# 1 Decoding

## 1.1 Task

The task of decoding in machine translation is to find the best scoring translation $e_{\text{best}} = \arg\max_e p(e|f)$.

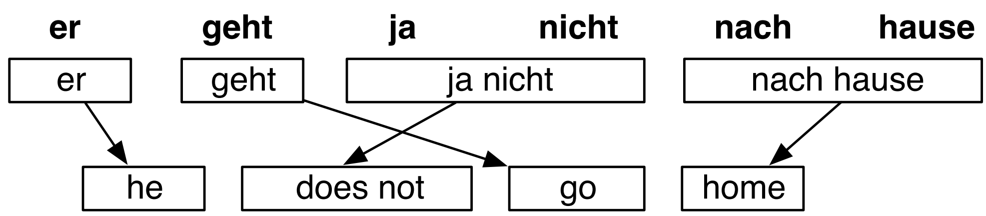
## 1.2 Evaluation

To find out what is a good translation for a given input, we have to look for two types of error which could prevent this.

**Search error** is the failure to find the best translation according to the model. $\Rightarrow$ $e_{\text{best}}$ cannot be found, despite a good translation model.

**Translation error** tells us that our translation model is bad. However, this type of error should not concern us in the Decoding chapter.

## 1.3 Decoding process




1. Pick a foreign input phrase, possibly out of sequence $\Rightarrow$ accounts for reordering!
2. Translate phrase $\Rightarrow$ uses phrase table!
3. Build English phrase in sequence $\Rightarrow$ evaluate using language model!

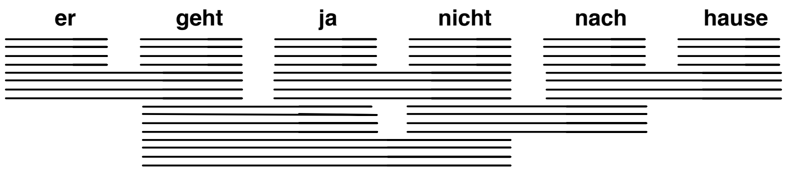
## 1.4 Incremental computation of $p(e|f)$ for each partial hypothesis

$$e_{\text{best}} = \arg\max_e \prod_{i=1}^{I} (\phi(\bar{f}_i|\bar{e}_i) d(\text{start}_i - \text{end}_{i-1} - 1)) \prod_{j=1}^{|e|} p_{\text{LM}}(e_j|e_1, \ldots, e_{j-1}) \tag{1}$$

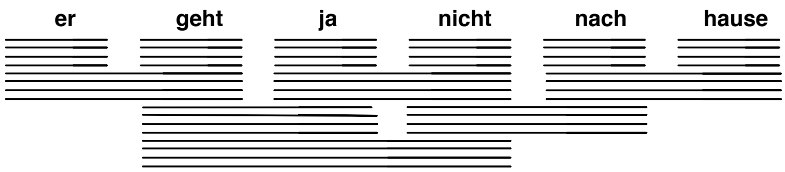
**Phrase translation:** Look up score $\phi(\bar{f}_i|\bar{e}_i)$ from phrase translation table.
**Reordering:** Compute $d(\text{start}_i - \text{end}_{i-1} - 1)$: previous phrase ends in $\text{end}_{i-1}$, current phrase starts at $\text{start}_i$.
**Language Model:** The n-gram language model needs to keep track of last $n-1$ words to compute $p_{\text{LM}}(w_i|w_{i-(n-1)}, \ldots, w_{i-1})$ for an added English word $w_i$.

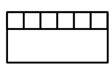
## 1.5 Beam Search

1. Consult phrase translation table for all possible input phrases, precompute **translation options** as all applicable phrase translations:

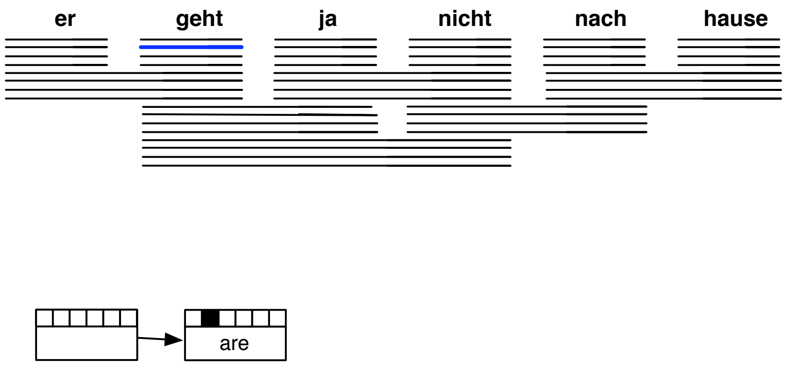



2. **Initial hypothesis:** No input phrase covered, no output produced:

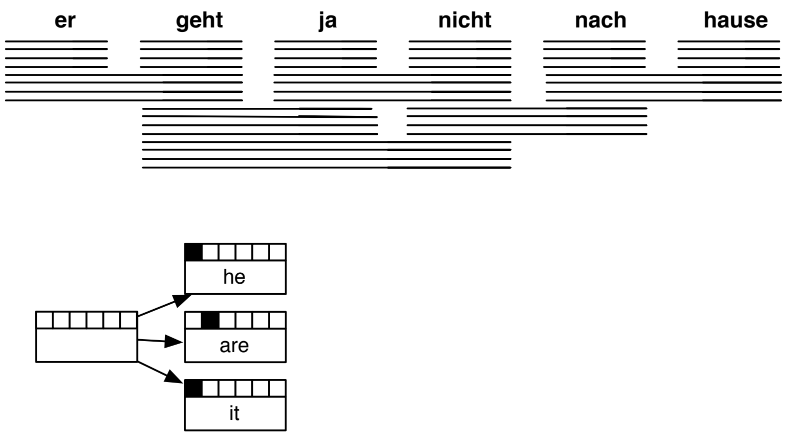


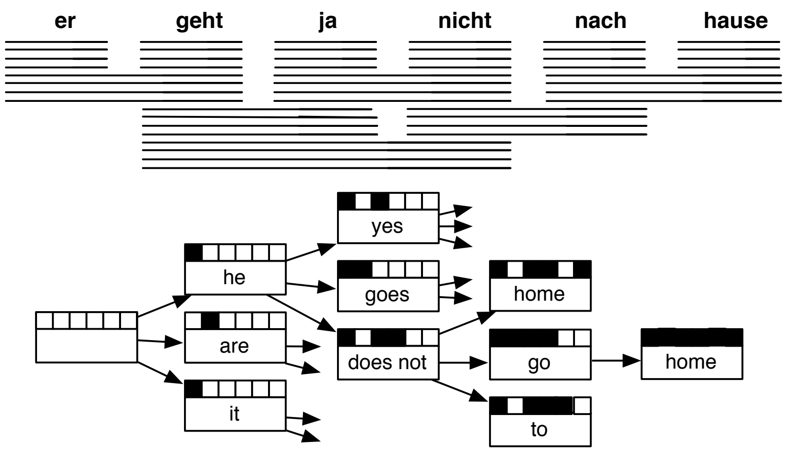
3. **Hypothesis expansion:** Pick translation option, create new hypothesis by constructing partial translation, mark off input:



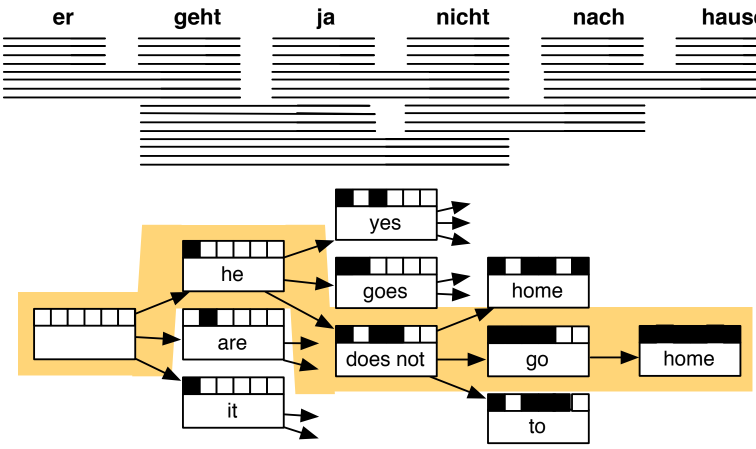

4. Create hypotheses for all other translation options:




5. Create hypotheses from already created partial hypotheses:



6. Find best path by backtracking from highest scoring complete hypothesis:




## 1.6 Computational Complexity of Decoding

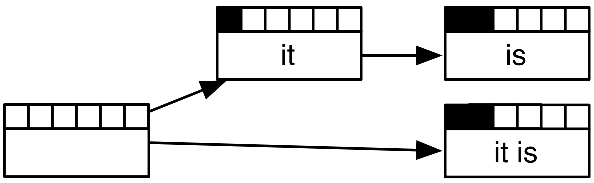
$$O(|\text{translation options}|^{\text{sentence length}}) \tag{2}$$

## 1.7 Recombination

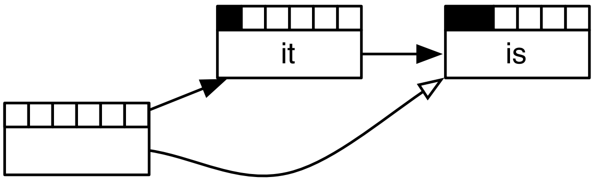
The idea of recombination is to have a risk-free reduction of our search space. If two hypotheses are indistinguishable in a subsequent search, we drop the hypothesis with the lower score.
There are **two cases** of "indistinguishable":

**case 1:**

- the same number of foreign words translated,
- the same English words in output:

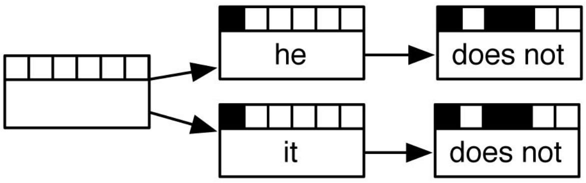



⇒ Drop the hypothesis with the worse score:

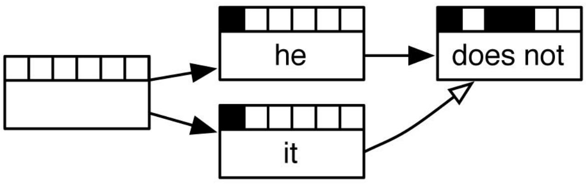



- the same number of foreign words translated,

**case 2:**
- the same last two words in output (assuming trigram lm),
- the same last foreign word translated:




⇒ Drop the hypothesis with the worse score:




### 1.7.1 Restrictions on Recombination

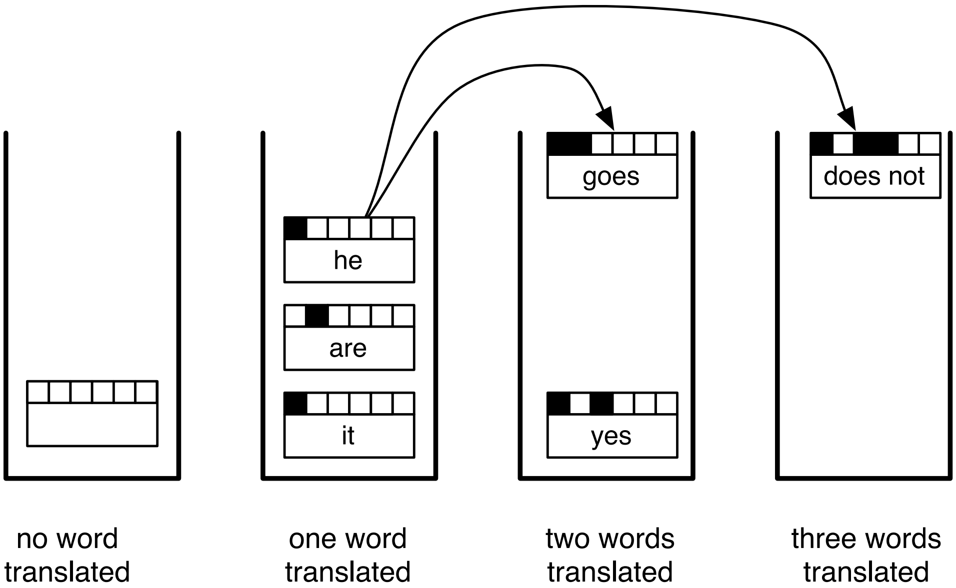
**Translation model:** Phrase translations are independent of each other.
⇒ no restrictions to recombination
**Language model:** The last $n-1$ words are used as history in $n$-gram the language model.
⇒ recombined hypotheses must match in their last $n-1$ words
**Reordering model:** The distance-based reordering model is based on the distance to the end position of the previous input phrase.
⇒ recombined hypotheses must have the same end position of the corresponding input phrase

**Problem:** Recombination reduces the search space, but our worst case still has exponential complexity.

**Solution:** **Pruning -** Remove bad hypotheses early and focus on efficiency.

## 1.8 Pruning by stacks: Stack Decoding

**Stacks:**
We put comparable hypotheses in a stack, for example hypotheses that have translated the same number of input words.




**Pseudocode:**

```
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
```

We want to limit the number of hypotheses in a stack by **pruning strategies**, i.e., we want to focus the **beam of light** that shines through the search space.

**k-best (histogram) pruning:** Sort the hypotheses with respect to their score and keep the $k$-best hypotheses.

**$\alpha$-best (threshold) pruning:** Sort the hypotheses with respect to their score and keep the hypotheses with scores of at least $\alpha$-fraction of the best score.
$\text{score}_{\text{hyp}} \geq \alpha \cdot \text{best score}$

## 1.9 Computational complexity:

$$O(\text{max. stack size} \times \text{number of translation options} \times \text{sentence length}) \quad (3)$$

Since the number of translation option is linear with the sentence length:

$$O(\text{max. stack size} \times \text{sentence length}^2) \quad (4)$$

⇒ **quadratic** complexity.

## 1.10 Further improvements in computational complexity: Reordering limits

The idea of reordering limits is to have more efficient decoding by limiting reordering to a maximal distance, which is typically 5-8 words. With this, the complexity is reduced to **linear**:

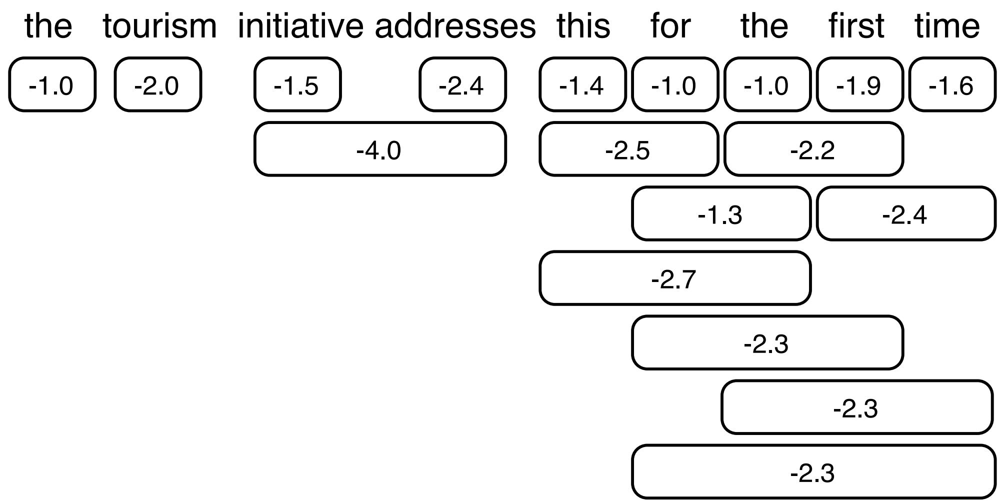
$$O(\text{max. stack size} \times \text{sentence length}) \quad (5)$$

## 1.11 Estimating Future Cost

Question: What's the cost of the rest of the sentence for a given hypothesis?
**Translation model:** The cost is known for all phrase pairs.

**Language model:** The output words are known, but their context is not, so we have an estimate without context.

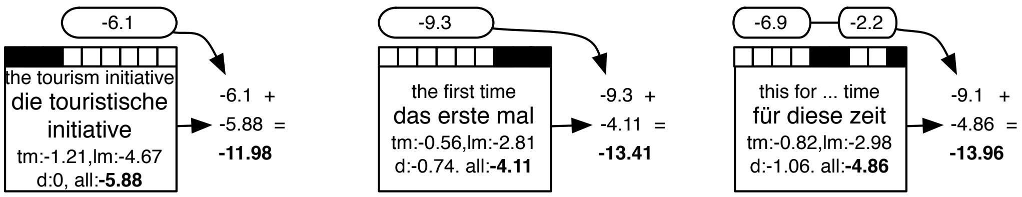
**Reordering model:** The cost is unknown and therefore ignored for future cost estimation.




Cost of the cheapest translation options for each input span (log-probabilities).

| **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 | | | | | | | | |

We compute the cost estimate for all contiguous spans by combining the cheapest options.




The hypothesis score and future cost estimate are combined for pruning.