Phrases and Sentences

1. Language models
2. Chunking
3. Structural descriptions
4. Parsing with phrase structure grammars
5. Probabilistic parsers
6. Parsing with dependency models
7. Principles and Parameters
8. Unification-based grammars
9. Semantics construction

Phrases and Sentences

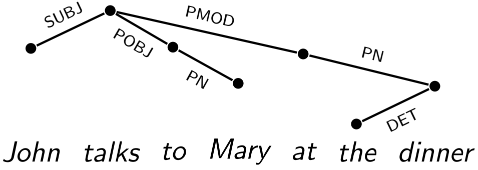
1. Language models
2. Chunking
3. Structural descriptions
4. Parsing with phrase structure grammars
5. Probabilistic parsers
6. Parsing with dependency models
7. Principles and Parameters
8. Unification-based grammars
9. Semantics construction

Parsing with Dependency Models

- Dependency modeling
- Dependency parsing as constraint satisfaction
- Structure-based dependency parsing
- History-based dependency parsing
- Parser combination

Dependency Modeling

- binary relationship between words



Dependency Modeling

- highly regular search space

root/nil	root/nil	root/nil	root/nil	root/nil
det/2	det/1	det/1	det/1	det/1
det/3	det/3	det/2	det/2	det/2
det/4	det/4	det/4	det/3	det/3
det/5	det/5	det/5	det/5	det/4
subj/2	subj/1	subj/1	subj/1	subj/1
subj/3	subj/3	subj/2	subj/2	subj/2
subj/4	subj/4	subj/4	subj/3	subj/3
subj/5	subj/5	subj/5	subj/5	subj/4
dobj/2	dobj/1	dobj/1	dobj/1	dobj/1
dobj/3	dobj/3	dobj/2	dobj/2	dobj/2
dobj/4	dobj/4	dobj/4	dobj/3	dobj/3
dobj/5	dobj/5	dobj/5	dobj/5	dobj/4
Diese	*Scheibe*	*ist*	*ein*	*Hit*
1	2	3	4	5

Dependency Modeling

- for every word the parser has to take at most three decisions
  - Where to attach?
  - With which label?
  - possibly: With which dictionary entry?

- projectivity assumption constrains the search space
  - non-projective structures as a major efficiency problem

Dependency Modeling

- advantages (Covington 2001, Nivre 2005)
  - straightforward mapping of head-modifier relationships to arguments in a semantic representation
  - parsing relates existing nodes to each other
    - no need to postulate additional ones
  - word-to-word attachment is a more fine-grained relationship compared to phrase structures
    - modelling constraints on partial "constituents"
    - factoring out dominance and linear order
    - well suited for incremental processing
  - non-projectivities can be treated appropriately
    - discontinuous constructions are not a (modeling) problem

Dependency Parsing as Constraint Satisfaction

- Constraint Grammar KARLSSON 1995
  - attaching possibly underspecified dependency relations to the word forms of an utterances

@+FMAINV	finite verb of a sentence
@SUBJ	grammatical subject
@OBJ	direct Object
@DN>	determiner modifying a noun to the right
@NN>	noun modifying a noun to the right

Dependency Parsing as Constraint Satisfaction

- typical CS problem:
  - constraints: conditions on the (mutual) compatibility of dependency labels
  - indirect definition of well-formedness: everything which does not violate a constraint explicitly is acceptable
- strong similarity to tagging procedures

Dependency Parsing as Constraint Satisfaction

- two important prerequisites for robust behaviour
  - inherent fail-soft property: the last remaining category is never removed even if it violates a constraint
  - possible structures and well-formedness conditions are fully decoupled: missing grammar rules do not lead to parse failures
- complete disambiguation cannot always be achieved

Bill	*saw*	*the*	*little*	*dog*	*in*	*the*	*park*
`@SUBJ`	`@+FMAINV`	`@DN>`	`@AN>`	`@OBJ`	`@<NOM`	`@DN>`	`@<P`
					`@<ADVL`		

Dependency Parsing as Constraint Satisfaction

- size of the grammar (English): 2000 Constraints
- quality

	without heuristics	with heuristics
precision	95.5%	97.4%
recall	99.7 ... 99.9%	99.6 ... 99.9%

Dependency Parsing as Constraint Satisfaction

- Constraint Dependency Grammar MARUYAMA 1990
- each word form of a sentence corresponds to a variable.
  $\rightarrow$ number of variables is a priori unknown.
  $\rightarrow$ no predefined meaning for variables.
- every constraint must hold for each variable or a combination thereof.
- values are taken from the domain $W \times L$
- constraints license linguistically meaningful structures
- parsing can be understood as structural disambiguation: find a complete variable assignment which satisfies all constraints

Parsing as Constraint Satisfaction

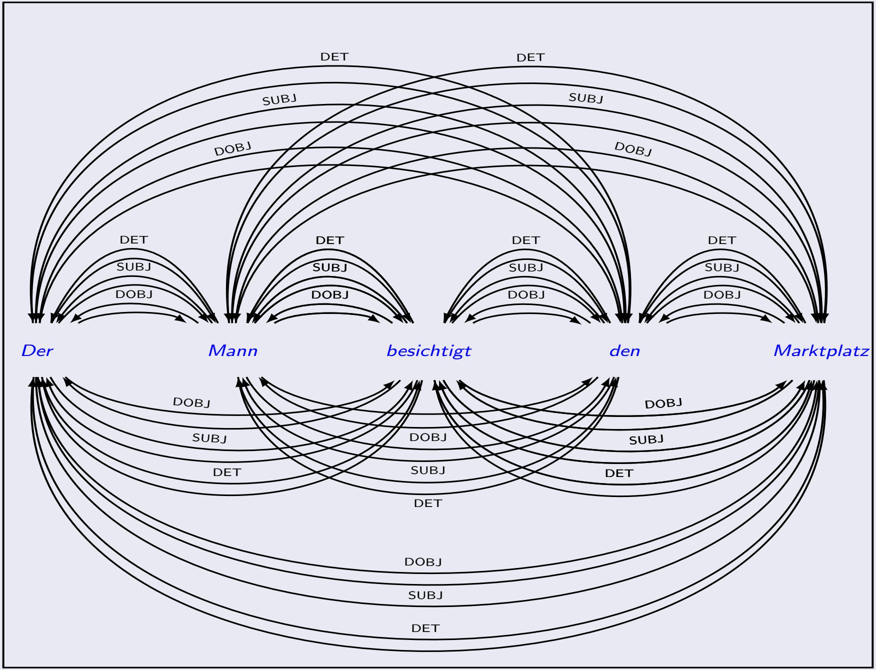
$\{X\}$: DetNom : Det : 0.0 :
$X{\downarrow}cat{=}det \;\rightarrow\; X{\uparrow}cat{=}noun \wedge X.label{=}\textsc{det}$

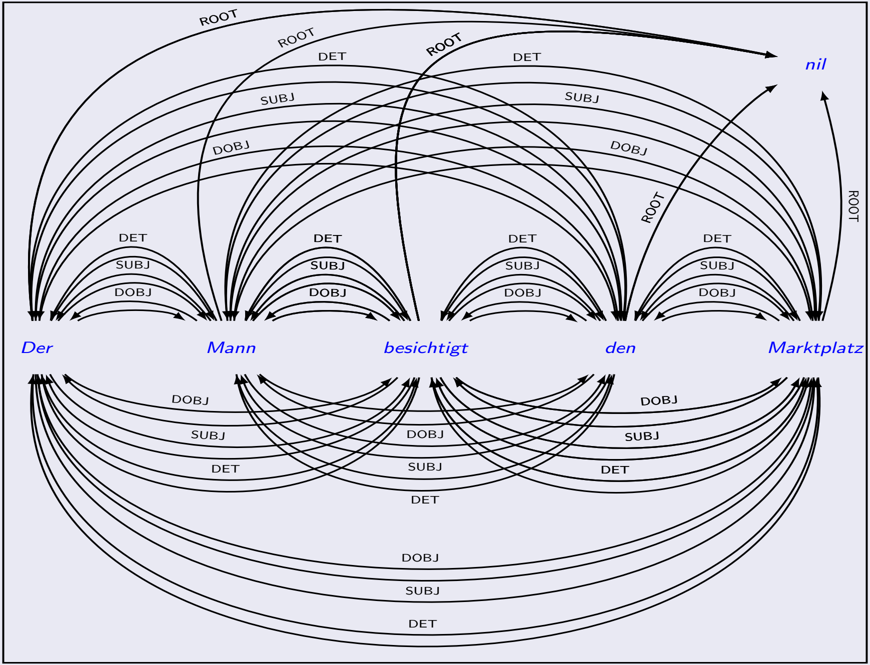
$\{X\}$: SubjObj : Verb : 0.0 :
$X{\downarrow}cat{=}noun$
$\rightarrow\; X{\uparrow}cat{=}vfin \wedge X.label{=}\textsc{subj} \vee X.label{=}\textsc{dobj}$

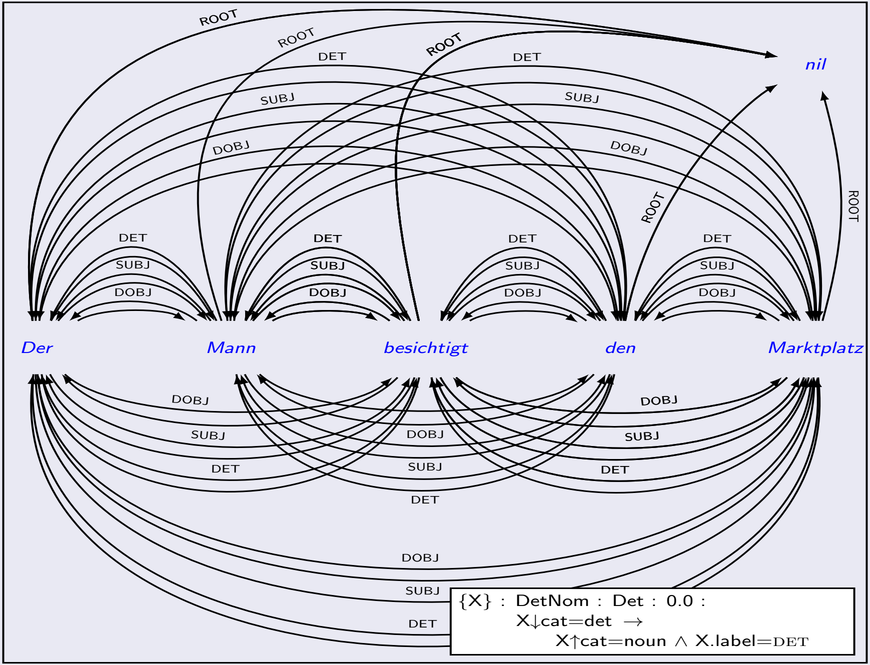
$\{X\}$: Root : Verb : 0.0 :
$X{\downarrow}cat{=}vfin \;\rightarrow\; X{\uparrow}cat{=}nil$

$\{X,Y\}$: Unique : General : 0.0 :
$X{\uparrow}id{=}Y{\uparrow}id \;\rightarrow\; X.label{\neq}Y.label$

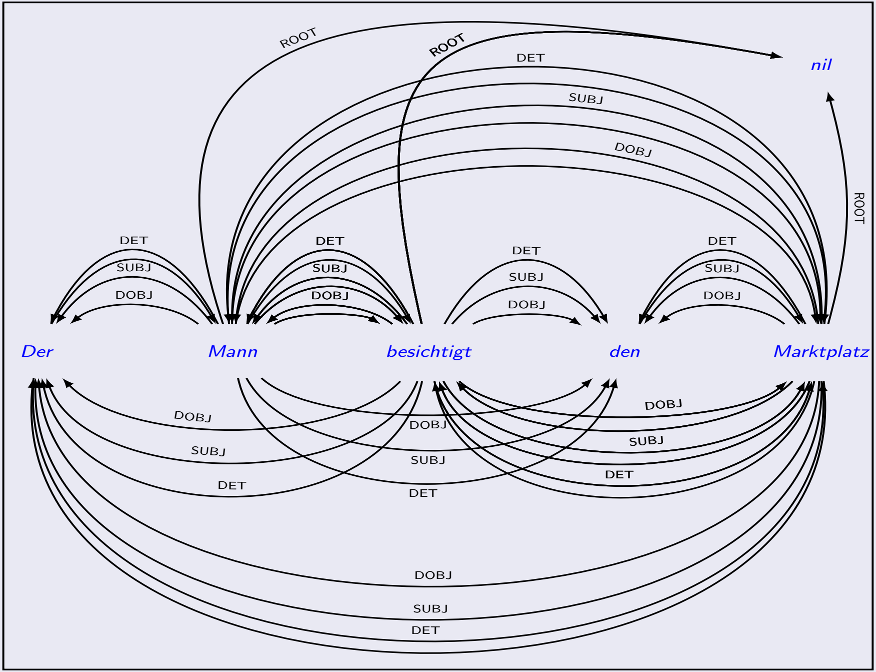
$\{X,Y\}$: SubjAgr : Subj : 0.0 :
$X.label{=}\textsc{subj} \wedge Y.label{=}\textsc{det} \wedge X{\downarrow}id{=}Y{\uparrow}id$
$\rightarrow\; Y{\uparrow}case{=}Y{\downarrow}case{=}nom$

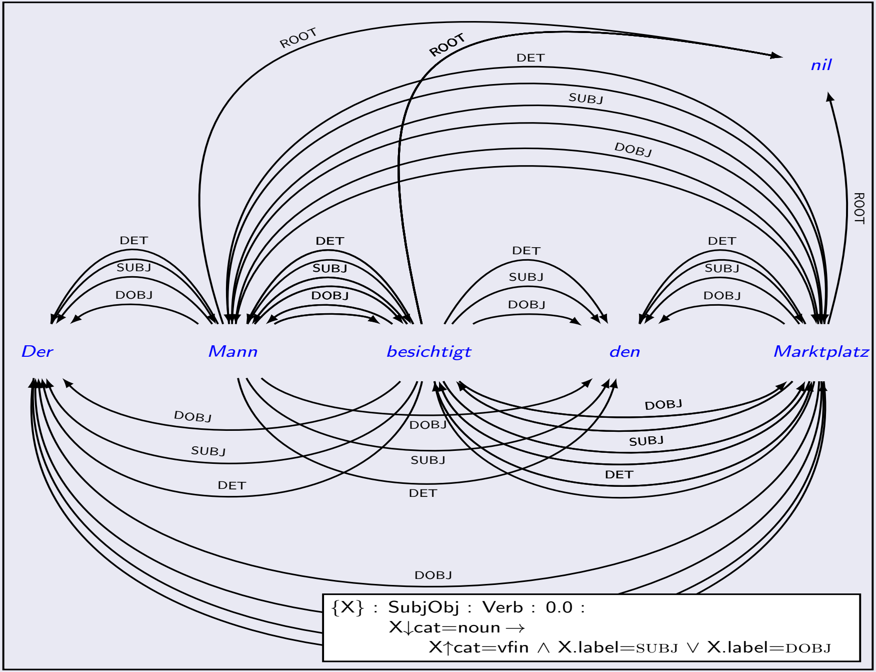

DET
DET
SUBJ
SUBJ
DOBJ
DOBJ
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
Der
Mann
besichtigt
den
Marktplatz
DOBJ
DOBJ
SUBJ
DOBJ
SUBJ
SUBJ
DET
DET
DET
DET
DOBJ
SUBJ
DET

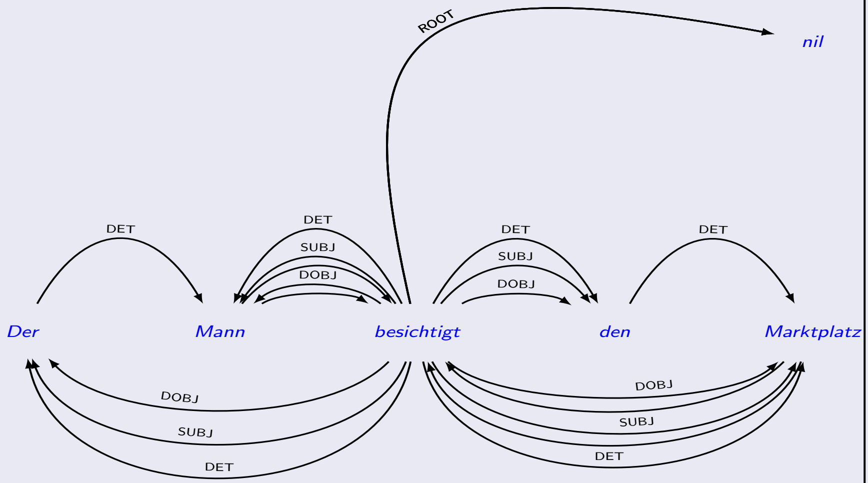

nil
Der
Mann
besichtigt
den
Marktplatz
ROOT
DET
SUBJ
DOBJ

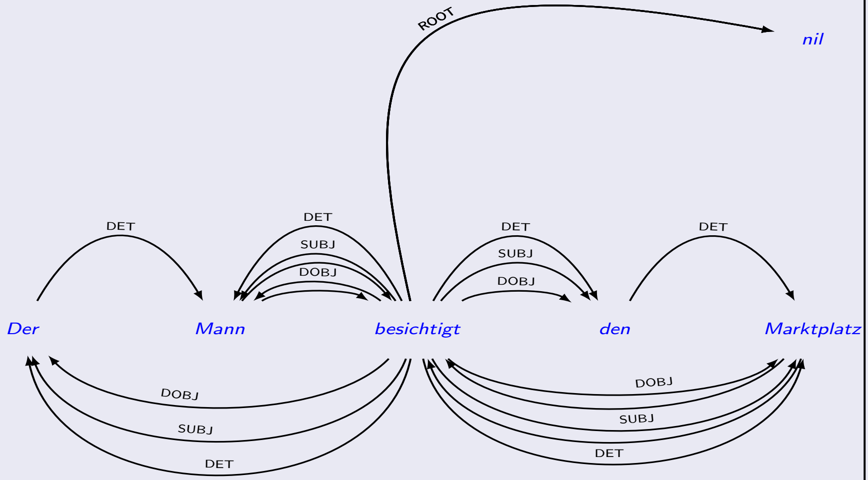


$\{X\}$: DetNom : Det : 0.0 :
$X{\downarrow}cat{=}det \rightarrow$
$X{\uparrow}cat{=}noun \wedge X.label{=}\textsc{det}$

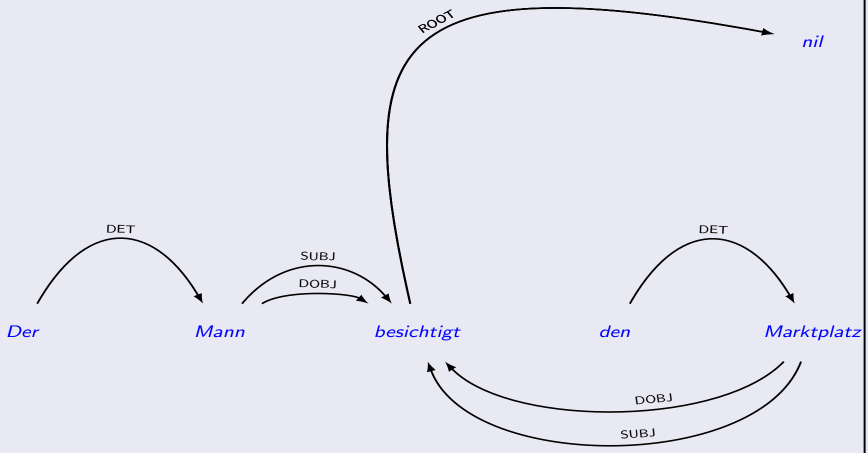

Der
Mann
besichtigt
den
Marktplatz
nil
ROOT
ROOT
ROOT
DET
SUBJ
DOBJ

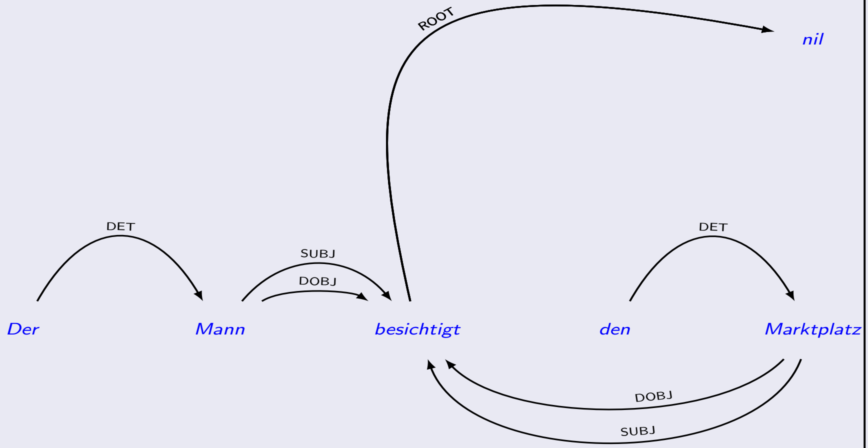

nil
ROOT
ROOT
ROOT
DET
SUBJ
DOBJ
Der
Mann
besichtigt
den
Marktplatz
{X} : SubjObj : Verb : 0.0 :
X↓cat=noun →
X↑cat=vfin ∧ X.label=SUBJ ∨ X.label=DOBJ


ROOT
nil
DET
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
DET
Der
Mann
besichtigt
den
Marktplatz
DOBJ
SUBJ
DET
DOBJ
SUBJ
DET

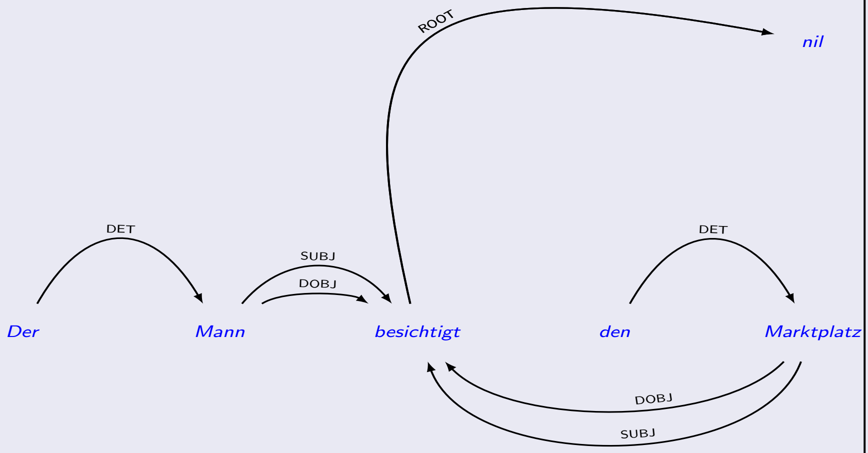


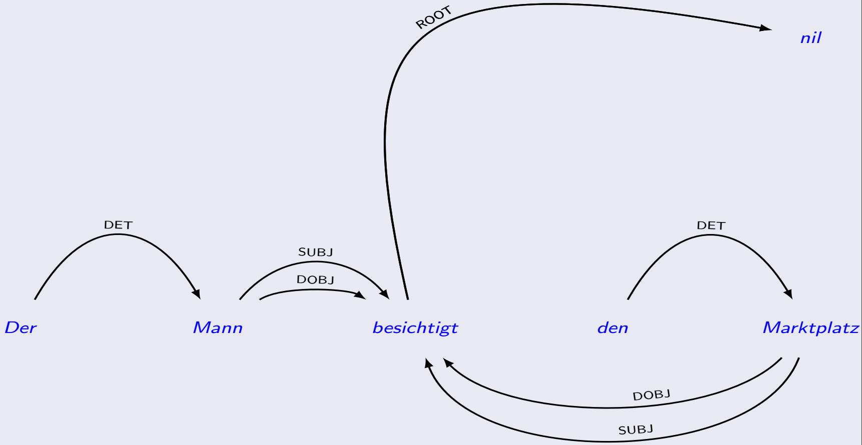
$\{X\}$: Root : Verb : 0.0 :
$X{\downarrow}cat{=}vfin \rightarrow X{\uparrow}cat{=}nil$


ROOT
nil
DET
SUBJ
DOBJ
DET
Der
Mann
besichtigt
den
Marktplatz
DOBJ
SUBJ

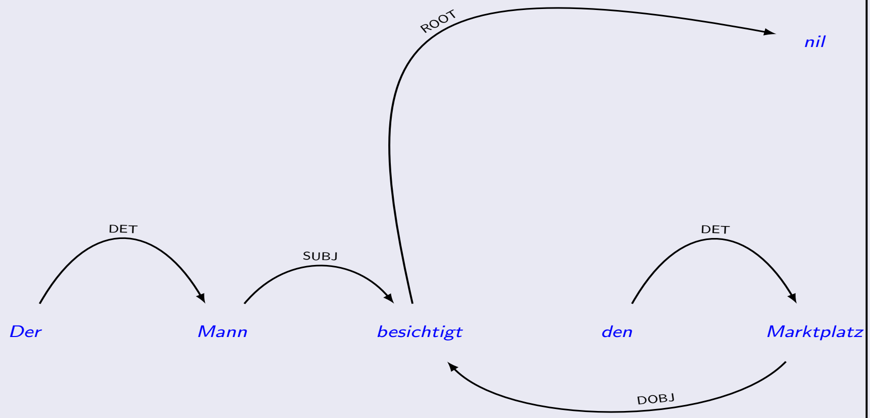


{X,Y} : Unique : General : 0.0 :
X↑id=Y↑id → X.label≠Y.label


ROOT
nil
DET
SUBJ
DOBJ
DET
Der
Mann
besichtigt
den
Marktplatz
DOBJ
SUBJ



$\{X,Y\}$: SubjAgr : Subj : 0.0 :
X.label=$\textsc{subj}$ $\wedge$ Y.label=$\textsc{det}$ $\wedge$ X$\downarrow$id=Y$\uparrow$id
$\rightarrow$ Y$\uparrow$case=Y$\downarrow$case=nom



ROOT
nil
DET
SUBJ
DET
Der
Mann
besichtigt
den
Marktplatz
DOBJ

Preferential reasoning

- Natural language grammar is not fully consistent
- Many conflicting requirements
  - e.g. minimizing distance: verb bracket vs. reference

Sie trägt den Termin, den wir vereinbart hatten, ein.

Preferential reasoning

- Natural language grammar is not fully consistent
- Many conflicting requirements
  - e.g. minimizing distance: verb bracket vs. reference

Sie trägt den Termin, den wir vereinbart hatten, ein.

Sie trägt den Termin ein, den wir vereinbart hatten.

Conflicts

Conflicts occur

- between levels of conceptualization
  e.g. syntax, information structure and semantics
- between different processing components
  e.g. tagger, chunker, PP-attacher
- between the model and the utterance
  e.g. modelling errors, not well-formed input
- between the utterance and the background knowledge
  e.g. misconceptions, lies
- across modalities
  e.g. seeing vs. hearing

Goal: achieve robustness and develop diagnostic capabilities

Conflicts

Why should we care about conflicts?

- they are pervasive
- they provide valuable information
  - for improving the system:
    e.g. through manual grammar development or reinforcement learning
  - about the proficiency of the speaker/writer:
    e.g. to derive remedial feedback
  - about the intentions of the speaker/writer:
    e.g. attention focussing by means of topicalization
  - for guiding the parser

Weighted Constraints

- conflict resolution requires weighted constraints
  - weights describe the importance of the constraint
  - how serious it is to violate the constraint
- differently strong constraints
  - hard constraints, must always be satisfied
  - strong constraints: agreement, word order, ...
  - weak constraints: preferences, defaults, ...

Preferential reasoning

- accumulating (multiplying) the weights for all constraints violated by a partial structure
  $\rightarrow$ numerical grading for single dependency relations and pairs of them

- combining local scores by multiplying them into a global one

$$w(t) = \prod_{e \in t} \prod_{c.\text{violates}(e,c)} w(c) \cdot \prod_{(e_i,e_j) \in t} \prod_{c.\text{violates}((e_i,e_j),c)} w(c)$$

- determining the optimal global structure

$$t(s) = \arg\max_{t} w(t)$$

$\rightarrow$ parsing becomes a constraint optimization problem

Preferential Reasoning

- writing constraints is counterintuitive
  - CFG: to extend coverage, *add* or *extend* a rule
  - CDG: to extend coverage, *remove* or *weaken* a constraint
- but: the parser itself supports grammar development providing *diagnostic information*
  - constraints violated by the optimal structure are identified

Global Constraints

- Most constraints are local ones (unary, binary)
- Sometimes global requirements need to be checked
  - existence/non-existence requirements (e.g. valencies)
  - conditions in a complex verb group
- Local search supports the application of global constraints
  - always a complete value assignment (i.e. a dependency tree) is available
- Three kinds of global constraints
  - *has*: downwards tree traversal
  - *is*: upwards path traversal
  - recursive constraints: can call other constraints to be checked elsewhere in the tree

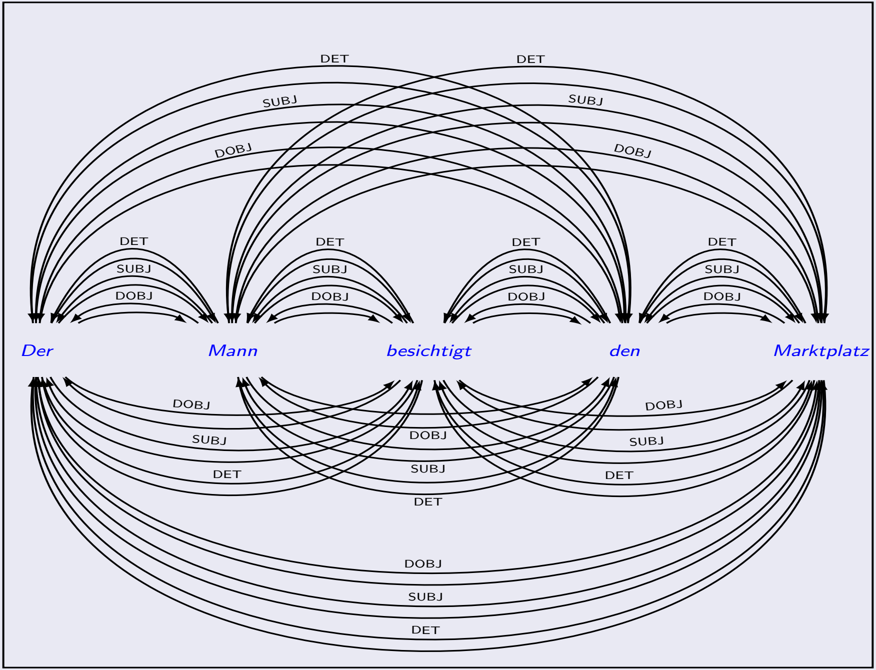
Weighted Constraints

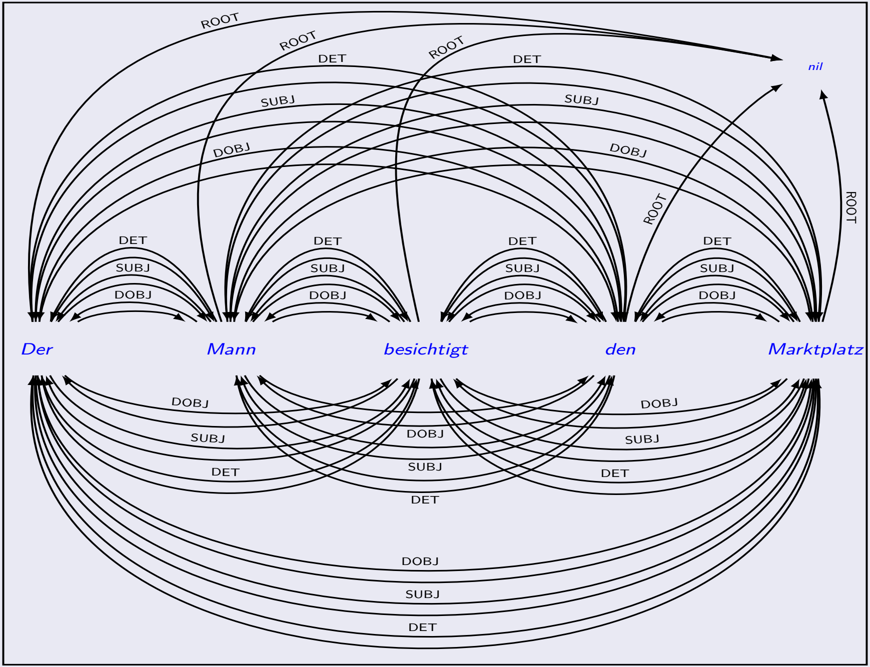
- different solution procedures available
  - **consistency**: works only for hard constraints
  - **pruning**: successively remove the least preferred dependency relations
  - **search**: determine the optimum dependency structure
  - **structural transformation**: apply local repairs to improve the overall score

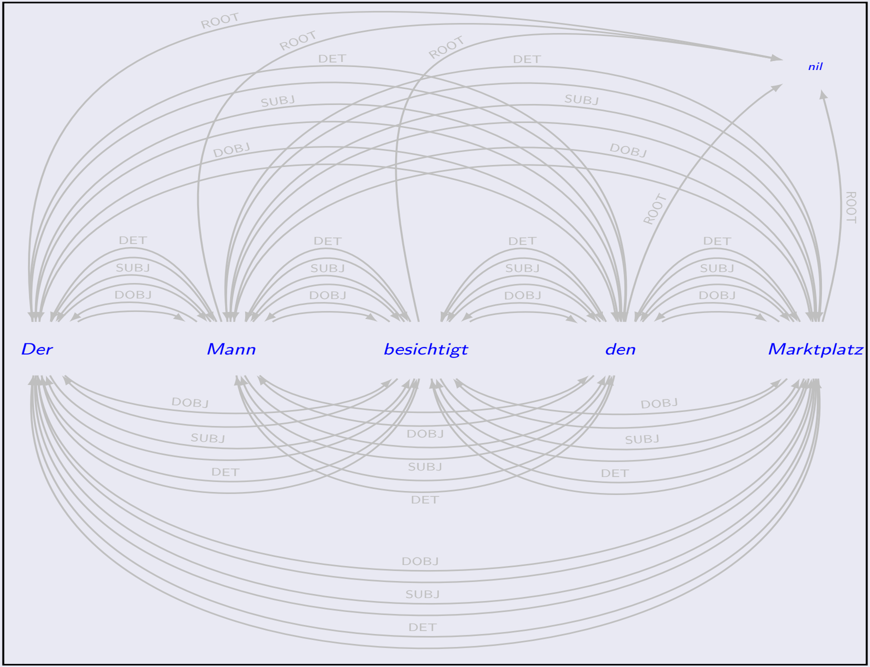
- strong quality requirements
  - a single prespecified solution has to be found (gold standard)
  - sometimes the gold standard differs from the optimal solution
  - modelling errors vs. search errors

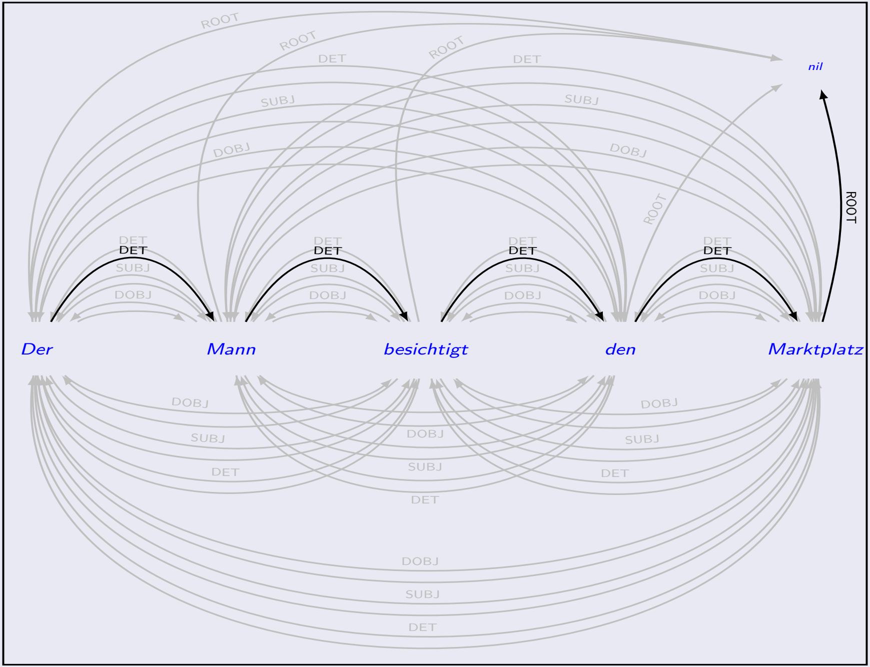
Solution Procedures

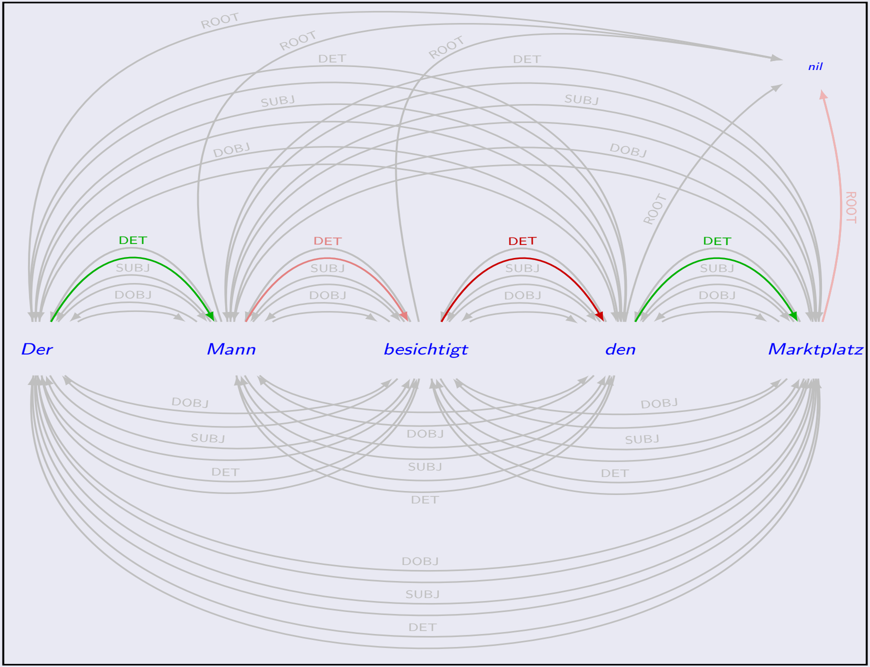
- the best method found so far:
  - local search with value exchange (frobbing)
  - gradient descent heuristics
  - with a tabu list
  - with limits (similar to branch and bound)
  - increasingly accepting degrading value selections to escape from local minima


DET
DET
SUBJ
SUBJ
DOBJ
DOBJ
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
Der
Mann
besichtigt
den
Marktplatz
DOBJ
DOBJ
SUBJ
DOBJ
SUBJ
DET
SUBJ
DET
DET
DOBJ
SUBJ
DET

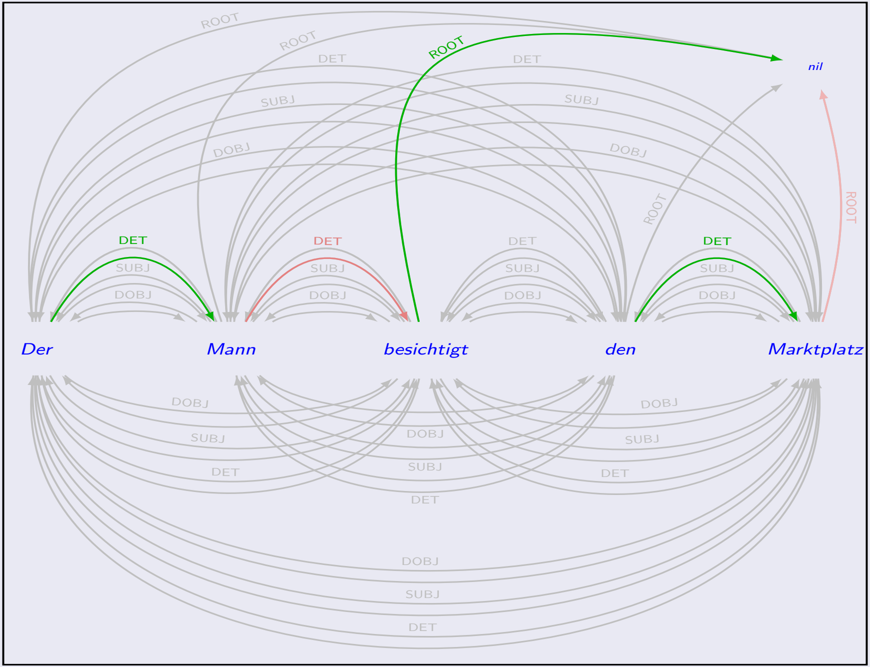

Der
Mann
besichtigt
den
Marktplatz
nil
ROOT
DET
SUBJ
DOBJ

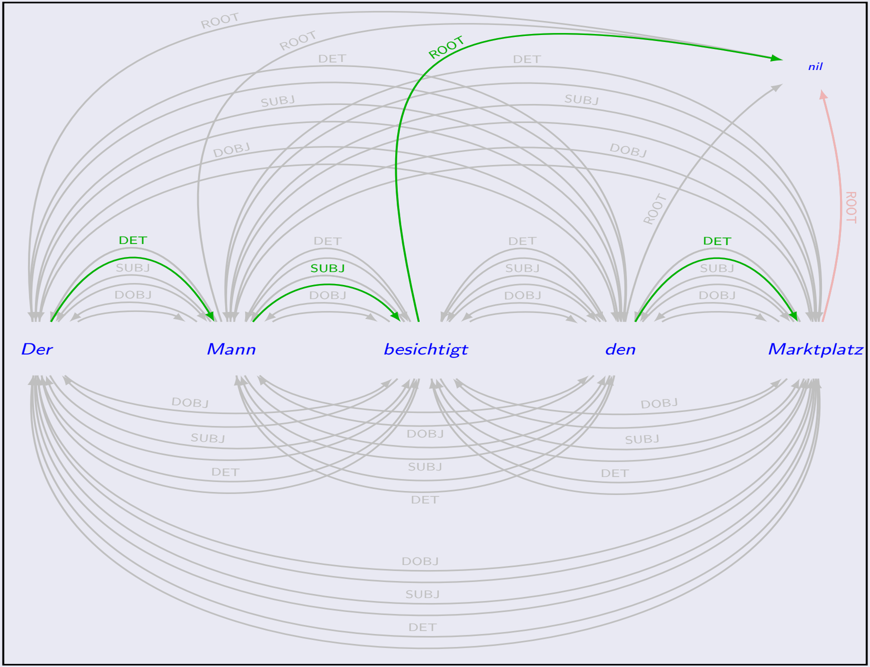

ROOT
ROOT
ROOT
nil
DET
DET
SUBJ
SUBJ
DOBJ
DOBJ
ROOT
ROOT
DET
SUBJ
DOBJ
Der
Mann
besichtigt
den
Marktplatz
DOBJ
SUBJ
DET
DOBJ
SUBJ
DET
DOBJ
SUBJ
DET
DOBJ
SUBJ
DET

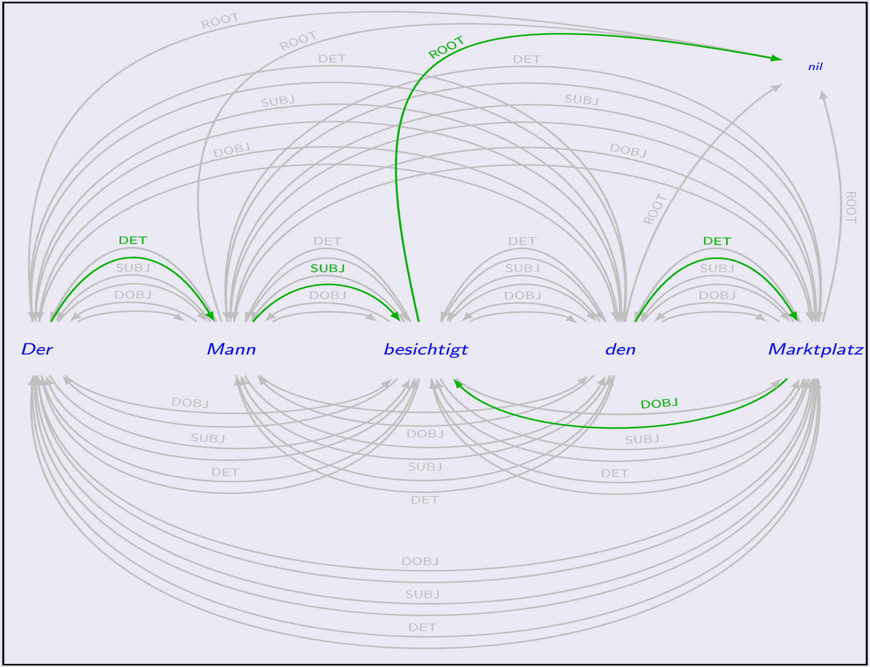

nil
ROOT
DET
SUBJ
DOBJ
Der
Mann
besichtigt
den
Marktplatz



nil
Der
Mann
besichtigt
den
Marktplatz
ROOT
DET
SUBJ
DOBJ


ROOT
ROOT
ROOT
DET
DET
nil
SUBJ
SUBJ
DOBJ
DOBJ
ROOT
ROOT
DET
DET
DET
DET
DET
SUBJ
SUBJ
SUBJ
SUBJ
DOBJ
DOBJ
DOBJ
DOBJ
Der
Mann
besichtigt
den
Marktplatz
DOBJ
DOBJ
SUBJ
DOBJ
SUBJ
SUBJ
DET
DET
SUBJ
DET
DET
DOBJ
SUBJ
DET


nil
Der
Mann
besichtigt
den
Marktplatz
ROOT
DET
SUBJ
DOBJ


ROOT
ROOT
ROOT
DET
DET
SUBJ
SUBJ
DOBJ
DOBJ
ROOT
ROOT
nil
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
DET
SUBJ
DOBJ
Der
Mann
besichtigt
den
Marktplatz
DOBJ
SUBJ
DET
DOBJ
SUBJ
DET
DOBJ
SUBJ
DET
DOBJ
SUBJ
DET

Non-local Transformations

- usually local transformations result in inacceptable structures
- sequences of repair steps have to be considered.
- e.g. swapping SUBJ and DOBJ

a)	syntax	...
diese_1	det/2	...
scheibe_2	**dobj/3**	...
ist_3	root/nil	...
ein_4	den/5	...
hit_5	**subj/3**	...

$\Longrightarrow$

b)	syntax	...
diese_1	det/2	...
scheibe_2	**subj/3**	...
ist_3	root/nil	...
ein_4	det/5	...
hit_5	**dobj/5**	...

Hybrid Parsing

- the bare constraint-based parser itself is weak

Hybrid Parsing

- the bare constraint-based parser itself is weak
- but: constraints can be used as interface to external predictor components

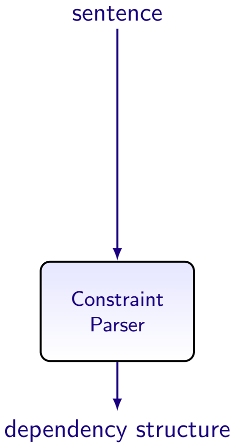
Hybrid Parsing

- the bare constraint-based parser itself is weak
- but: constraints can be used as interface to external predictor components
- predictors are all probabilistic, thus inherently unreliable
  $\rightarrow$ can their information still be useful?

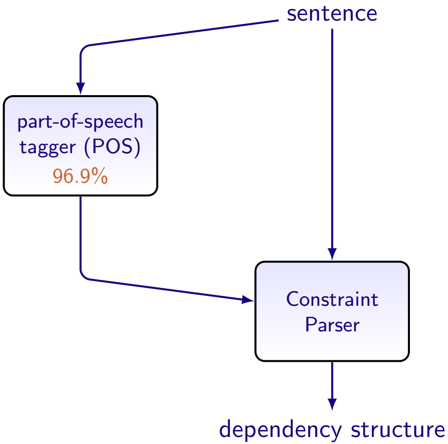
Hybrid Parsing

- the bare constraint-based parser itself is weak
- but: constraints can be used as interface to external predictor components
- predictors are all probabilistic, thus inherently unreliable
  $\rightarrow$ can their information still be useful?
- several predictors $\rightarrow$ consistency cannot be expected

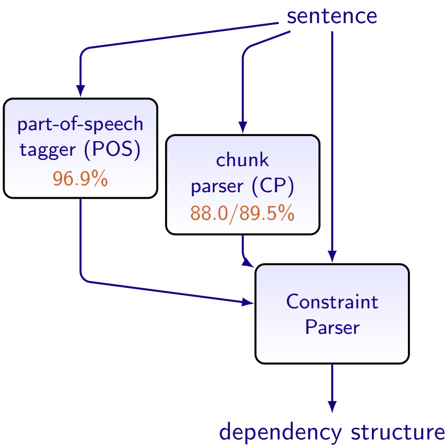
Hybrid Parsing



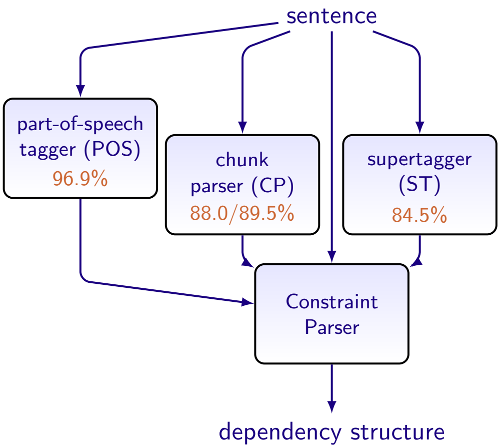
Hybrid Parsing



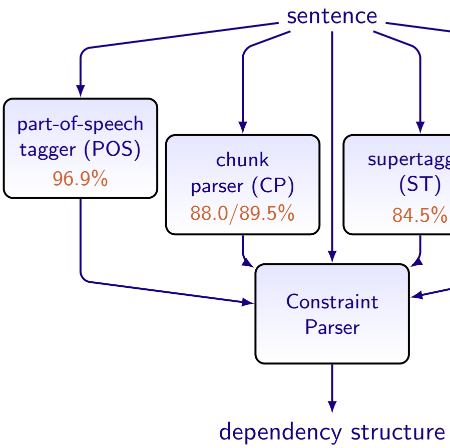
Hybrid Parsing



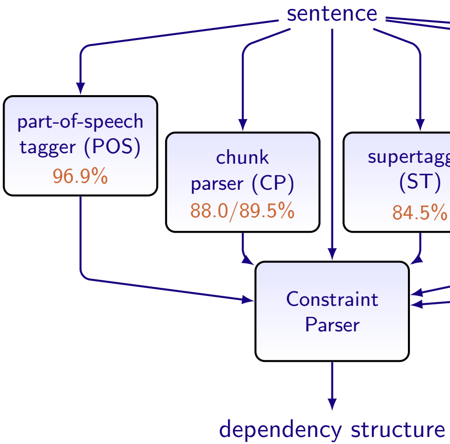
Hybrid Parsing



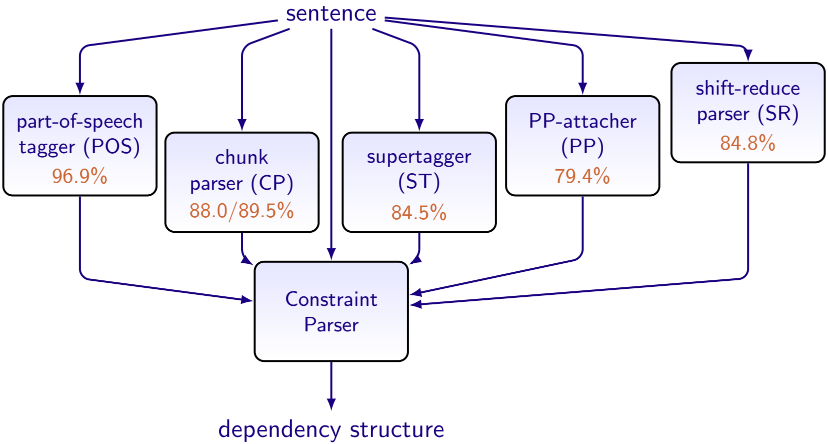
Hybrid Parsing



Hybrid Parsing



Hybrid Parsing



- predictor scores are mapped to constraint weights

Hybrid Parsing

- results on a 1000 sentence newspaper testset (Foth 2006)

	accuracy	
Predictors	unlabelled	labelled
0: none	72.6%	68.3%
1: POS only	89.7%	87.9%
2: POS+CP	90.2%	88.4%
3: POS+PP	90.9%	89.1%
4: POS+ST	92.1%	90.7%
5: POS+SR	91.4%	90.0%
6: POS+PP+SR	91.6%	90.2%
7: POS+ST+SR	92.3%	90.9%
8: POS+ST+PP	92.1%	90.7%
9: all five	92.5%	91.1%

Hybrid Parsing

- results on a 1000 sentence newspaper testset (Foth 2006)

	accuracy	
Predictors	unlabelled	labelled
0: none	72.6%	68.3%
1: POS only	89.7%	87.9%
2: POS+CP	90.2%	88.4%
3: POS+PP	90.9%	89.1%
4: POS+ST	92.1%	90.7%
5: POS+SR	91.4%	90.0%
6: POS+PP+SR	91.6%	90.2%
7: POS+ST+SR	92.3%	90.9%
8: POS+ST+PP	92.1%	90.7%
9: all five	92.5%	91.1%

Hybrid Parsing

- results on a 1000 sentence newspaper testset (FOTH 2006)

	accuracy	
Predictors	unlabelled	labelled
0: none	72.6%	68.3%
1: POS only	89.7%	87.9%
2: POS+CP	90.2%	88.4%
3: POS+PP	90.9%	89.1%
4: POS+ST	92.1%	90.7%
5: POS+SR	91.4%	90.0%
6: POS+PP+SR	91.6%	90.2%
7: POS+ST+SR	92.3%	90.9%
8: POS+ST+PP	92.1%	90.7%
9: all five	92.5%	91.1%

- net gain although the individual components are unreliable

Hybrid Parsing

- robust across different corpora (Foth 2006)

text type	sentences	average length	accuracy unlabelled	accuracy labelled
law text	1145	18.4	90.7%	89.6%

Hybrid Parsing

- robust across different corpora (Foth 2006)

text type	sentences	average length	accuracy unlabelled	accuracy labelled
law text	1145	18.4	90.7%	89.6%
online news	10000	17.3	92.0%	90.9%

Hybrid Parsing

- robust across different corpora (FOTH 2006)

text type	sentences	average length	accuracy unlabelled	accuracy labelled
law text	1145	18.4	90.7%	89.6%
online news	10000	17.3	92.0%	90.9%
Bible text	2709	15.9	93.0%	91.2%

Hybrid Parsing

- robust across different corpora (Foth 2006)

text type	sentences	average length	accuracy unlabelled	accuracy labelled
law text	1145	18.4	90.7%	89.6%
online news	10000	17.3	92.0%	90.9%
Bible text	2709	15.9	93.0%	91.2%
trivial literature	9547	13.8	94.2%	92.3%

Hybrid Parsing

- robust across different corpora (Foth 2006)

text type	sentences	average length	accuracy unlabelled	accuracy labelled
law text	1145	18.4	90.7%	89.6%
online news	10000	17.3	92.0%	90.9%
Bible text	2709	15.9	93.0%	91.2%
trivial literature	9547	13.8	94.2%	92.3%
Verbmobil dialogues	1316	8	90.3%	86.3%

Hybrid Parsing

- robust across different corpora (Foth 2006)

text type	sentences	average length	accuracy unlabelled	accuracy labelled
law text	1145	18.4	90.7%	89.6%
online news	10000	17.3	92.0%	90.9%
Bible text	2709	15.9	93.0%	91.2%
trivial literature	9547	13.8	94.2%	92.3%
Verbmobil dialogues	1316	8	90.3%	86.3%
serious literature	68	34	78.0%	75.4%

Relative Importance of Information Sources

Class	Purpose	Example	Importance
agree	rection and agreement	subjects have nominative case	1.02
cat	category cooccurrence	prepositions do not modify each other	1.13
dist	locality principles	prefer the shorter of two attachments	1.01
exist	valency	finite verbs must have subjects	1.04
init	hard constraints	appositions are nominals	3.70
lexical	word-specific rules	“entweder” requires following “oder”	1.02
order	word-order	determiners precede their regents	1.11
pos	POS tagger integration	prefer the predicted category	1.77
pref	default assumptions	assume nominative case by default	1.00
proj	projectivity	disprefer nonprojective coordinations	1.09
punc	punctuation	subclauses are marked with commas	1.03
root	root subordinations	only verbs should be tree roots	1.72
sort	sortal restrictions	“sein” takes only local predicatives	1.00
uniq	label cooccurrence	there can be only one determiner	1.00
zone	crossing of marker words	conjunctions must be leftmost dependents	1.00

Relative Importance of Information Sources

Class	Purpose	Example	Importance
init	hard constraints	appositions are nominals	3.70
pos	POS tagger integration	prefer the predicted category	1.77
root	root subordinations	only verbs should be tree roots	1.72
cat	category cooccurrence	prepositions do not modify each other	1.13
order	word-order	determiners precede their regents	1.11
proj	projectivity	disprefer nonprojective coordinations	1.09
exist	valency	finite verbs must have subjects	1.04
punc	punctuation	subclauses are marked with commas	1.03
agree	rection and agreement	subjects have nominative case	1.02
lexical	word-specific rules	"entweder" requires following "oder"	1.02
dist	locality principles	prefer the shorter of two attachments	1.01
pref	default assumptions	assume nominative case by default	1.00
sort	sortal restrictions	"sein" takes only local predicatives	1.00
uniq	label cooccurrence	there can be only one determiner	1.00
zone	crossing of marker words	conjunctions must be leftmost dependents	1.00

Selling Points

- robustness against ungrammatical input
- inherent diagnostic abilities:
  constraint violations can be interpreted as error diagnoses
  - transformation-based parsing is conflict-driven
  - crucial for interactive grammar development
  - applications for second language learning
- inherent anytime properties
  - interruptable
  - processing time can be traded for parsing accuracy

Selling Points

- framework for soft information fusion
  - syntax, semantics, information structure, ...
  - shallow processing components
- achieves always full disambiguation
- partial results can be obtained if needed
- you have to be **very** patient

Graph-based Dependency Parsing

- MST-parser (McDonald)

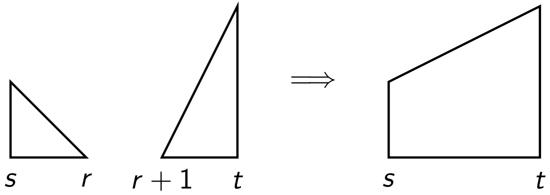
- large margin learning → scoring of candidate edges
- first order (unary) / second order (binary) constraints

- two step approach:
  - computation of bare attachments
  - labeling edges as a classification task

- problem: combining second order constraints and non-projective parsing
- projective tree building: Eisner (1996)
  - parse the left and the right dependents independently
  - join the partial trees later

Graph-based Dependency Parsing

- to build an incomplete subtree from word index s to t find a word index r ($s \leq r < t$) which maximizes the sum of the scores of the two complete subtrees plus the score of the edge from s to t

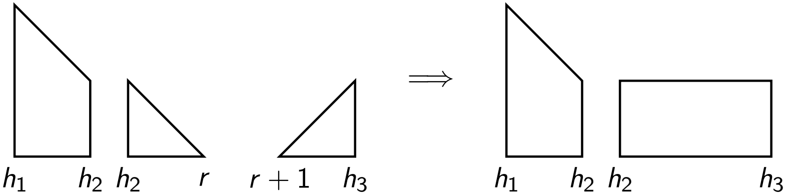


Graph-based Dependency Parsing

- extension to second order constraints:
  - establishing a dependency in two phases
  - sibling creation + head attachment

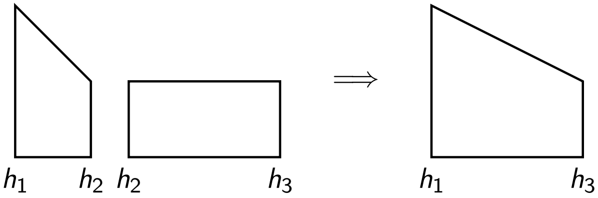
Graph-based Dependency Parsing

- extension to second order constraints:
  - establishing a dependency in two phases
  - sibling creation + head attachment
- to establish an edge between h_3 and h_1, given that an edge between h_2 and h_1 had already been established, find a word index r ($h_2 \leq r < h_3$) that maximizes the score of making h_2 and h_3 sibling nodes



Graph-based Dependency Parsing

- delay the completion of an item until all the sibling nodes have been collected



MST-Parser

- generating non-projective attachments by tree transformation
- labeling of edges as a classification task

Graph-based Dependency Parsing

- re-evaluation of MST on the WCDG annotations

Graph-based Dependency Parsing

- re-evaluation of MST on the WCDG annotations
- with interpunction

	accuracy[%]	
	structural	labelled
MST parser	91.9	89.1
WCDG (POS tagger only)	89.7	87.9
WCDG (all predictors)	92.5	91.1

- without interpunction

	accuracy[%]	
	structural	labelled
MST on NEGRA	90.5	87.5
MST on TIGER (CoNLL 2006)	90.4	87.3

Transition-based Dependency Parsing

- MaltParser Nivre (2004): choice between four parser actions:
  shift / left-attach + reduce / right-attach + shift / reduce

Transition-based Dependency Parsing

- MaltParser Nivre (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce

Jetzt

Transition-based Dependency Parsing

- MaltParser NIVRE (2004): choice between four parser actions:
  shift / left-attach + reduce / right-attach + shift / reduce

Jetzt

Jetzt

Transition-based Dependency Parsing

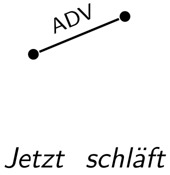
- MaltParser NIVRE (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce

Jetzt *schläft*

Jetzt

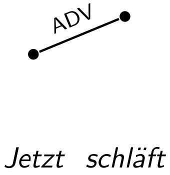
Transition-based Dependency Parsing

- MaltParser NIVRE (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce



Transition-based Dependency Parsing

- MaltParser NIVRE (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce



Transition-based Dependency Parsing

- MaltParser Nivre (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce

Transition-based Dependency Parsing

- MaltParser Nivre (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce

Transition-based Dependency Parsing

- MaltParser NIVRE (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce

Jetzt schläft das Kind

das
schläft

Transition-based Dependency Parsing

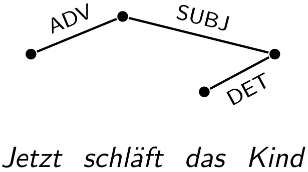
- MaltParser NIVRE (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce

Transition-based Dependency Parsing

- MaltParser Nivre (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce

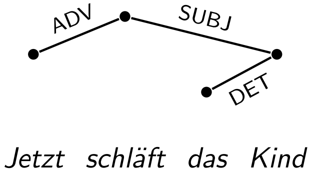
Transition-based Dependency Parsing

- MaltParser NIVRE (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce



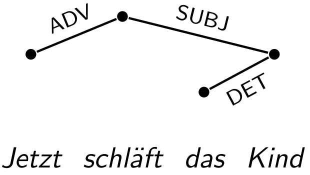
Transition-based Dependency Parsing

- MaltParser Nivre (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce



Transition-based Dependency Parsing

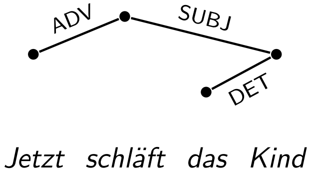
- MaltParser Nivre (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce



- support vector machine trained on the parse history to predict the best next parser action

Transition-based Dependency Parsing

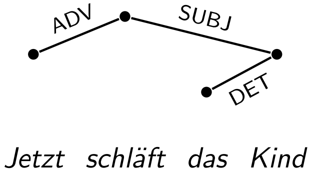
- MaltParser Nivre (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce



- support vector machine trained on the parse history to predict the best next parser action
- parser takes deterministic decisions: eager processing

Transition-based Dependency Parsing

- MaltParser NIVRE (2004): choice between four parser actions: shift / left-attach + reduce / right-attach + shift / reduce



- support vector machine trained on the parse history to predict the best next parser action
- parser takes deterministic decisions: eager processing
- fully left-to-right incremental processing, but with delay

Parser Combination

- Co-parsing
- Reparsing
- Co-training

Co-Parsing

- Khmylko et al. (2007)
- WCDG has proven useful to integrate external predictions
- so far, all predictors consider
  - partial aspects of the parsing problem
    tagger, supertagger, pp-attacher, ...,
  - or use a different representation
    projective vs. non-projective

Co-Parsing

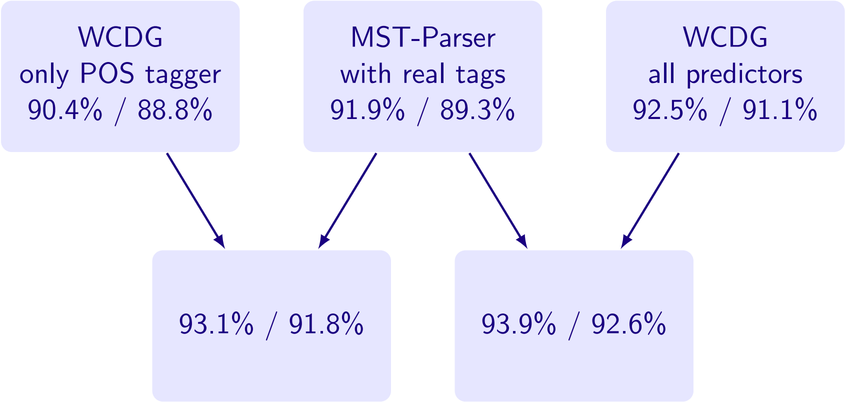
- Khmylko et al. (2007)
- WCDG has proven useful to integrate external predictions
- so far, all predictors consider
  - partial aspects of the parsing problem
    tagger, supertagger, pp-attacher, ...,
  - or use a different representation
    projective vs. non-projective
- What happens ...
  - ... if two parsers for exactly the same task are combined?
  - ... if the predictor becomes superior?

Co-Parsing

- using the output of MST to guide WCDG
- three additional constraints
  - Is the modifiee the same?
  - Is the root node the same?
  - Is the label the same?
- separate constraint weights for attachment and label

Hybrid Parsing

- What happens if the predictor becomes superior?



- high degree of synergy

Reparsing

- Sagae and Lavie (2006)

- combining the results of arbitrary many dependency parsers into a common dependency graph
- combination by joining the node and edge sets
  - as long as one of the parse trees is a valid dependency tree, the graph is connected

Reparsing

- edges receive a weight
  - if an edge is proposed by more than one parser, weights are added
- three different weighting schemes
  - W1: all dependencies receive the same weight
  - W2: the dependencies from different parsers receive different weights
  - W3: the dependencies receive different weights with respect to the parser that generated them and the POS tag of the modifier

Reparsing

- four different parsers
  - LR: deterministic shift-reduce parser (left-to-right)
  - LR: deterministic shift-reduce parser (right-to-left)
  - LRRL: deterministic multi-pass shift-reduce parser (left-to-right and right-to-left)
  - MST-Parser

Reparsing

System	Accuracy	Root Acc.
LR	91.0	92.6
RL	90.1	86.3
LRRL	89.6	89.1
MST	90.9	94.2
Reparse W1	91.8	96.0
Reparse W2	92.1	95.9
Reparse W3	**92.7**	**96.6**

Co-Training

- McDonald and Nivre (2008/2010)
- MST-Parser and MaltParser have roughly the same accuracy
- but commit (partly) complementary errors
  - MST-Parser: higher accuracy near the root of the tree
  - MaltParser: higher accuracy near the leaves
- caused by alternative training/inference approaches
  - MST-Parser: global learning and exhaustive search, but locally restricted features
  - MaltParser: rich feature set, but local learning and greedy inference

Co-Training

- combination by providing parser A with information about the result of parser B (and vive versa) *during training*
  - enriching the feature set by additional features
- two different systems
  - MaltParser informed by MST-Parser: MALT_{MST}
  - MST-Parser informed by MaltParser: MST_{MALT}

Co-Training

Language	MST	MST_{Malt}	Malt	Malt_{MST}	oracle graph	oracle arc
Arabic	66.91	**68.64** (+1.73)	66.71	67.80 (+1.09)	70.3	75.8
Bulgarian	87.57	**89.05** (+1.48)	87.41	88.59 (+1.18)	90.7	92.4
Chinese	85.90	**88.43** (+2.53)	86.92	87.44 (+0.52)	90.8	91.5
Czech	80.18	**82.26** (+2.08)	78.42	81.18 (+2.76)	84.2	86.6
Danish	84.79	**86.67** (+1.88)	84.77	85.43 (+0.66)	87.9	89.6
Dutch	79.19	**81.63** (+2.44)	78.59	79.91 (+1.32)	83.5	86.4
German	87.34	**88.46** (+1.12)	85.82	87.66 (+1.84)	89.9	92.0
Japanese	90.71	91.43 (+0.72)	91.65	**92.20** (+0.55)	93.2	94.1
Portugese	86.82	87.50 (+0.68)	87.60	**88.64** (+1.04)	90.0	91.6
Slovene	73.44	**75.94** (+2.50)	70.30	74.24 (+3.94)	77.2	80.7
Spanish	82.25	**83.99** (+1.74)	81.29	82.41 (+1.12)	85.4	88.2
Swedish	82.55	**84.66** (+2.11)	84.58	84.31 (–0.27)	86.8	88.8
Turkish	63.19	64.29 (+1.10)	65.58	**66.28** (+0.70)	69.3	72.6
Average	80.83	**82.53** (+1.70)	80.75	82.01 (+1.27)	84.5	86.9