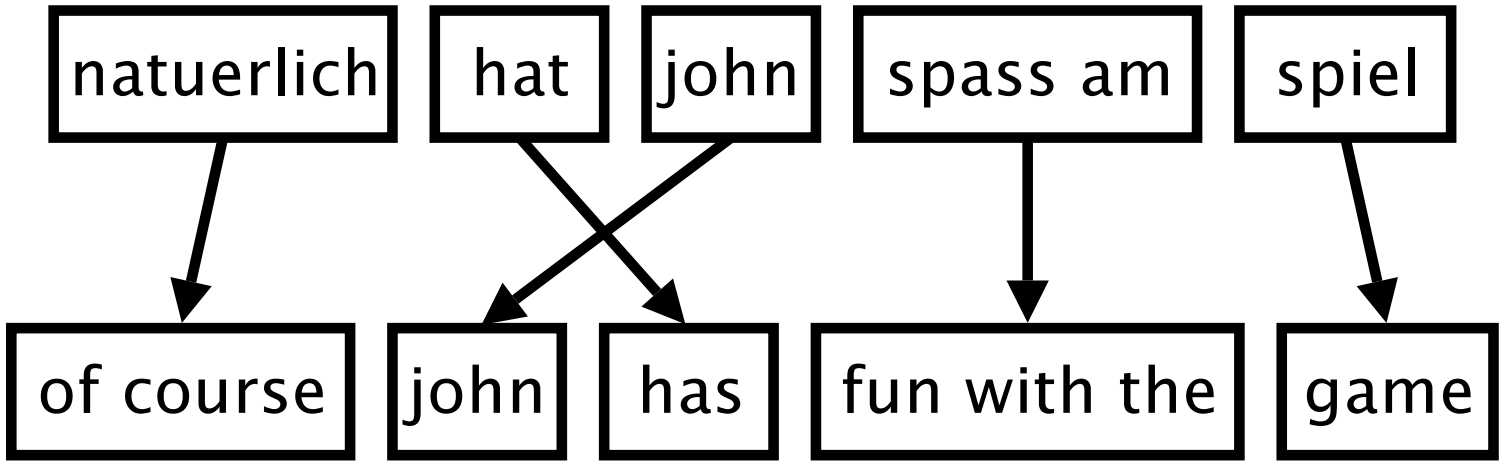
Phrase-based models

Statistical Machine Translation

Motivation

- Word-Based Models translate *words* as atomic units
- Phrase-Based Models translate *phrases* as atomic units
- Advantages:
  - many-to-many translation can handle non-compositional phrases
  - use of local context in translation
  - the more data, the longer phrases can be learned
- "Standard Model", used by Google Translate and others

Phrase-Based Model



- Foreign input is segmented in phrases
- Each phrase is translated into English
- Phrases are reordered

Phrase Translation Table

- Main knowledge source: table with phrase translations and their probabilities
- Example: phrase translations for natuerlich

Translation	**Probability** $\phi(\bar{e}\|\bar{f})$
of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05

Real Example

- Phrase translations for den Vorschlag learned from the Europarl corpus:

English	$\phi(\bar{e}\|\bar{f})$	English	$\phi(\bar{e}\|\bar{f})$
the proposal	0.6227	the suggestions	0.0114
's proposal	0.1068	the proposed	0.0114
a proposal	0.0341	the motion	0.0091
the idea	0.0250	the idea of	0.0091
this proposal	0.0227	the proposal ,	0.0068
proposal	0.0205	its proposal	0.0068
of the proposal	0.0159	it	0.0068
the proposals	0.0159	...	...

  - lexical variation (proposal vs suggestions)
  - morphological variation (proposal vs proposals)
  - included function words (the, a, ...)
  - noise (it)

Linguistic Phrases?

- Model is not limited to linguistic phrases
  (noun phrases, verb phrases, prepositional phrases, ...)

- Example non-linguistic phrase pair

  spass am → fun with the

- Prior noun often helps with translation of preposition

- Experiments show that limitation to linguistic phrases hurts quality

Probabilistic Model

- Find best target sentence

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

- Using Bayes rule

$$p(e|f) = \frac{p(f|e)p(e)}{p(f)}$$

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

  - reverse translation model $p(\mathbf{f}|\mathbf{e})$
  - language model $p(\mathbf{e})$

- **'Noisy Channel'** model

Log-Linear Model

- General framework for computing 'score' of a translation

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

$$log\ p(\mathbf{e}|\mathbf{f}) = \frac{1}{Z} \sum_{i=1}^{n} \lambda_i h_i(e, f)$$

$$= \frac{\cancel{1}}{\cancel{Z}} \sum_{i=1}^{n} \lambda_i h_i(e, f)$$

- Our feature functions
  - $h_1 = \log p(e|f)$ (translation model)
  - $h_2 = \log p(e)$ (language model)
  - $h_3 = \log p(f|e)$ (reverse translation model)
  - .....

Tuning

- Find optimal weights

$$log\ p(\mathbf{e}|\mathbf{f}) = \sum_{\mathbf{i=1}}^{\mathbf{n}} \lambda_\mathbf{i} \mathbf{h_i}(\mathbf{e}, \mathbf{f})$$

- Linear Optimization
  - Minimum Error Rate Training (MERT)

1: **repeat**
2: Decode test set
3: Rerank n-best list to maximum translation quality
4: **until** No change

Decoding

- Transform the source sentence into target sentence, given model

$$p(\mathbf{e}|\mathbf{f}) = \frac{1}{Z} \sum_{i=1}^{n} \lambda_i h_i(e, f)$$

- 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

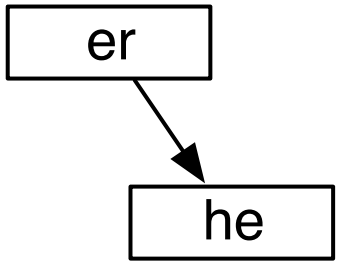
- Task: translate this sentence from German into English

er geht ja nicht nach hause

Translation Process

- Task: translate this sentence from German into English

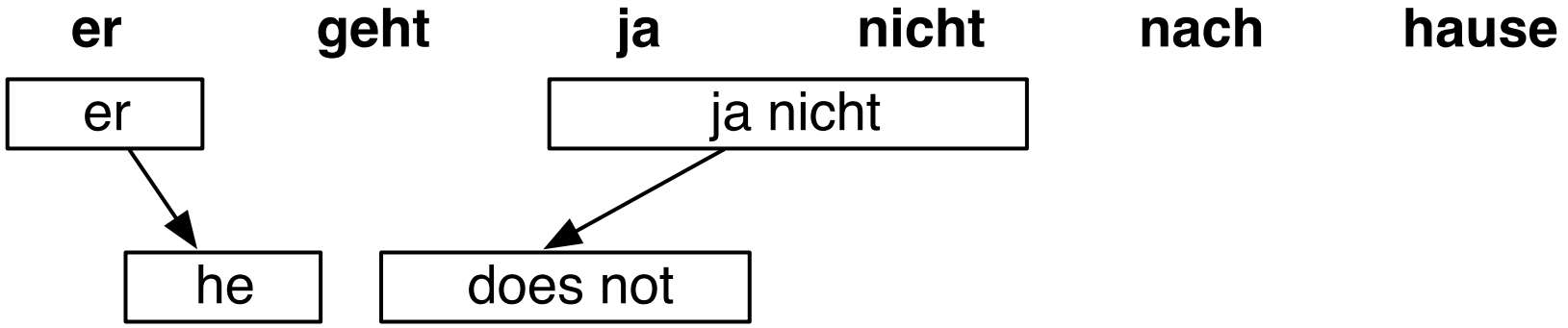
er geht ja nicht nach hause



- Pick phrase in input, translate

Translation Process

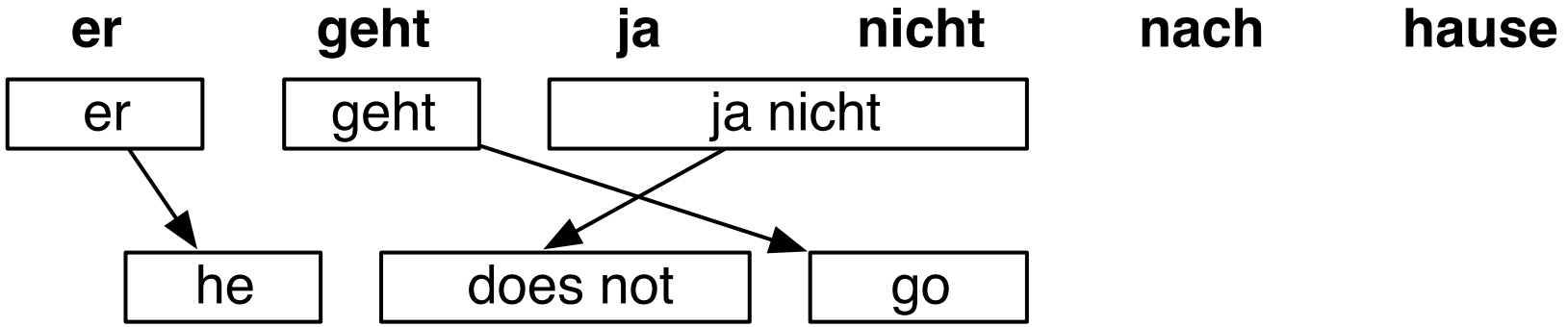
- Task: translate this sentence from German into English



- 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

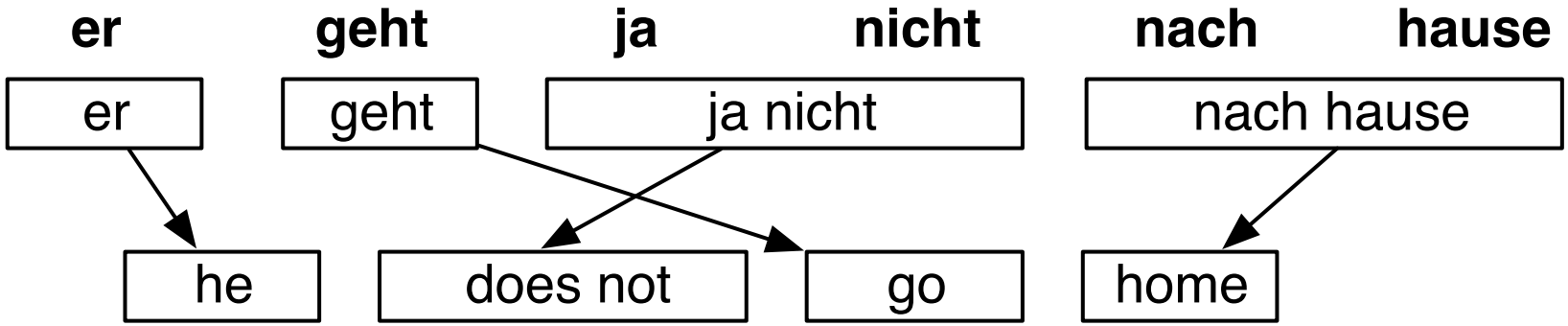
- Task: translate this sentence from German into English



- Pick phrase in input, translate

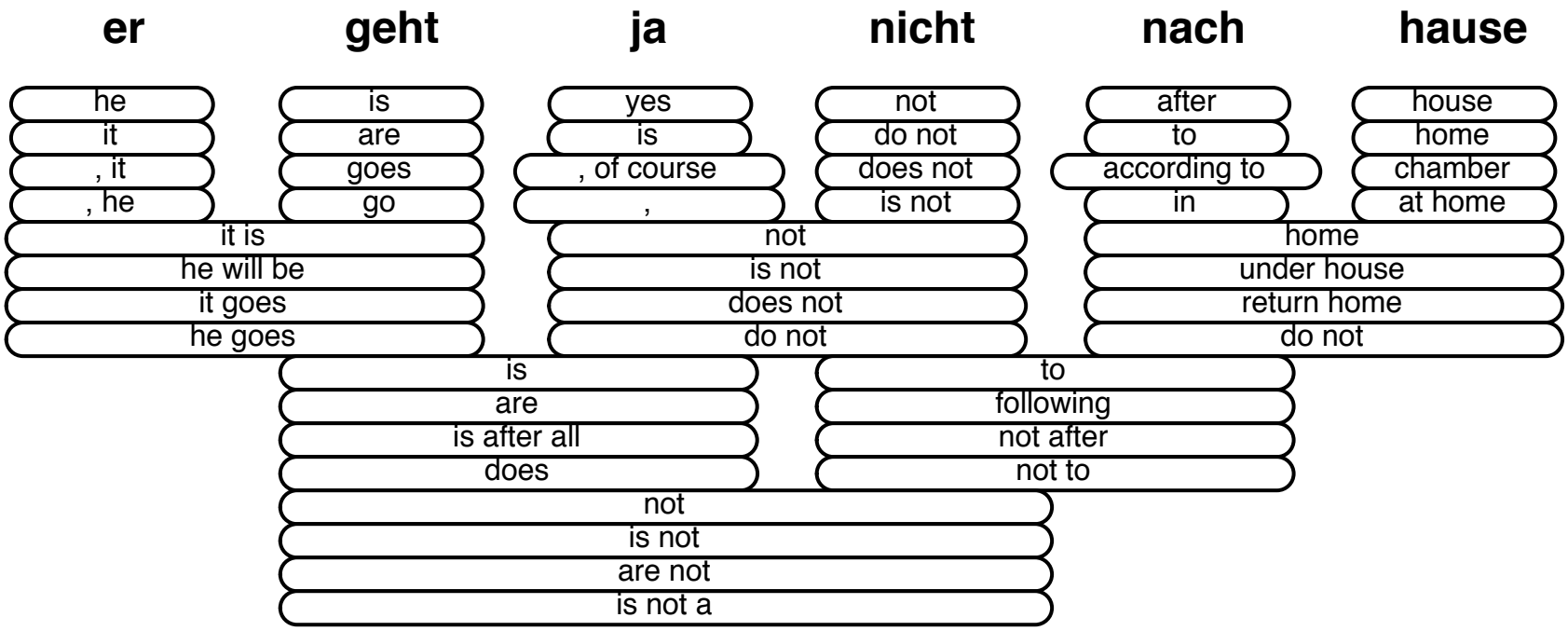
Translation Process

- Task: translate this sentence from German into English



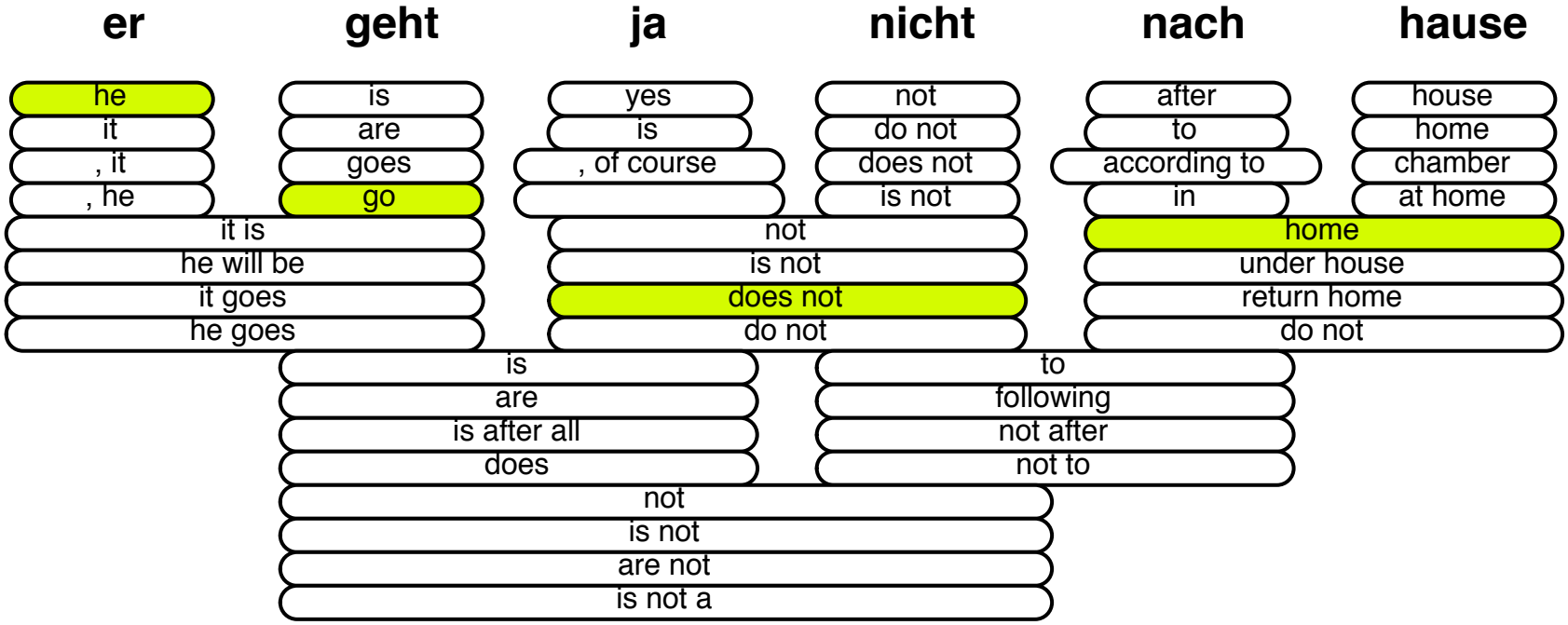
- Pick phrase in input, translate

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

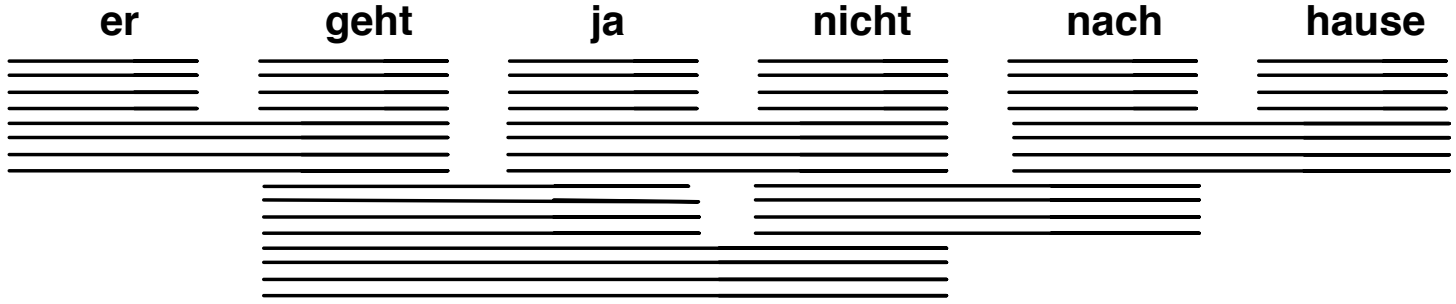
Translation Options



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

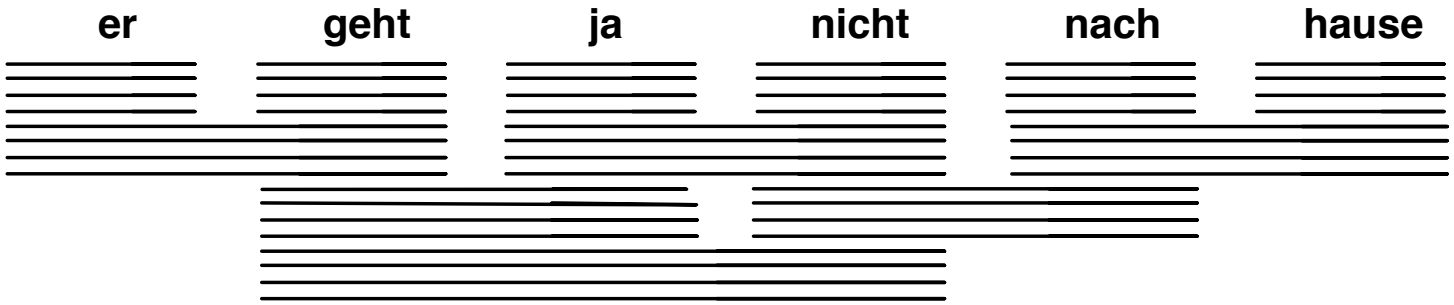
→ Search problem solved by heuristic beam search

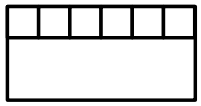
Decoding: Precompute Translation Options



consult phrase translation table for all input phrases

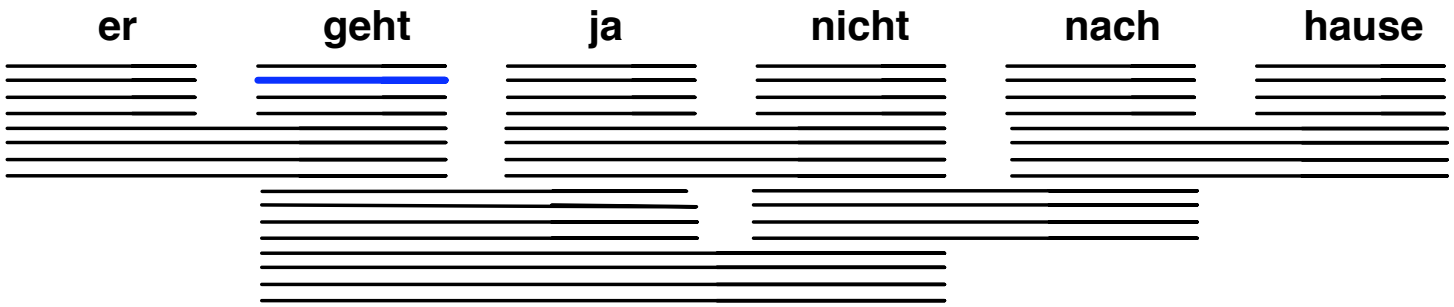
Decoding: Start with Initial Hypothesis

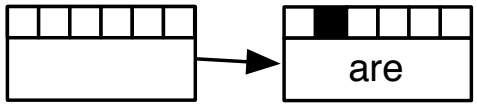




initial hypothesis: no input words covered, no output produced

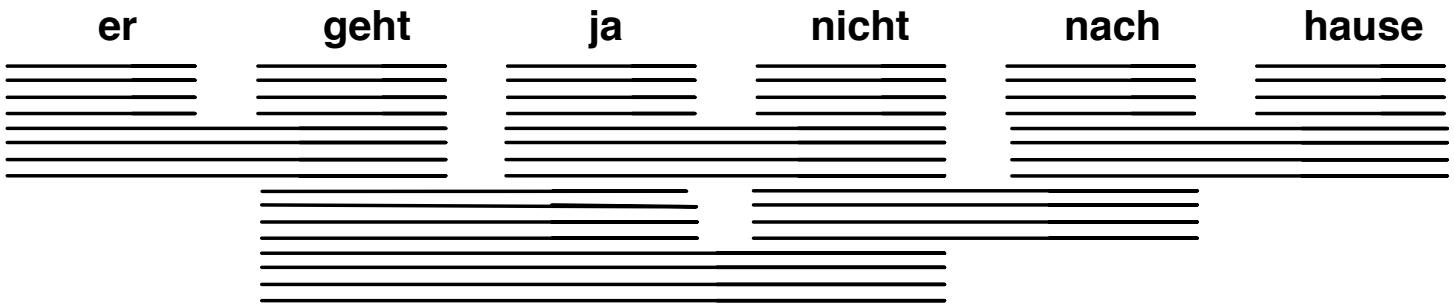
Decoding: Hypothesis Expansion

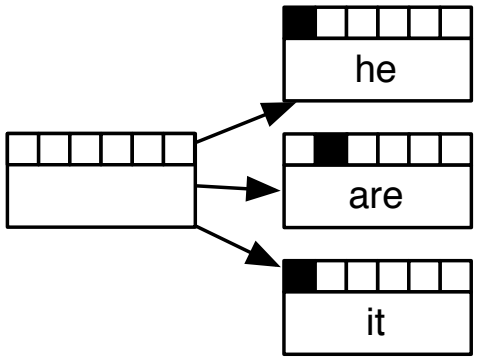




pick any translation option, create new hypothesis

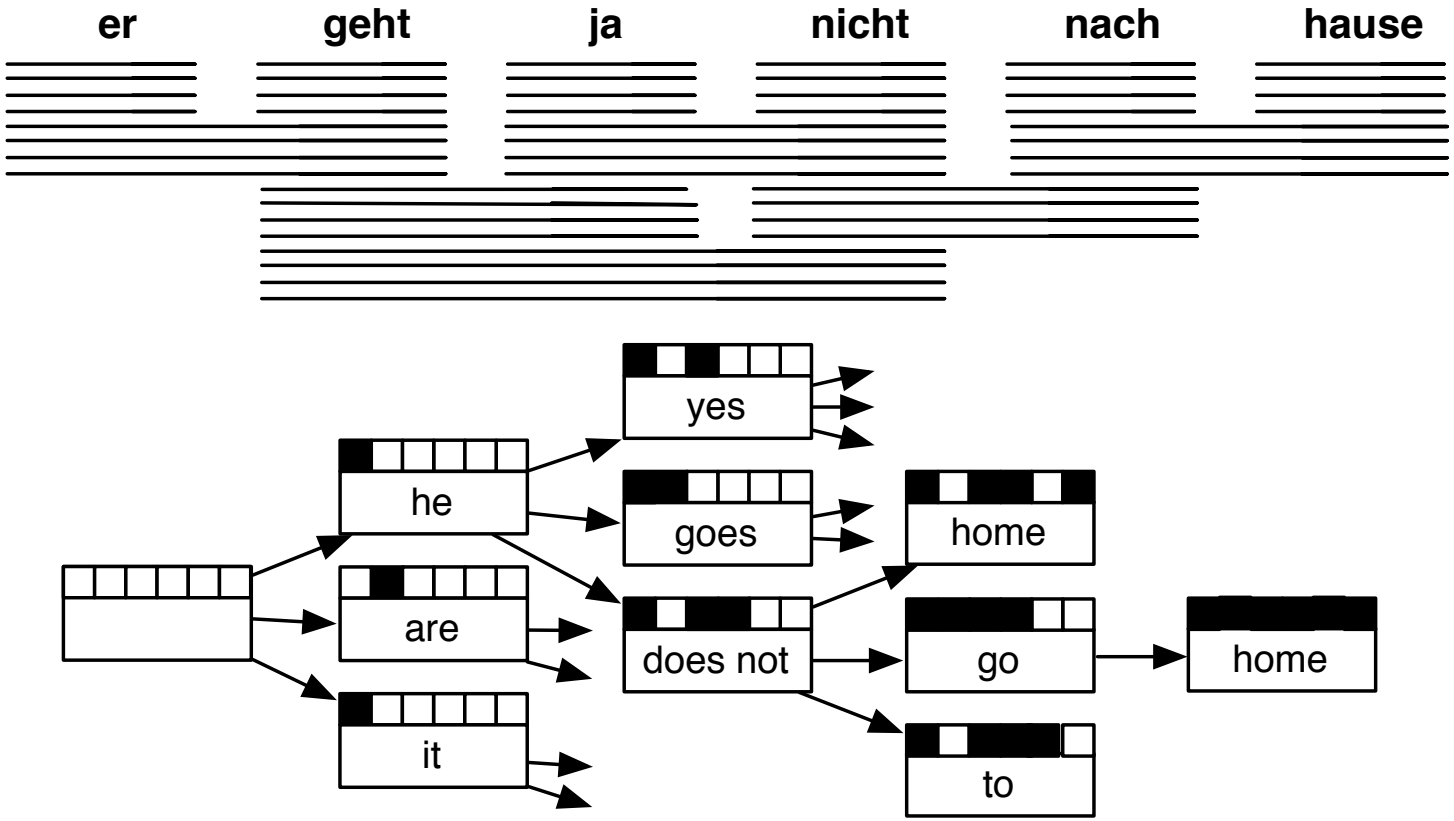
Decoding: Hypothesis Expansion





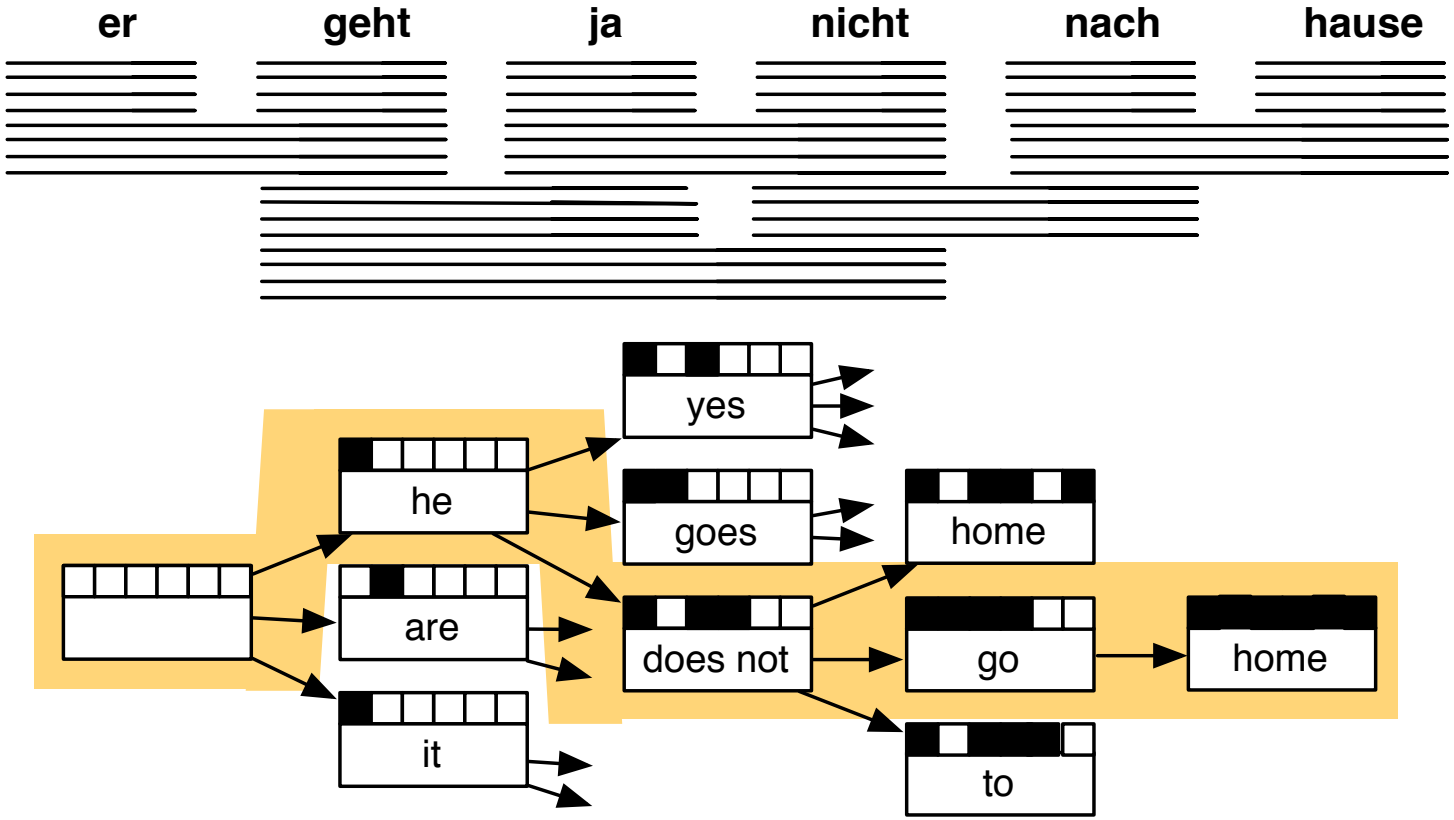
create hypotheses for all other translation options

Decoding: Hypothesis Expansion



also create hypotheses from created partial hypothesis

Decoding: Find Best Path



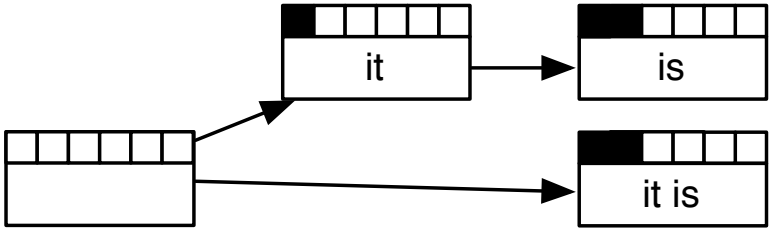
backtrack from highest scoring complete hypothesis

Computational Complexity

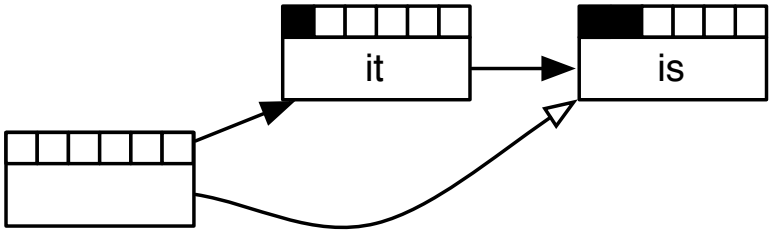
- The suggested process creates exponential number of hypothesis
- Machine translation decoding is NP-complete
- Reduction of search space:
  - recombination (risk-free)
  - pruning (risky)

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



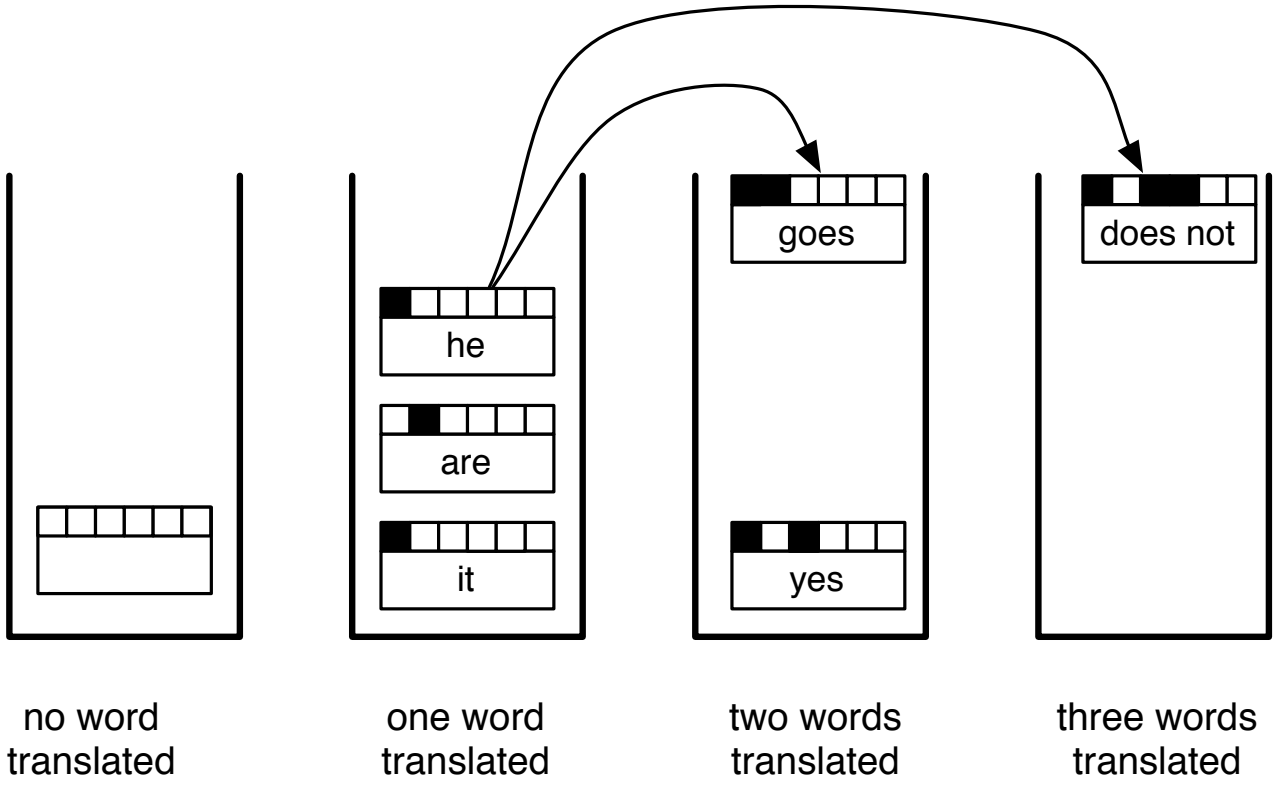
Pruning

- 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

```
1: place empty hypothesis into stack 0
2: for all stacks 0...n − 1 do
3:    for all hypotheses in stack do
4:       for all translation options do
5:          if applicable then
6:             create new hypothesis
7:             place in stack
8:             recombine with existing hypothesis if possible
9:             prune stack if too big
10:         end if
11:      end for
12:   end for
13: 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

Summary

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