Deep DARE

German Research Center for Artificial Intelligence (DFKI)
Language Technology Lab

Peter Adolphs, Feiyu Xu, Hans Uszkoreit, Li Hong

Presenter: Yi Zhang

1) Introduction
2) DARE
3) Parsers
4) Graph Rules
5) Experiments
6) Conclusions

1) Introduction

2) DARE

3) Parsers

4) Graph Rules

5) Experiments

6) Conclusions

Introduction – What is Relation Extraction?

- Task: find mentions of specific semantic relations between entities in raw text
- Example relations:
  birthplace, marriage, management succession, prize winning, …
- Approximation of full natural language understanding: focussed on a limited set of relevant semantic relations

„Investigation“ by Paul Vladuchick

Introduction – DARE

- DARE: Domain Adaptive Relation Extraction (Xu, 2008)
- Especially suited for relations with higher arity
- Learns relation extraction rules from raw text
- Bootstrapping framework
- Minimally supervized

- Related Work:
  - Adapted from DIPRE (Brin, 1998) and Snowball (Agichtein & Gravano, 2000)
  - Extended and enriched with linguistic analysis

Introduction – Linguistic Challenges

- 38.2% of extraction errors in Xu (2008) are due to errors made by the dependency parser (MINIPAR)
- More detailed analyses are required for recognizing and treating valency-changing operations on the semantics (modality, negation, reports, and their scopal interaction)
- Goal: increase precision and analysis depth by using deep NLP methods (HPSG)

„ Varios Series - 72“ by Jef Harris

1) Introduction

2) DARE

3) Parsers

4) Graph Rules

5) Experiments

6) Conclusions

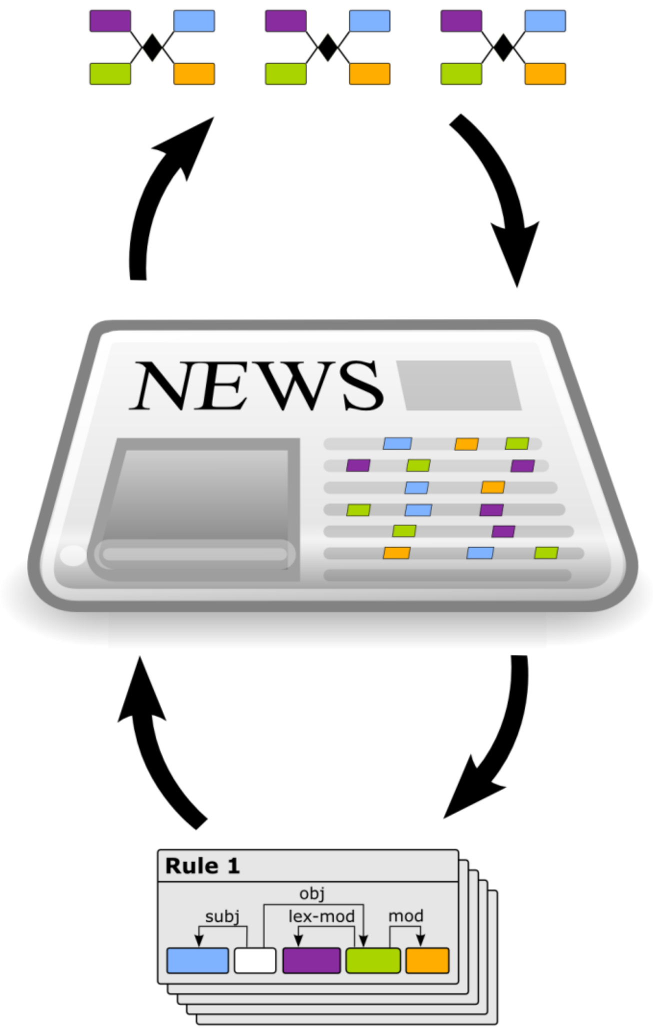
DARE – Overview

- Requirements:
  - A target relation
  - Some instances of the target relation → „semantic seeds“
  - Corpus with named-entity and parsing annotations
- Two main phases:
  - **Rule learning**: relation extraction rules are learned from free text
  - **Relation extraction**: relation extraction rules are used to extract new relation instances

DARE – Overview Rule Learning

- For each seed, find sentences mentioning its arguments
- Find subtrees connecting the seed's arguments
- Postulate rules for the relation and its projections by generalizing the subtrees
- Optionally rank and filter the rules according to their complexity and productivity

- Use learned relation extraction rules to extract relation instances
- Bootstrapping:
  - Use new relation instances to learn further rules
  - Continue until a fixpoint is reached
- Once learned, rules can be used for relation extraction on running text



- Relation extraction rules represented with feature structures
- Components:
  - Rule name
  - Rule body: the actual structure to be matched
  - Output: the target semantics
- Compositional rule format, allowing for subrule calls (recognizing relation projections)

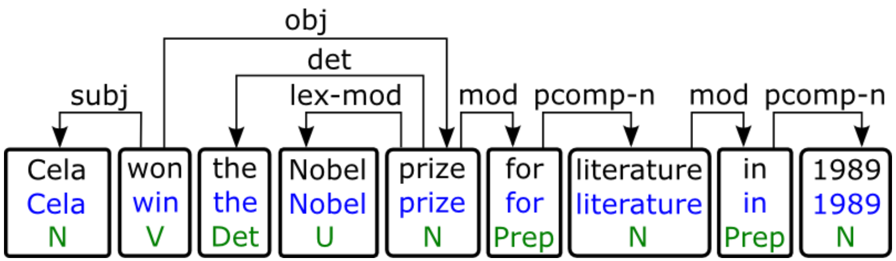
`Rule name::` recipient_prize_area_year_1

`Rule body::`

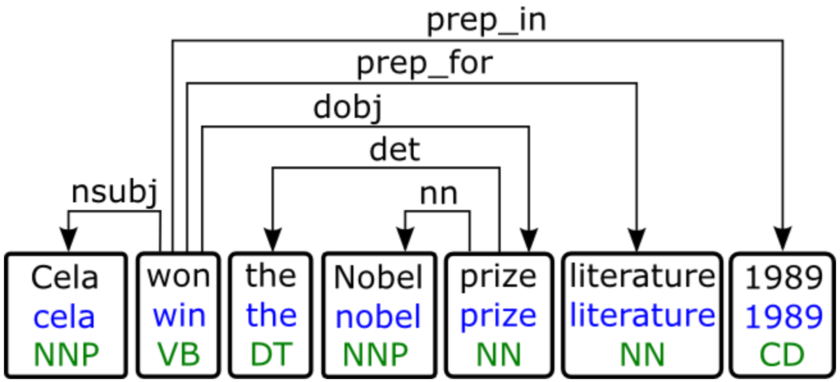
$$\left[\begin{array}{ll} \text{head} & \left[\begin{array}{ll} \text{pos} & \text{verb} \\ \text{mode} & \text{active} \\ \text{lex-form} & \text{“win”} \end{array}\right] \\ \text{daughters} & \left\langle \left[\text{subject} \; \left[\text{head} \; \boxed{1} \text{ Person}\right]\right], \left[\begin{array}{ll} \text{object} & \left[\begin{array}{ll} \text{head} & \left[\begin{array}{ll} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“prize”} \end{array}\right] \\ \text{rule} & \text{year\_prize\_area\_1::} \; \langle \boxed{4} Year, \boxed{2} Prize, \boxed{3} Area \rangle \end{array}\right] \end{array}\right] \right\rangle \end{array}\right]$$

`Output::` $\langle \boxed{1} Recipient, \boxed{2} Prize, \boxed{3} Area, \boxed{4} Year \rangle$

- MINIPAR (Lin, 2003)
- Broad-coverage parser for English
- Constraint-based parsing algorithm (reminiscent of chart parsing with rewrite rules)
- Parse results available in dependency format
- Partial results possible

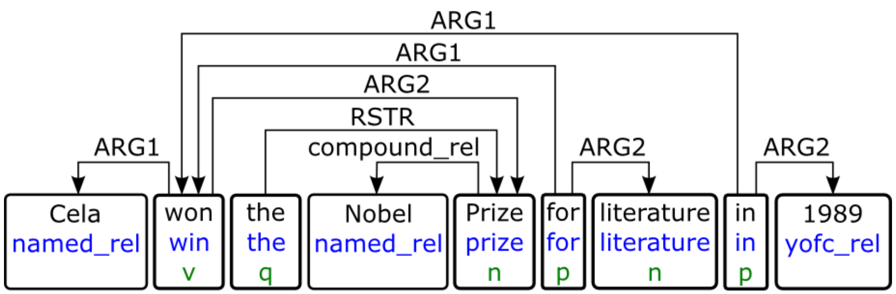


- Stanford Parser (Klein & Manning, 2003)
- Package with different parse strategies
- We use the unlexicalized PCFG parser
- Trees converted to labelled dependency representation (de Marneffe et al, 2006; de Marneffe & Manning, 2008)
- Tree simplifications tailored towards semantic tasks: functional edges collapsed



Parsers – PET + ERG

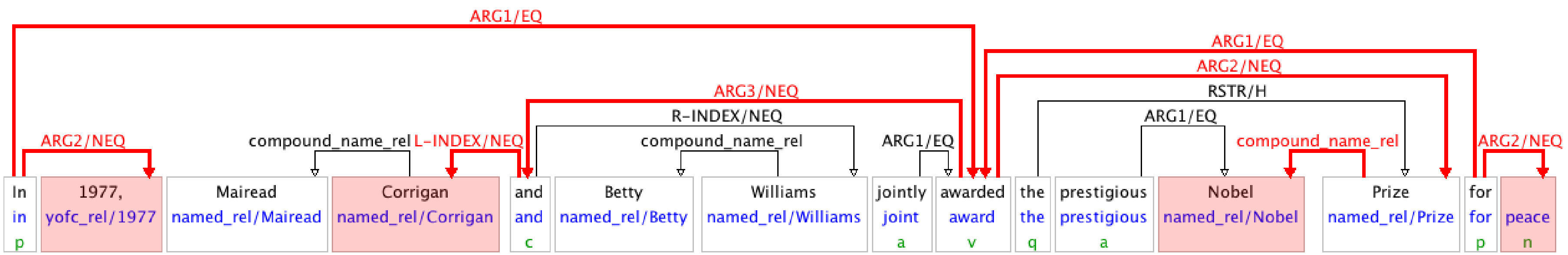
- We parsed with PET + ERG (Callmeier, 2002; Flickinger, 2000)
- DMRS (Copestake, 2008) is a dependency-style semantic representation
- We applied further simplifications to yield classical token-to-token dependencies
- Resulting structures are often genuine graphs



1) Introduction
2) DARE
3) Parsers
4) Graph Rules
5) Experiments
6) Conclusions

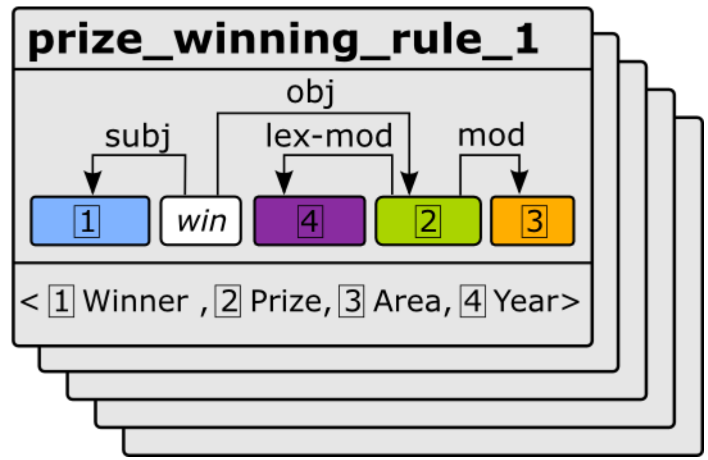
Graph Rules – Overview

- Adequate analyses of linguistic phenomena such as relative clauses or subject or object control constructions require that some nodes are shared, i.e. a graph structure.
- We extended the original DARE rule representation to match arbitrary graph substructures and adapted the rule learning procedure accordingly.



❖ Rule components:

- Rule name
- Rule body: Graph G=(N,E)
  - N: set of nodes with (possibly underspecified) features such as stem, part-of-speech or NE type
  - E: set of (possibly labelled) edges
- Output: mapping from argument nodes to target semantics



- For a given n-ary seed $S = (s_1, \ldots, s_n)$, find all sentences that mention the seed's arguments.
- For each sentence with dependency graph G, collect set T of all terminal nodes that represent arguments in S.
- For each acceptable combination of seed argument terminal nodes $C = \{t_1, \ldots, t_n\}$ ($m \geq 2$), find a shortest path S_i between t_i and t_{i+1} for $0 < i < m$.
- Extract the pattern subgraph $P_C = (N_C, E_C)$ from G with
  $N_C = \text{U}_i\, \text{N}(S_i)$ where $\text{N}(S_i)$ is the set of nodes in path S_i
  $E_C = \text{U}_i\, \text{E}(S_i)$ where $\text{E}(S_i)$ is the set of edges in path S_i
- Generalize nodes in N_C: keep stem, part-of-speech and named-entity type where applicable

1) Introduction
2) DARE
3) Parsers
4) Graph Rules
5) Experiments
6) Conclusions

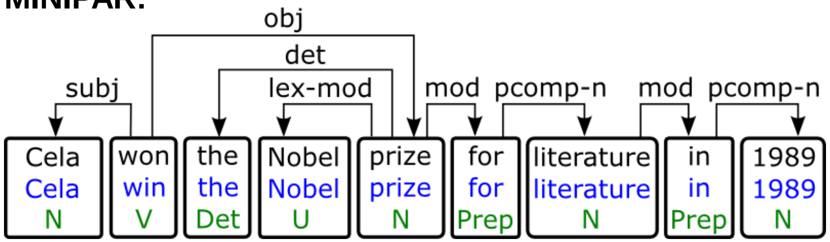
Experiments – Task and Data

- Target relation: prize winning
  - Who has won which prize for which achievement in which year?
  - Arguments:
    <winner, prize, field, year>
- Nobel Prize award corpus
  - Nobel Prize because gold relation instances are easily available
  - Already used in previous experiments (Xu et al., 2007)
  - Newswire texts (BBC, CNN and New York Times)
  - Contains only potentially relevant documents (mentioning „Nobel“)
  - Size: 2,864 relevant documents; 2,896 relevant sentences
  - Annotated for event mentions

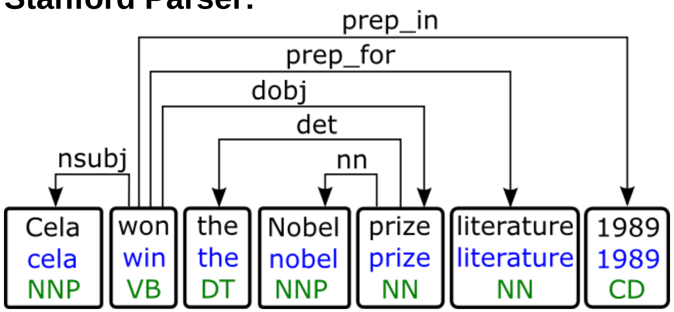
Experiments – Processing Setup

- Preprocessing:
  - Sentence and token segmentatoin (jTok)
  - Named entity recognition (SProUT, OpenCalais)
  - Coreference analysis (SProUT)
- Parsers:
  - MINIPAR 0.5
  - Stanford Parser 1.6.5
  - ERG 1010 with chart mapping, TnT unknown word handling
- Parse Coverage:
  - MINIPAR: 99.79%
  - Stanford Parser: 99.79%
  - PET + ERG: 71.71% (less robust on preprocessing errors)

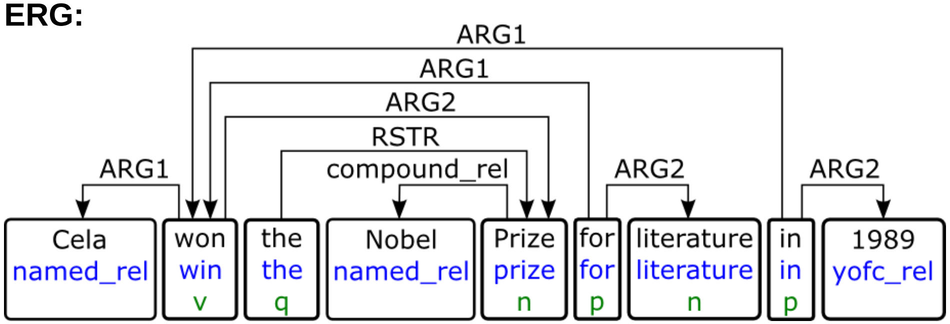
MINIPAR:



Stanford Parser:

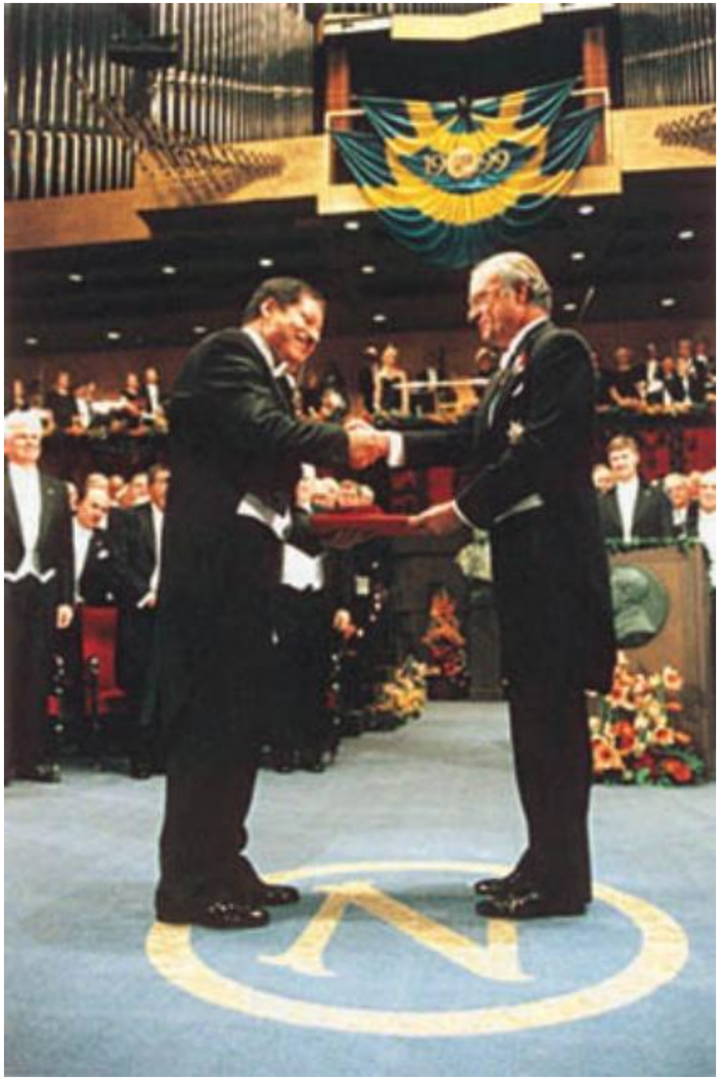


ERG:



Experiments – General Procedure

- Learn and apply rules based on dependency structures for different parsers separately
- Split corpus into learning and evaluation corpus
  - Equal-sized learning and evaluation corpora
  - Allows to assess reusability of relation extraction rules
  - Previous results evaluated learning performance on whole corpus
- Experiments with different seeds:
  - exactly one semantic seed <Ahmed Zewail, Nobel, chemistry, 1999>
  - 99 randomly chosen Nobel prize winning events
  - all Nobel prize winning events

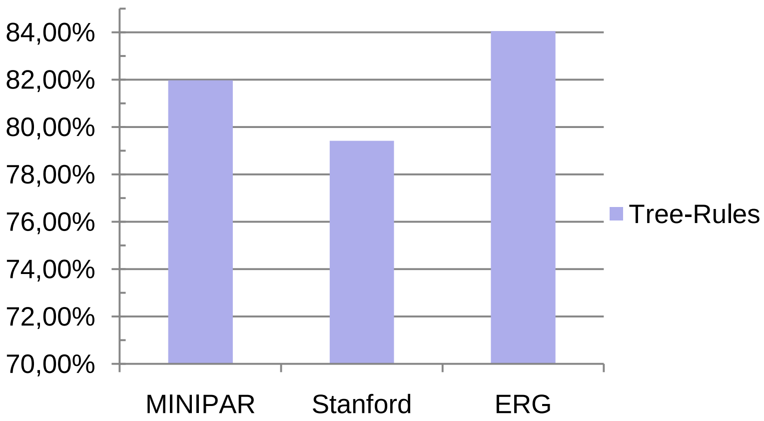


Experiments – Evaluation Setup

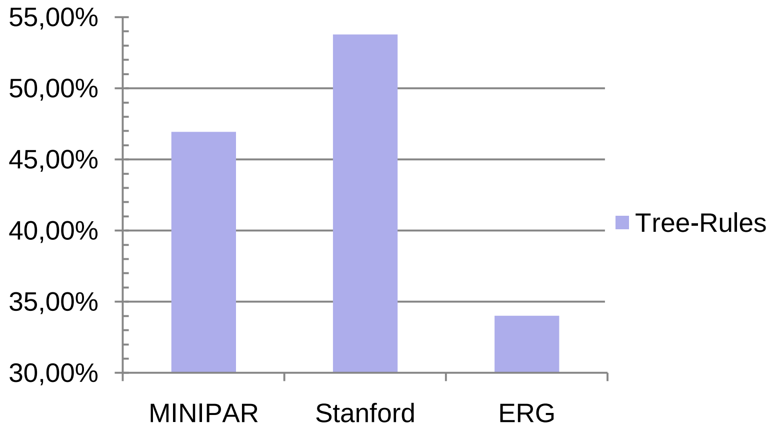
- Mention evaluation:
  - Evaluate pairs of <corpus sentence, extracted relation instance> against gold
  - Extraction considered successful if compatible with gold (extractions of lower arity are not penalized)
  - Measures: precision, recall, f-score and average arity of extractions
- NB: Previous results based on extracted relation instances only

Experiments – Evaluation (1)

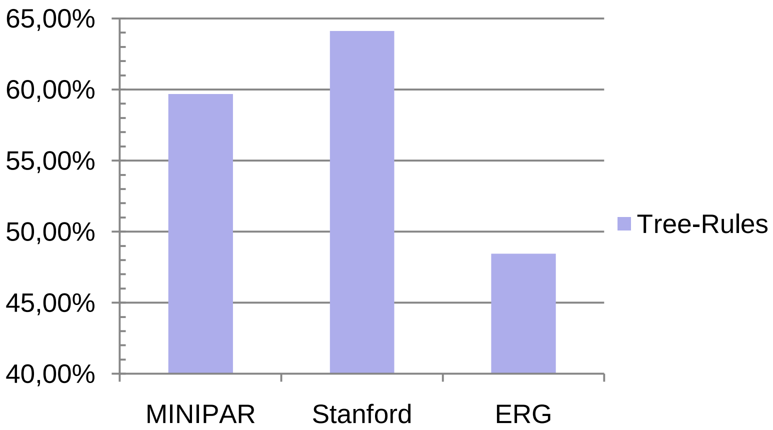
Precision



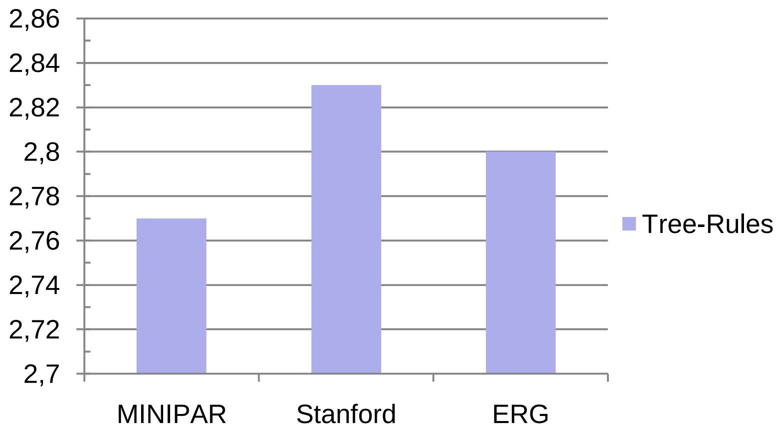
Recall



F-Score



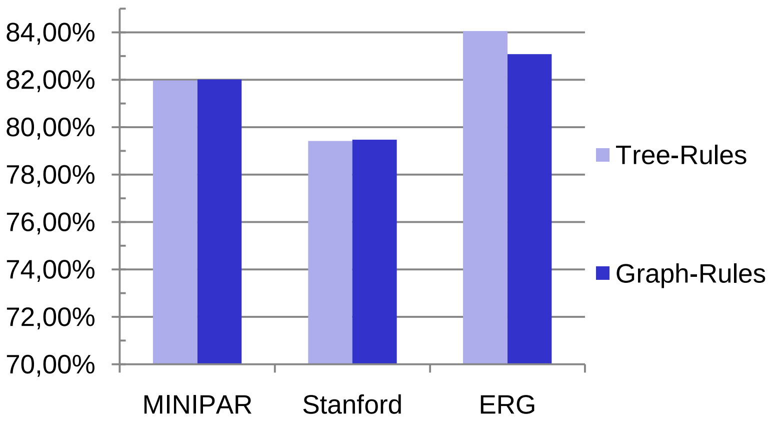
Average Arity



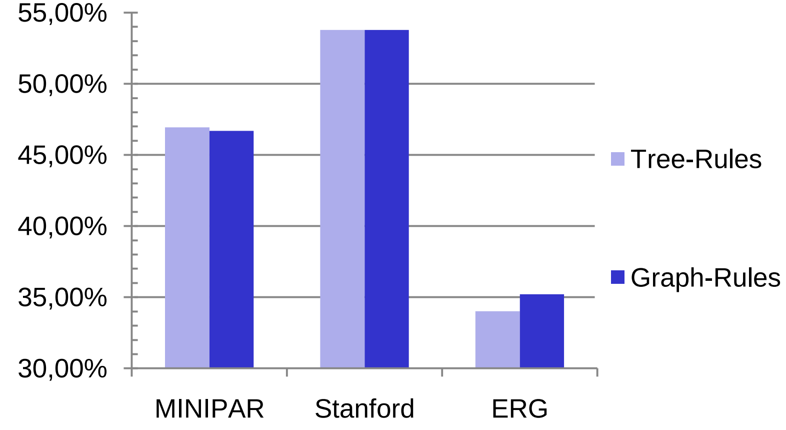
Corpus: All relevant sentences
1 semantic seed
Tree rules only

Experiments – Evaluation (2)

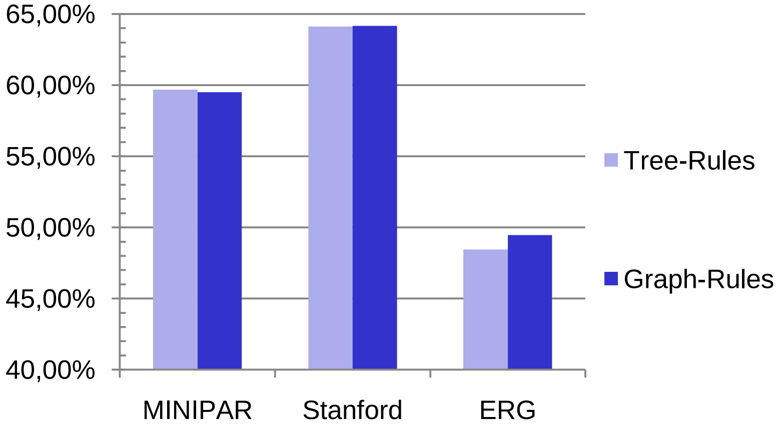
Precision



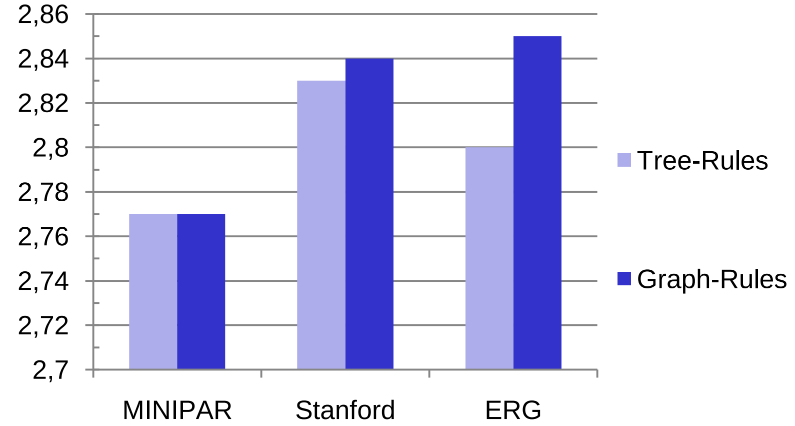
Recall



F-Score



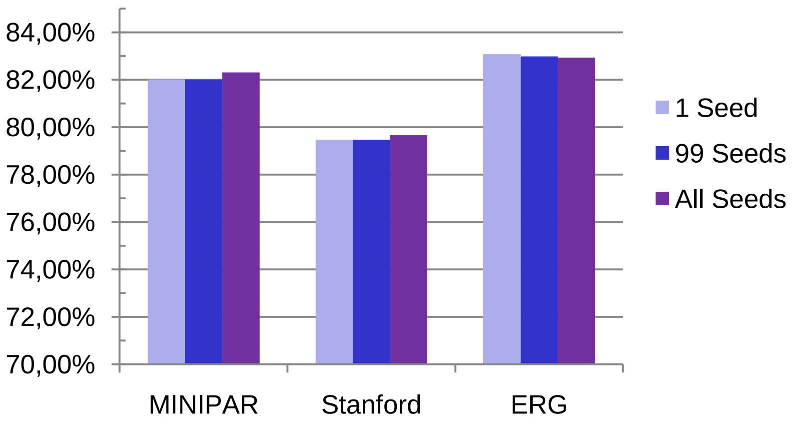
Average Arity



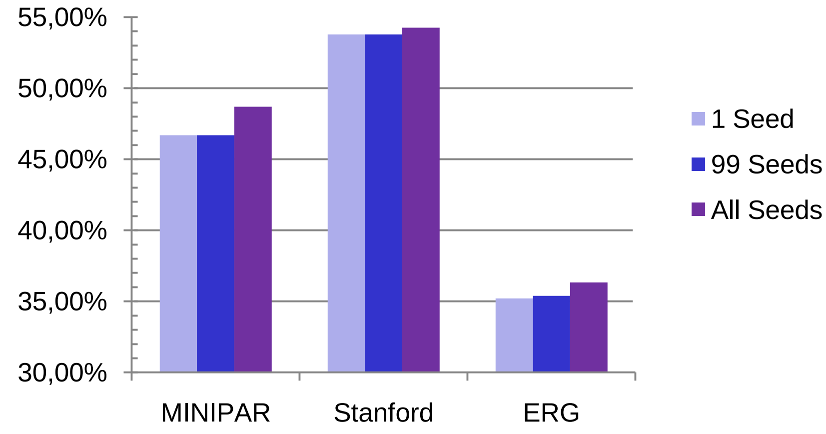
Corpus: All relevant sentences
1 semantic seed
Tree rules & **graph rules**

Experiments – Evaluation (3)

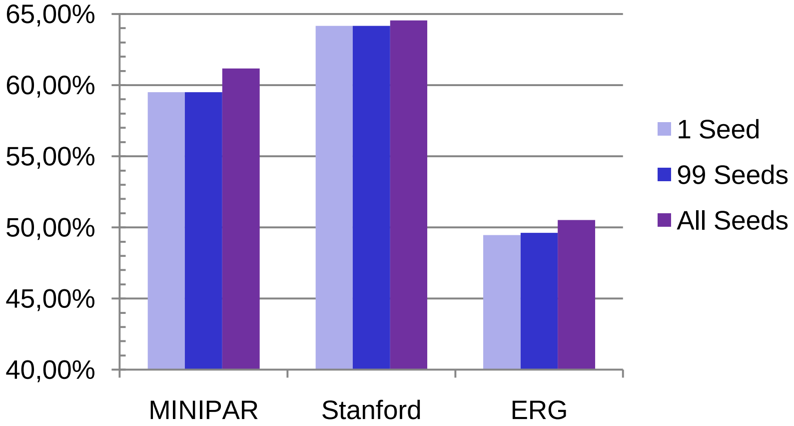
Precision



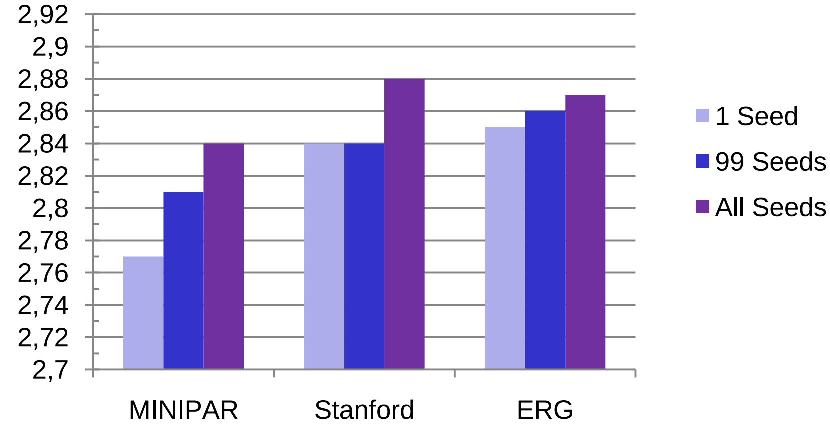
Recall



F-Score



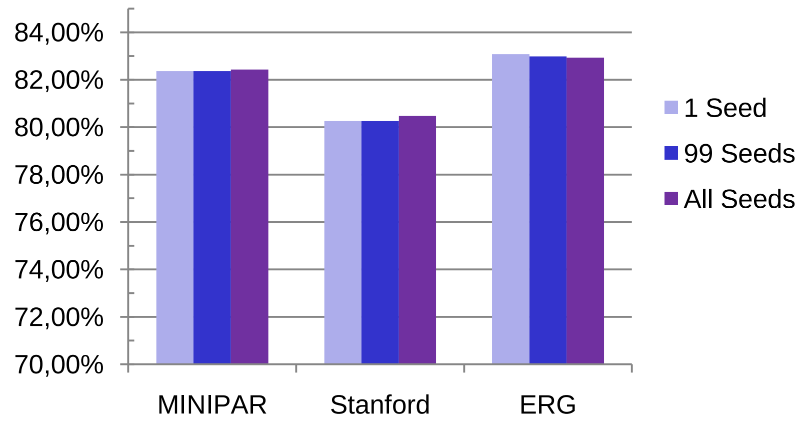
Average Arity



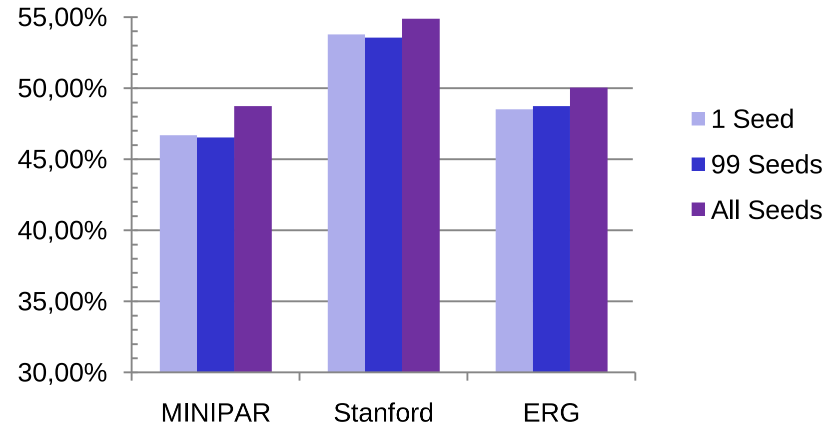
Corpus: All relevant sentences
1 / 99 / all seeds
Graph rules only

Experiments – Evaluation (4)

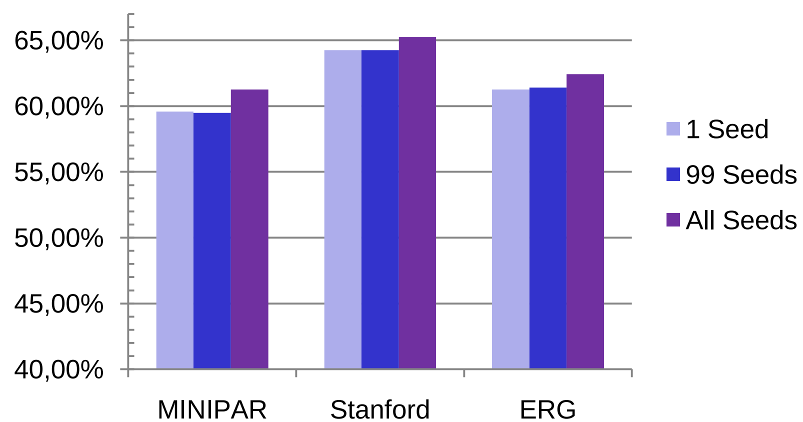
Precision



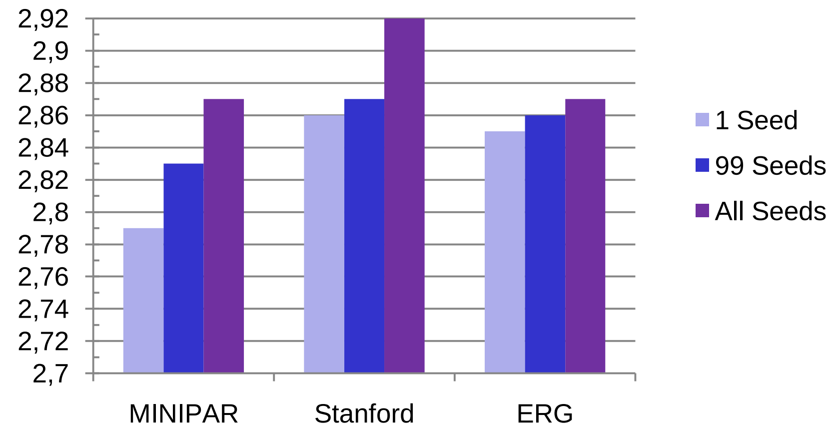
Recall



F-Score



Average Arity

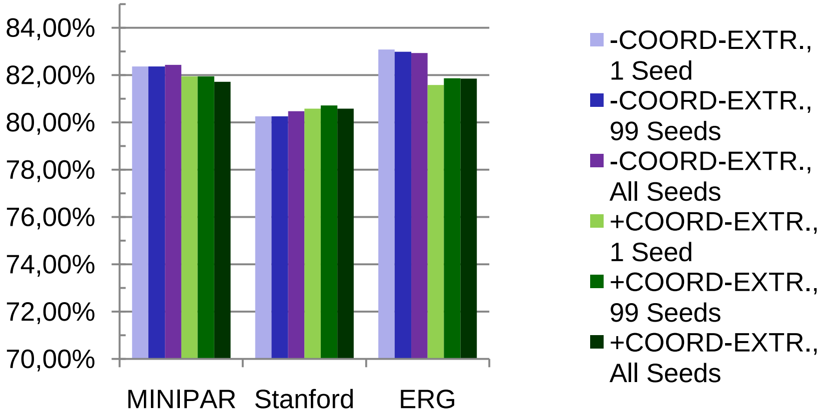


Corpus: All relevant **HPSG-parsable** sentences
1 / 99 / all seeds
Graph rules only

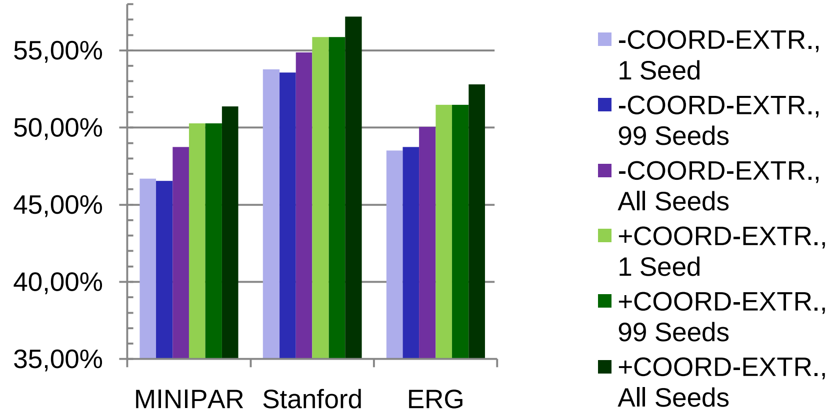
- Systematic differences of coordination analysis for MINIPAR / Stanford Parser vs ERG
  - MINIPAR / Stanford Parser: first conjunct of dependent conjunctions are linked to the head, remaining conjuncts are linked to first conjunct with a conjunction edge
  - ERG: extra conjunction node, conjuncts are linked to the conjunction node
- Effect:
  - MINIPAR / Stanford Parser RE rules learned from a structure without conjunction are also used to extract first conjunct in conjunction structures
  - Conjunction seeds may help to learn more complex RE rules for conjunction structures in the bootstrapping
- Solution: interpret coordination structures during RE

Experiments – Evaluation (5)

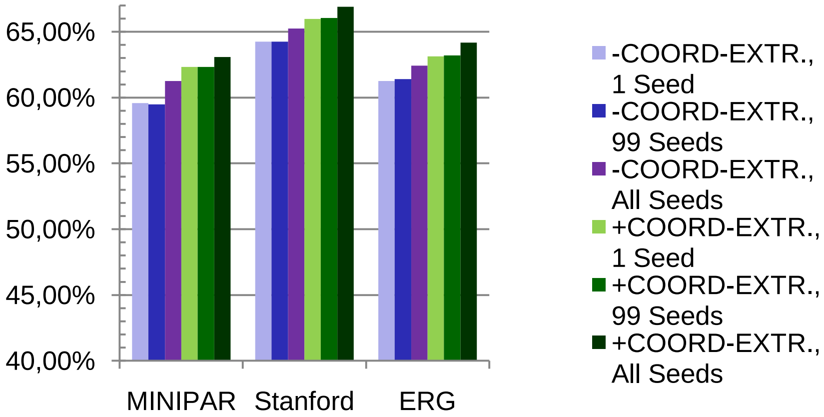
Precision



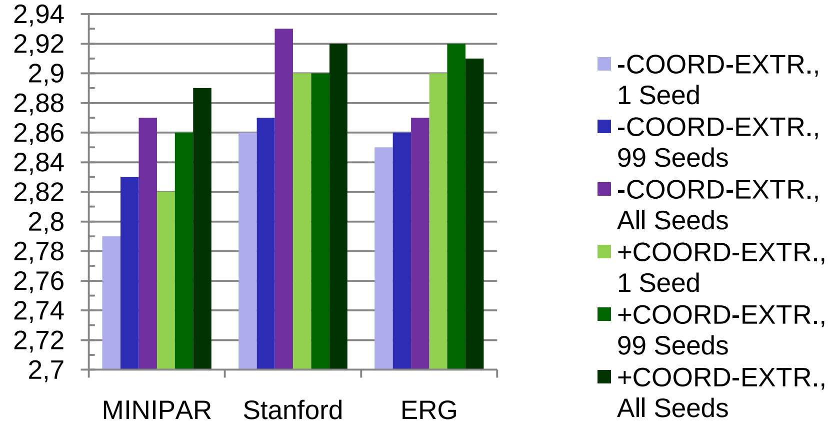
Recall



F-Score



Average Arity



Corpus: All relevant HPSG-parsable sentences
1 / 99 / all seeds
Graph rules **-/+ Extraction from Coordinations**

Experiments – Summary

- Graph rules are beneficial for relation extraction with ERG
- Relation extraction on top of ERG analyes delivers highest precision results, but on the cost of recall; cf. for 1 seed:
  - Precision: +3.8% for ERG if compared to Stanford Parser
  - Recall: -18.6% for ERG if compared to Stanford Parser
- Rule learning & relation extraction on HPSG-parsable corpus:
  - Comparable results for Stanford Parser (precision even improves)
  - Stanford Parser still performs best
- Extraction from coordinations:
  - All parsers benefit from this extraction strategy:
    recall and f-score improve
  - Precision improves for Stanford Parser
  - Precision drops surprisingly for ERG (reading selection?)

1) Introduction
2) DARE
3) Parsers
4) Graph Rules
5) Experiments
6) Conclusions

Conclusions

- ERG can be sucessfully employed for substantially improving precision in the relation extraction task
- Stanford typed dependency relations are better suited out of the box for semantic applications such as DARE
- Graph-based relation extraction rules set the ground for hybrid relation extraction systems
  - Represent annotations by arbitrary parsers in an annotation graph, on which graph-based relation extraction rules operate
  - Combination of several parsers promises to overcome coverage gaps of HPSG (by using more shallow parsers) and benefiting from more detailed analyses (when using HPSG)

Next Steps

- Pin down advantages of each parser to distinguishing criteria, in order to learn RE rules from the merged output of a parser ensemble
- Use ERG analyses to detect valancy-changing semantic operations such as modality, negation and their scopal interactions

THANK YOU FOR YOUR ATTENTION

Questions?