

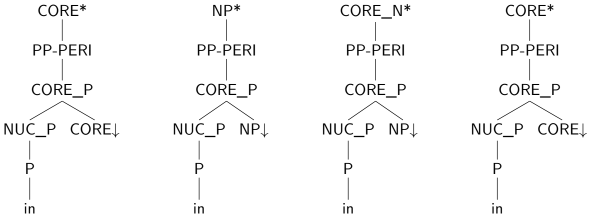
Automatic RRG Grammar Extraction

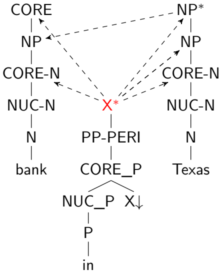
Parsing experiments

Issues

Summary & Outlook

Attachment ambiguities





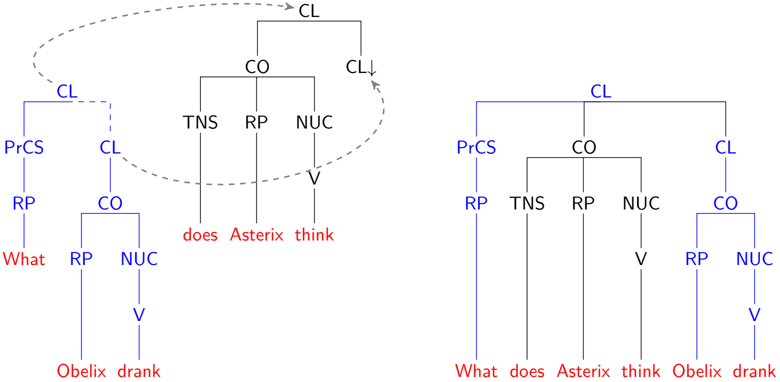
syntactic information needed that might not be in RRGBank

Size of the grammar

	Gold Grammar		Silver Grammar	
	w feats	w/o feats	w feats	w/o feats
Sentences	395		1480	
avg. sentence length	6.1		8.0	
token-supertag pairs	1526	1497	6288	6044
avg. number of parses	6.9	12.7	1166	2939

the number of parses per sentence increases with the size of the grammar

Extraction of wrapping substitution trees



- ⋆ discontinuous constituents are marked with traces in PTB;
- ⋆ no special marking in RRGBank;
- ⋆ transfer traces from PTB to RRG trees in RRGbank?

Outline

Design of RRG Grammars

Automatic RRG Grammar Extraction

Parsing experiments

Issues

Summary & Outlook

Summary

- ⋆ automatically extracted elementary trees from RRGBank;
- ⋆ experiments with exhaustive parsing of 395/1480 sentences;
- ⋆ parsing w/o edge features → too many results;
- ⋆ some edge features already rule out bad results.

Future Work

- ⋆ adding more edge features = better results;
- ⋆ extract wrapping elementary trees;
- ⋆ 100s of results → not satisfying;
- ⋆ ambiguity and annotation/extraction mistakes have bad consequences;
- ⋆ use unlexicalized elementary trees (= supertags);
- ⋆ probabilistic grammar and parsing
  → A* parsing algorithm ParTAGe by Waszczuk (2017) [5];
- ⋆ Web GUI.

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 the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018), December 13–14, 2018, Oslo University, Norway*, number 155, pages 5–16. Linköping University Electronic Press.

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.

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

Xia, F. (1999). Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-99)*, pages 398–403.