Computational Linguistics

1. Natural Language and the Computer
2. Words and Wordforms
3. Phrases and Sentences
4. Discourse: Texts and Dialogs

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 Strategies

- rule application from left to right: top-down analysis
  - derivation of a sentence from the start symbol

    S
    NP VP
    N V NP
    John sees NP
    John sees Mary

- rule application from right to left: bottom up analysis
  - derivation of the start symbol from the sentence:

    John sees Mary
    N V N
    NP V NP
    NP VP
    S

Parsing strategies

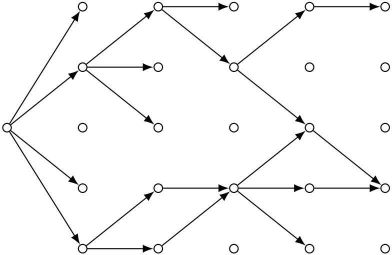
- all alternatives for rule applications need to be checked
- ambiguities do not allow local decisions
- lexical ambiguities: *green/VINF/VFIN/NN/ADJ/ADV*
- structural ambiguities as a consequence of lexical ones

Parsing strategies

- purely structural ambiguities
  [$_{NP}$ the man [$_{PP}$ with the hat [$_{PP}$ on the stick]]]
  [$_{NP}$ the man [$_{PP}$ with the hat] [$_{PP}$ on the stick]]
  . . . , weil [$_{NP}$ dem Sohn des Meisters] [$_{NP}$ Geld] fehlt.
  . . . , weil [$_{NP}$ dem Sohn] [$_{NP}$ des Meisters Geld] fehlt.
- local ambiguities can be resolved during subsequent analysis steps
- global ambiguities remain until the analysis finishes

Parsing strategies

- parsing as search
    - alternative rule applications create a search space



- recombination of paths reduces computation time
    - without recombination: $\mathcal{O}(k^n)$
    - with recombination: $\mathcal{O}(n^3)$
- two paths can be recombined
    - if they consists of the same sequence of constituent types
    - aligned to the same substrings of terminal symbols

Parsing strategies

- expectation driven (top-down, expand-reduce)
  - problem: left/right recursive rules cause termination problems
    - even in case of indirect recursion:

      $X \rightarrow Y\ a$

      $Y \rightarrow X$
  - solution: transformation into a weakly equivalent grammar without left/right recursion
    - linguistically motivated derivation structure is lost
    - workaround: generating a separated structure by means of unification

Parsing strategies

- data driven (bottom-up, shift-reduce)
  - problem: empty productions (linguistically motivated) X $\rightarrow \epsilon$
    - perhaps "licensing" empty categories by lexical nodes
  - problem: unary rules which form a cycle
    - avoid them completely

Parsing strategies

- depth-first
    - alternative rule applications are tried later on
    - storing them on a stack
- breadth-first
    - alternative rule applications are tried in "parallel"
    - maintaining the alternatives in a queue

Parsing strategies

- left-to-right
    - input is processed beginning from its left side
- right-to-left
    - input is processed beginning from its right side

Parsing strategies

- mixed strategies
    - Left-Corner-Parsing: top-down analysis activating a rule by its left corner
    - robust parsing for erroneous input:
      bottom-up analysis and subsequent top-down reconstruction in case of failure (MELLISH 1989)
    - island parsing: bidirectional analysis starting from reliable hypotheses (e.g. for speech recognition results)

Chart parsing

- effciency problem: repetition of analysis steps on alternative analysis paths
- recombination of search paths is required
- data
  - German with head-final verb group
  - unmarked case: subclause ordering

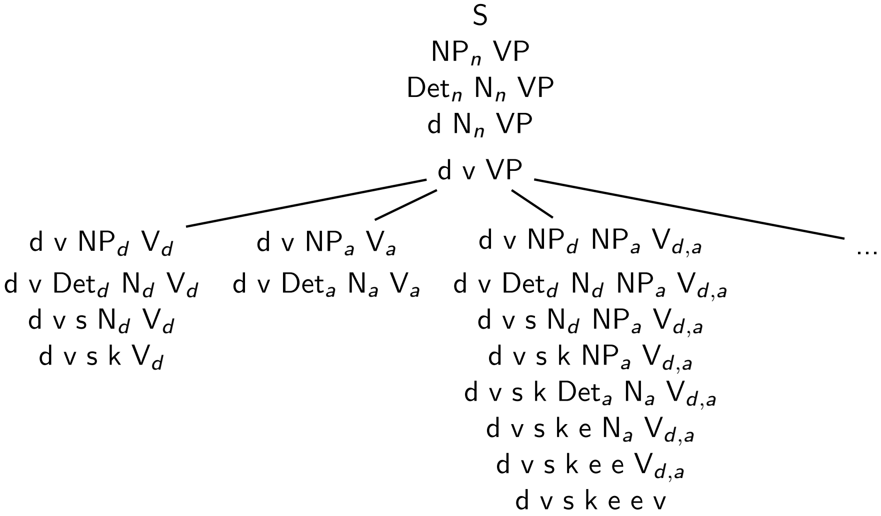
..., *weil der Vater seine Kinder liebt*.
..., weil der Vater seinen Kindern glaubt.
..., weil der Vater seinen Kindern ein Eis versprach.
..., weil der Vater seinen Kindern mit einer Strafe droht.

Chart parsing

- grammar

  S' $\rightarrow$ Konj S
  S $\rightarrow$ NP_n VP
  VP $\rightarrow$ NP_a V_a
  VP $\rightarrow$ NP_d V_d
  VP $\rightarrow$ NP_d NP_a $\mathrm{V}_{d,a}$
  VP $\rightarrow$ NP_d $\mathrm{PP}_{mit,d}$ $\mathrm{V}_{d,mit}$
  NP_X $\rightarrow$ Det_X N_X
  $\mathrm{PP}_{X,Y}$ $\rightarrow$ P_X NP_Y
- Example analysis: top-down, depth-first
  ... der Vater seinen Kindern ein Eis versprach.

Chart parsing



S

$\mathrm{NP}_n\ \mathrm{VP}$

$\mathrm{Det}_n\ \mathrm{N}_n\ \mathrm{VP}$

$\mathrm{d}\ \mathrm{N}_n\ \mathrm{VP}$

$\mathrm{d\ v\ VP}$

$\mathrm{d\ v}\ \mathrm{NP}_d\ \mathrm{V}_d$

$\mathrm{d\ v}\ \mathrm{Det}_d\ \mathrm{N}_d\ \mathrm{V}_d$

$\mathrm{d\ v\ s}\ \mathrm{N}_d\ \mathrm{V}_d$

$\mathrm{d\ v\ s\ k}\ \mathrm{V}_d$

$\mathrm{d\ v}\ \mathrm{NP}_a\ \mathrm{V}_a$

$\mathrm{d\ v}\ \mathrm{Det}_a\ \mathrm{N}_a\ \mathrm{V}_a$

$\mathrm{d\ v}\ \mathrm{NP}_d\ \mathrm{NP}_a\ \mathrm{V}_{d,a}$

$\mathrm{d\ v}\ \mathrm{Det}_d\ \mathrm{N}_d\ \mathrm{NP}_a\ \mathrm{V}_{d,a}$

$\mathrm{d\ v\ s}\ \mathrm{N}_d\ \mathrm{NP}_a\ \mathrm{V}_{d,a}$

$\mathrm{d\ v\ s\ k}\ \mathrm{NP}_a\ \mathrm{V}_{d,a}$

$\mathrm{d\ v\ s\ k}\ \mathrm{Det}_a\ \mathrm{N}_a\ \mathrm{V}_{d,a}$

$\mathrm{d\ v\ s\ k\ e}\ \mathrm{N}_a\ \mathrm{V}_{d,a}$

$\mathrm{d\ v\ s\ k\ e\ e}\ \mathrm{V}_{d,a}$

$\mathrm{d\ v\ s\ k\ e\ e\ v}$

...

Chart parsing

- well-formed substring table (chart)
    - directed acyclic graph (DAG) with
        - one source (beginning of the sentence)
        - one sink (end of the sentence) and
        - a total precedence relation on the nodes
    - edges correspond to successfully recognized constituents

Chart parsing

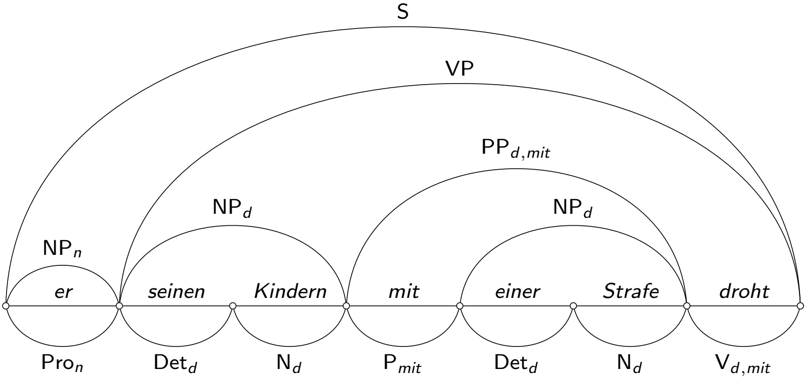


Chart parsing

	1	2	3	4	5	6	7
0	er Pro_n NP_n						S
1		seinen Det_d	NP_d				VP
2			Kindern N_d				
3				mit P_{mit}		PP_{mit}	
4					einer Det_d	NP_d	
5						Strafe N_d	
6							droht $\text{V}_{d,mit}$

Chart parsing

- Cocke-Younger-Kasami algorithm (KASAMI 1965, YOUNGER 1967)
- grammar in Chomsky-normalform
  - binary branching rules: X → Y Z
  - pre-terminal/lexical rules: X → a

Chart parsing

- properties of the CYK algorithm
  1. length o the derivation is constant:
     n lexical rules + n-1 binary branching rules
  2. number of binary partitionings of a sentence is constant: n-1

     `((a) (b c d))`
     `((a b) (c d))`
     `((a b c) (d))`
  3. no structural ambiguities due to different segmentations of the sentence

     VP → NP NP V
     VP → NP V
     VP → V

Tabellenparsing

- CYK algorithm

  1. initialization of the table

```
for $i = 0$ to $n - 1$:
    $\text{CHART}_{i,i+1} \Leftarrow \{ \text{X} \mid \text{X} \in \text{V}_T \text{ and } \text{w}_{i+1} \in \text{X} \}$
```

  2. computation of the remaining entries

```
for $k = 2$ to $n$:
    for $i = 0$ to $n - k$:
        $j \Leftarrow i + k$
        $\text{CHART}_{i,j} \Leftarrow \{ \text{A} \mid (\text{A} \rightarrow \text{X Y}) \in \text{R} \wedge \exists \text{ m . } (\text{X} \in \text{CHART}_{i,m}$
                  $\wedge \text{ Y} \in \text{CHART}_{m,j}, \quad \text{mit } i < m < j \}$
if $\text{S} \in \text{CHART}_{0,n}$
    then RETURN(*true*)
    else RETURN(*false*)
```

Chart parsing

- bottom-up analysis
  - time complexity $\mathcal{O}(n^3)$
  - memory complexity $\mathcal{O}(n^2)$
  - achieved by reycling of intermediate results (recombination)
- disadvantage: still constituents are generated which cannot be integrated into a larger structure (dead ends)
  → EARLEY parser

Chart parsing

- active chart
  - extension: even incomplete attempts of rule applications are recorded in the chart
  - active edges:
    open expectations for the right context
    notation: $\langle$ a, b, A $\rightarrow$ B . C D $\rangle$
  - inactive edges:
    completely satisfied expectations for the right context
    notation: $\langle$ a, b, A $\rightarrow$ B C D . $\rangle$

Chart parsing

- TD rule (initialisation)

  For all rules A $\rightarrow$ w_1 where A is a start symbol of the grammar, add an edge $\langle$ 0, 0, A $\rightarrow$. w_1 $\rangle$ to the chart.

- rule: S $\rightarrow$ NP_n VP

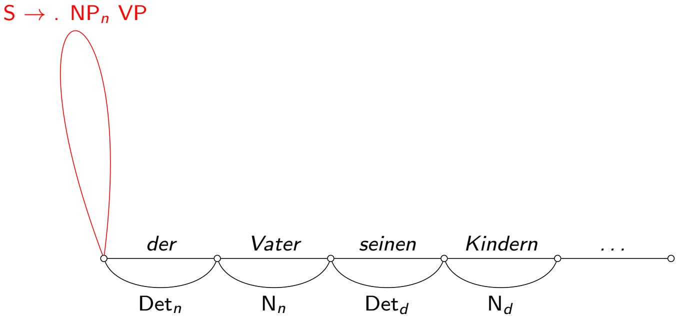


Chart parsing

- TD-rule (edge introduction)

  When adding a rule $\langle$ i, j, A $\rightarrow$ w_1 . B w_2 $\rangle$ to the chart, add for each rule B $\rightarrow$ w_3 an edge $\langle$ j, j, B $\rightarrow$. w_3 $\rangle$.
- rule: $NP_X \rightarrow Det_X\ N_X$

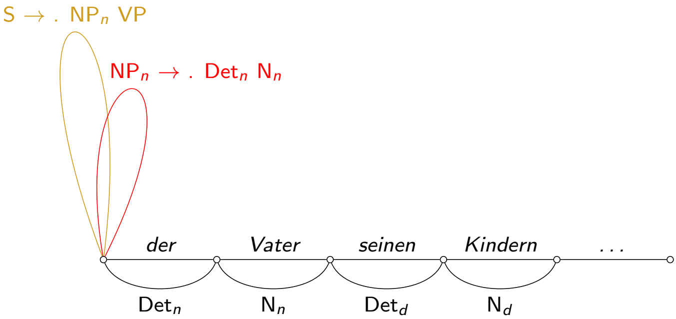


Chart parsing

- fundamental rule (edge expansion)

  If the chart contains two edges $\langle$ i, j, A $\rightarrow$ w_1 . B w_2 $\rangle$ and $\langle$ j, k, B $\rightarrow$ w_3 . $\rangle$, add a third edge $\langle$ i, k, A $\rightarrow$ w_1 B . $w_2$$\rangle$.

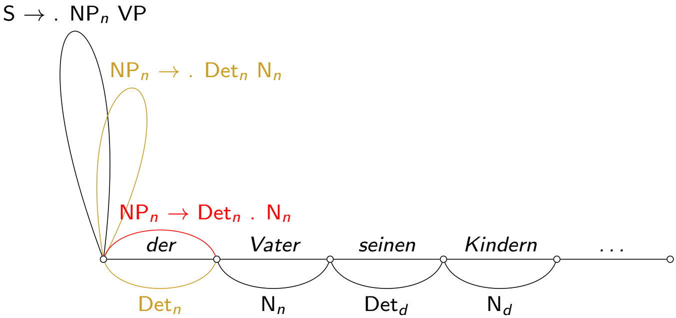


Chart parsing

- repeated application of the fundamental rule

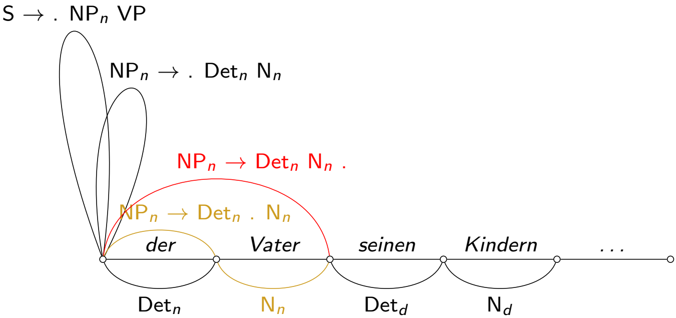


Chart parsing

- repeated application of the fundamental rule

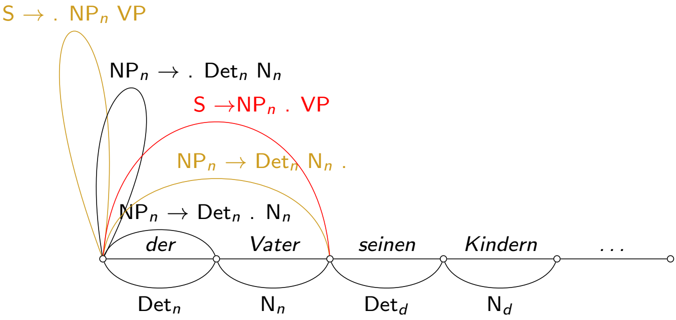


Chart parsing

- repeated application of the top-down rule

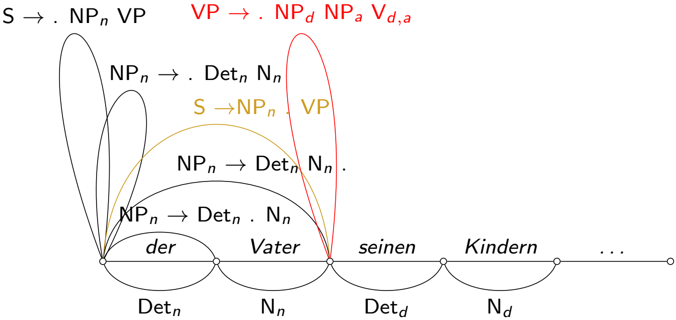


Chart-Parsing

- repeated application of the top-down rule

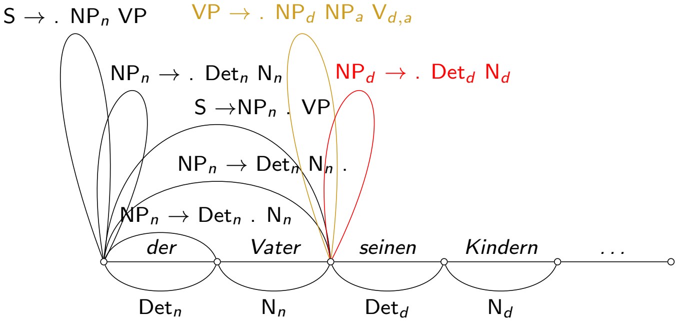


Chart parsing

- repeated application of the fundamental rule

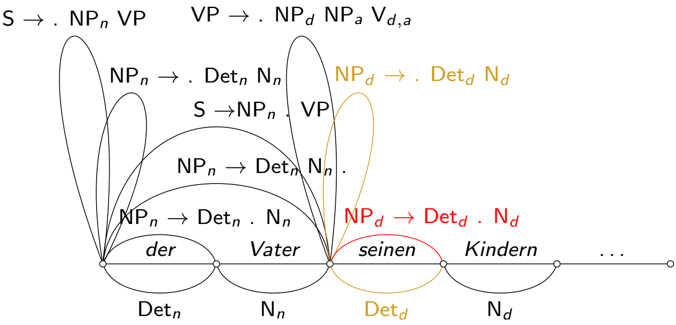


Chart parsing

- repeated application of the fundamental rule

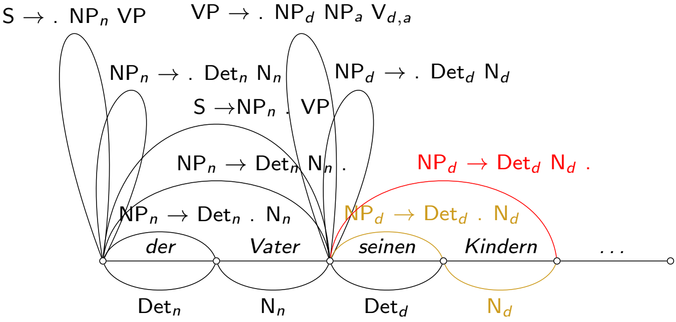


Chart parsing

- repeated application of the fundamental rule

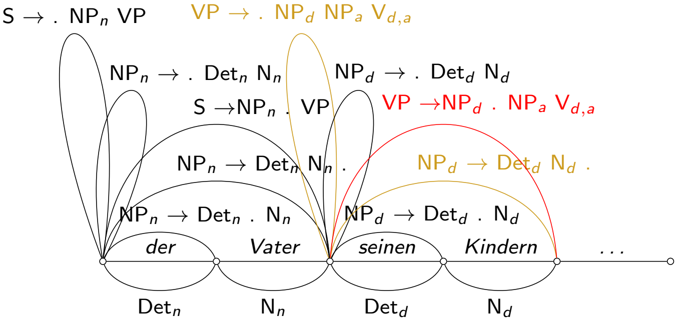
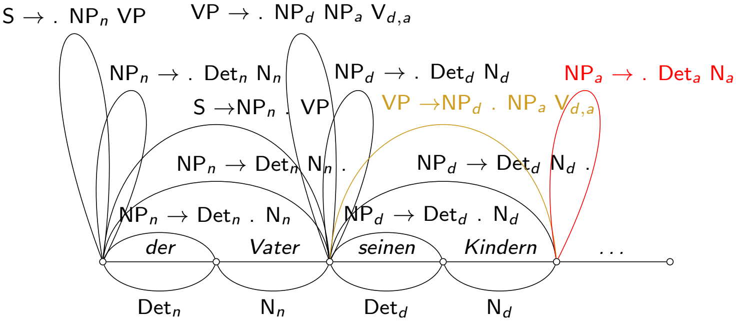


Chart parsing

- repeated application of the top-down rule



$S \rightarrow .\ NP_n\ VP$

$VP \rightarrow .\ NP_d\ NP_a\ V_{d,a}$

$NP_n \rightarrow .\ Det_n\ N_n$

$NP_d \rightarrow .\ Det_d\ N_d$

$NP_a \rightarrow .\ Det_a\ N_a$

$S \rightarrow NP_n\ .\ VP$

$VP \rightarrow NP_d\ .\ NP_a\ V_{d,a}$

$NP_n \rightarrow Det_n\ N_n\ .$

$NP_d \rightarrow Det_d\ N_d\ .$

$NP_n \rightarrow Det_n\ .\ N_n$

$NP_d \rightarrow Det_d\ .\ N_d$

der *Vater* *seinen* *Kindern* ...

Det_n N_n Det_d N_d

Chart parsing

- Earley algorithm (Earley 1970)
  - for arbitrary context free grammars
    - including recursion, cycles and ϵ-rules
  - mixed top-down/bottom-up strategy, to avoid adding of edges (constituents) which cannot be incorporated into larger ones
    1. top-down condition:
       only edges are added for which the left context is compatible with the requirements of the grammar
    2. bottom-up condition:
       the already applied part of the rule is compatible with the input data

Chart parsing

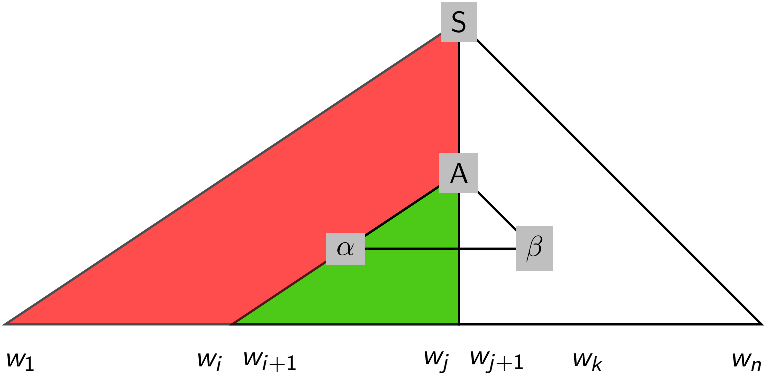


Chart parsing

- elementary operations
  - expand (top-down rule, edge introduction)
  - complete (fundamental rule, edge expansion)
  - shift (introduction of lexical edges)
- different search strategies (depth-first/breadth-first/best-first) are possible depending on the agenda management

Chart parsing

- EARLEY-Algorithmus
  1. initialization

```
for all $(S \rightarrow \beta) \in R$: $CHART_{0,0} \Leftarrow \langle S, \emptyset, \beta \rangle$
Apply EXPAND to the previously generated edges
until no new edges can be added.
```

  2. computation of the remaining edges

```
for $j = 1, \ldots, n$:
    for $i = 0, \ldots, j$:
        compute $CHART_{i,j}$:
            1. apply SHIFT to all relevant edges in $CHART_{i,j-1}$
            2. apply EXPAND and COMPLETE until no new
               edges can be produced.
if $\langle S, \beta, \emptyset \rangle \in CHART_{0,n}$
    then RETURN(true) else RETURN(false)
```

Chart parsing

- a chart-based algorithm is only a recognizer
- extending it to a real parser:
  - extraction of structural descriptions (trees, derivations) from the chart in a separate step
  - basis: maintaining a pointer from an edge to the activating edge in the fundamental rule
  - "collecting" the trees starting with all inactive S-edges

Chart parsing

- time complexity
  - $\mathcal{O}(n^3 \cdot |G^2|)$
  - for deterministic grammars: $\mathcal{O}(n^2)$
  - in many relevant cases: $\mathcal{O}(n)$
- complexity result is only valid for constructing the chart
- tree extraction might require exponential effort in case of exponentially many results

Chart parsing

- space complexity
  - $\mathcal{O}(n^2)$
  - due to the reuse of intermediate results
    - holds only for atomic non-terminal symbols

- chart is a general data structur to maintain intermediate results during parsing

  - alternative parsing strategies are possible
  - e.g. bottom-up

Chart parsing

- bottom-up rule (edge introduction)

  When adding a rule $\langle$ i, j, B $\rightarrow$ w_1 $\rangle$ for every rule A $\rightarrow$ B w_2 add another edge $\langle$ i, i, A $\rightarrow$. B w_2 $\rangle$

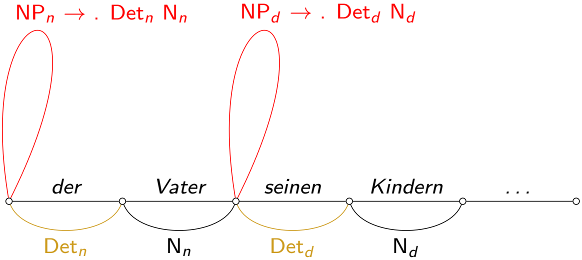


Chart parsing

- application of the fundamental rule

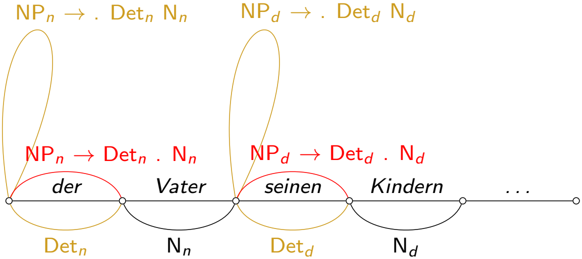


Chart parsing

- application of the fundamental rule

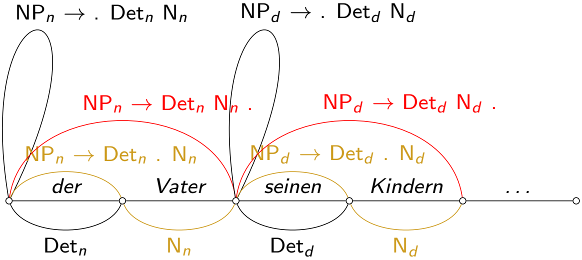


Chart parsing

- Application of the bottom-up rule

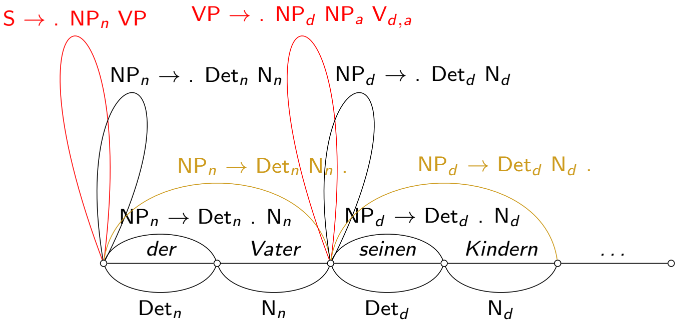


Chart parsing

- application of the fundamental rule

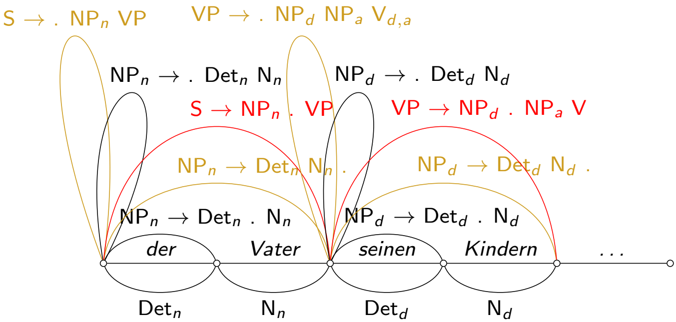


Chart parsing

- parsing is a monotonic procedure of information gathering
  - edges are never deleted from the chart
  - even unsuccessful rule applications are kept

  - edges which cannot be expanded further

- duplicating analysis effort is avoided
  - edge is only added to the chart if not already there

Chart parsing

- agenda
  - list of active edges
  - can be sorted according to different criteria
  - stack: depth-first
  - queue: breadth-first
  - TD-rule: expectation-driven analysis
  - BU-rule: data -driven analysis

Chart parsing

- flexible control for hybrid strategies
- left-corner parsing
    - TD-parsing, but only those rules are activated, which can derive a given lexical category (left corner) directly or indirectly
    - mapping between rules and their possible left corners is computed from the grammar at compile time
    - variant: head-corner parsing

Chart parsing

- best-first parsing
    - sorting the agenda according to confidence values
        - hypothesis scores of speech recognition
        - rule weights (e.g. relative frequency in a tree bank)

Phrases and Sentences

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

Probabilistic Parsers

- common problem of all purely symbolic parser
    - high degree of output ambiguity
    - even in case of (very) fine-grained syntactic modelling
    - despite of a dissatisfyingly low coverage
- coverage and degree of output ambiguity are typically highly correlated

Probabilistic Parsers

- output ambiguity
  - *Hinter dem Betrug werden die gleichen Täter vermutet, die während der vergangenen Tage in Griechenland gefälschte Banknoten in Umlauf brachten.*
  - *The same criminals are supposed to be behind the deceit who in Greece over the last couple of days brought falsified money bills into circulation.*

Probabilistic Parsers

- output ambiguity
  - *Hinter dem Betrug werden die gleichen Täter vermutet, die während der vergangenen Tage in Griechenland gefälschte Banknoten in Umlauf brachten.*
  - *The same criminals are supposed to be behind the deceit who in Greece over the last couple of days brought falsified money bills into circulation.*
  - Paragram (KUHN UND ROHRER 1997): 92 readings

Probabilistic Parsers

- output ambiguity
  - *Hinter dem Betrug werden die gleichen Täter vermutet, die während der vergangenen Tage in Griechenland gefälschte Banknoten in Umlauf brachten.*
  - *The same criminals are supposed to be behind the deceit who in Greece over the last couple of days brought falsified money bills into circulation.*
  - Paragram (KUHN UND ROHRER 1997): 92 readings
  - Gepard (LANGER 2001): 220 readings

Probabilistic Parsers

- output ambiguity
  - *Hinter dem Betrug werden die gleichen Täter vermutet, die während der vergangenen Tage in Griechenland gefälschte Banknoten in Umlauf brachten.*
  - *The same criminals are supposed to be behind the deceit who in Greece over the last couple of days brought falsified money bills into circulation.*
  - Paragram (KUHN UND ROHRER 1997): 92 readings
  - Gepard (LANGER 2001): 220 readings
  - average ambiguity for a corpus of newspaper texts: 78
    with an average sentence length of 11.43 syntactic words (Gepard)

Probabilistic Parsers

- output ambiguity
  - *Hinter dem Betrug werden die gleichen Täter vermutet, die während der vergangenen Tage in Griechenland gefälschte Banknoten in Umlauf brachten.*
  - *The same criminals are supposed to be behind the deceit who in Greece over the last couple of days brought falsified money bills into circulation.*
  - Paragram (Kuhn und Rohrer 1997): 92 readings
  - Gepard (Langer 2001): 220 readings
  - average ambiguity for a corpus of newspaper texts: 78
    with an average sentence length of 11.43 syntactic words (Gepard)
  - extreme case: $6.4875 \cdot 10^{22}$ for a single sentence (Block 1995)

Probabilistic Parsers

- sources of ambiguity:
  - lexical ambiguity
  - attachment
    - *We saw the Eiffel Tower flying to Paris.*
  - coordination:
    - *old men and women*
  - NP segmentation
    - *. . . der Sohn des Meisters Geld*

Probabilistic Parsers

- example: PP-attachment
  the ball with the dots in the bag on the table
- grows exponentially (catalan) with the number of PPs

$$C(n) = \frac{1}{n+1} \binom{2n}{n}$$

# PPs	# parses
2	2
3	5
4	14
5	132
6	469
7	1430
8	4867

Probabilistic Parsers

- coverage
  - partial parser (WAUSCHKUHN 1996): 56.5% of the sentences
  - Gepard: 33.51%
  - on test suites (better lexical coverage, shorter and less ambiguous sentences) up to 66%

Probabilistic Parsers

- alternative: probabilistic context-free grammars (PCFG)
- estimation of derivation probabilities for all rules

$$\Pr(N \rightarrow \zeta)$$

or

$$\Pr(N \rightarrow \zeta | N) \quad \text{mit} \quad \sum_{\zeta} \Pr(N \rightarrow \zeta) = 1$$

- e.g.

S → NP VP	0.8
S → Aux NP VP	0.15
S → VP	0.05

Probabilistic Parsers

- language models: assigning a probability to a terminal string

$$\Pr(w_{1,n}) = \sum_{t_{1,n}} \Pr(t_{1,n})$$

(several derivations for a sentence)

$$= \sum_{t_{1,n}} \prod_{r_j \in t_{1,n}} \Pr(r_j)$$

- determining the most probable word form sequence

Probabilistic Base Model

- disambiguation: determining of the most probable derivation

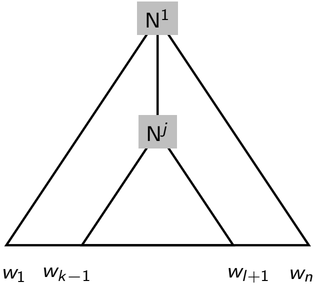
$$
\begin{aligned}
t_{1,n} &= \arg \max_{t_{1,n} \in T} \Pr(t_{1,n}) \\
&= \arg \max_{t_{1,n} \in T} \prod_{r_j \in t_{1,n}} \Pr(r_j)
\end{aligned}
$$

Probabilistic Parsers

- independence assumption:

$$\Pr(N^j_{k,l} \rightarrow \zeta | N^1, \ldots., N^{j-1}, w_1, \ldots, w_{k-1}, w_{l+1}, \ldots, w_n)$$

$$= \Pr(N^j_{k,l} \rightarrow \zeta)$$



Parser Evaluation

- PARSEVAL-metric (Black et al. 1991)
- comparison with a reference annotation (*gold standard*)
- labelled recall

$$LR = \frac{\text{\# correct constituents in the output}}{\text{\# constituents in the gold standard}}$$

- labelled precision

$$LP = \frac{\text{\# correct constituents in the output}}{\text{\# constituents in the output}}$$

Parser Evaluation

- crossing brackets
  a constituent of a parse tree contains parts of two constituents from the reference, but not the complete ones.

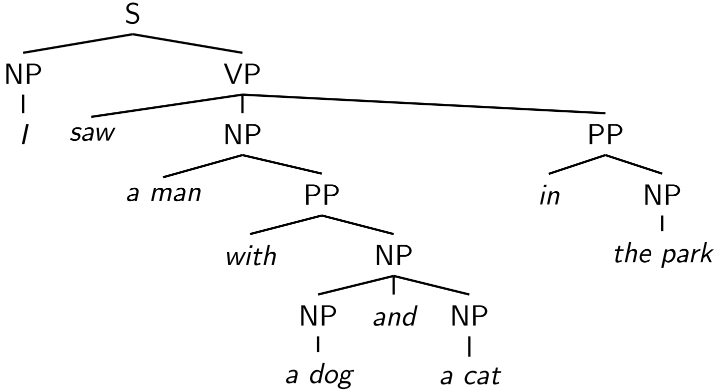
```
output:         [ [ A  B     C ] [ D  E ] ]
gold standard:  [ [ A  B ] [ C     D  E ] ]
```

$$CB = \frac{\#\ \text{crossing brackets}}{\#\ \text{sentences}}$$

$$0CB = \frac{\#\ \text{sentences without crossing brackets}}{\#\ \text{sentences}}$$

Parser Evaluation

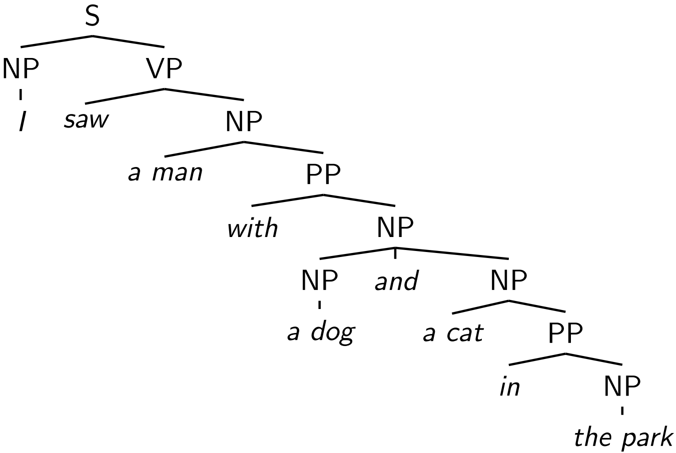
- How meaningful are the results?
- gold standard:



[I [saw [[a man] [with [[a dog] and [a cat]]]] [in [the park]]]]

Parser Evaluation

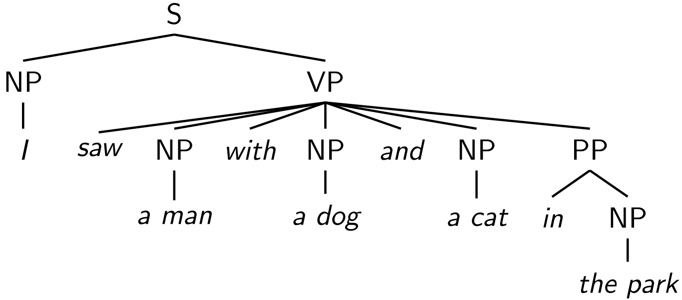
- 1st result: one erroneous attachment



[I [saw [[a man] [with [[a dog] and [[a cat] [in [the park]]]]]]]]

Parser Evaluation

- 2nd result: almost flat analysis
  - the parser tries to avoid any decisions on attachments



[I [saw [a man] with [a dog] and [a cat] [in [the park]]]]

Parser Evaluation

- 1st result
  [I [saw [[a man] [with [[a dog] and [[a cat] [in [the park]]]]]]]]
  [I [saw [[a man] [with [[a dog] and [a cat]]]] [in [the park]]]]
  $LR = \frac{7}{10} = 0.7 \qquad LP = \frac{7}{11} = 0.64 \qquad CB = \frac{3}{1} = 3$
- 2nd result
  [I [saw [a man] with [a dog] and [a cat] [in [the park]]]]
  [I [saw [[a man] [with [[a dog] and [a cat]]]] [in [the park]]]]
  $LR = \frac{7}{10} = 0.7 \qquad LP = \frac{7}{7} = 1 \qquad CB = \frac{0}{1} = 0$
- alternative (Lin 1996):
  transformation of the PS-tree into a dependency tree and evaluation of attachment errors

Parser Evaluation

- training: estimation of rule-application probabilities
- simplest case: treebank grammars
  (CHARNIAK 1996)

$$\Pr(N \to \zeta | N) = \frac{C(N \to \zeta)}{\sum_{\xi} C(N \to \xi)} = \frac{C(N \to \zeta)}{C(N)}$$

- Penn treebank: 10605 rules, among them 3943 only seen once
- results for sentences with up to 40 word forms:
  - LR = 80.4%, LP = 78.8%
  - constituents without crossing brackets: 87.7%

Probabilistic Parsers

- parsing with a modified EARLEY/CYK algorithm
- dynamic programming:
  - recursively constructing the parsing table and selecting the locally optimal interpretation

Probabilistic Parsers

- problem: independence assumption is systematically wrong
    - subject is more often pronominalized than the object
        - particularly in spoken language
        - consequence of the information structure
    - subcategorisation preferences disambiguate attachment problems
        - attachment to an NP is more frequent that attachment to the verb (2:1)
        - but: some verbs enforce an attachment of certain prepositions

    Moscow sent more than 100.000 soldiers into Afghanistan.

        - *send* requires a direction (*into*)
          → modelling of lexical dependencies becomes necessary

Probabilistic Parsers

- lexical dependencies cannot be expressed in a PCFG
  - only stochastic dependence on the dominating non-terminal

$$\Pr(N \to \zeta | N)$$

- extending the stochastic model with additional conditions

Probabilistic Parsers

- → lexicalised rule-application probabilities (CHARNIAK 2000)

  $$\Pr(N \rightarrow \zeta | N, h(r))$$

- additionally considering the dependence (CHARNIAK 2000, COLLINS 1999)
  - on the head of the immediately dominating phrase level

    $$\Pr(r = N \rightarrow \zeta | N, h(r), h(m(r)))$$

  - on the head of the two dominating phrase levels

    $$\Pr(r = N \rightarrow \zeta | N, h(r), h(m(r)), h(m(m(r))))$$

Probabilistic Parsers

- problem: data sparseness
  - backoff
  - smoothing
  - stochastic modelling of the dependency of the sister nodes from the head as a Markov process (COLLINS 1999)

Probabilistic Parsers

- quality (CHARNIAK 2000)

sentence length $\leq$ 40					
parser	LR	LP	CB	0CB	2CB
COLLINS 1999	88.5	88.7	0.92	66.7	87.1
CHARNIAK 2000	90.1	90.1	0.74	70.1	89.6

sentence length $\leq$ 100					
parser	LR	LP	CB	0CB	2 CB
COLLINS 1999	88.1	88.3	1.06	64.0	85.1
CHARNIAK 2000	89.6	89.5	0.88	67.6	87.7