Syntax-Based Decoding 2

Philipp Koehn

14 November 2017

flashback: syntax-based models

Synchronous Context Free Grammar Rules

- Nonterminal rules

$$\mathsf{NP} \rightarrow \mathsf{DET}_1\ \mathsf{NN}_2\ \mathsf{JJ}_3 \mid \mathsf{DET}_1\ \mathsf{JJ}_3\ \mathsf{NN}_2$$

- Terminal rules

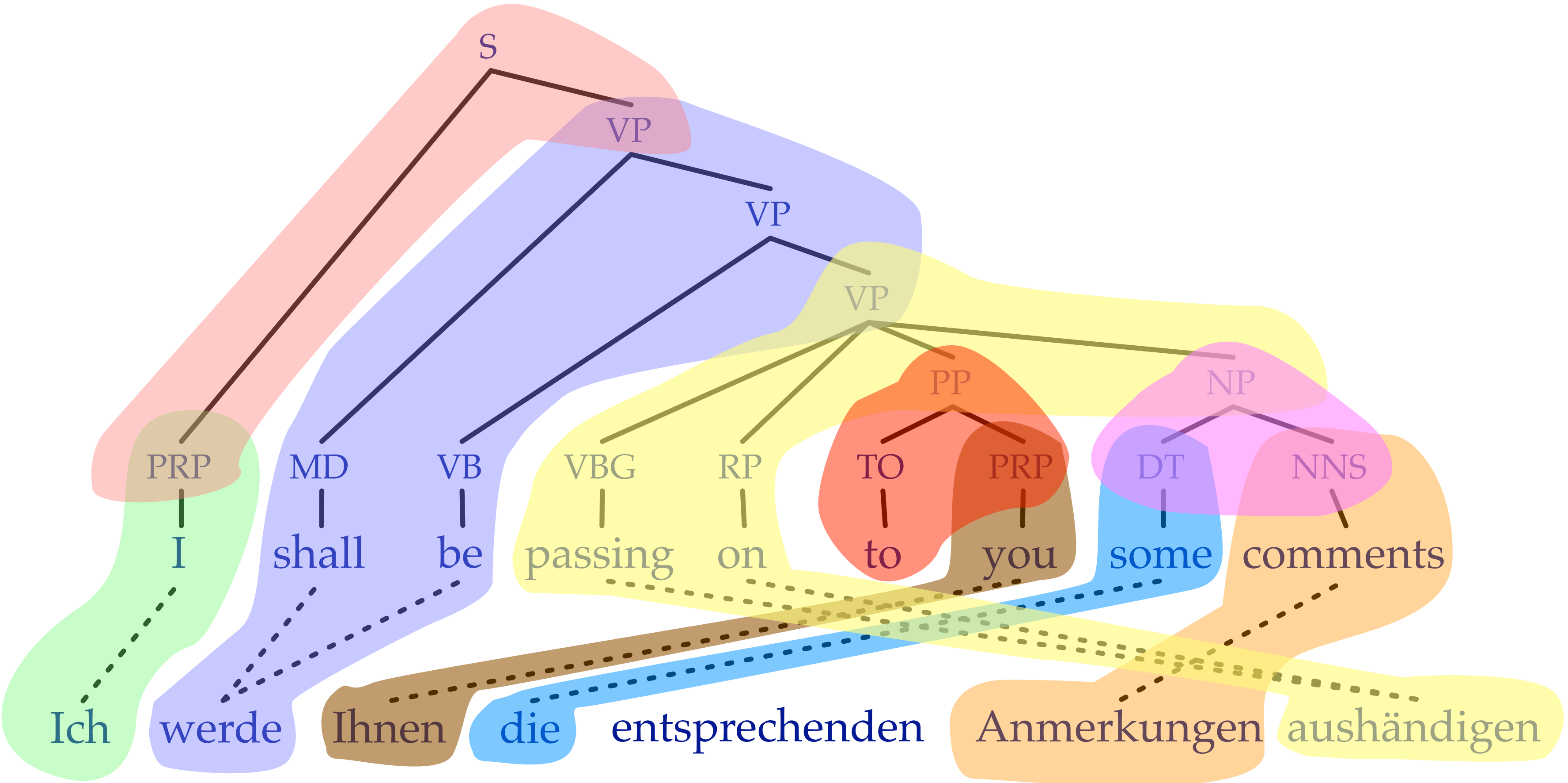
$$\mathsf{N} \rightarrow \text{maison} \mid \text{house}$$

$$\mathsf{NP} \rightarrow \text{la maison bleue} \mid \text{the blue house}$$

- Mixed rules

$$\mathsf{NP} \rightarrow \text{la maison } \mathsf{JJ}_1 \mid \text{the } \mathsf{JJ}_1 \text{ house}$$

Extracting Minimal Rules

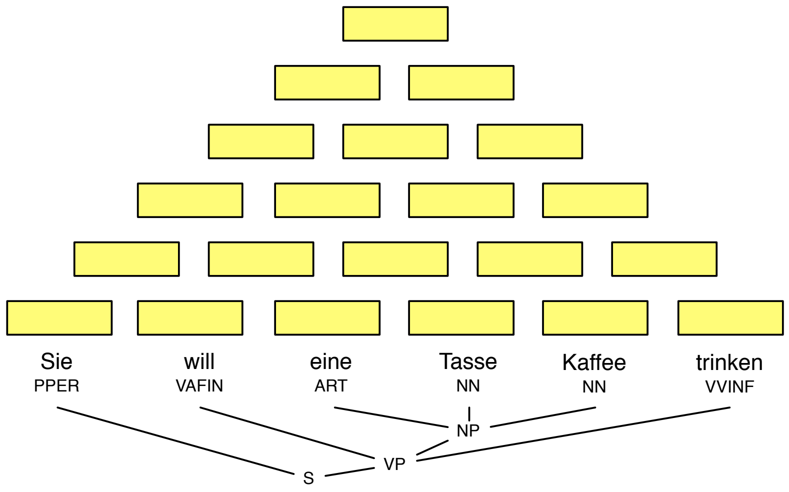


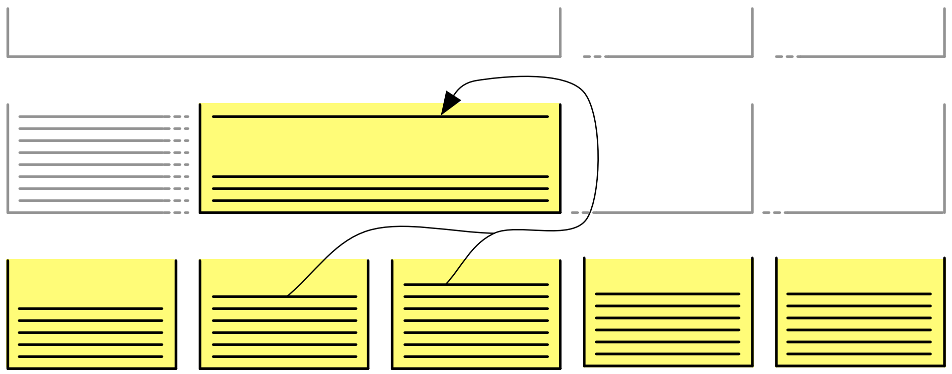
Extracted rule: $\text{S} \rightarrow \text{X}_1\ \text{X}_2 \mid \text{PRP}_1\ \text{VP}_2$

DONE — note: one rule per alignable constituent

flashback: decoding

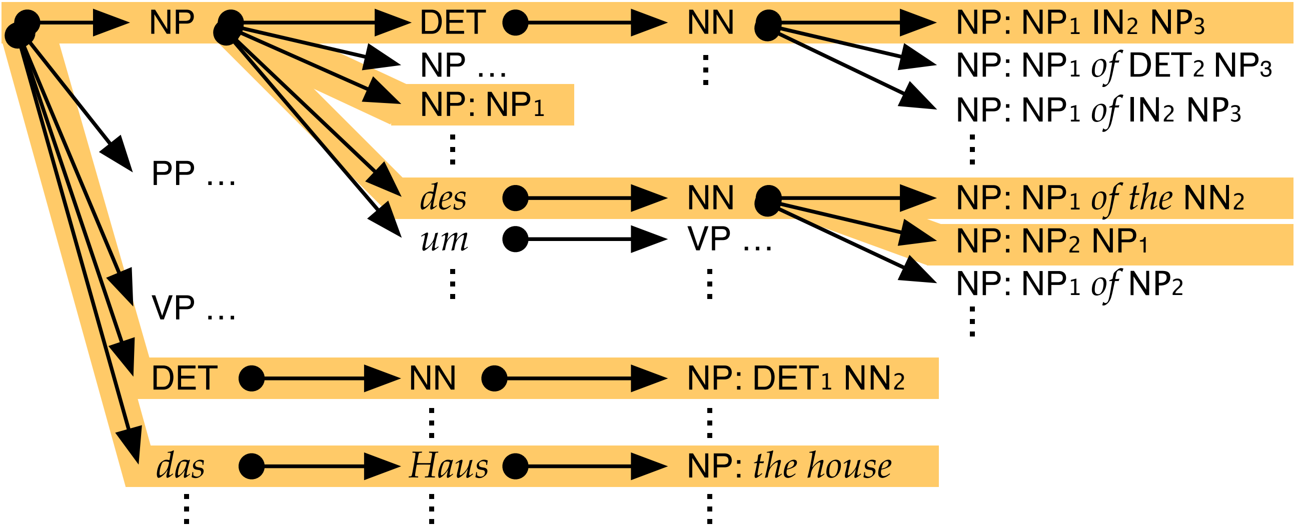
Chart Organization





- Chart consists of cells that cover contiguous spans over the input sentence
- For each span, a stack of (partial) translations is maintained
- Bottom-up: a higher stack is filled, once underlying stacks are complete

Prefix Tree for Rules



Highlighted Rules

$\text{NP} \rightarrow \text{NP}_1\ \text{DET}_2\ \text{NN}_3 \mid \text{NP}_1\ \text{IN}_2\ \text{NN}_3$

$\text{NP} \rightarrow \text{NP}_1 \mid \text{NP}_1$

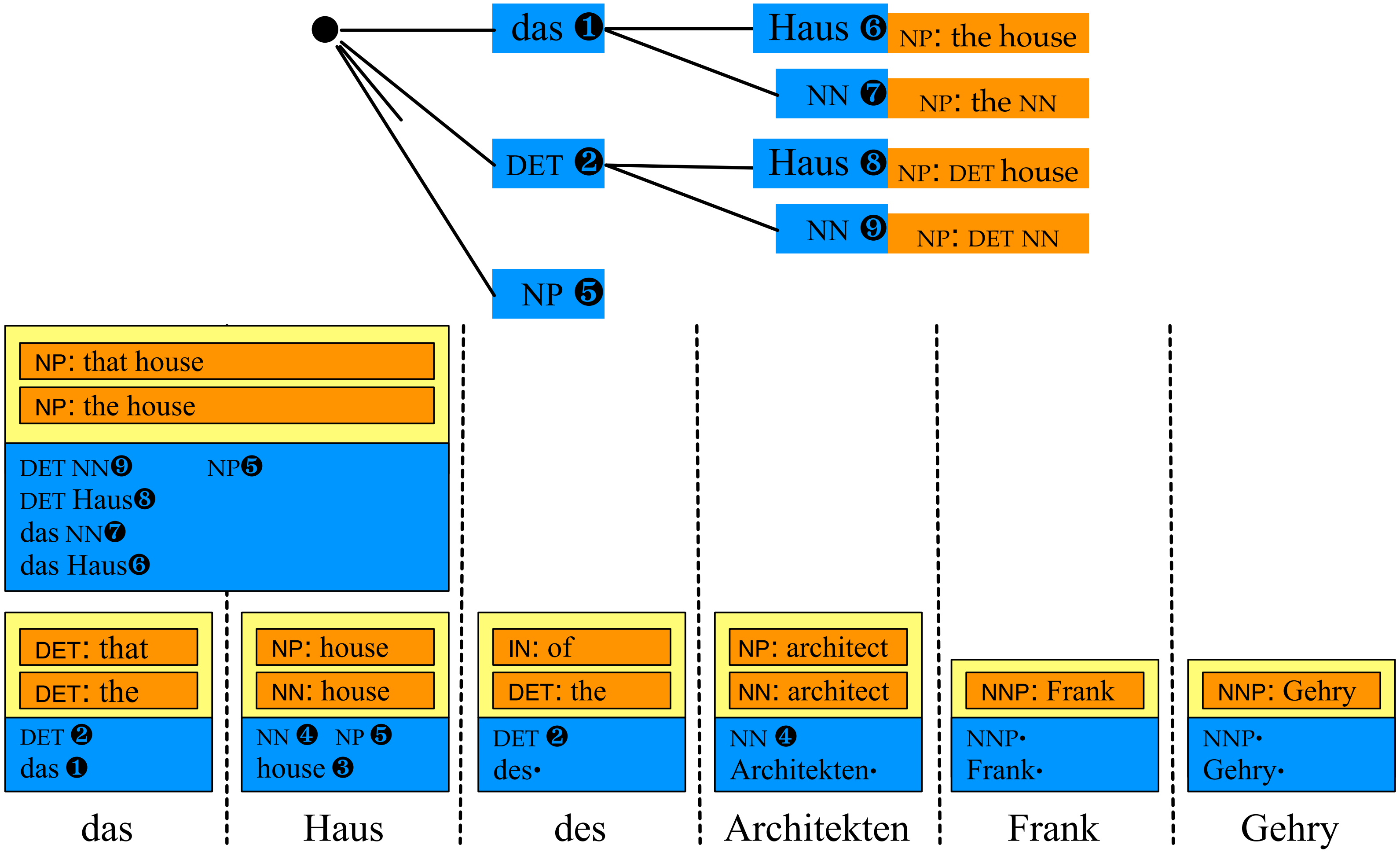
$\text{NP} \rightarrow \text{NP}_1$ des $\text{NN}_2 \mid \text{NP}_1$ of the NN_2

$\text{NP} \rightarrow \text{NP}_1$ des $\text{NN}_2 \mid \text{NP}_2\ \text{NP}_1$

$\text{NP} \rightarrow \text{DET}_1\ \text{NN}_2 \mid \text{DET}_1\ \text{NN}_2$

$\text{NP} \rightarrow$ das Haus | the house

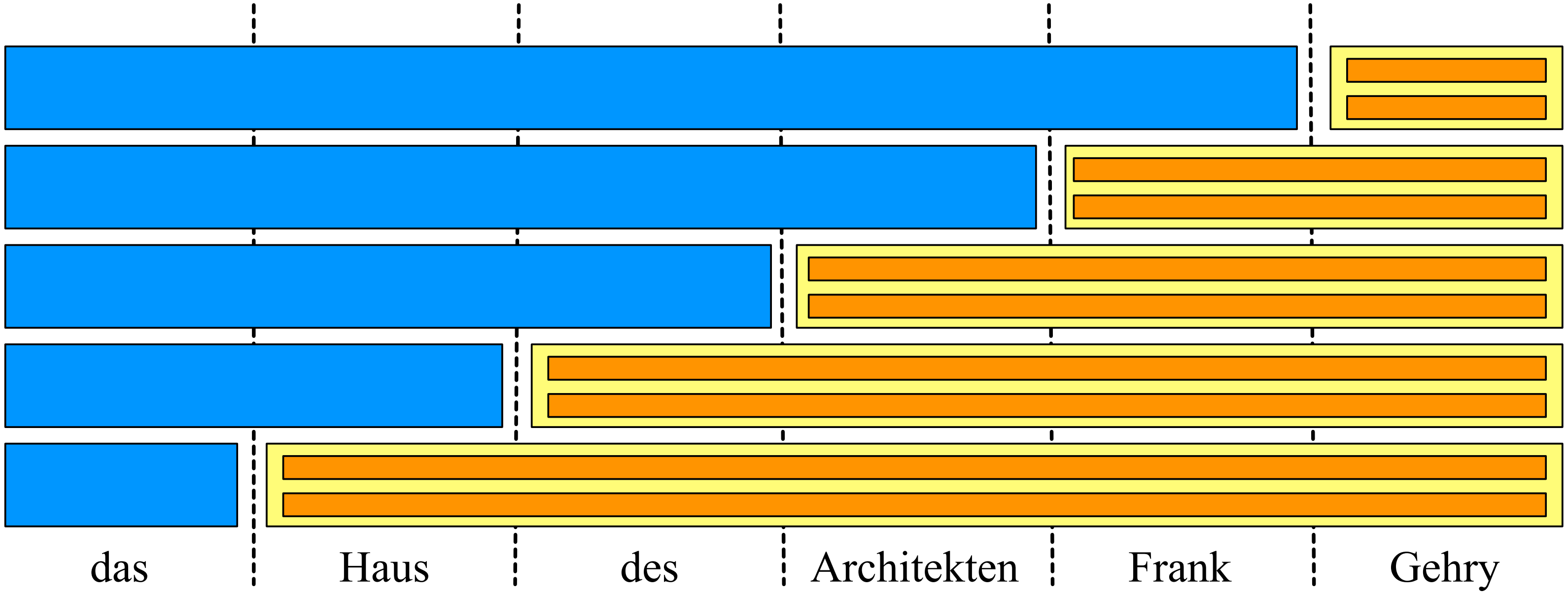
CYK+ Parsing for SCFG



Processing One Span

Extend lists of dotted rules with cell constituent labels

span's dotted rule list (with same start)
plus neighboring
span's constituent labels of hypotheses (with same end)



pruning

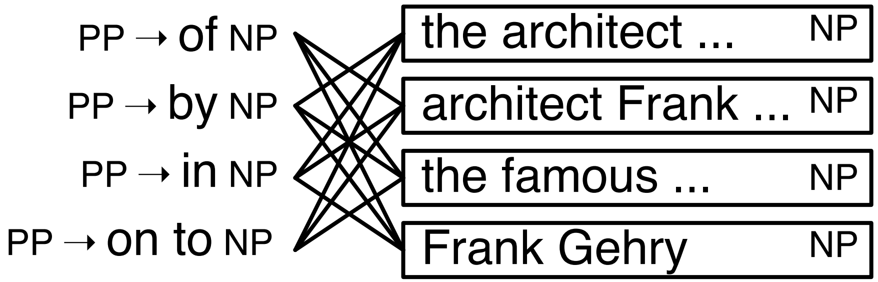
Where are we now?

- We know which rules apply
- We know where they apply (each non-terminal tied to a span)
- But there are still many choices
  - many possible translations
  - each non-terminal may match multiple hypotheses

  → number choices exponential with number of non-terminals

Rules with One Non-Terminal

Found applicable rules PP $\rightarrow$ des X | ... NP ...



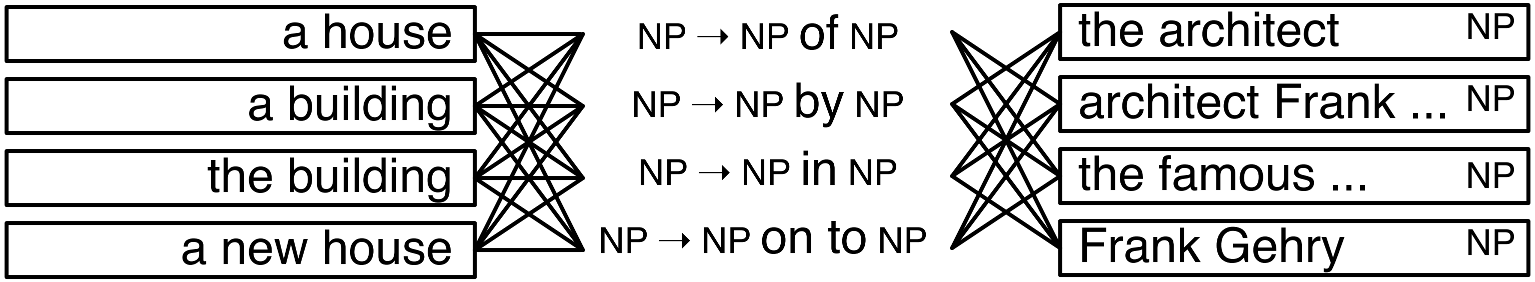
- Non-terminal will be filled any of h underlying matching hypotheses
- Choice of t lexical translations

$\Rightarrow$ Complexity $O(ht)$

(note: we may not group rules by target constituent label,
so a rule NP $\rightarrow$ des X | the NP would also be considered here as well)

Rules with Two Non-Terminals

Found applicable rule NP → X_1 des X_2 | NP_1 ... NP_2



- Two non-terminal will be filled any of h underlying matching hypotheses each

- Choice of t lexical translations

⇒ Complexity $O(h^2t)$ — a three-dimensional ”cube” of choices

(note: rules may also reorder differently)

Cube Pruning

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0				
a building 1.3				
the building 2.2				
a new house 2.6				

Arrange all the choices in a ”cube”

(here: a square, generally a orthotope, also called a hyperrectangle)

Create the First Hypothesis

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1			
a building 1.3				
the building 2.2				
a new house 2.6				

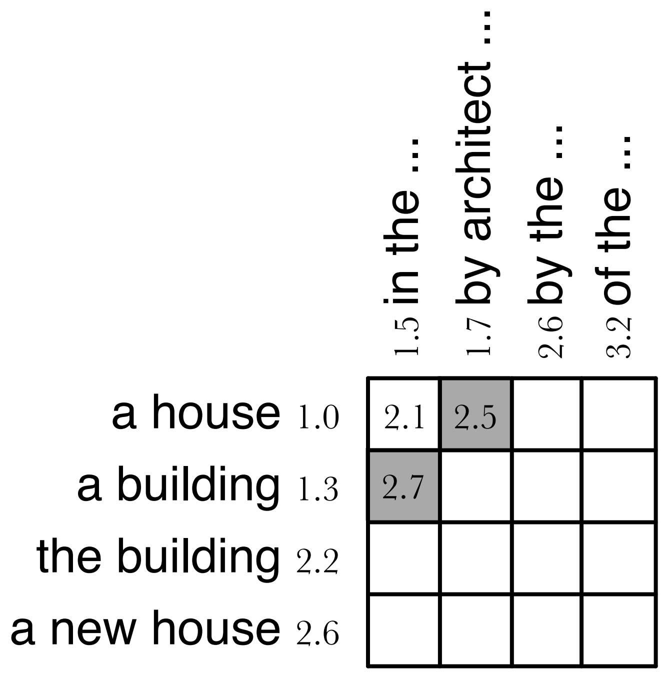
- Hypotheses created in cube: (0,0)

Add (”Pop”) Hypothesis to Chart Cell

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1			
a building 1.3				
the building 2.2				
a new house 2.6				

- Hypotheses created in cube: ϵ
- Hypotheses in chart cell stack: (0,0)

Create Neighboring Hypotheses



	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1	2.5		
a building 1.3	2.7			
the building 2.2				
a new house 2.6				

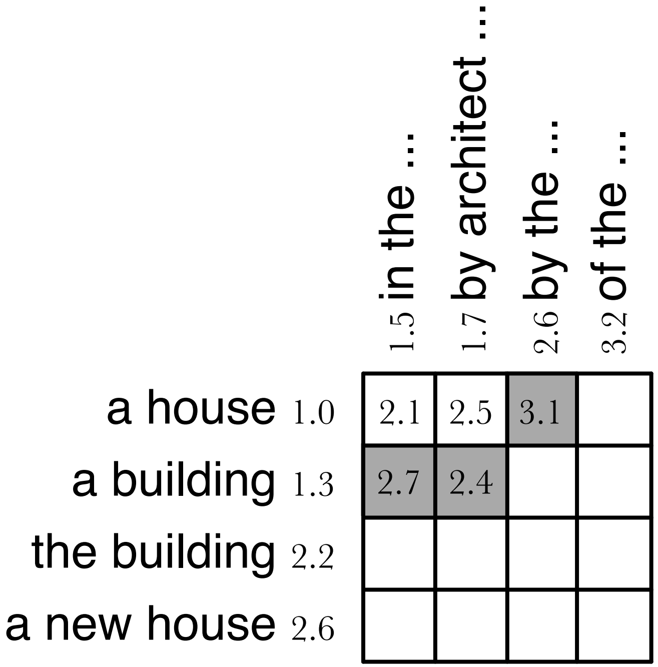
- Hypotheses created in cube: (0,1), (1,0)
- Hypotheses in chart cell stack: (0,0)

Pop Best Hypothesis to Chart Cell

	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1	2.5		
a building 1.3	2.7			
the building 2.2				
a new house 2.6				

- Hypotheses created in cube: (0,1)
- Hypotheses in chart cell stack: (0,0), (1,0)

Create Neighboring Hypotheses

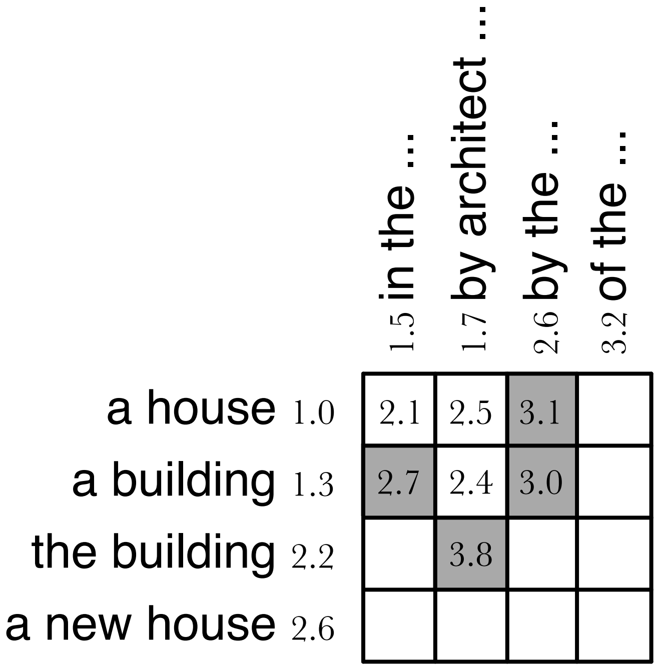


	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1	2.5	3.1	
a building 1.3	2.7	2.4		
the building 2.2				
a new house 2.6				

- Hypotheses created in cube: (0,1), (1,1), (2,0)

- Hypotheses in chart cell stack: (0,0), (1,0)

More of the Same



	1.5 in the ...	1.7 by architect ...	2.6 by the ...	3.2 of the ...
a house 1.0	2.1	2.5	3.1	
a building 1.3	2.7	2.4	3.0	
the building 2.2		3.8		
a new house 2.6				

- Hypotheses created in cube: (0,1), (1,2), (2,1), (2,0)
- Hypotheses in chart cell stack: (0,0), (1,0), (1,1)

Queue of Cubes

- Several groups of rules will apply to a given span
- Each of them will have a cube
- We can create a queue of cubes

⇒ Always pop off the most promising hypothesis, regardless of cube

- May have separate queues for different target constituent labels

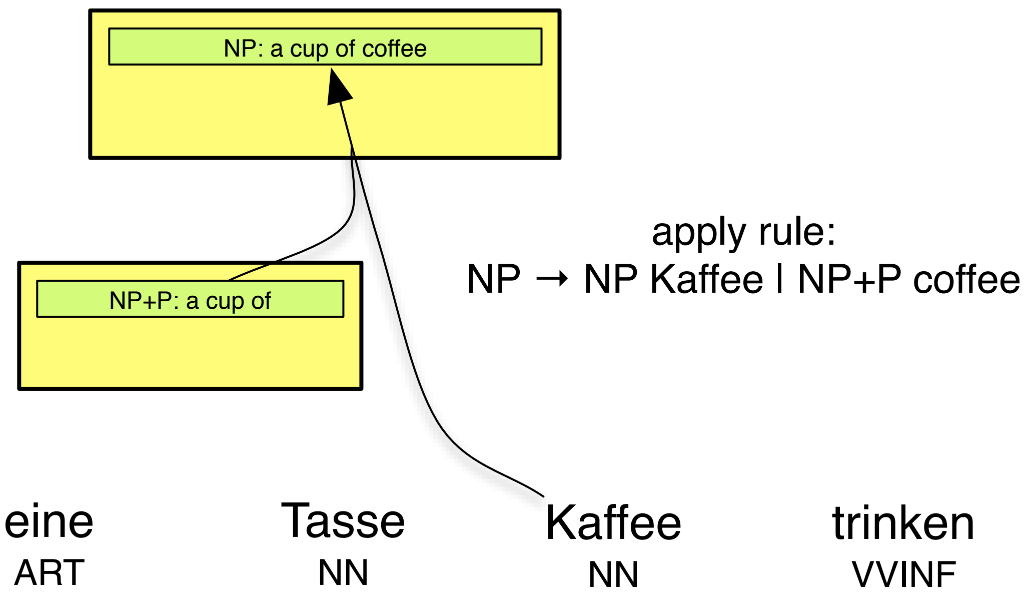
Bottom-Up Chart Decoding Algorithm

```
for all spans (bottom up) do
    extend dotted rules
    for all dotted rules do
        find group of applicable rules
        create a cube for it
        create first hypothesis in cube
        place cube in queue
    end for
    for specified number of pops do
        pop off best hypothesis of any cube in queue
        add it to the chart cell
        create its neighbors
    end for
    extend dotted rules over constituent labels
end for
```

recombination and pruning

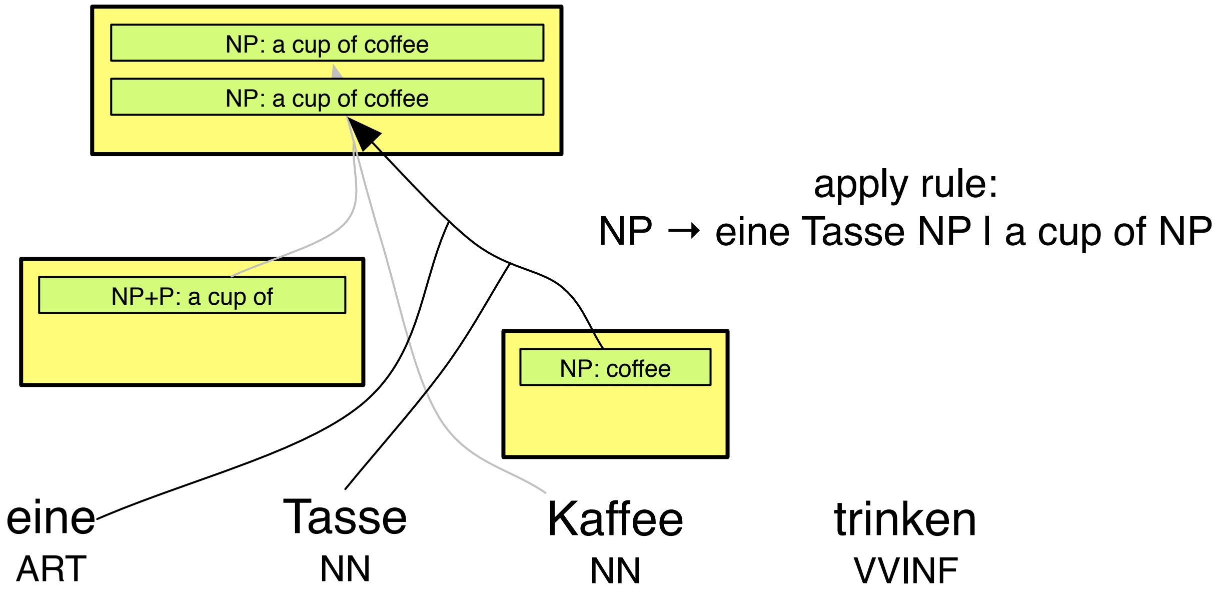
Dynamic Programming

Applying rule creates new hypothesis



Dynamic Programming

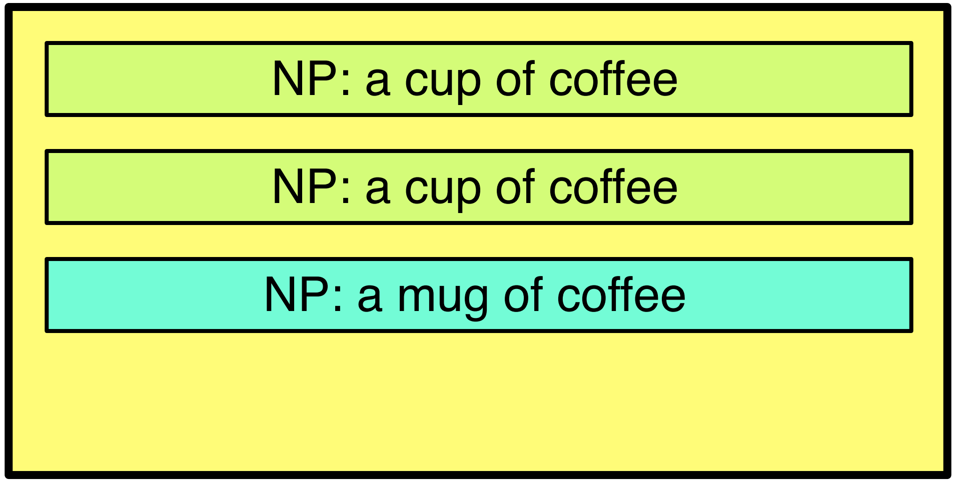
Another hypothesis



Both hypotheses are indistiguishable in future search
→ can be recombined

Recombinable States

Recombinable?



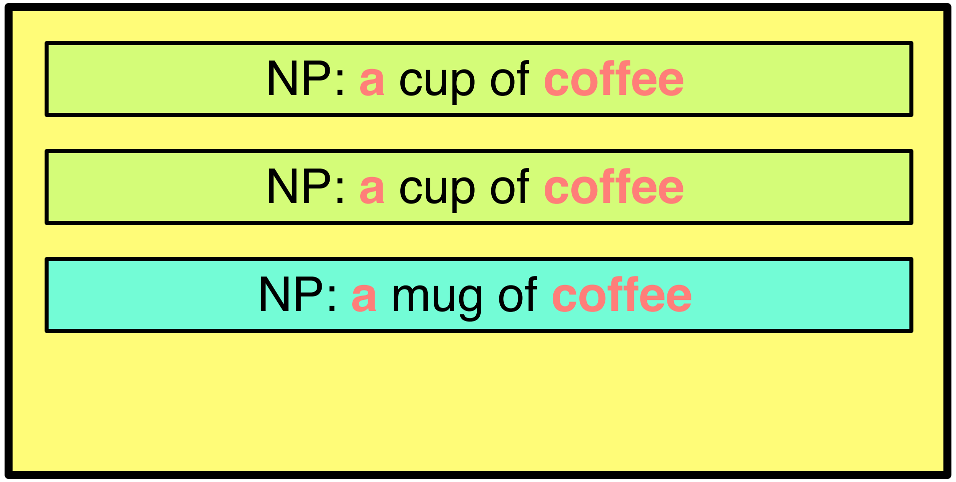
NP: a cup of coffee

NP: a cup of coffee

NP: a mug of coffee

Recombinable States

Recombinable?



NP: a cup of coffee

NP: a cup of coffee

NP: a mug of coffee

Yes, iff max. 2-gram language model is used

Recombinability

Hypotheses have to match in

- span of input words covered
- output constituent label
- first n–1 output words

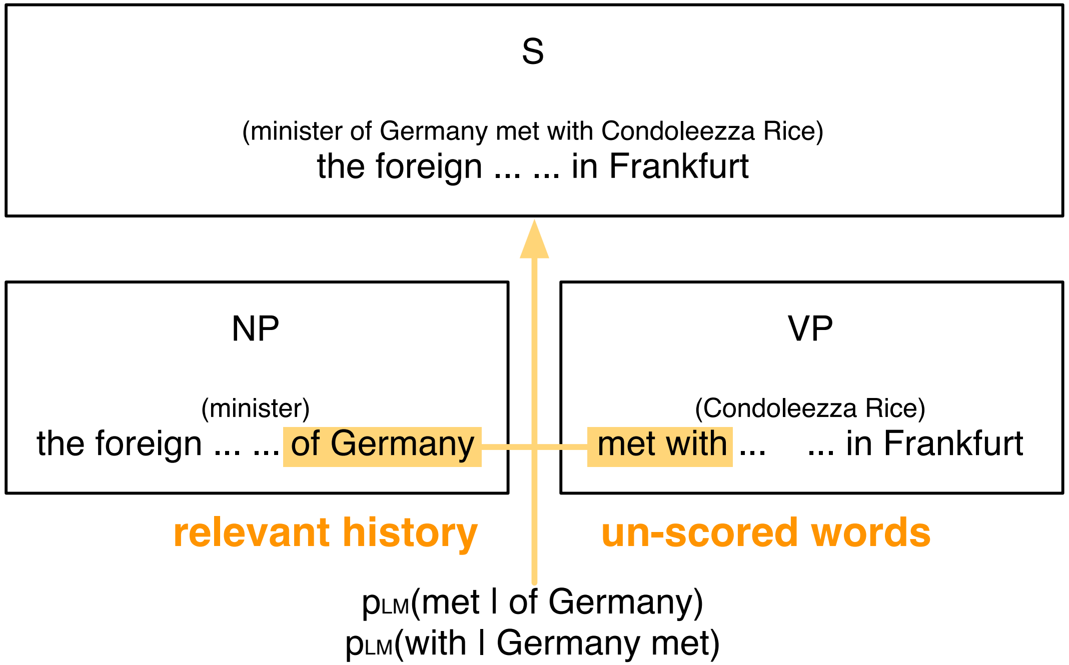
  not properly scored, since they lack context
- last n–1 output words

  still affect scoring of subsequently added words, just like in phrase-based decoding

(n is the order of the n-gram language model)

Language Model Contexts

When merging hypotheses, internal language model contexts are absorbed


S
(minister of Germany met with Condoleezza Rice)
the foreign ... ... in Frankfurt
NP
(minister)
the foreign ... ... of Germany
VP
(Condoleezza Rice)
met with ... ... in Frankfurt
relevant history
un-scored words
pLM(met | of Germany)
pLM(with | Germany met)

Stack Pruning

- Number of hypotheses in each chart cell explodes

⇒ need to discard bad hypotheses
e.g., keep 100 best only

- Different stacks for different output constituent labels?

- Cost estimates
  - translation model cost known
  - language model cost for internal words known
    → estimates for initial words
  - outside cost estimate?
    (how useful will be a NP covering input words 3–5 later on?)

scope 3 pruning

How Often Does a Rule Apply?

- Lexical rule → only once in sentence

NP → la maison bleue | the blue house

- One non-terminal bounded by words → only once in sentence

NP → la NN_1 bleue | the blue NN_1

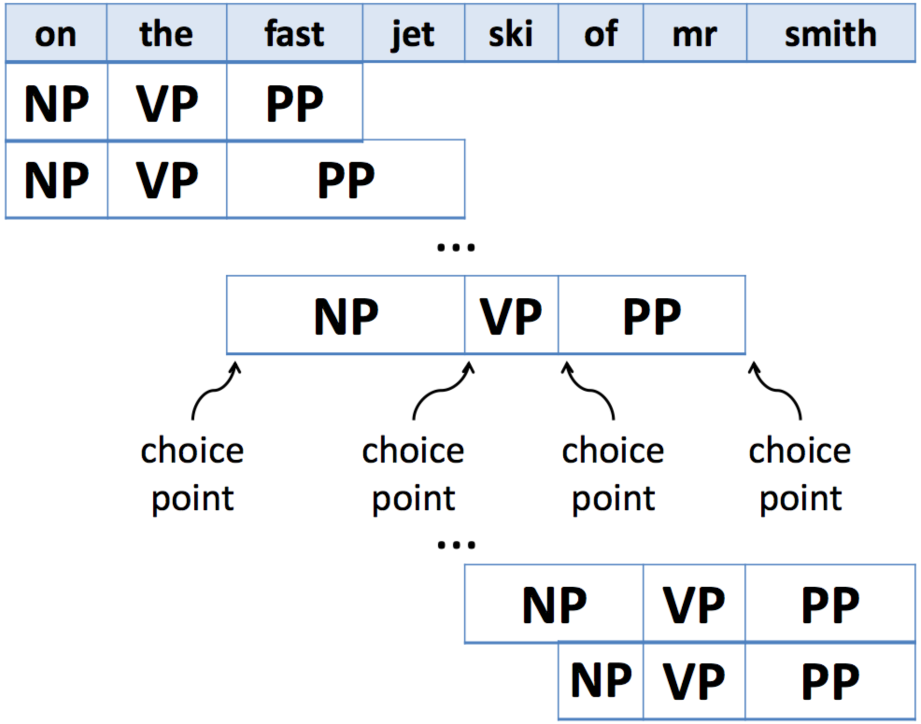
- One non-terminal at edge of rule → non-terminal can cover $O(n)$ words

NP → la NN_1 | the NN_1

- Two non-terminals at edges → combined choices for both non-terminals $O(n^2)$

NP → DET_1 maison JJ_2 | DET_1 JJ_2 house

Choice Points



- 4 choice points → $O(n^4)$ application contexts
- Too many choice points → rule applied to many times

Recall: Hierarchical Rule Extraction

- Having only one non-terminal symbol X

- Restrictions to limit complexity
  - at most 2 nonterminal symbols
  - no neighboring non-terminals on the source side
  - span at most 15 words (counting gaps)

⇒ At most 2 choice points (”scope 2”)

Rule Binarization

- Convert grammar to Chomsky Normal Form (CNF) — scope 3

- Only allow two types of rules

  $$\begin{aligned} \mathsf{A} &\rightarrow \text{word} \\ \mathsf{A} &\rightarrow \mathsf{B}\ \mathsf{C} \end{aligned}$$

  (Note: for our rules, we would allow additional terminals)

- Convert rules with more non-terminals

  $$\mathsf{A} \rightarrow \mathsf{X}\ \mathsf{Y}\ \mathsf{Z}$$
  $$\Downarrow$$
  $$\begin{aligned} \mathsf{A} &\rightarrow \mathsf{X}\ \mathsf{Q} \\ \mathsf{Q} &\rightarrow \mathsf{Y}\ \mathsf{Z} \end{aligned}$$

  (Q is a new non-terminal, specific to this rule)

- But:
  - increases the number of non-terminals (”grammar constant”)
  - can be tricky for SCFG rules

Scope 3 Pruning

- Remove all rules with scope > 3

- Less restrictive than CNF
  e.g., allows:

$$\mathtt{A} \rightarrow \mathtt{DET}_1 \text{ maison } \mathtt{JJ}_2 \text{ sur la } \mathtt{NN}_3 \;\middle|\; \mathtt{DET}_1\ \mathtt{JJ}_2 \text{ house on the } \mathtt{NN}_3$$

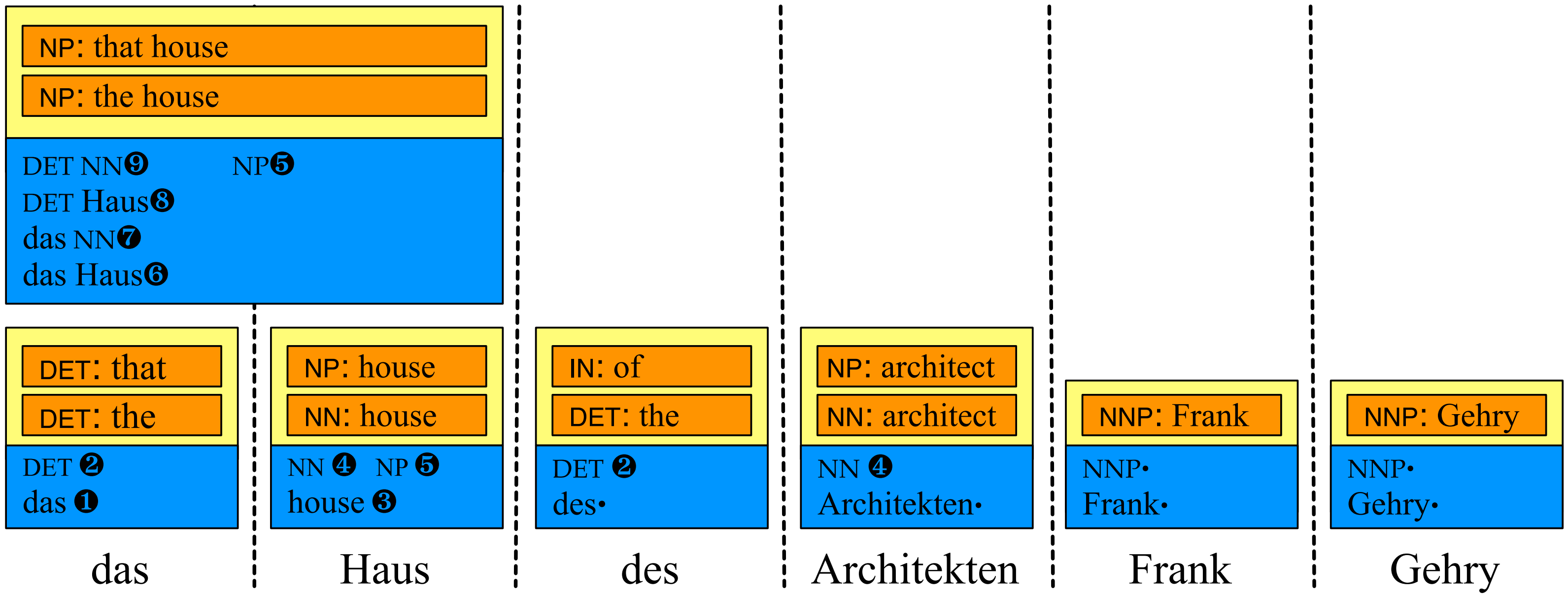
  (2 choice points at edges)

- Better speed/quality trade-off than synchronous binarization

recursive cky+

CKY+

- Two charts: (1) hypothesis chart, (2) dotted rule chart



NP: that house
NP: the house

DET NN❾ NP❺
DET Haus❽
das NN❼
das Haus❻

DET: that
DET: the
DET ❷
das ❶

NP: house
NN: house
NN ❹ NP ❺
house ❸

IN: of
DET: the
DET ❷
des•

NP: architect
NN: architect
NN ❹
Architekten•

NNP: Frank
NNP•
Frank•

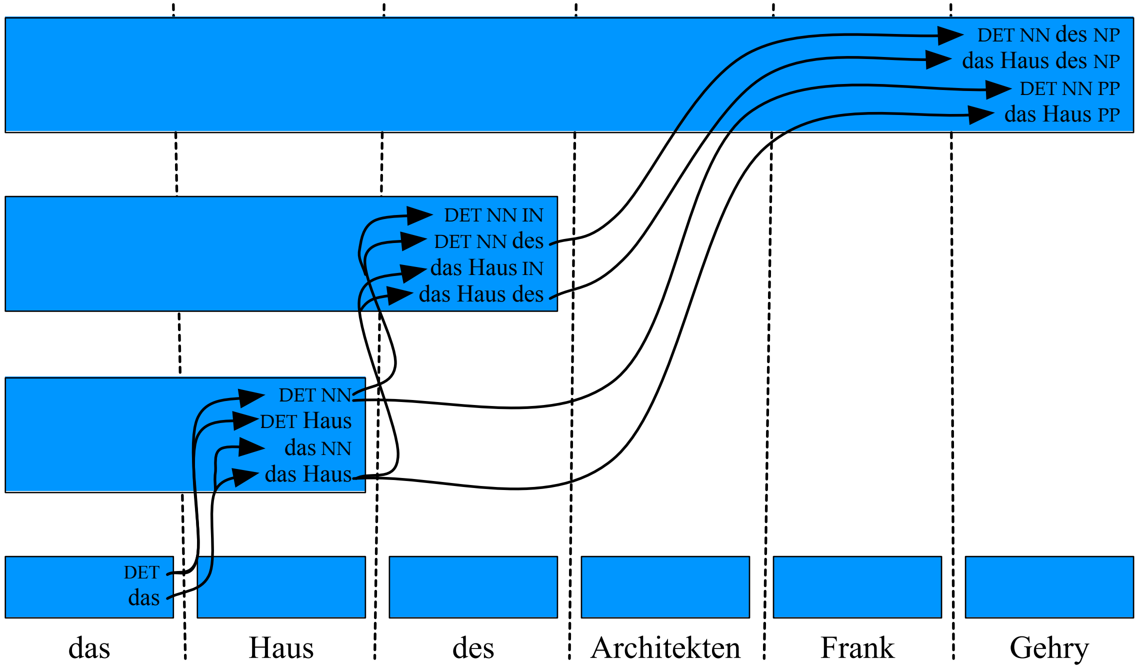
NNP: Gehry
NNP•
Gehry•

das Haus des Architekten Frank Gehry

- Dotted rule chart allows dynamic programming of rules with same prefix

Expansion of Dotted Rules

- Dotted rules are expanded recursively

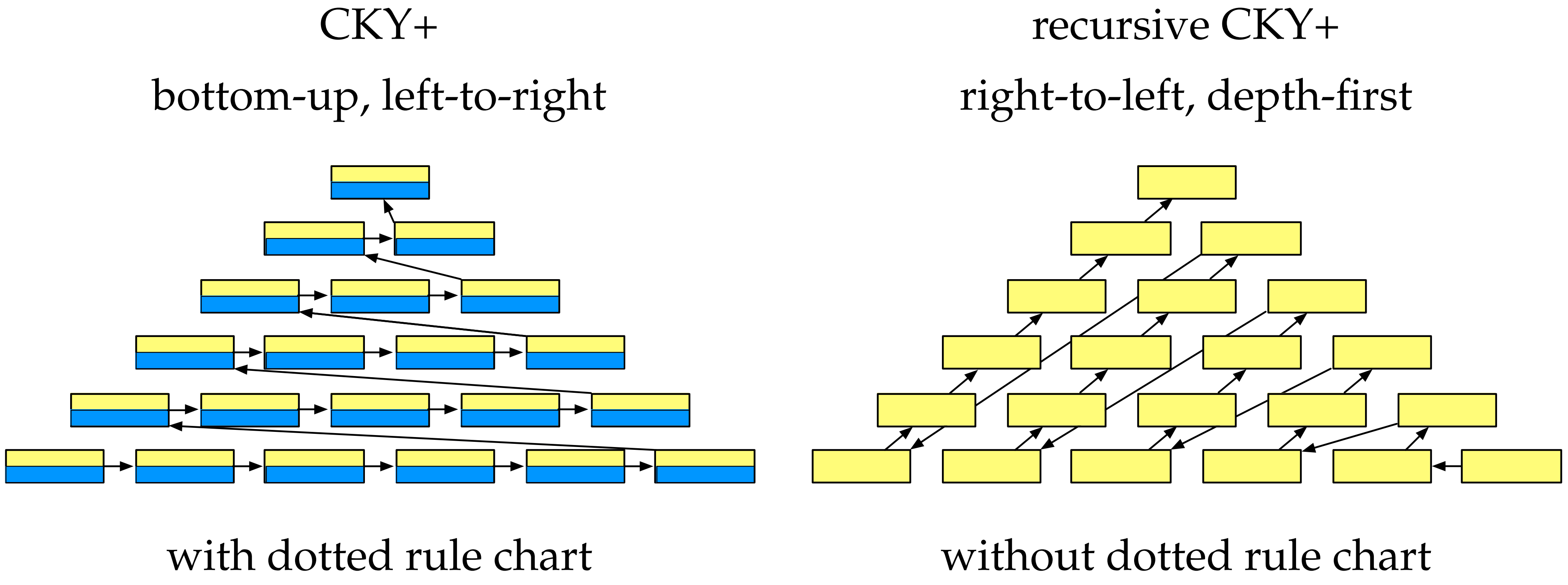


- Dotted rules are stored with each chart cell

Recursive CKY+

- Recursive CKY+ (Sennrich, 2014) removes need for dotted rule chart

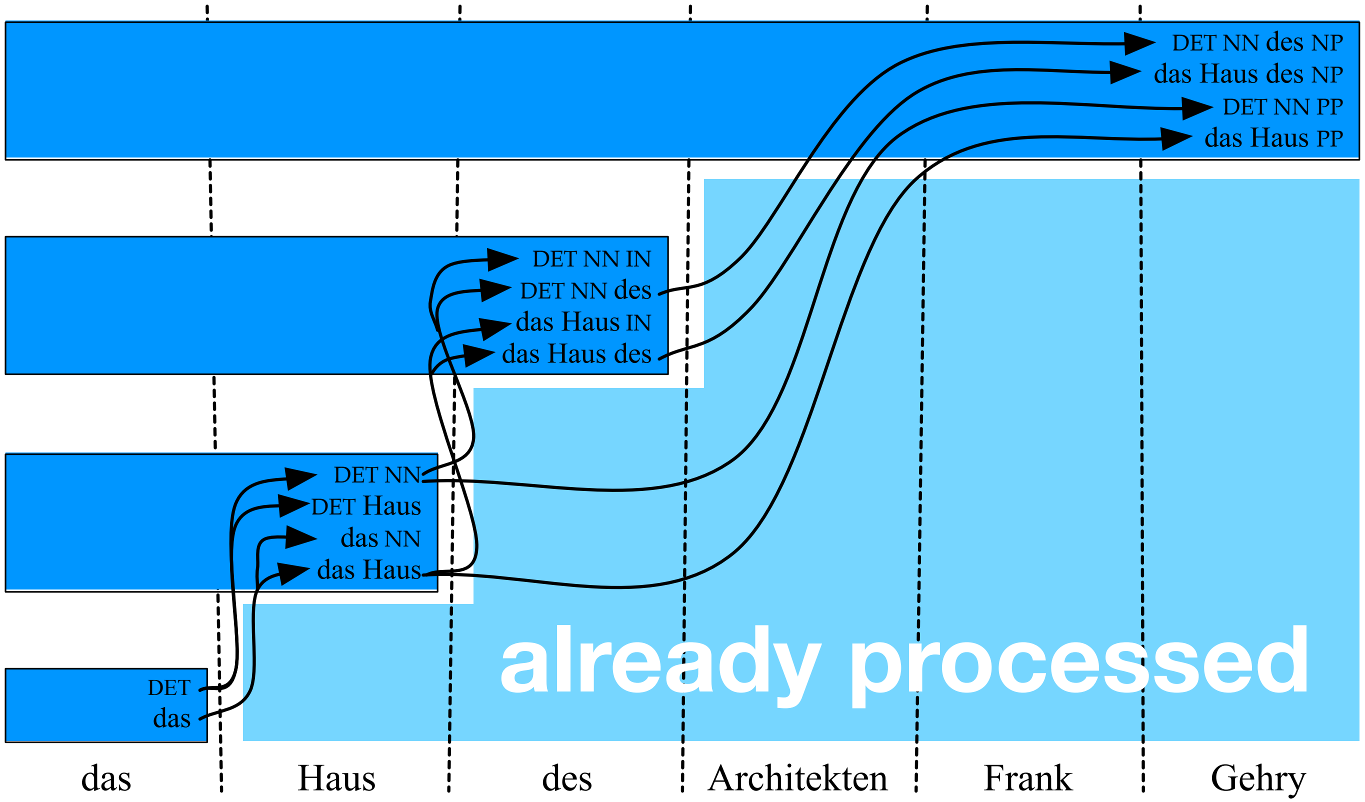
- Chart traversal is re-arranged



Recursive CKY+

- Rule expansion by recursive function calls

- Rules can be immediately expanded, because all needed cells already processed



search strategies

Two-Stage Decoding

- First stage: decoding without a language model (-LM decoding)
  - may be done exhaustively
  - eliminate dead ends
  - optionably prune out low scoring hypotheses

- Second stage: add language model
  - limited to packed chart obtained in first stage

- Note: essentially, we do two-stage decoding for each span at a time
  - stage 1: find applicable rules
  - stage 2: cube pruning

Coarse-to-Fine

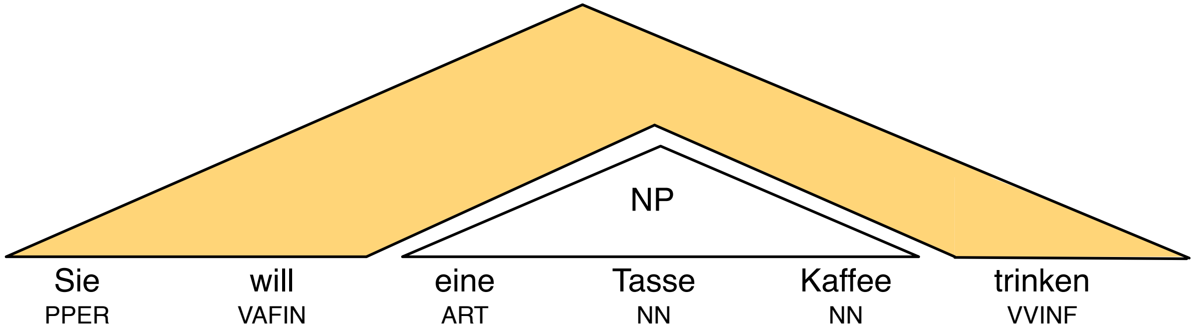
- Decode with increasingly complex model

- Examples
  - reduced language model [Zhang and Gildea, 2008]
  - reduced set of non-terminals [DeNero et al., 2009]
  - language model on clustered word classes [Petrov et al., 2008]

Outside Cost Estimation

- Which spans should be more emphasized in search?

- Initial decoding stage can provide outside cost estimates



- Use min/max language model costs to obtain admissible heuristic (or at least something that will guide search better)

Open Questions

- What causes the high search error rate?
- Where does the best translation fall out the beam?
- How accurate are LM estimates?
- Are particular types of rules too quickly discarded?
- Are there systemic problems with cube pruning?