Statistical Machine Translation
LING-462/COSC-482
Week 5:
Decoding and tree-based models

Achim Ruopp
achim.ruopp@Georgetown.edu

Agenda

- Language in ten minutes: Derek Acosta
- Decoding

- Break -

- Tree-based models
- HW3: Decoding
- Internships tips/inquiries

Statistical Machine Translation

- What we have now: statistical models
  - Word-based translation models
  - Phrase-based translation models
  - N-gram language models
  - Noisy channel model
  - Log-linear model
- Next: decoding
  - How do we find the most-likely or top-n most likely translations?

DECODING

Decoding

- We have a mathematical model for translation

$$p(\mathbf{e}|\mathbf{f})$$

- Task of decoding: find the translation $\mathbf{e}_{\text{best}}$ with highest probability

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\ p(\mathbf{e}|\mathbf{f})$$

- Two types of error
  - the most probable translation is bad $\rightarrow$ fix the model
  - search does not find the most probably translation $\rightarrow$ fix the search

- Decoding is evaluated by search error, not quality of translations (although these are often correlated)

Translation Process

- Task: translate this sentence from German into English

er **geht** **ja** **nicht** **nach** **hause**

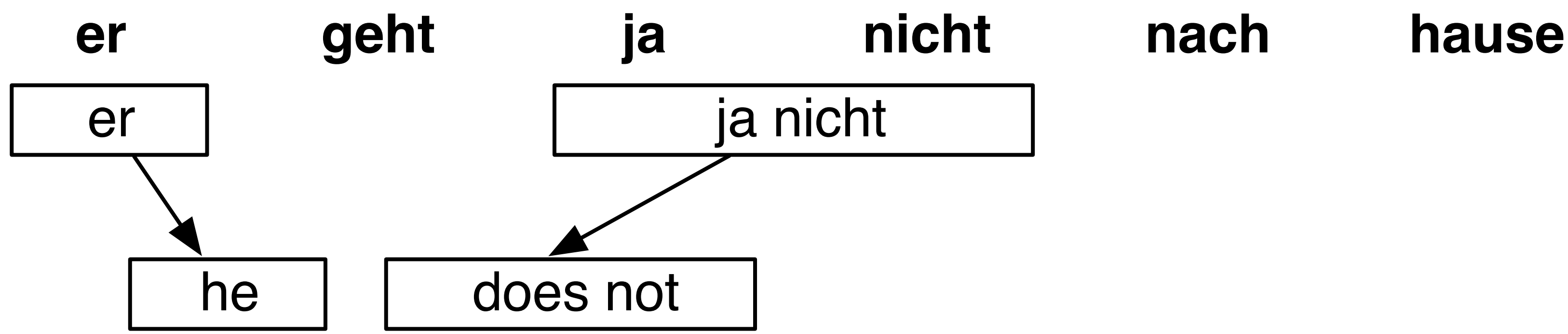
Translation Process

- Task: translate this sentence from German into English

- Pick phrase in input, translate

Translation Process

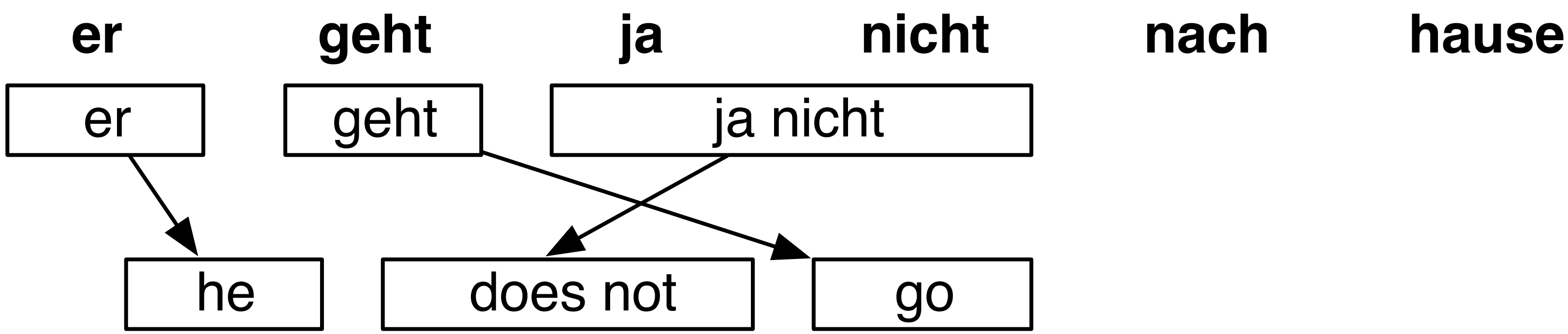
- Task: translate this sentence from German into English



- Pick phrase in input, translate
  - it is allowed to pick words out of sequence reordering
  - phrases may have multiple words: many-to-many translation

Translation Process

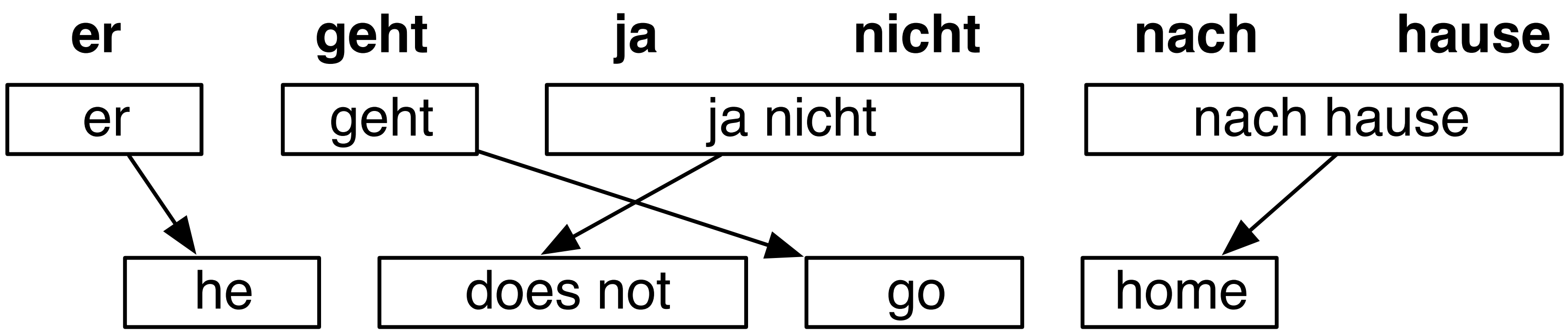
- Task: translate this sentence from German into English



- Pick phrase in input, translate

Translation Process

- Task: translate this sentence from German into English



- Pick phrase in input, translate

Computing Translation Probability

- Probabilistic model for phrase-based translation:

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}} \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\; d(start_i - end_{i-1} - 1)\; p_{\text{LM}}(\mathbf{e})$$

- Score is computed incrementally for each partial hypothesis

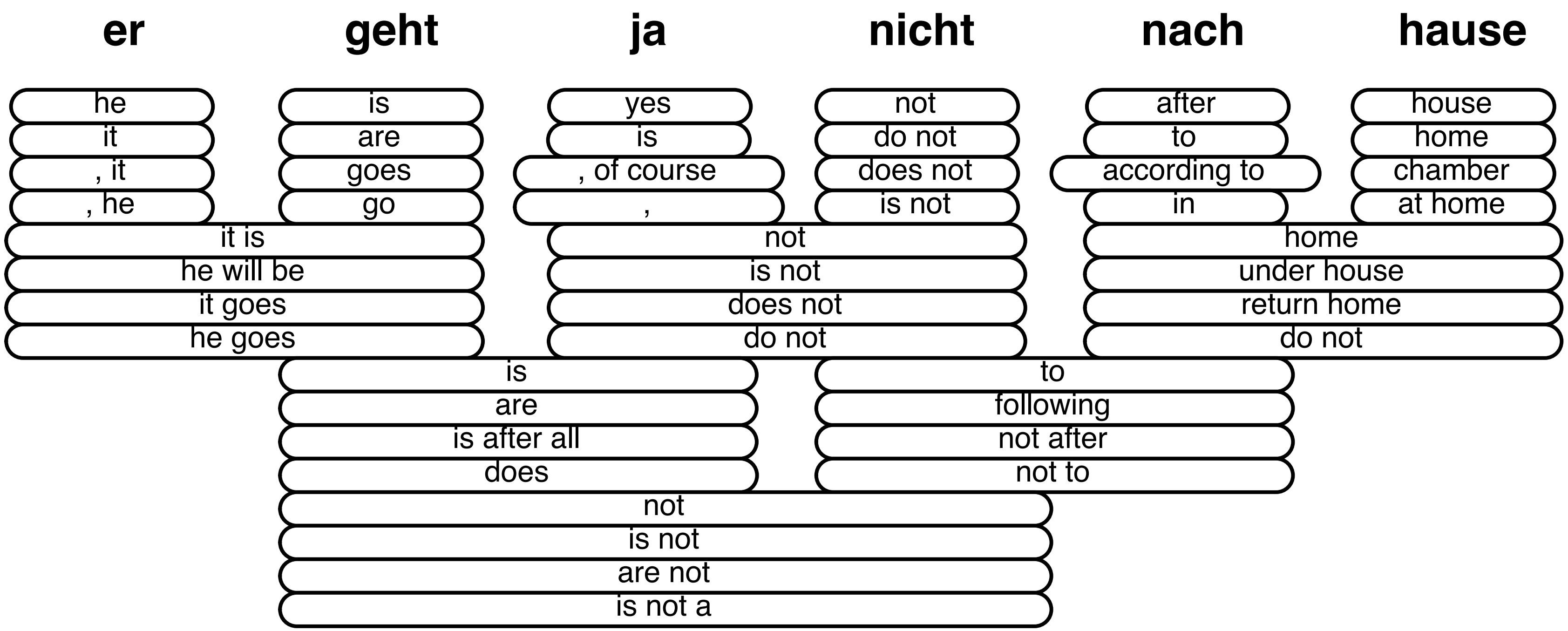
- Components

  Phrase translation Picking phrase $\bar{f}_i$ to be translated as a phrase $\bar{e}_i$
  $\rightarrow$ look up score $\phi(\bar{f}_i|\bar{e}_i)$ from phrase translation table

  Reordering Previous phrase ended in end_{i-1}, current phrase starts at $start_i$
  $\rightarrow$ compute $d(start_i - end_{i-1} - 1)$

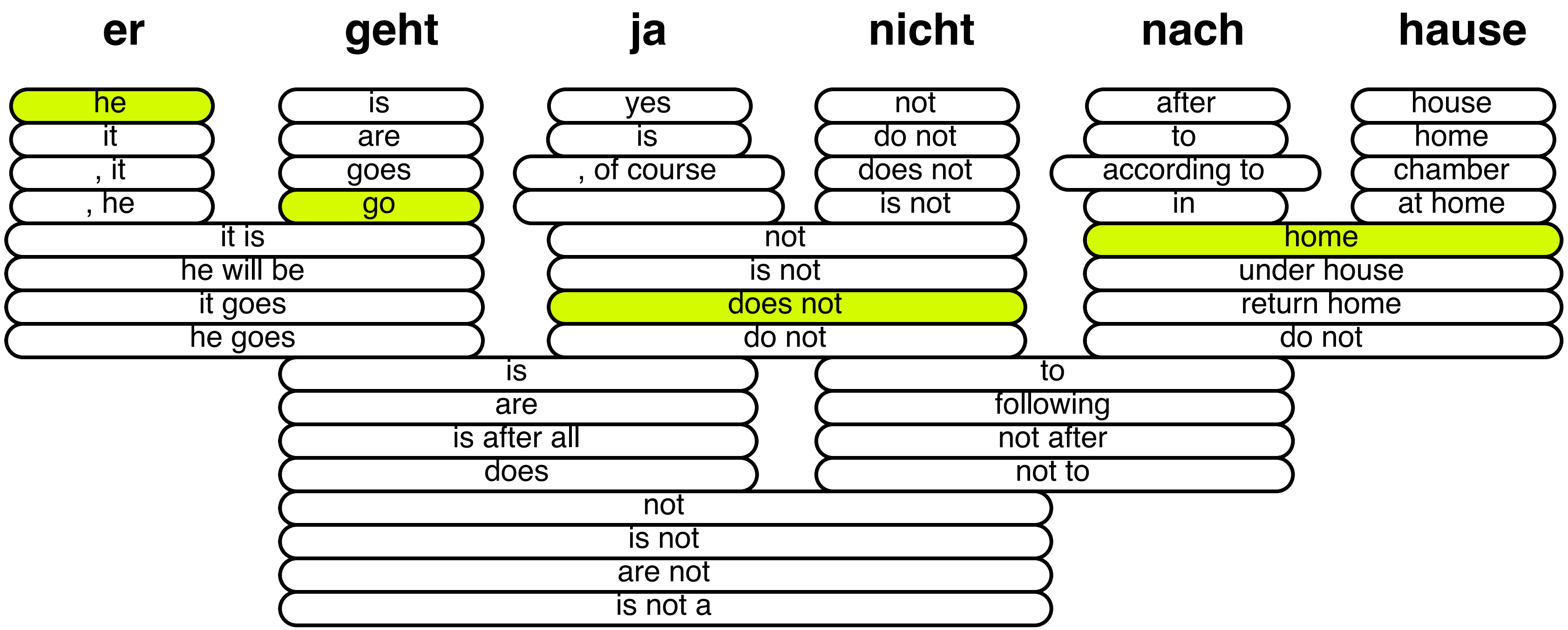
  Language model For n-gram model, need to keep track of last $n-1$ words
  $\rightarrow$ compute score $p_{\text{LM}}(w_i|w_{i-(n-1)}, ..., w_{i-1})$ for added words w_i

Translation Options



- Many translation options to choose from
  - in Europarl phrase table: 2727 matching phrase pairs for this sentence
  - by pruning to the top 20 per phrase, 202 translation options remain

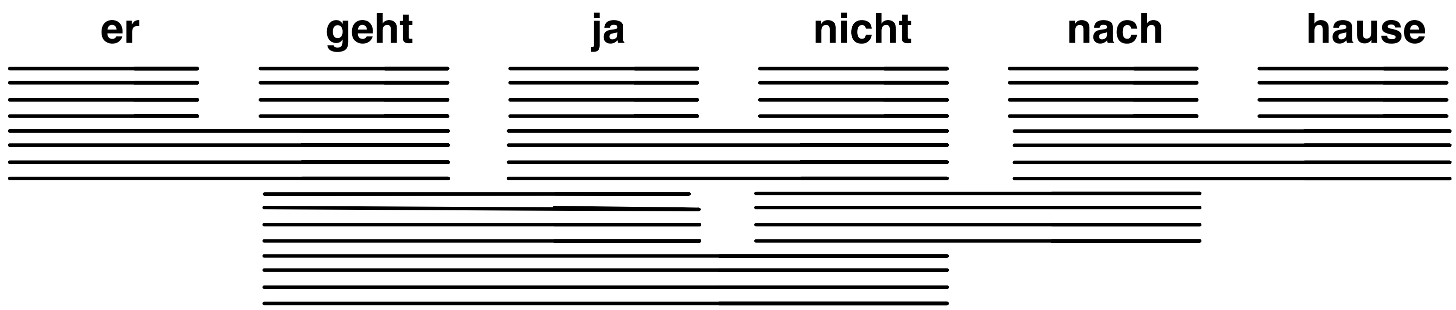
Translation Options



- The machine translation decoder does not know the right answer
  - picking the right translation options
  - arranging them in the right order

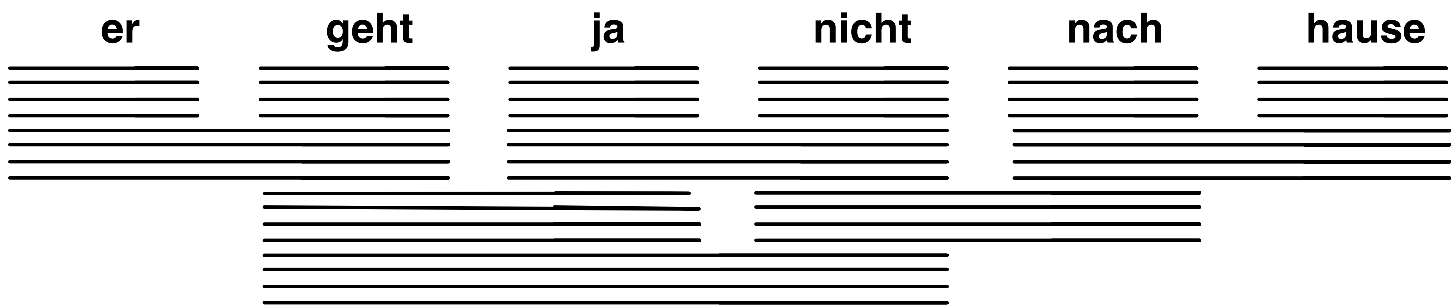
→ Search problem solved by heuristic beam search

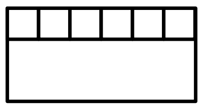
Decoding: Precompute Translation Options



consult phrase translation table for all input phrases

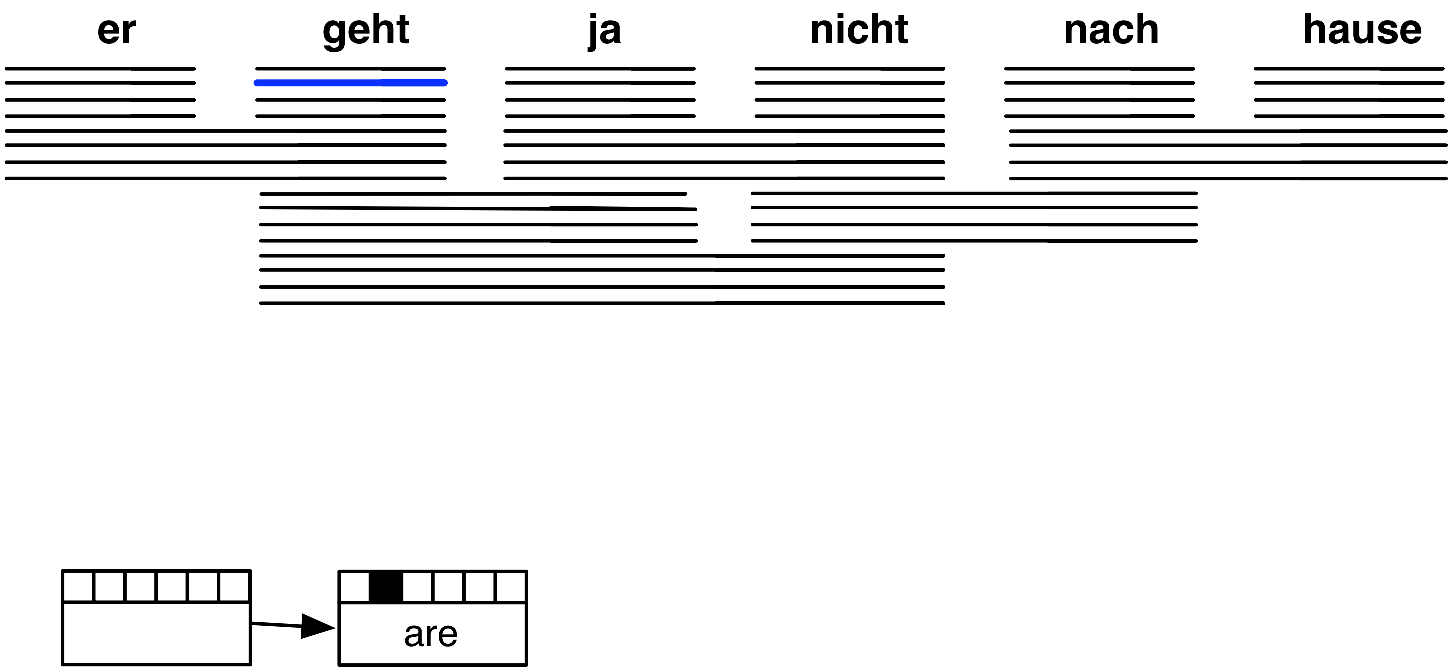
Decoding: Start with Initial Hypothesis





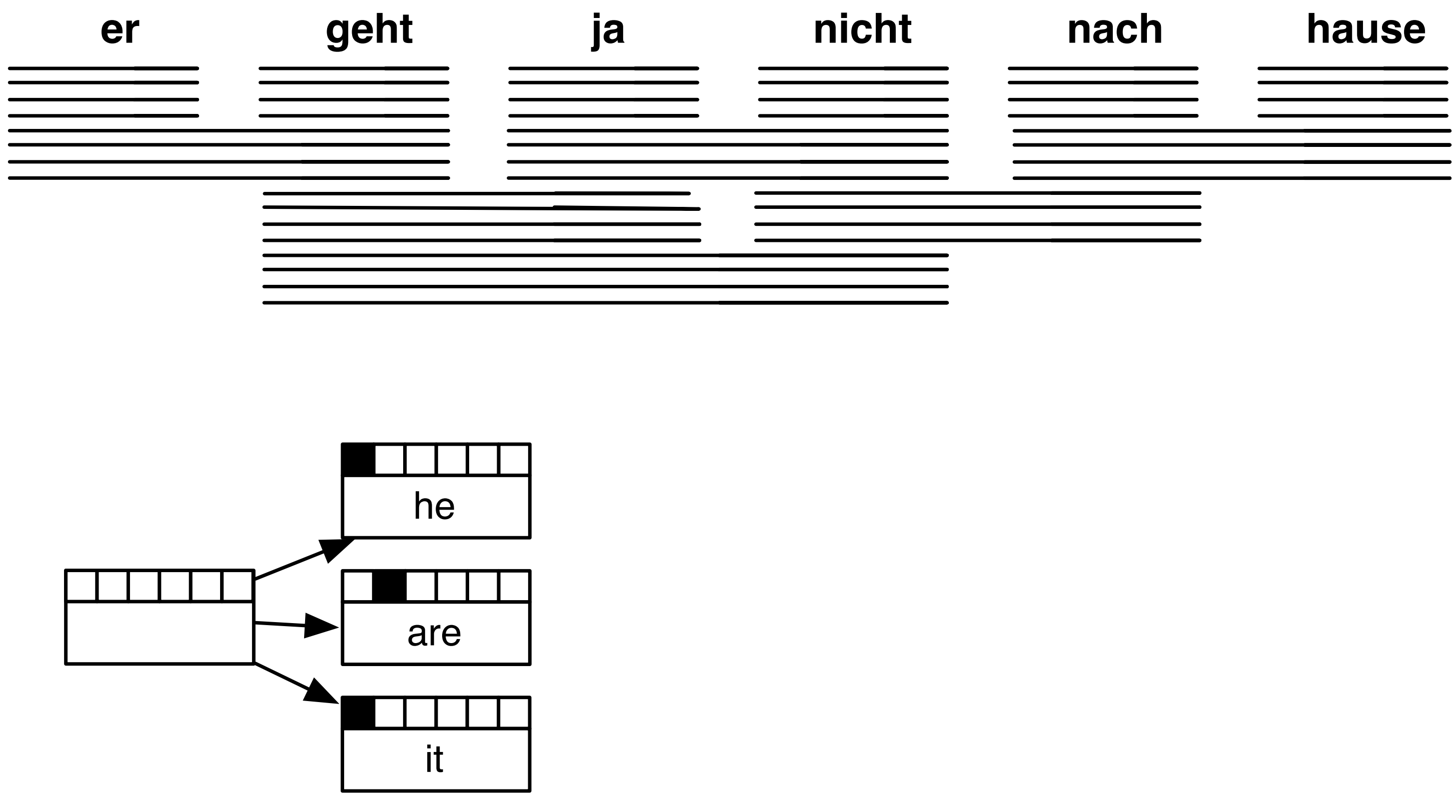
initial hypothesis: no input words covered, no output produced

Decoding: Hypothesis Expansion



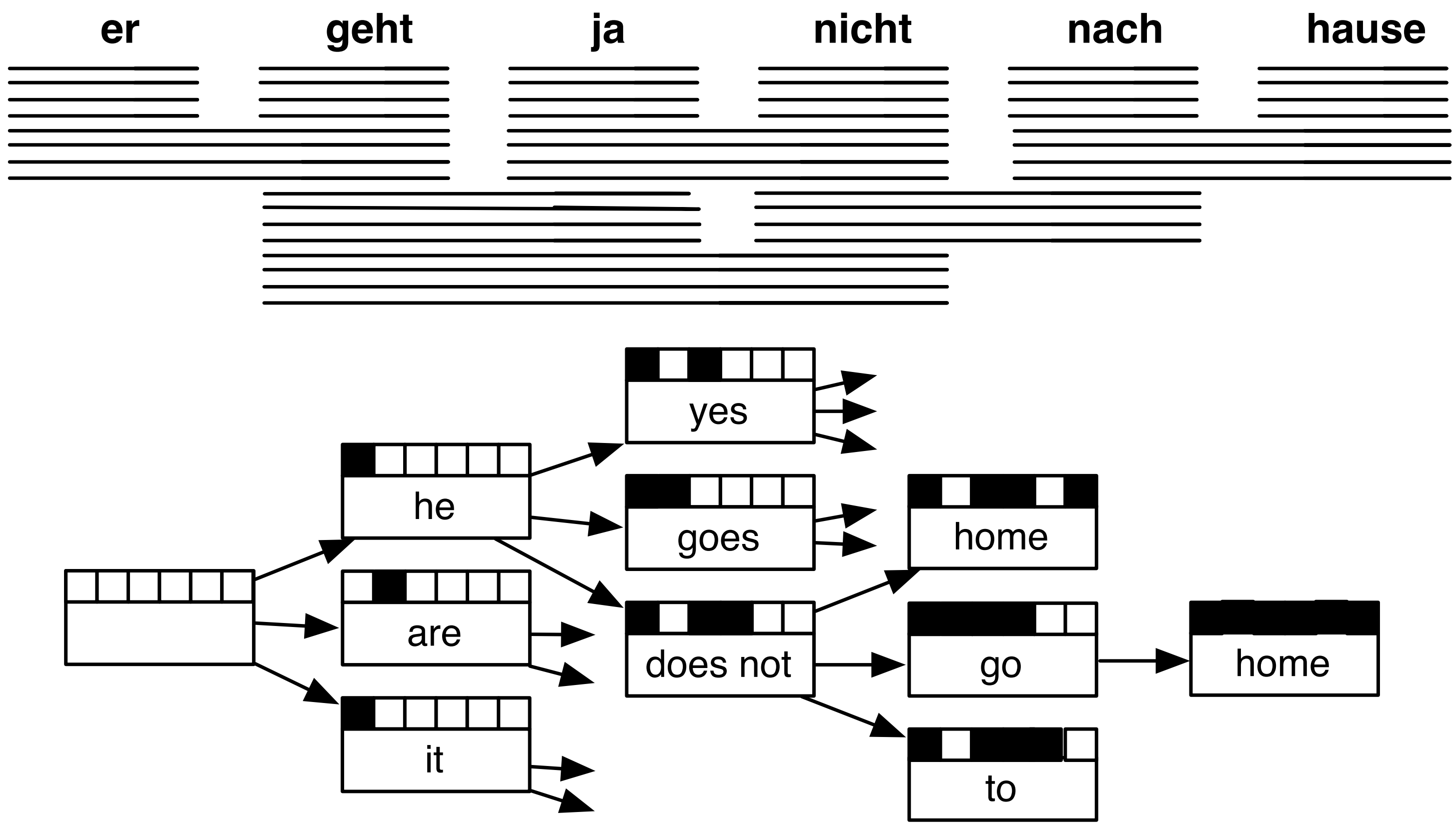
pick any translation option, create new hypothesis

Decoding: Hypothesis Expansion



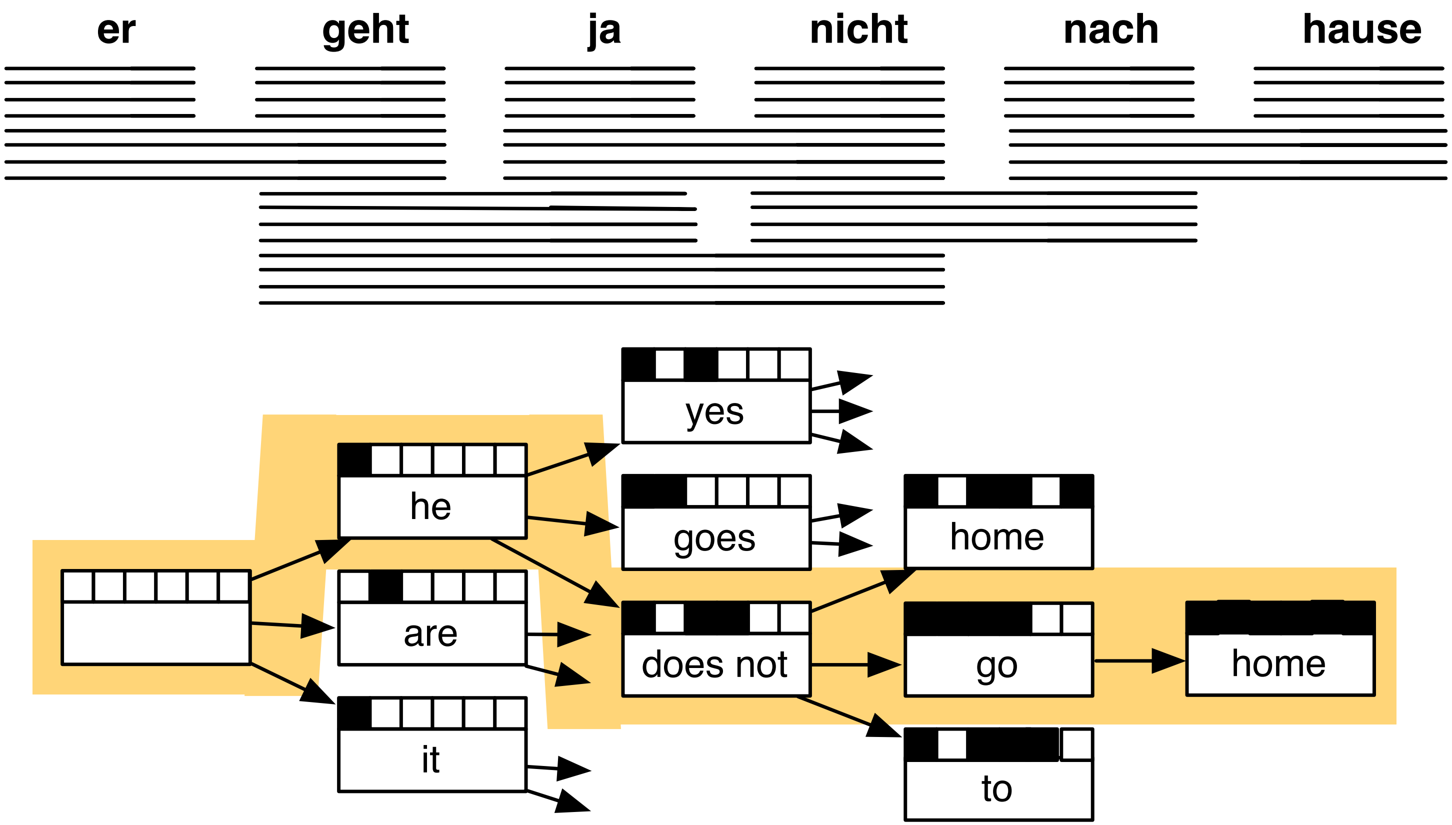
create hypotheses for all other translation options

Decoding: Hypothesis Expansion



also create hypotheses from created partial hypothesis

Decoding: Find Best Path



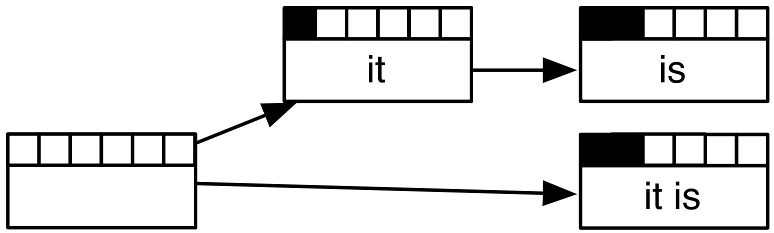
backtrack from highest scoring complete hypothesis

Computational Complexity

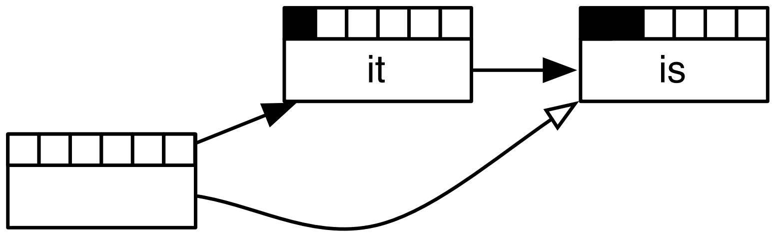
- The suggested process creates exponential number of hypothesis
- Machine translation decoding is NP-complete
- Reduction of search space:
  - recombination (risk-free)
  - pruning (risky)

Recombination

- Two hypothesis paths lead to two matching hypotheses
  - same number of foreign words translated
  - same English words in the output
  - different scores

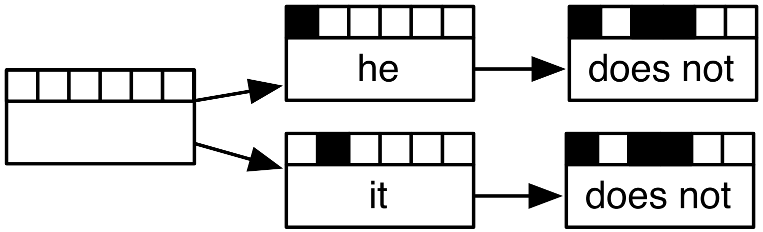


- Worse hypothesis is dropped

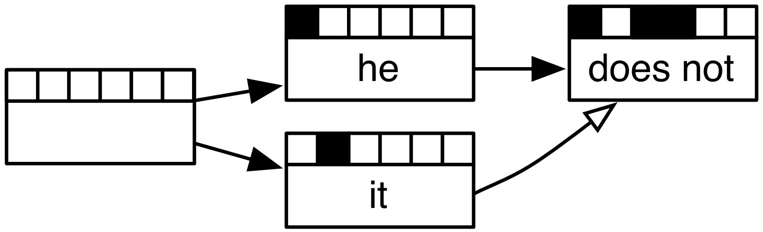


Recombination

- Two hypothesis paths lead to hypotheses indistinguishable in subsequent search
  - same number of foreign words translated
  - same last two English words in output (assuming trigram language model)
  - same last foreign word translated
  - different scores



- Worse hypothesis is dropped



Restrictions on Recombination

- **Translation model:** Phrase translation independent from each other

  $\rightarrow$ no restriction to hypothesis recombination

- **Language model:** Last $n-1$ words used as history in n-gram language model

  $\rightarrow$ recombined hypotheses must match in their last $n-1$ words

- **Reordering model:** Distance-based reordering model based on distance to end position of previous input phrase

  $\rightarrow$ recombined hypotheses must have that same end position

- Other feature function may introduce additional restrictions

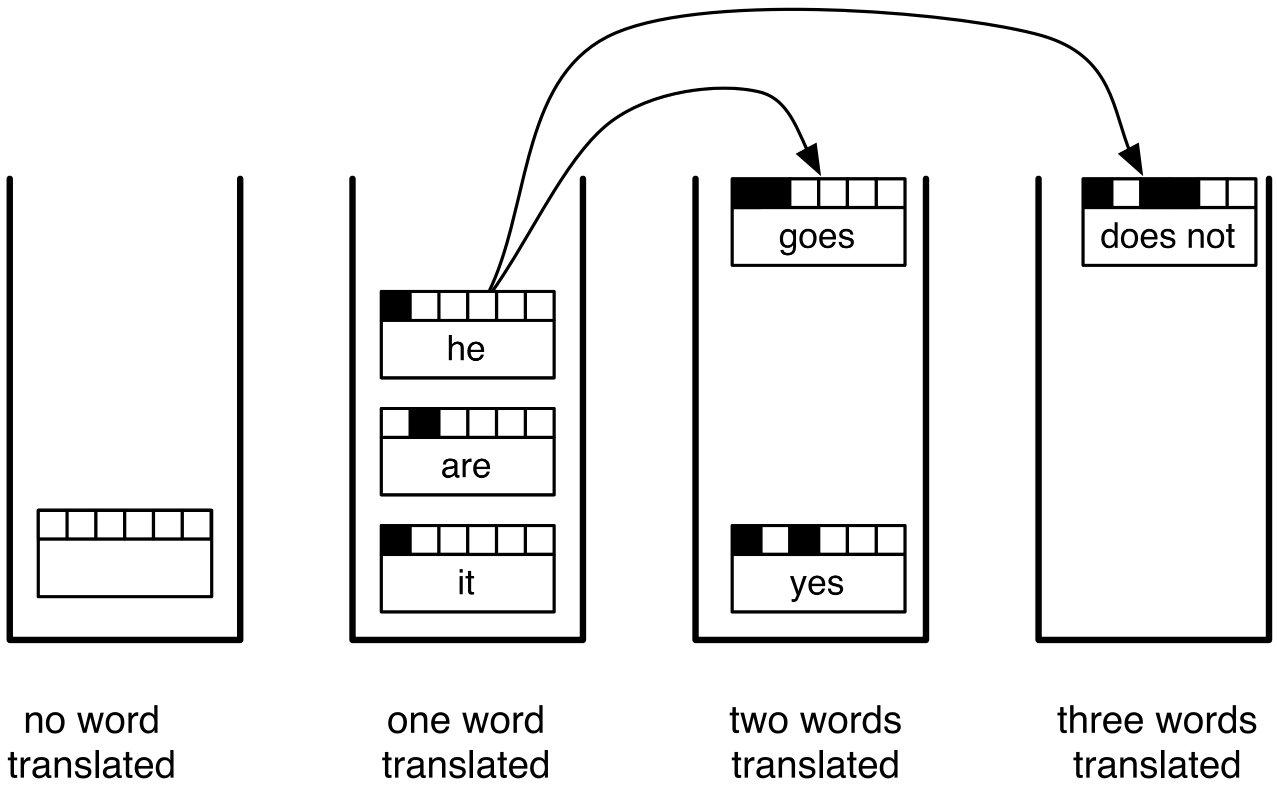
Pruning

- Recombination reduces search space, but not enough

  (we still have a NP complete problem on our hands)

- Pruning: remove bad hypotheses early
  - put comparable hypothesis into stacks
    (hypotheses that have translated same number of input words)
  - limit number of hypotheses in each stack

Stacks



- Hypothesis expansion in a stack decoder
  - translation option is applied to hypothesis
  - new hypothesis is dropped into a stack further down

Stack Decoding Algorithm

```
place empty hypothesis into stack 0
for all stacks 0...n − 1 do
    for all hypotheses in stack do
        for all translation options do
            if applicable then
                create new hypothesis
                place in stack
                recombine with existing hypothesis if possible
                prune stack if too big
            end if
        end for
    end for
end for
```

Pruning

- Pruning strategies
  - histogram pruning: keep at most k hypotheses in each stack
  - stack pruning: keep hypothesis with score $\alpha \times$ best score ($\alpha < 1$)

- Computational time complexity of decoding with histogram pruning

$$O(\text{max stack size} \times \text{translation options} \times \text{sentence length})$$

- Number of translation options is linear with sentence length, hence:

$$O(\text{max stack size} \times \text{sentence length}^2)$$

- Quadratic complexity

Reordering Limits

- Limiting reordering to maximum reordering distance

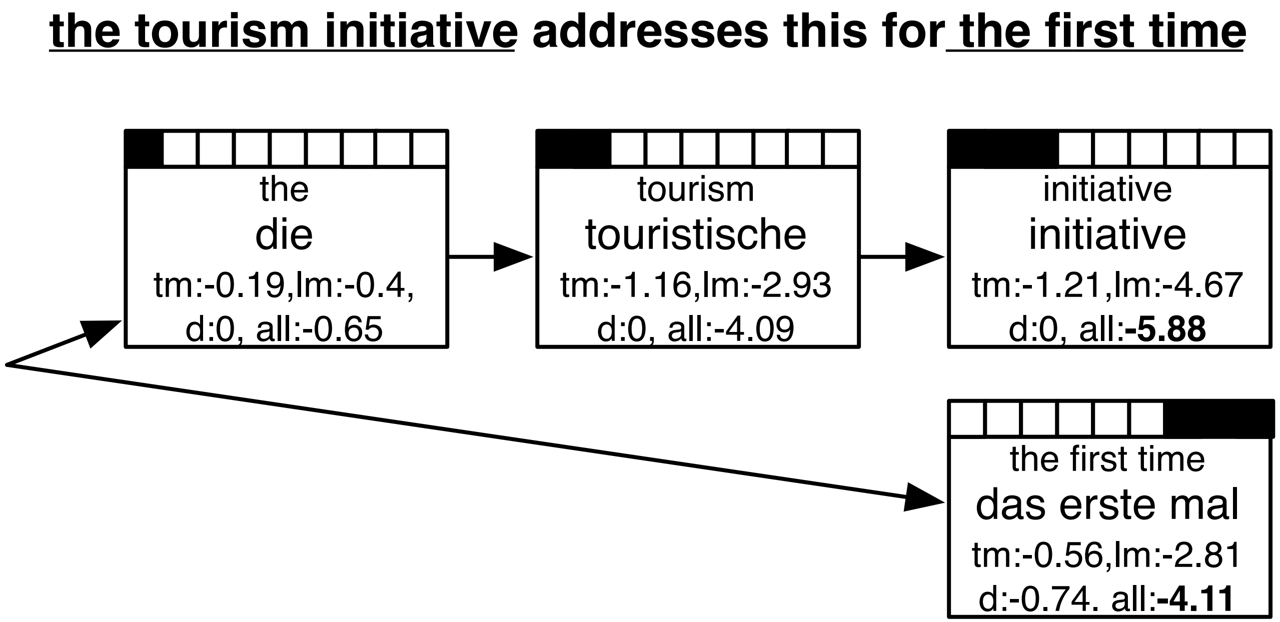
- Typical reordering distance 5–8 words
  - depending on language pair
  - larger reordering limit hurts translation quality

- Reduces complexity to linear

$$O(\text{max stack size} \times \text{sentence length})$$

- Speed / quality trade-off by setting maximum stack size

Translating the Easy Part First?

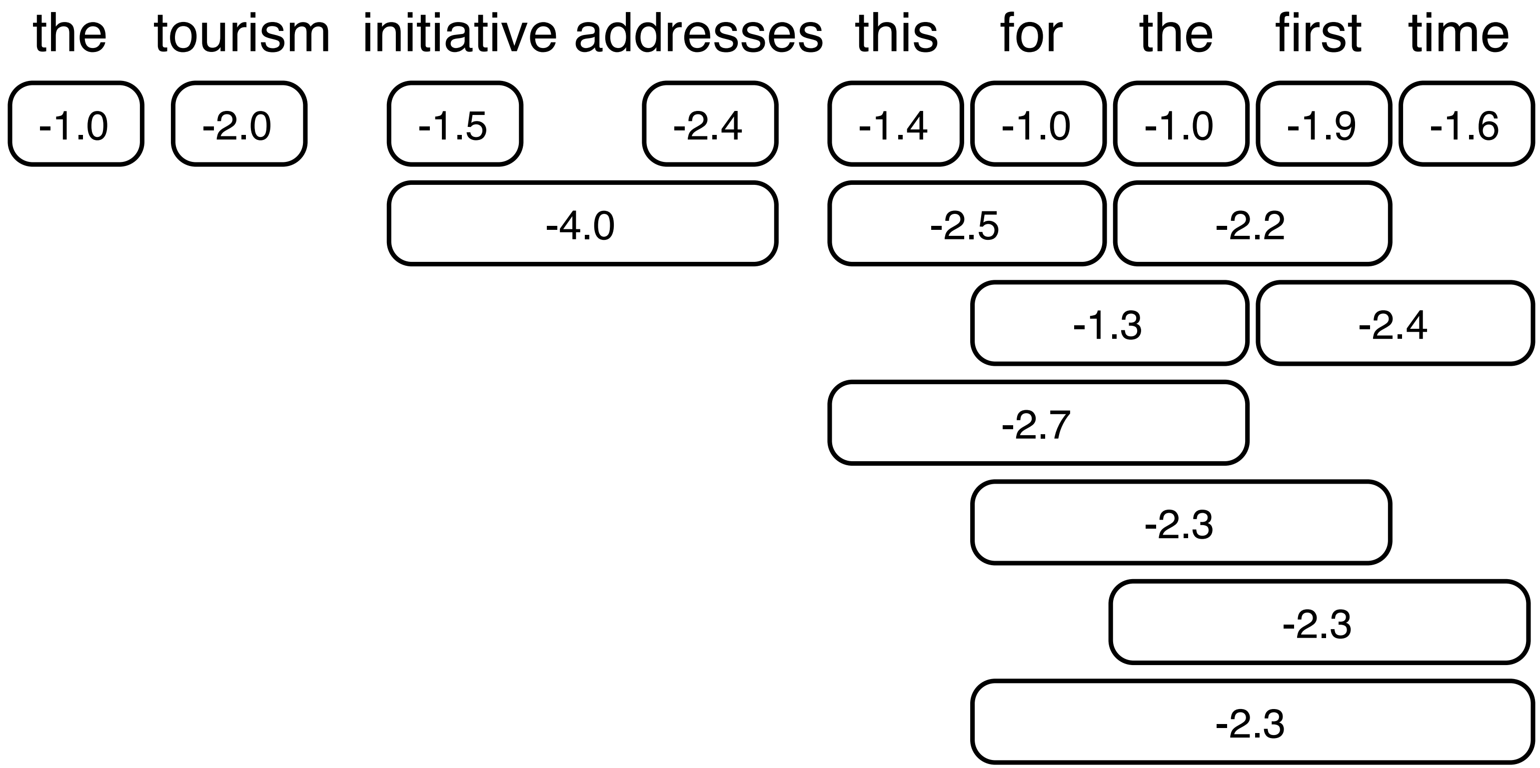


both hypotheses translate 3 words
worse hypothesis has better score

Estimating Future Cost

- Future cost estimate: how expensive is translation of rest of sentence?
- Optimistic: choose cheapest translation options
- Cost for each translation option
  - **translation model**: cost known
  - **language model:** output words known, but not context
    $\rightarrow$ estimate without context
  - **reordering model:** unknown, ignored for future cost estimation

Cost Estimates from Translation Options



cost of cheapest translation options for each input span (log-probabilities)

Statistical Machine Translation, Philipp Koehn

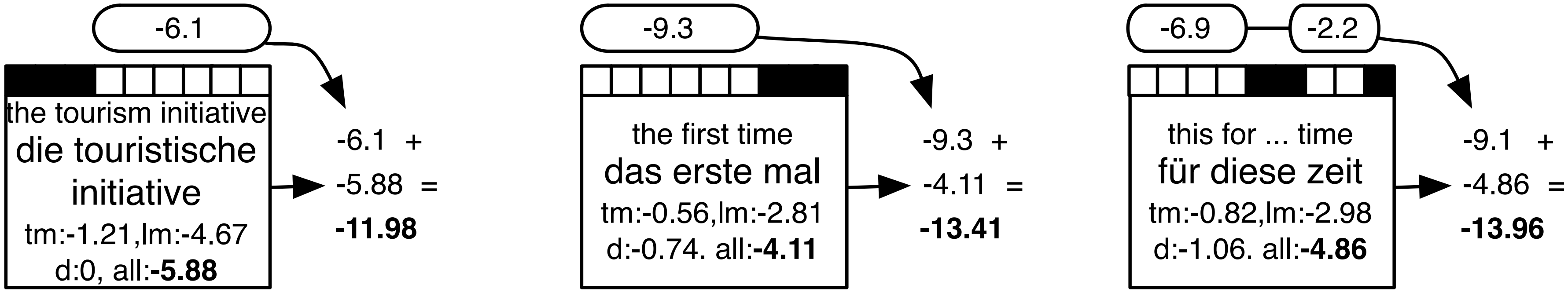
Cost Estimates for all Spans

- Compute cost estimate for all contiguous spans by combining cheapest options

first word	future cost estimate for n words (from first)								
	1	2	3	4	5	6	7	8	9
the	-1.0	-3.0	-4.5	-6.9	-8.3	-9.3	-9.6	-10.6	-10.6
tourism	-2.0	-3.5	-5.9	-7.3	-8.3	-8.6	-9.6	-9.6	
initiative	-1.5	-3.9	-5.3	-6.3	-6.6	-7.6	-7.6		
addresses	-2.4	-3.8	-4.8	-5.1	-6.1	-6.1			
this	-1.4	-2.4	-2.7	-3.7	-3.7				
for	-1.0	-1.3	-2.3	-2.3					
the	-1.0	-2.2	-2.3						
first	-1.9	-2.4							
time	-1.6								

- Function words cheaper (the: -1.0) than content words (tourism -2.0)
- Common phrases cheaper (for the first time: -2.3) than unusual ones (tourism initiative addresses: -5.9)

Combining Score and Future Cost

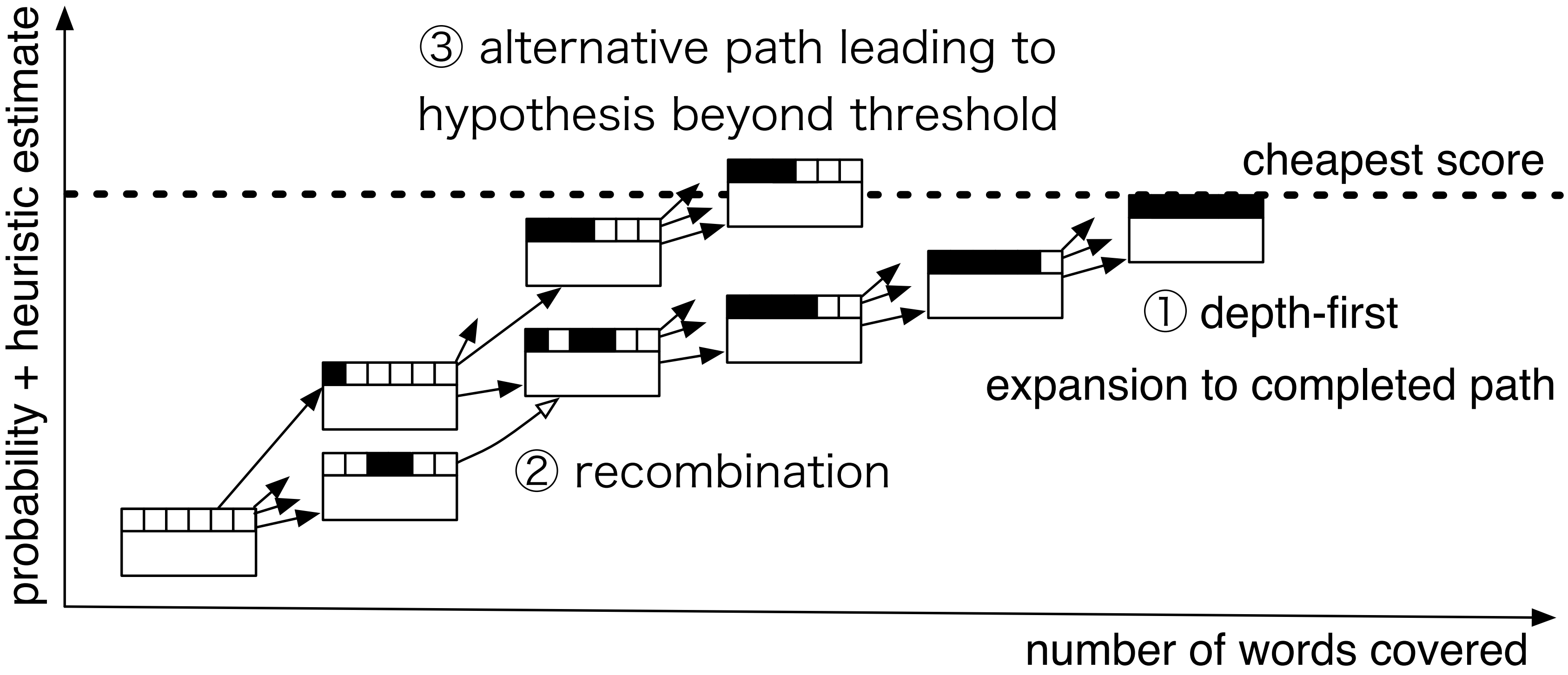


- Hypothesis score and future cost estimate are combined for pruning
  - left hypothesis starts with hard part: the tourism initiative
    score: -5.88, future cost: -6.1 → total cost -11.98
  - middle hypothesis starts with easiest part: the first time
    score: -4.11, future cost: -9.3 → total cost -13.41
  - right hypothesis picks easy parts: this for ... time
    score: -4.86, future cost: -9.1 → total cost -13.96

Other Decoding Algorithms

- A* search
- Greedy hill-climbing
- Using finite state transducers (standard toolkits)

A* Search



- Uses *admissible* future cost heuristic: never overestimates cost
- Translation agenda: create hypothesis with lowest score + heuristic cost
- Done, when complete hypothesis created

Greedy Hill-Climbing

- Create one complete hypothesis with depth-first search (or other means)
- Search for better hypotheses by applying change operators
  - change the translation of a word or phrase
  - combine the translation of two words into a phrase
  - split up the translation of a phrase into two smaller phrase translations
  - move parts of the output into a different position
  - swap parts of the output with the output at a different part of the sentence
- Terminates if no operator application produces a better translation

Summary

- Translation process: produce output left to right
- Translation options
- Decoding by hypothesis expansion
- Reducing search space
  - recombination
  - pruning (requires future cost estimate)
- Other decoding algorithms

Decoding Demo

- http://mt-class.org/jhu/stack-decoder/
- Coded by Matt Post http://cs.jhu.edu/~post/
- Install from https://github.com/mjpost/stack-decoder

TREE-BASED MODELS

Tree-Based Models

- Traditional statistical models operate on sequences of words

- Many translation problems can be best explained by pointing to syntax
  - reordering, e.g., verb movement in German–English translation
  - long distance agreement (e.g., subject-verb) in output

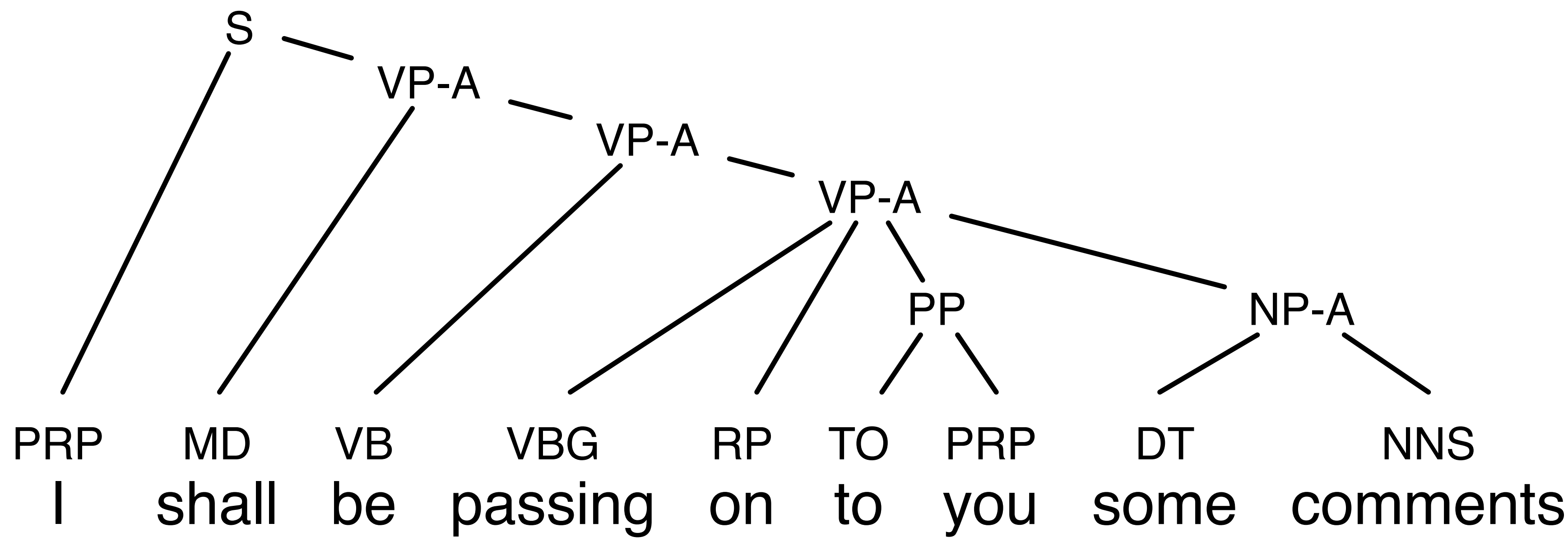
⇒ Translation models based on tree representation of language
  - significant ongoing research
  - state-of-the art for some language pairs

Phrase Structure Grammar

- Phrase structure
  - noun phrases: the big man, a house, ...
  - prepositional phrases: at 5 o'clock, in Edinburgh, ...
  - verb phrases: going out of business, eat chicken, ...
  - adjective phrases, ...

- Context-free Grammars (CFG)
  - non-terminal symbols: phrase structure labels, part-of-speech tags
  - terminal symbols: words
  - production rules: $\text{NT} \rightarrow [\text{NT},\text{T}]+$
    example: $\text{NP} \rightarrow \text{DET NN}$

Phrase Structure Grammar



Phrase structure grammar tree for an English sentence
(as produced Collins' parser)

Synchronous Phrase Structure Grammar

- English rule

$$\text{NP} \longrightarrow \text{DET JJ NN}$$

- French rule

$$\text{NP} \longrightarrow \text{DET NN JJ}$$

- Synchronous rule (indices indicate alignment):

$$\text{NP} \longrightarrow \text{DET}_1 \ \text{NN}_2 \ \text{JJ}_3 \mid \text{DET}_1 \ \text{JJ}_3 \ \text{NN}_2$$

Synchronous Grammar Rules

- Nonterminal rules

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

- Terminal rules

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

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

- Mixed rules

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

Tree-Based Translation Model

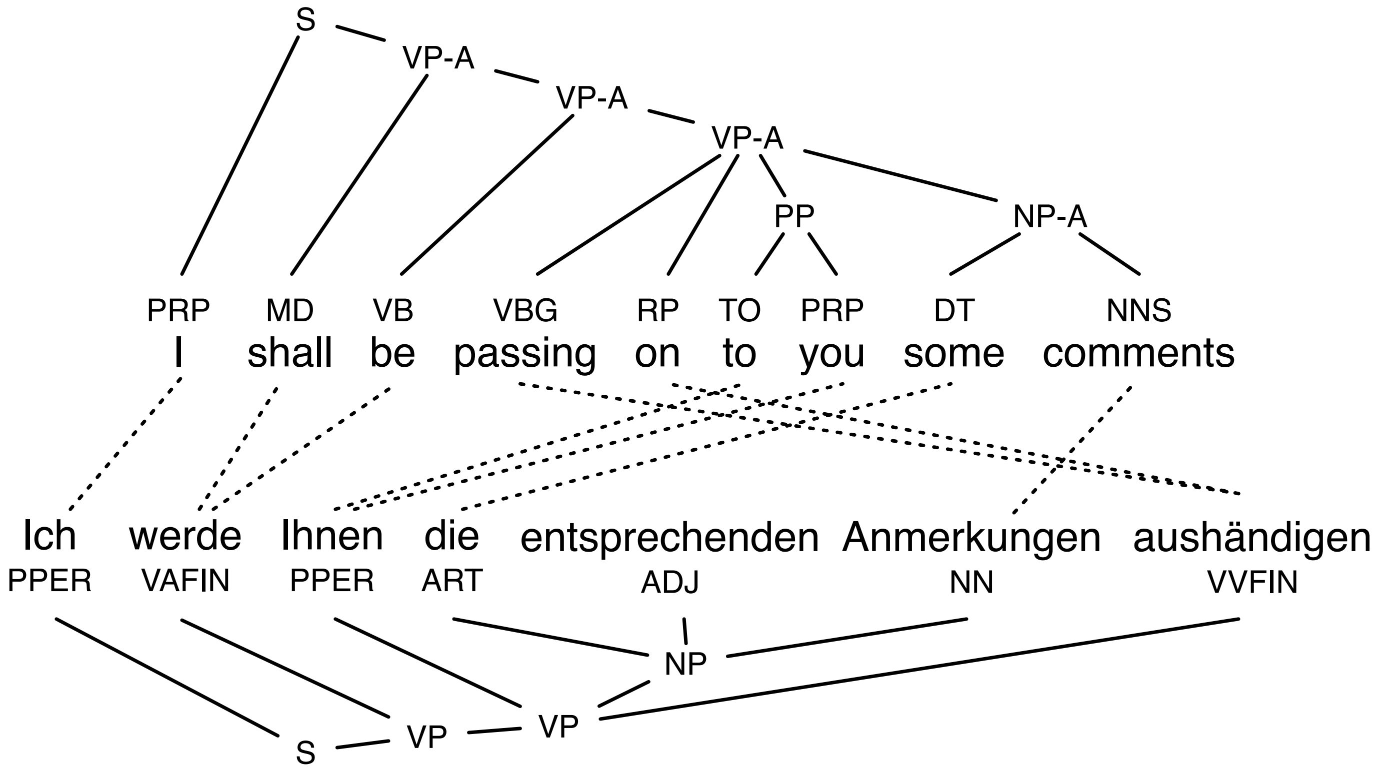
- Translation by parsing
  - synchronous grammar has to parse entire input sentence
  - output tree is generated at the same time
  - process is broken up into a number of rule applications

- Translation probability

$$\text{SCORE}(\text{TREE}, \text{E}, \text{F}) = \prod_i \text{RULE}_i$$

- Many ways to assign probabilities to rules

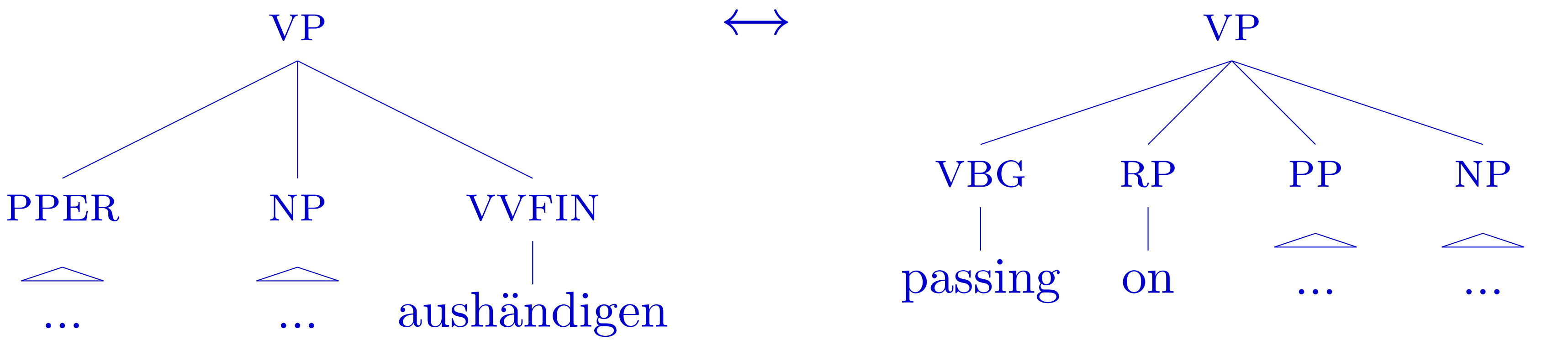
Aligned Tree Pair



Phrase structure grammar trees with word alignment
(German–English sentence pair.)

Reordering Rule

- Subtree alignment



- Synchronous grammar rule

$$\text{VP} \rightarrow \text{PPER}_1\ \text{NP}_2\ \text{aushändigen} \ \big|\ \text{passing on}\ \text{PP}_1\ \text{NP}_2$$

- Note:
  - one word aushändigen mapped to two words passing on ok
  - but: fully non-terminal rule not possible (one-to-one mapping constraint for nonterminals)

Another Rule

- Subtree alignment

- Synchronous grammar rule (stripping out English internal structure)

$$\text{PRO/PP} \rightarrow \text{Ihnen} \mid \text{to you}$$

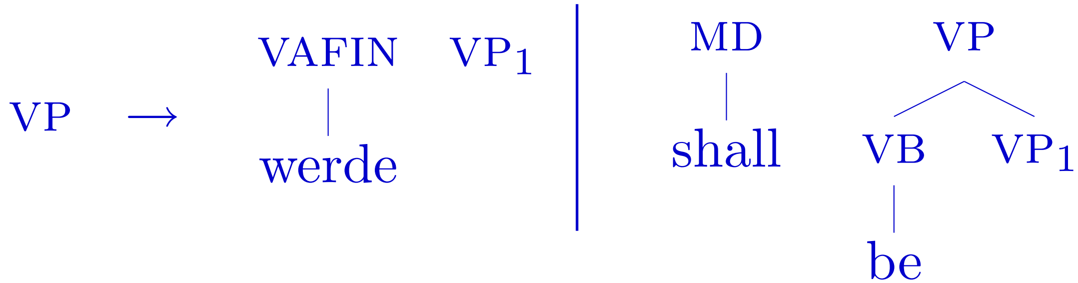
- Rule with internal structure

Another Rule

- Translation of German werde to English shall be

- Translation rule needs to include mapping of VP

⇒ Complex rule



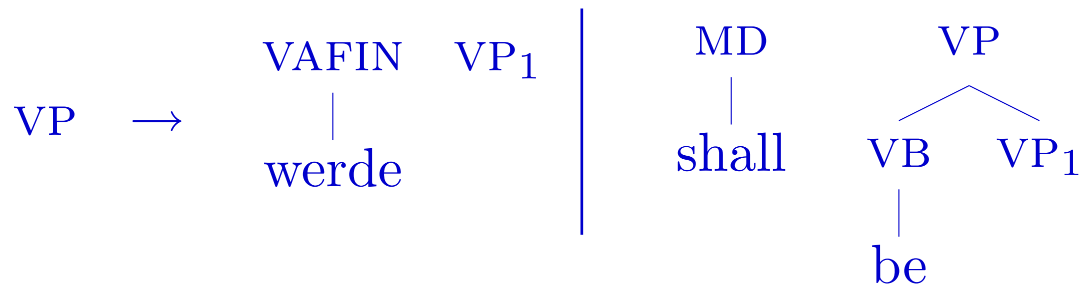
Internal Structure

- Stripping out internal structure

$$\text{VP} \rightarrow \text{werde VP}_1 \;|\; \text{shall be VP}_1$$

$\Rightarrow$ synchronous context free grammar

- Maintaining internal structure



$\Rightarrow$ synchronous tree substitution grammar

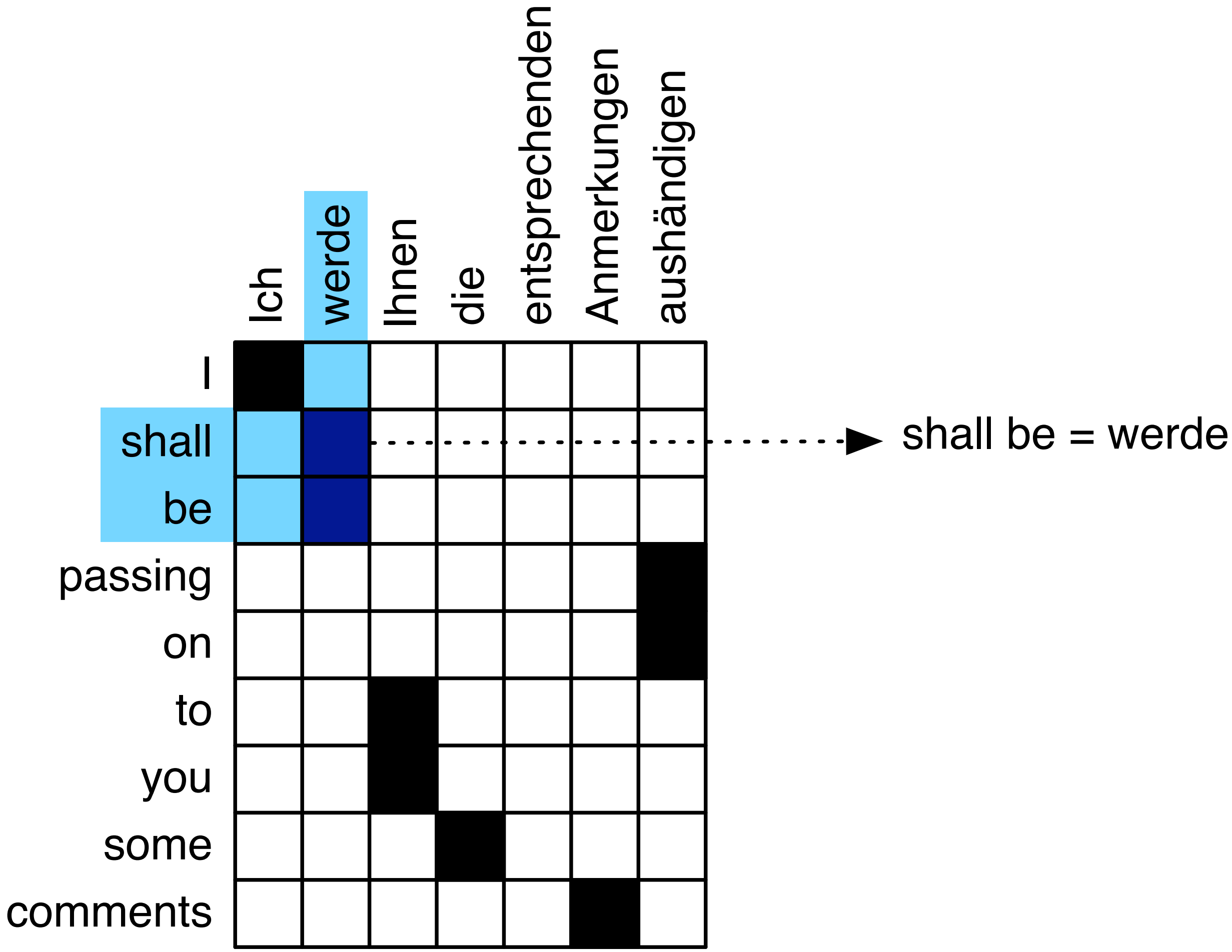
Learning Synchronous Grammars

- Extracting rules from a word-aligned parallel corpus

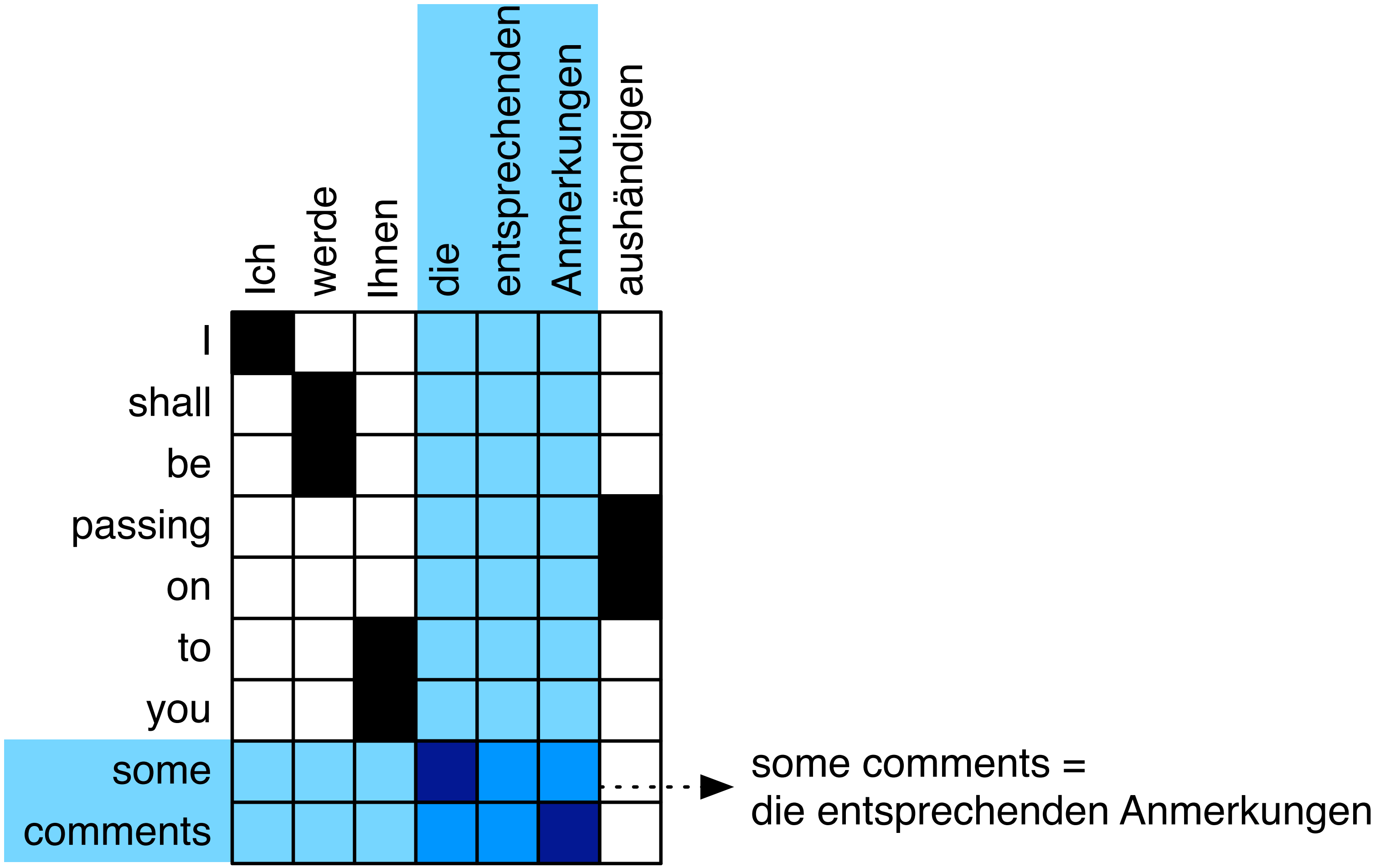
- First: Hierarchical phrase-based model
  - only one non-terminal symbol X
  - no linguistic syntax, just a formally syntactic model

- Then: Synchronous phrase structure model
  - non-terminals for words and phrases: NP, VP, PP, ADJ, ...
  - corpus must also be parsed with syntactic parser

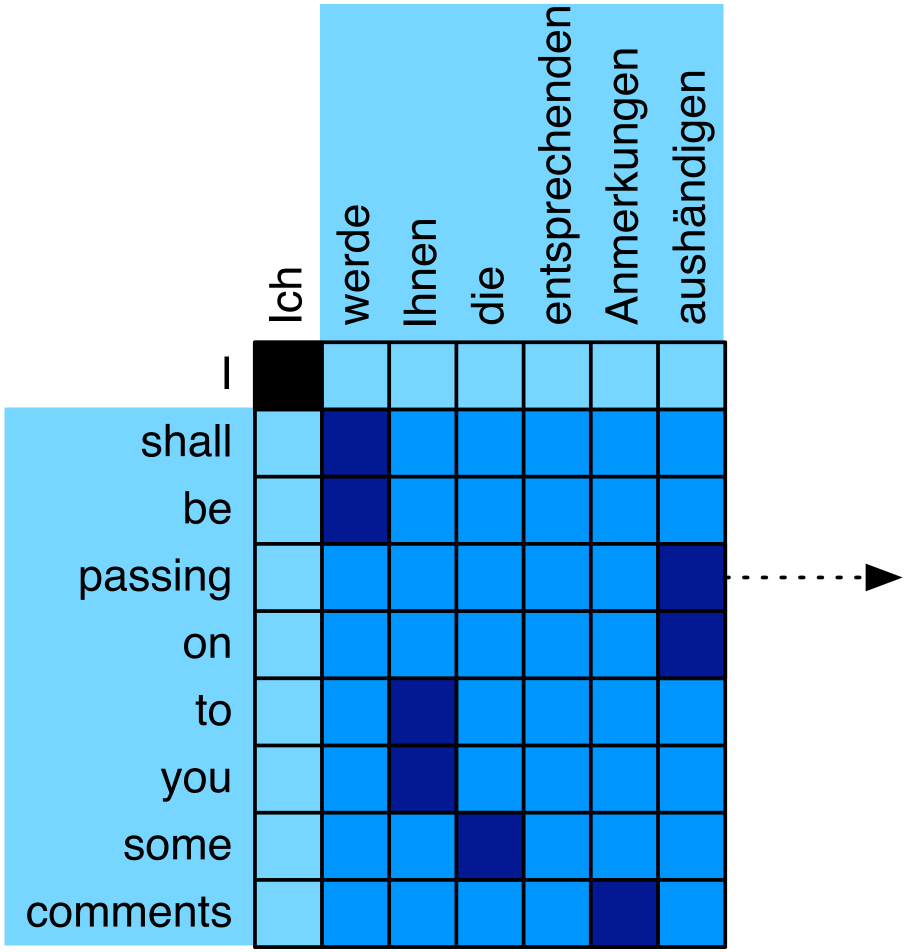
Extracting Phrase Translation Rules



Extracting Phrase Translation Rules

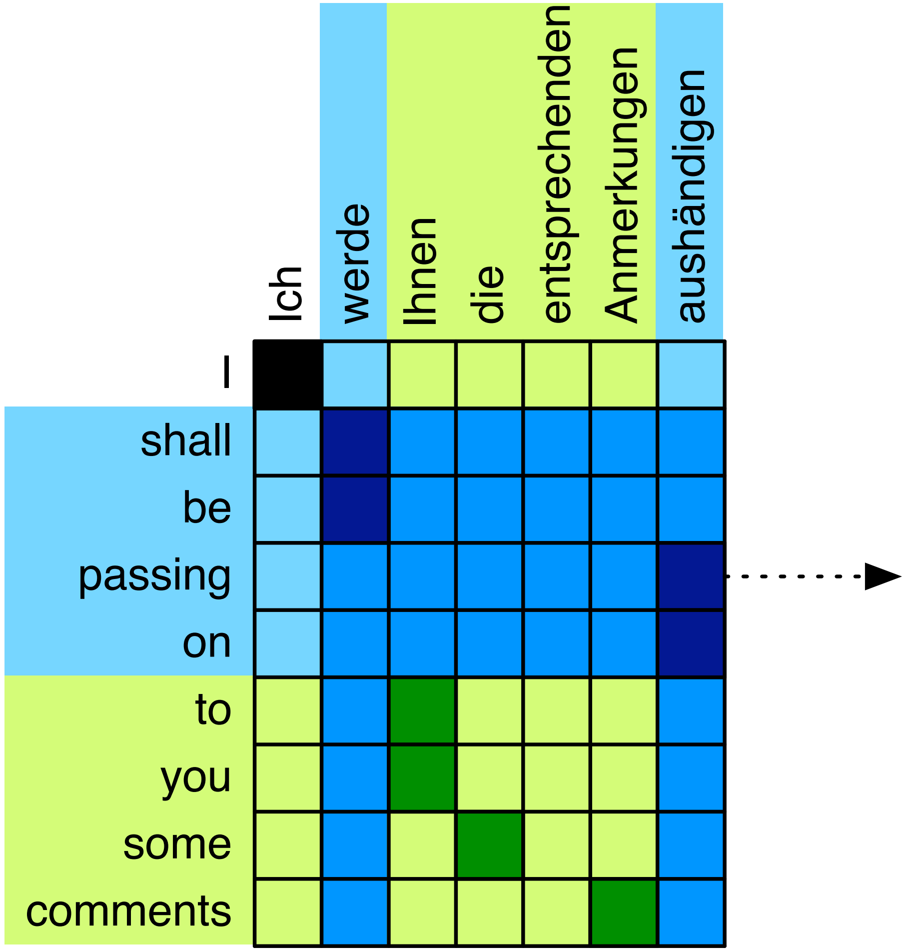


Extracting Phrase Translation Rules



werde Ihnen die entsprechenden Anmerkungen aushändigen
= shall be passing on to you some comments

Extracting Hierarchical Phrase Translation Rules



subtracting subphrase

werde X aushändigen
= shall be passing on X

Formal Definition

- Recall: consistent phrase pairs

$$
\begin{aligned}
(\bar{e}, \bar{f}) \text{ consistent with } A \Leftrightarrow \\
& \forall e_i \in \bar{e} : (e_i, f_j) \in A \rightarrow f_j \in \bar{f} \\
& \text{AND } \forall f_j \in \bar{f} : (e_i, f_j) \in A \rightarrow e_i \in \bar{e} \\
& \text{AND } \exists e_i \in \bar{e}, f_j \in \bar{f} : (e_i, f_j) \in A
\end{aligned}
$$

- Let P be the set of all extracted phrase pairs $(\bar{e}, \bar{f})$

Statistical Machine Translation, Philipp Koehn

Formal Definition

- Extend recursively:

$$\text{if } (\bar{e}, \bar{f}) \in P \text{ AND } (\bar{e}_{\text{SUB}}, \bar{f}_{\text{SUB}}) \in P$$
$$\text{AND } \bar{e} = \bar{e}_{\text{PRE}} + \bar{e}_{\text{SUB}} + \bar{e}_{\text{POST}}$$
$$\text{AND } \bar{f} = \bar{f}_{\text{PRE}} + \bar{f}_{\text{SUB}} + \bar{f}_{\text{POST}}$$
$$\text{AND } \bar{e} \neq \bar{e}_{\text{SUB}} \text{ AND } \bar{f} \neq \bar{f}_{\text{SUB}}$$
$$\text{add } (e_{\text{PRE}} + \text{X} + e_{\text{POST}}, f_{\text{PRE}} + \text{X} + f_{\text{POST}}) \text{ to } P$$

(note: any of e_{PRE}, e_{POST}, f_{PRE}, or f_{POST} may be empty)

- Set of hierarchical phrase pairs is the closure under this extension mechanism

Statistical Machine Translation, Philipp Koehn

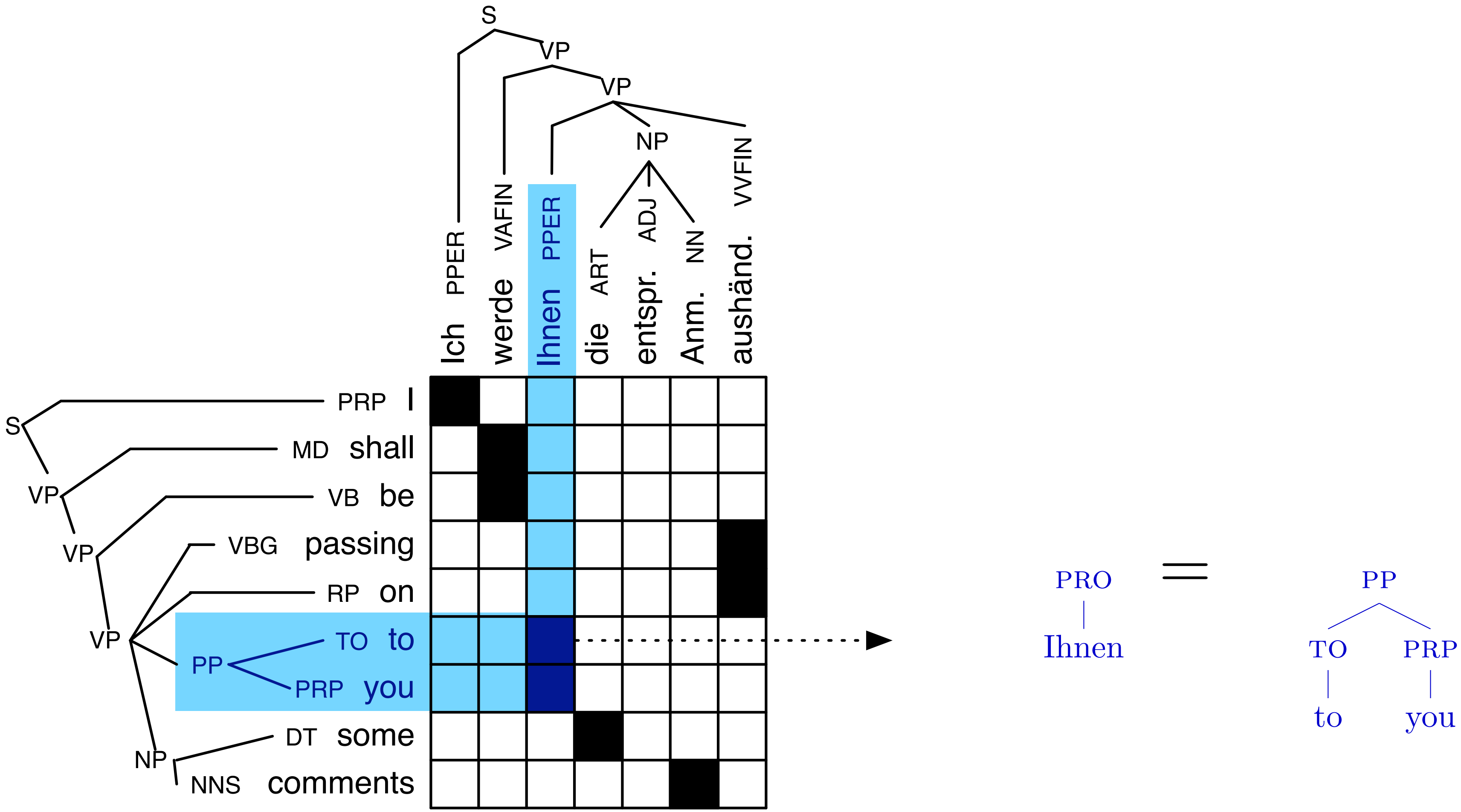
Comments

- Removal of multiple sub-phrases leads to rules with multiple non-terminals, such as:

$$\text{Y} \rightarrow \text{X}_1 \ \text{X}_2 \ \mid \ \text{X}_2 \ \textit{of} \ \text{X}_1$$

- Typical restrictions to limit complexity [Chiang, 2005]
  - at most 2 nonterminal symbols
  - at least 1 but at most 5 words per language
  - span at most 15 words (counting gaps)

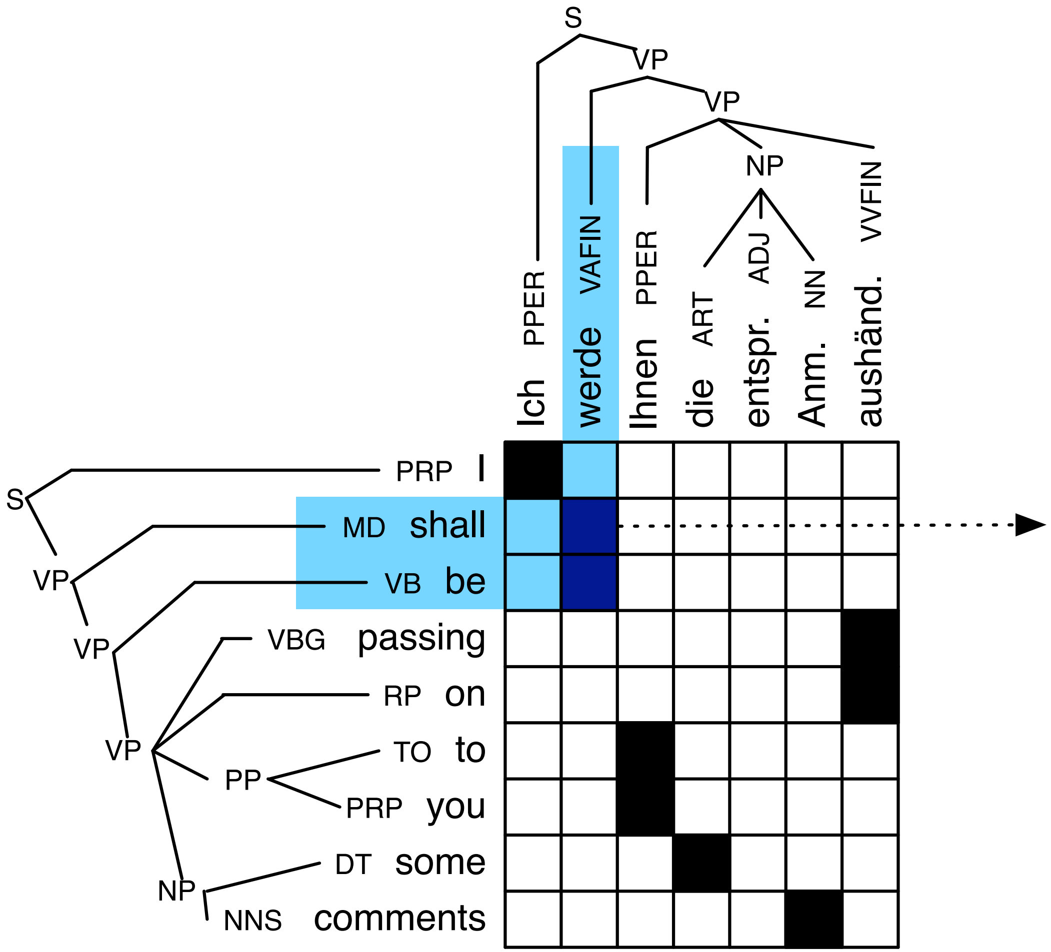
Learning Syntactic Translation Rules



Constraints on Syntactic Rules

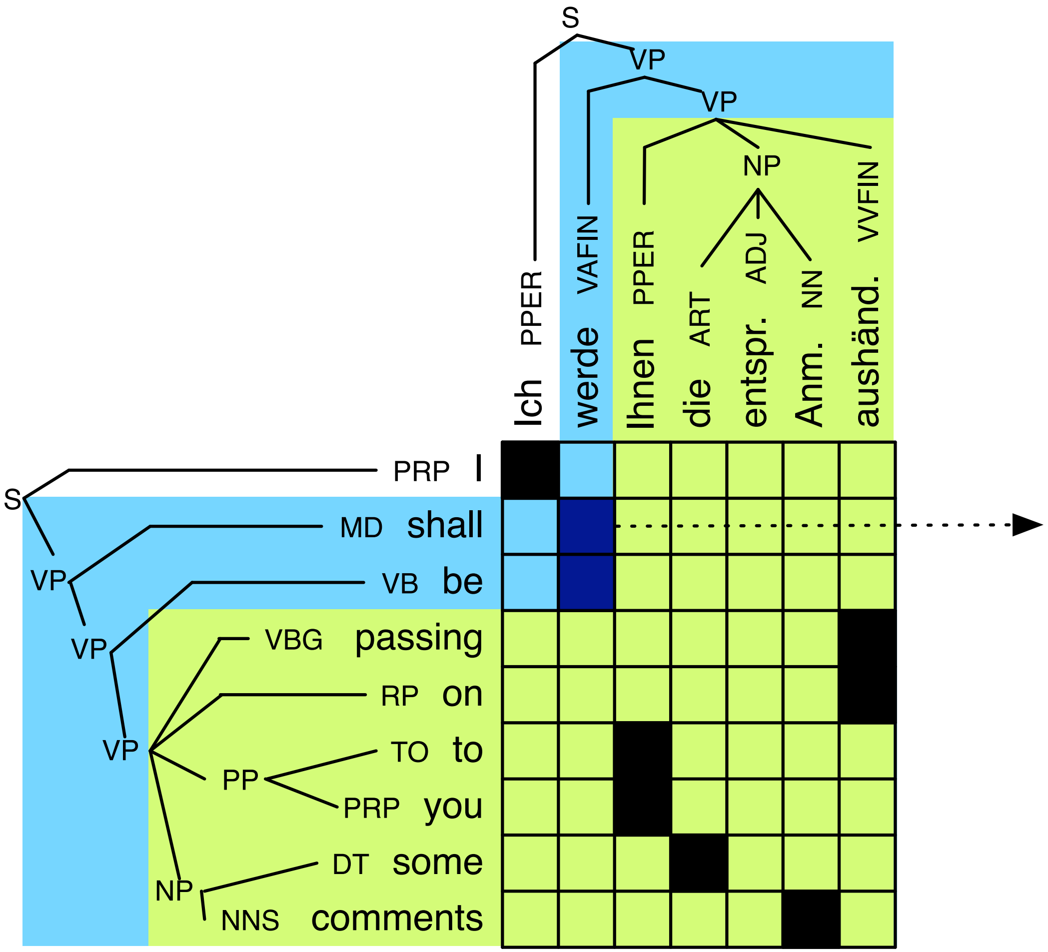
- Same word alignment constraints as hierarchical models
- Hierarchical: rule can cover any span
  $\Leftrightarrow$ syntactic rules must cover constituents in the tree
- Hierarchical: gaps may cover any span
  $\Leftrightarrow$ gaps must cover constituents in the tree
- Much less rules are extracted (all things being equal)

Impossible Rules

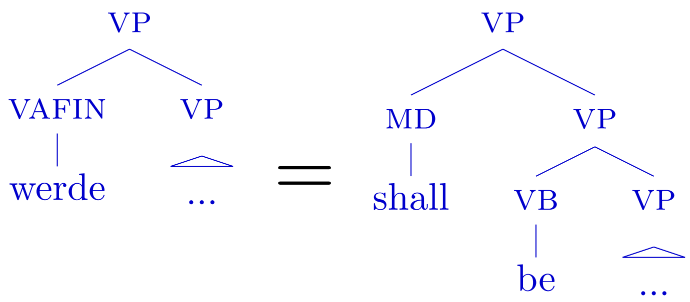


English span not a constituent
no rule extracted

Rules with Context



Rule with this phrase pair requires syntactic context



Too Many Rules Extractable

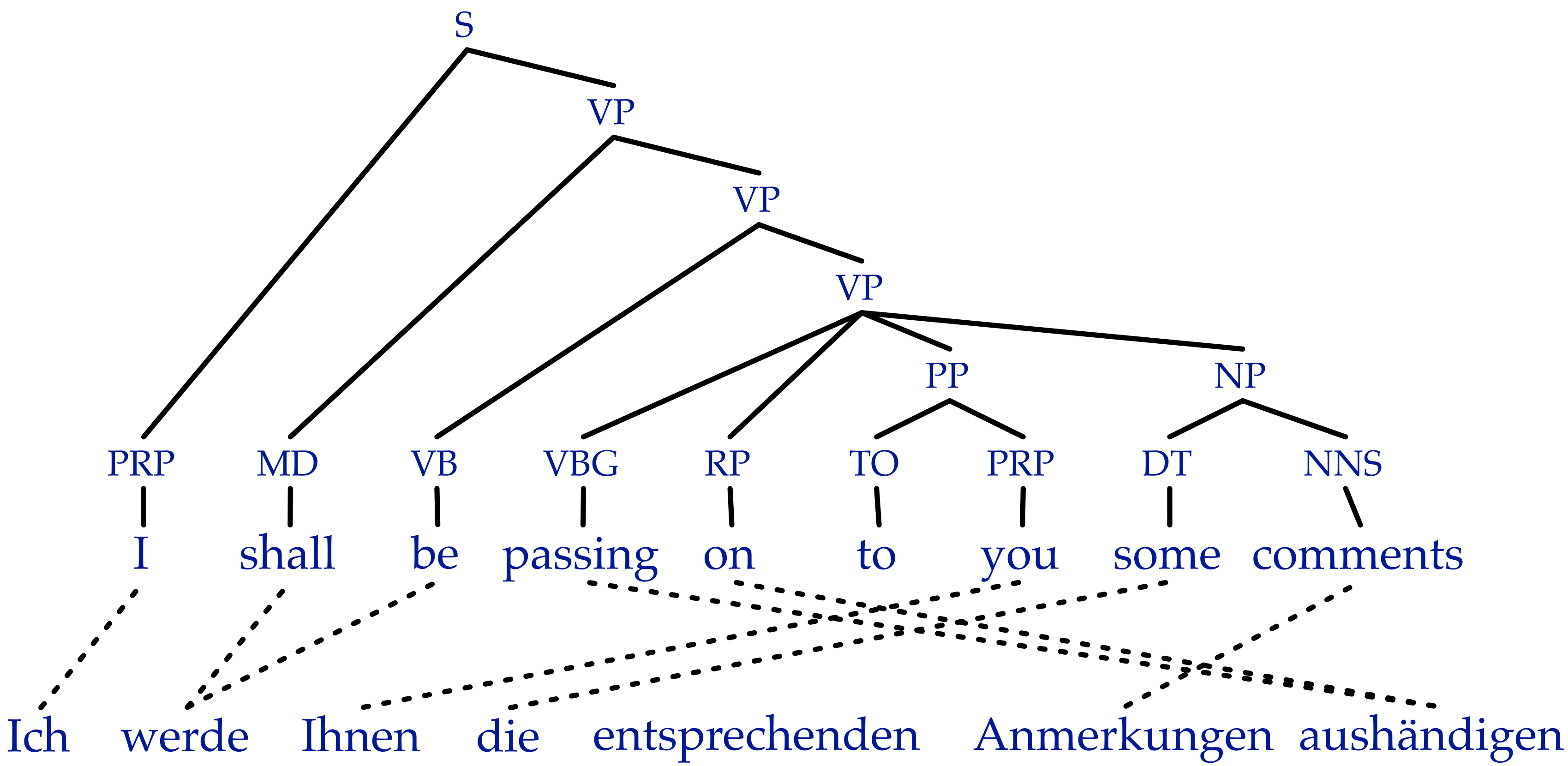
- Huge number of rules can be extracted
  (every alignable node may or may not be part of a rule $\longrightarrow$ exponential number of rules)

- Need to limit which rules to extract

- Option 1: similar restriction as for hierarchical model
  (maximum span size, maximum number of terminals and non-terminals, etc.)

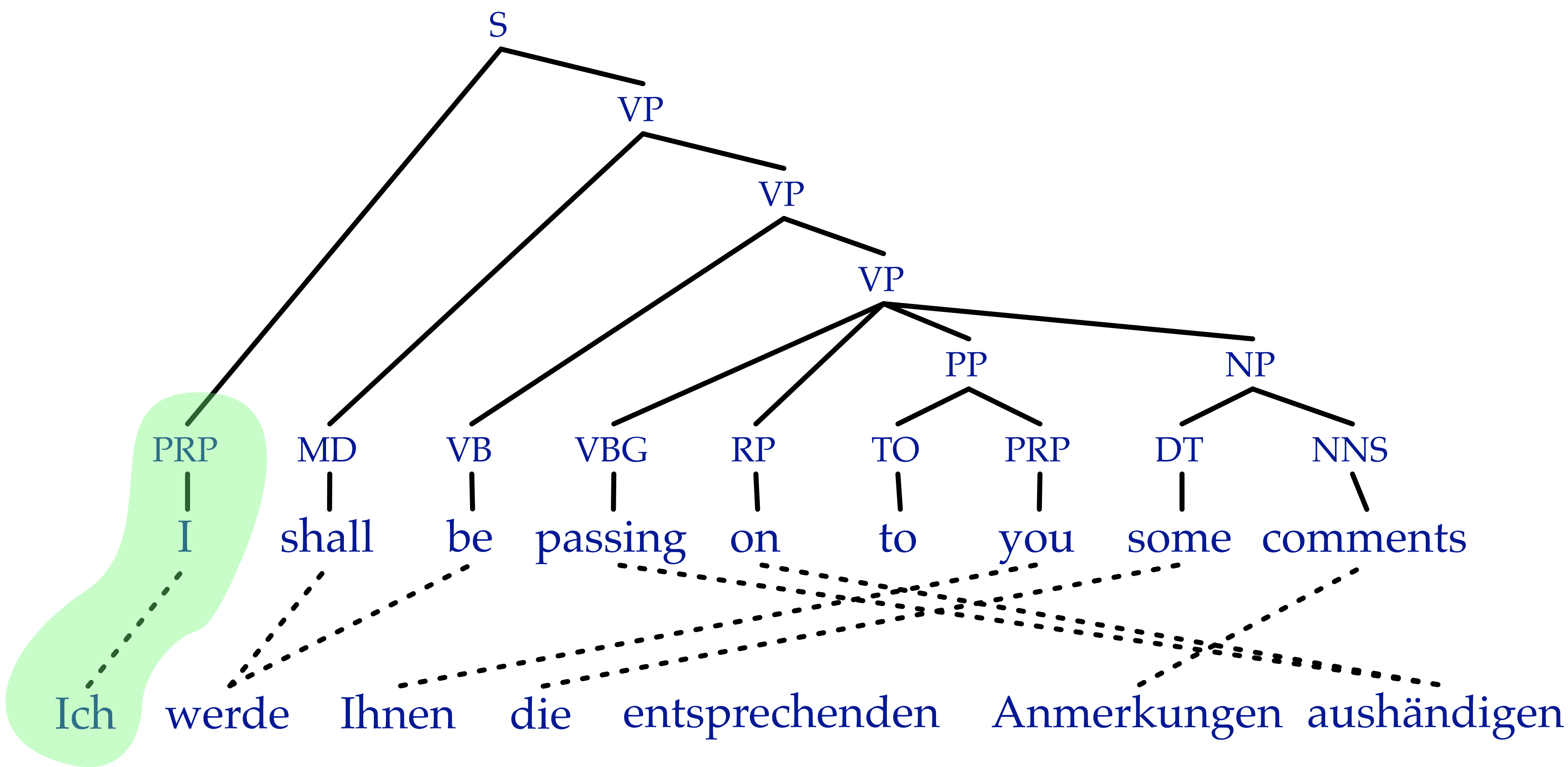
- Option 2: only extract minimal rules ("GHKM" rules)

Minimal Rules



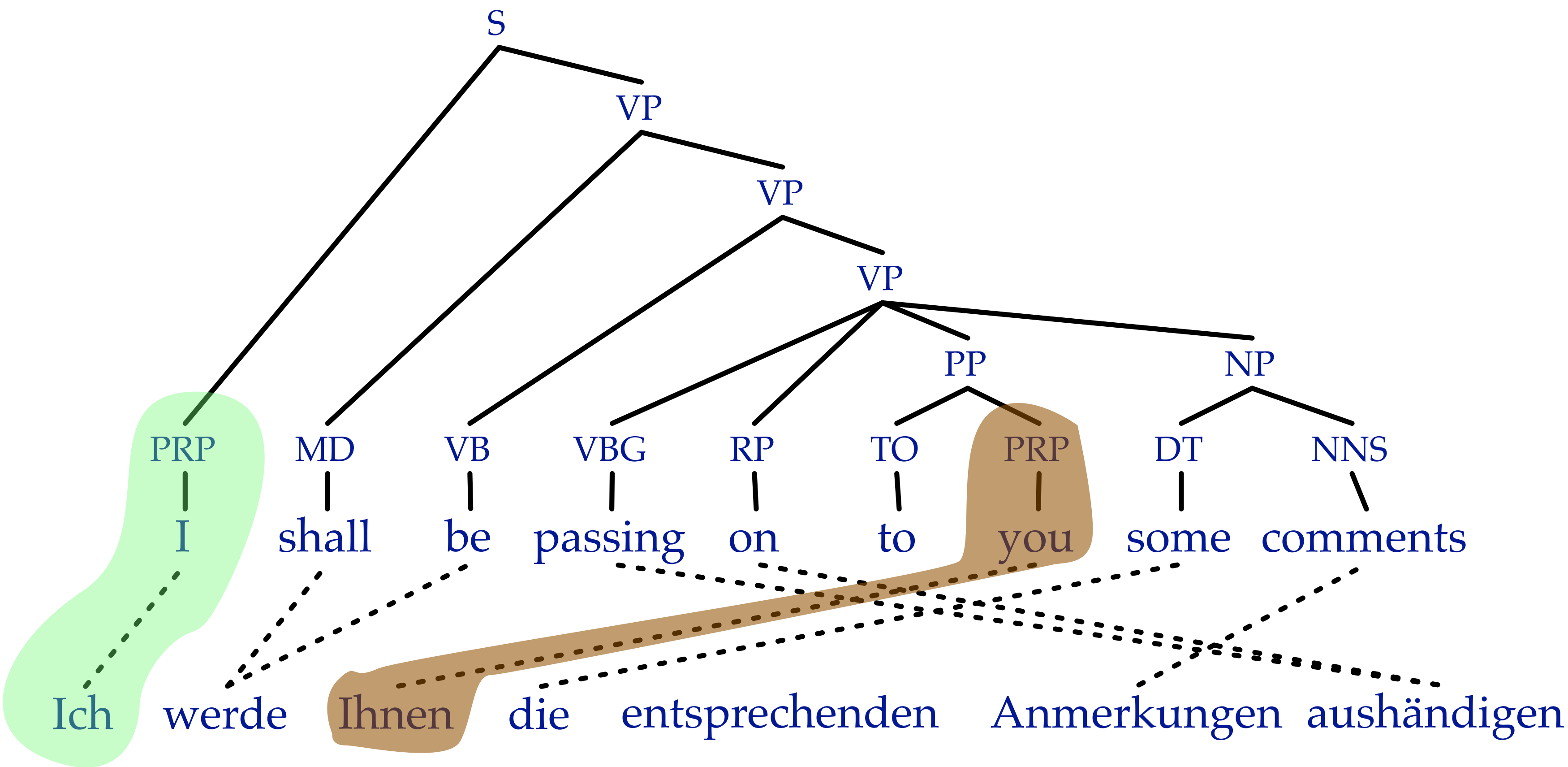
Extract: set of smallest rules required to explain the sentence pair

Lexical Rule



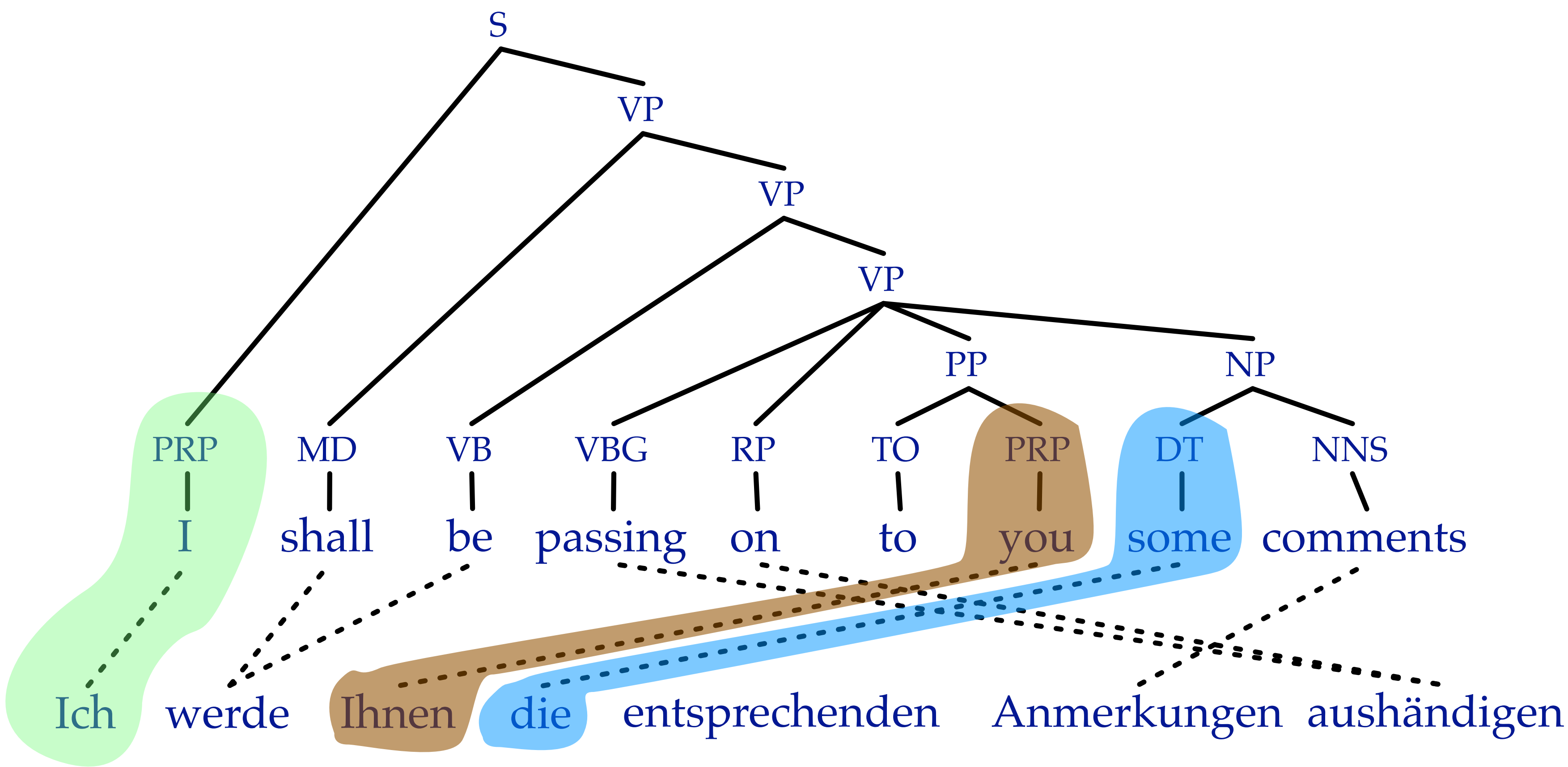
Extracted rule: PRP → Ich | I

Lexical Rule



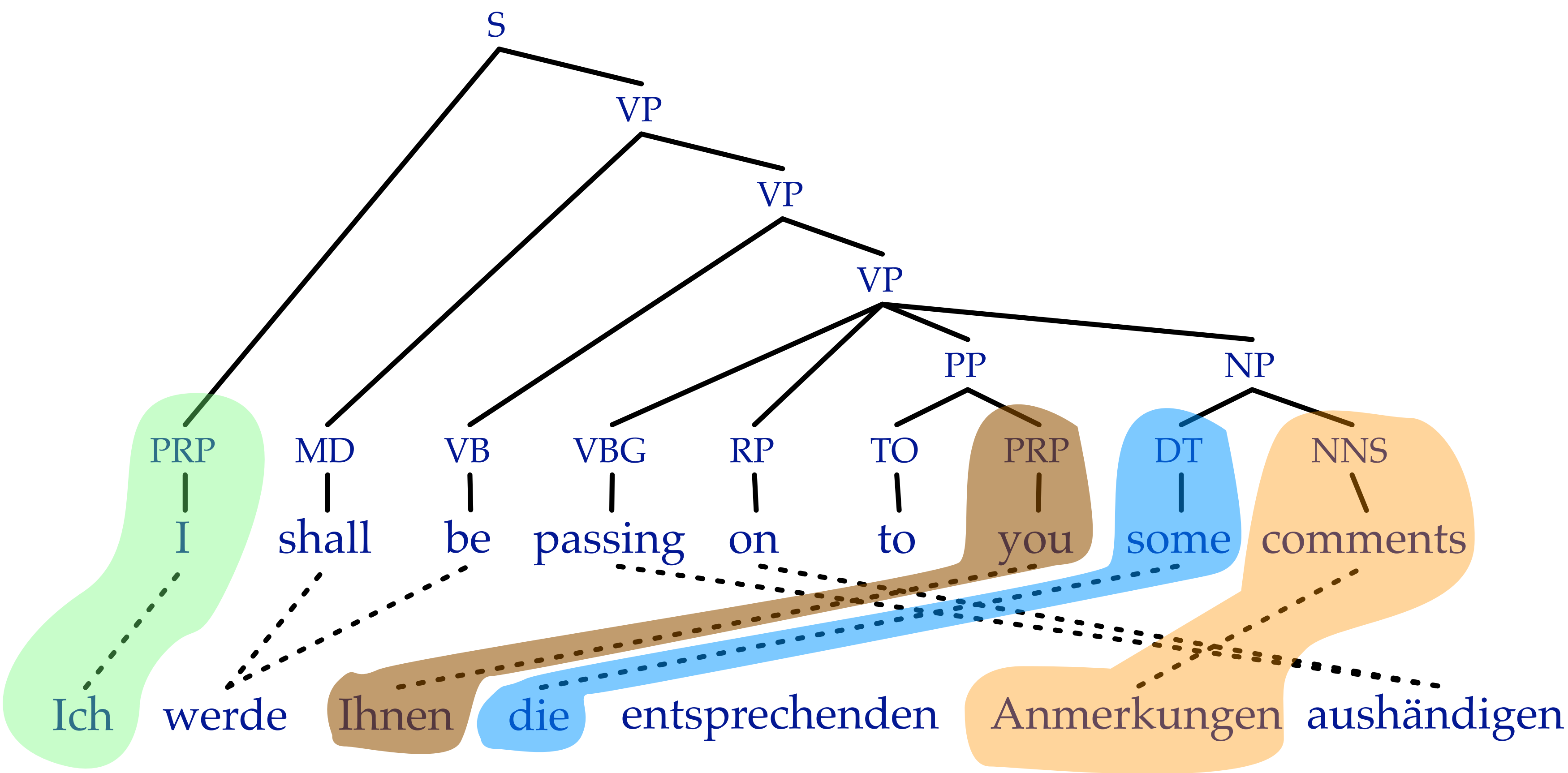
Extracted rule: PRP → Ihnen | you

Lexical Rule



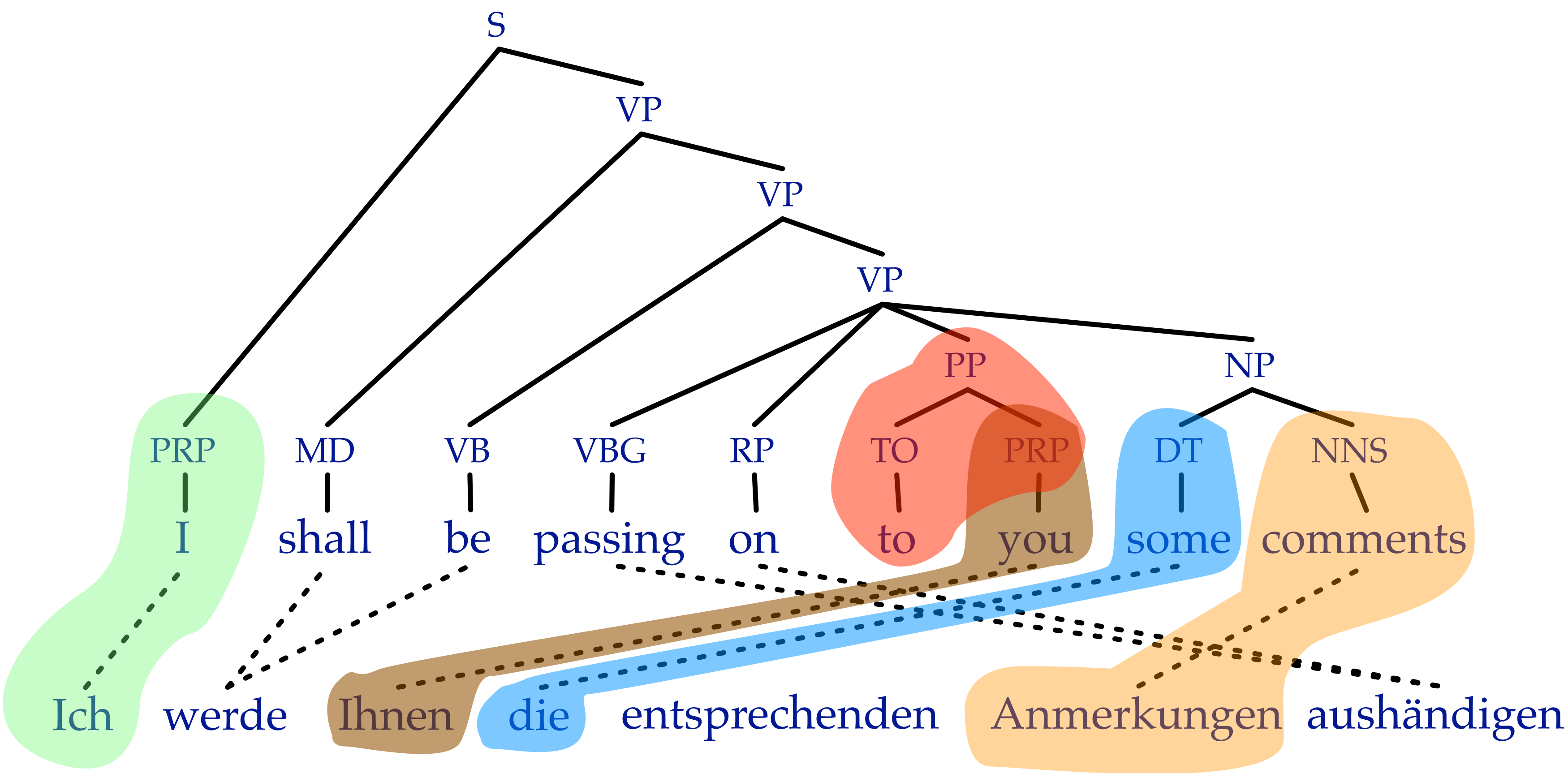
Extracted rule: DT $\rightarrow$ die | some

Lexical Rule



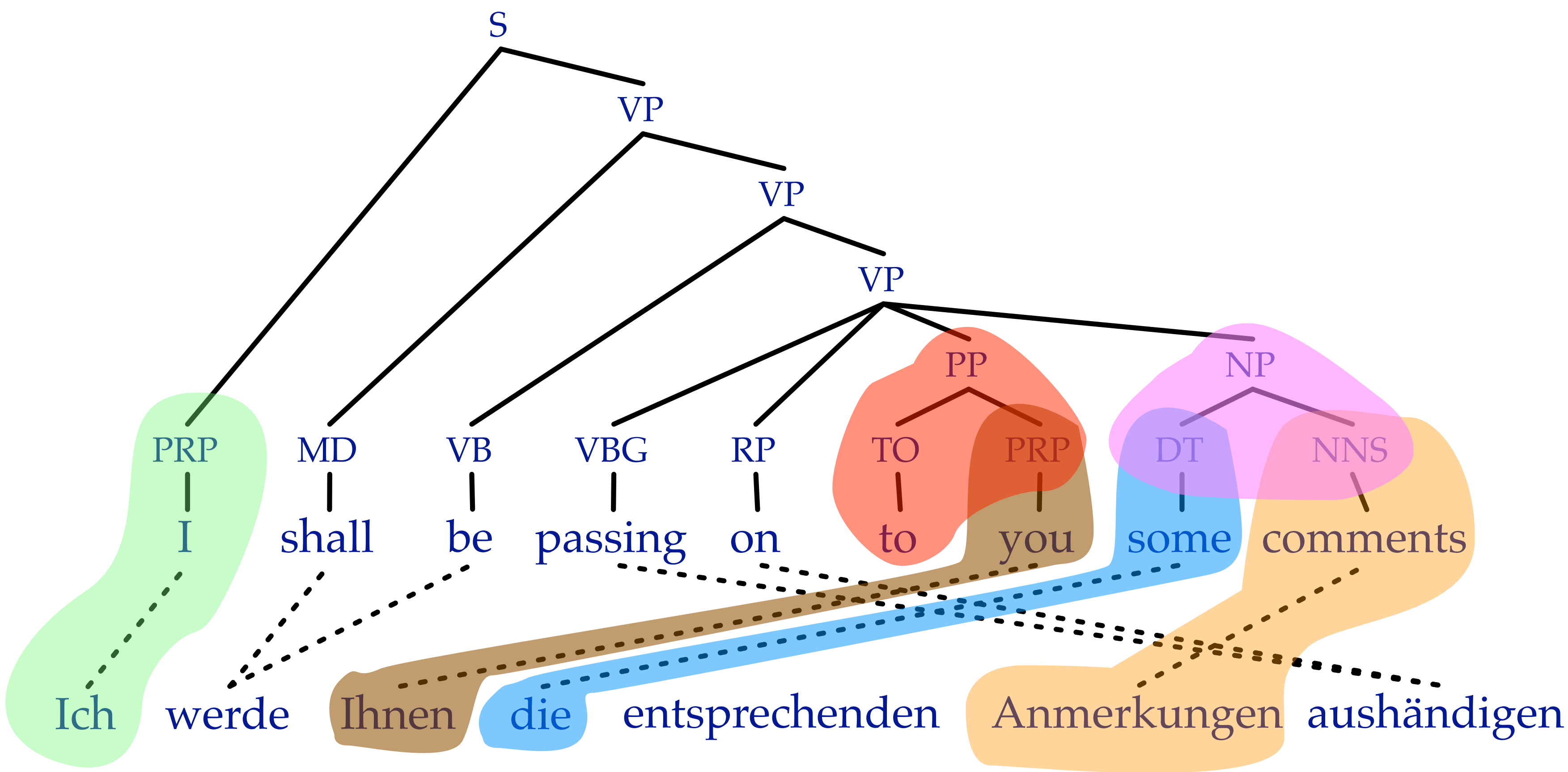
Extracted rule: NNS → Anmerkungen | comments

Insertion Rule



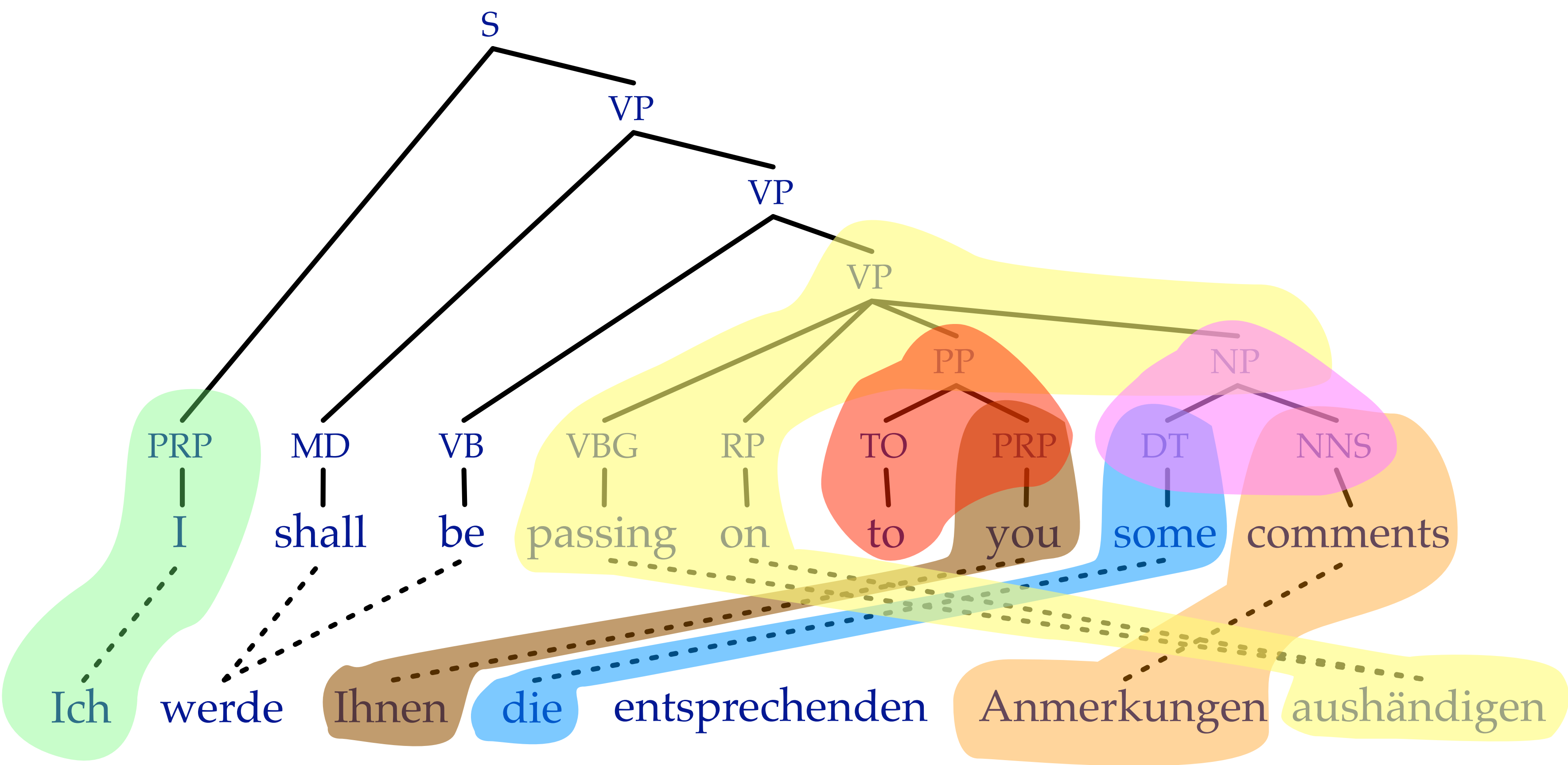
Extracted rule: PP → X | to PRP

Non-Lexical Rule



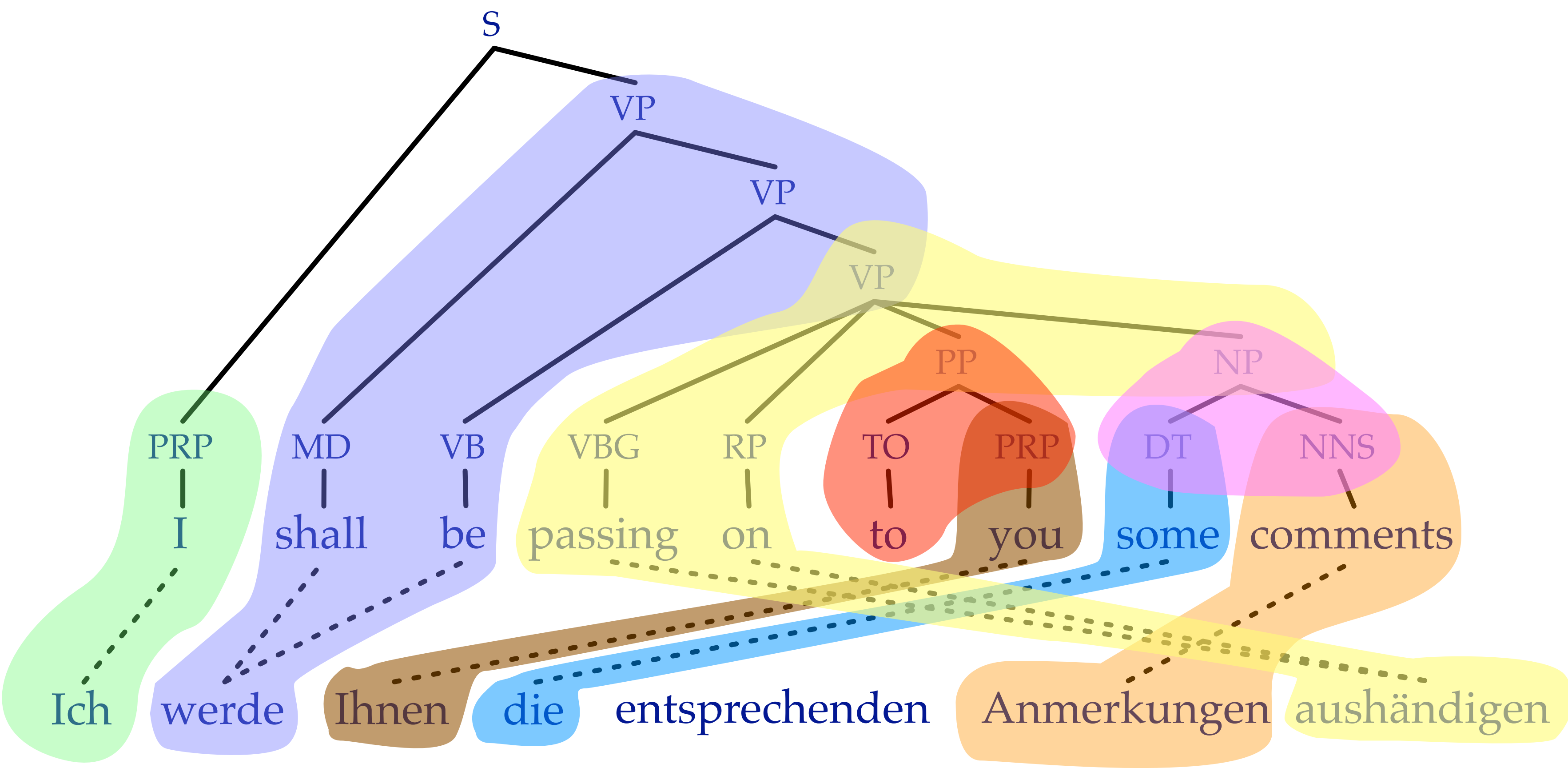
Extracted rule: $\text{NP} \rightarrow \text{X}_1 \ \text{X}_2 \mid \text{DT}_1 \ \text{NNS}_2$

Lexical Rule with Syntactic Context



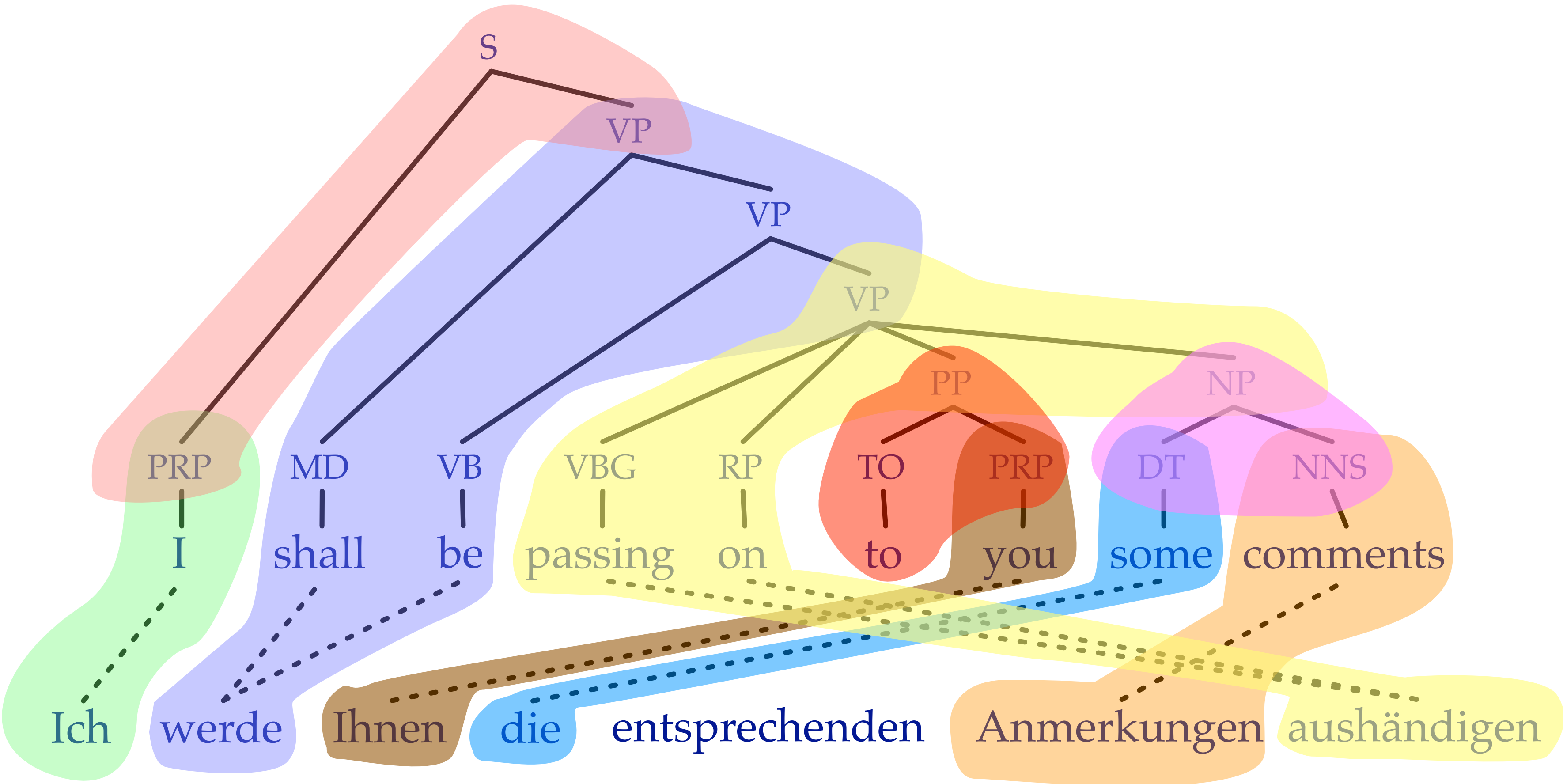
Extracted rule: VP → X_1 X_2 aushändigen | passing on PP_1 NP_2

Lexical Rule with Syntactic Context



Extracted rule: VP → werde X | shall be VP (ignoring internal structure)

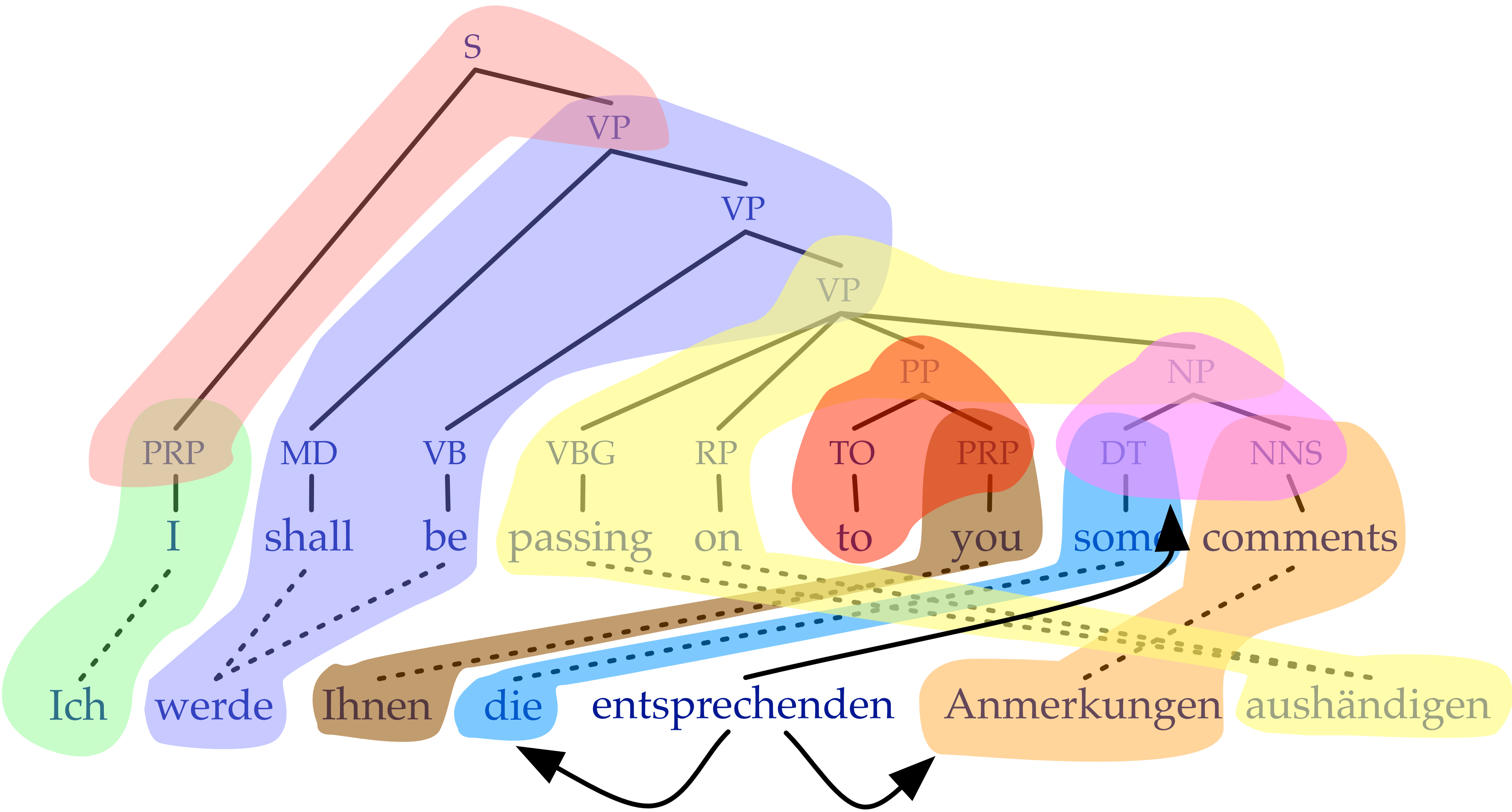
Non-Lexical Rule



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

DONE — note: one rule per alignable constituent

Unaligned Source Words



Attach to neighboring words or higher nodes → additional rules

Too Few Phrasal Rules?

- Lexical rules will be 1-to-1 mappings (unless word alignment requires otherwise)

- But: phrasal rules very beneficial in phrase-based models

- Solutions
  - combine rules that contain a maximum number of symbols
    (as in hierarchical models, recall: "Option 1")
  - compose minimal rules to cover a maximum number of non-leaf nodes

Composed Rules

- Current rules

$X_1\ X_2$ = NP → (DT_1 NNS_1)

die = DT → (some)

entsprechenden Anmerkungen = NNS → (comments)

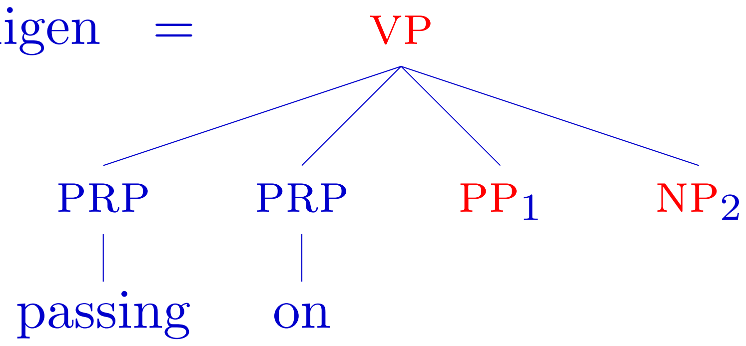
- Composed rule

die entsprechenden Anmerkungen = NP → (DT → (some) NNS → (comments))

(1 non-leaf node: NP)

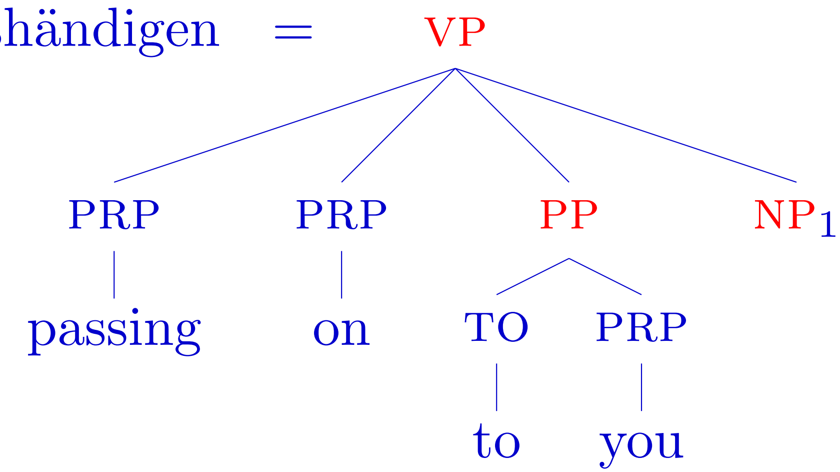
Composed Rules

- Minimal rule: $\text{x}_1\ \text{x}_2$ aushändigen =



3 non-leaf nodes:
VP, PP, NP

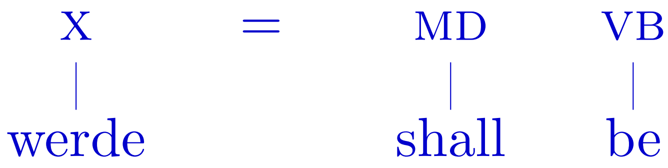
- Composed rule: Ihnen x_1 aushändigen =



3 non-leaf nodes:
VP, PP and NP

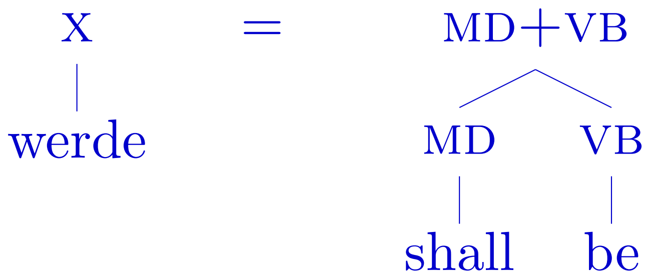
Relaxing Tree Constraints

- Impossible rule



- Create new non-terminal label: MD+VB

⇒ New rule



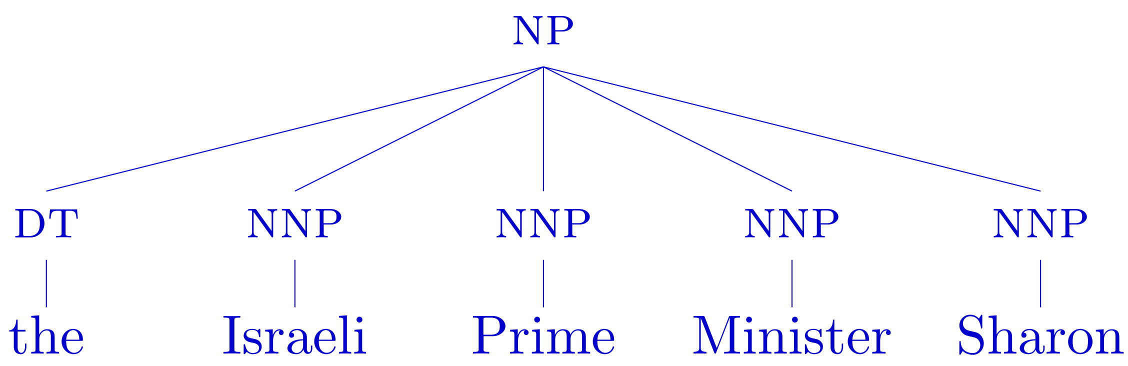
Zollmann Venugopal Relaxation

- If span consists of two constituents , join them: X+Y
- If span conststs of three constituents, join them: X+Y+Z
- If span covers constituents with the same parent X and include
  - every but the first child Y, label as X\Y
  - every but the last child Y, label as X/Y
- For all other cases, label as FAIL

⇒ More rules can be extracted, but number of non-terminals blows up

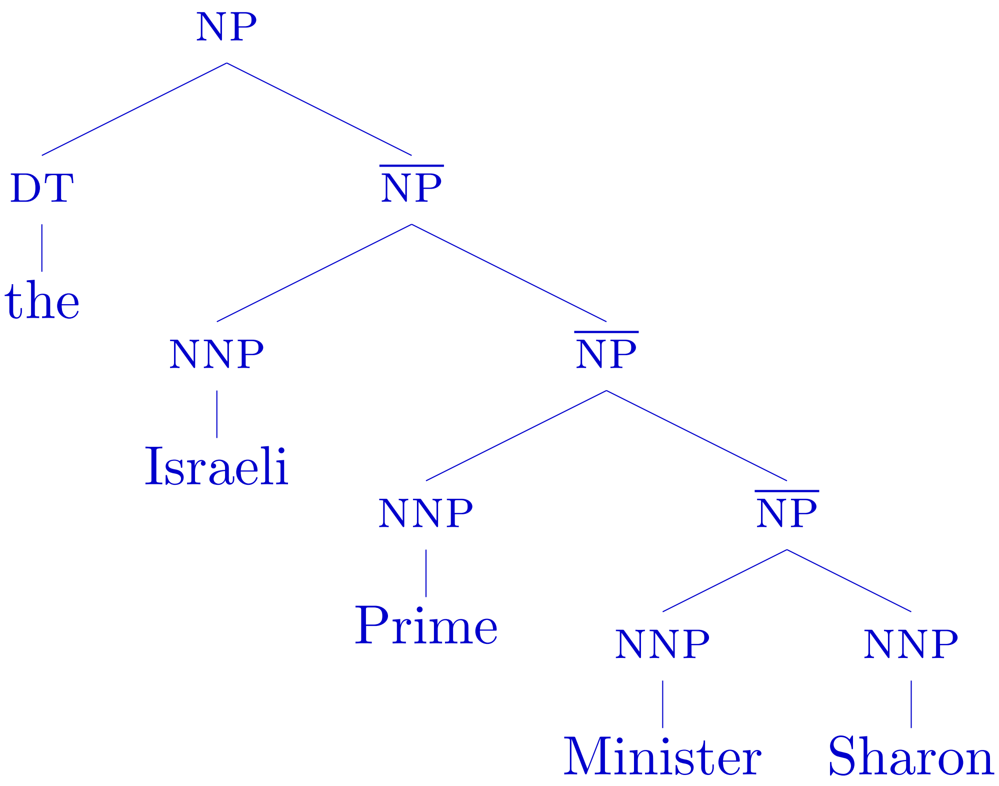
Special Problem: Flat Structures

- Flat structures severely limit rule extraction



- Can only extract rules for individual words or entire phrase

Relaxation by Tree Binarization



More rules can be extracted

Left-binarization or right-binarization?

Scoring Translation Rules

- Extract all rules from corpus
- Score based on counts
  - joint rule probability: $p(\text{LHS}, \text{RHS}_f, \text{RHS}_e)$
  - rule application probability: $p(\text{RHS}_f, \text{RHS}_e | \text{LHS})$
  - direct translation probability: $p(\text{RHS}_e | \text{RHS}_f, \text{LHS})$
  - noisy channel translation probability: $p(\text{RHS}_f | \text{RHS}_e, \text{LHS})$
  - lexical translation probability: $\prod_{e_i \in \text{RHS}_e} p(e_i | \text{RHS}_f, a)$

Statistical Machine Translation, Philipp Koehn