Decoding

IT Systems Engineering | Universität Potsdam

Dr. Mariana Neves
(adapted from the original slides
of Prof. Philipp Koehn)

December 14th, 2015

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

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

- NP-complete problem

Heuristic search

- There is no guarantee that we will find the best translation.

Decoding

- Two types of errors in translations:
  - the most probable translation is bad → fix the model (model error)
  - search does not find the most probably translation → fix the search (search error)

- 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**

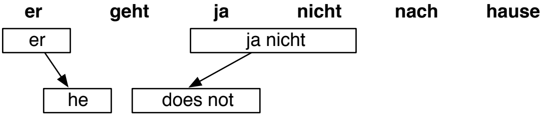
Translation Process

- Task: translate this sentence from German into English

- Pick phrase in input, translate

Translation Process

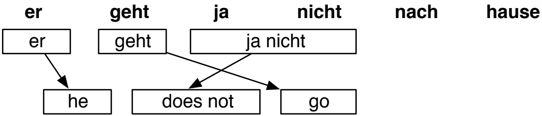
- Task: translate this sentence from German into English



- 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

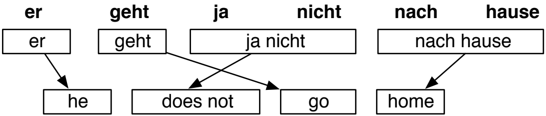
- Task: translate this sentence from German into English



- 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

Computing Translation Probability

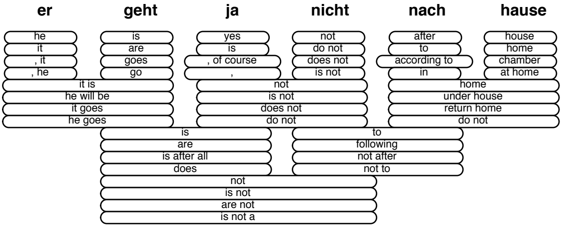
- Components:

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 $d(start_i - end_{i-1} - 1)$

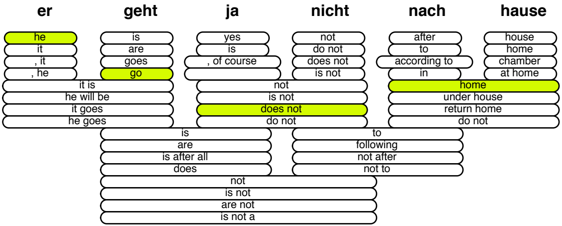
Language model For n-gram model, need to keep track of last $n-1$ words
$\rightarrow$ compute score $p_{\mathrm{LM}}(w_i|w_{i-(n-1)}, \ldots, w_{i-1})$ for added words w_i

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

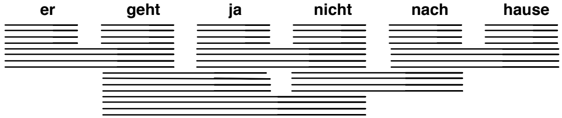
Translation Options



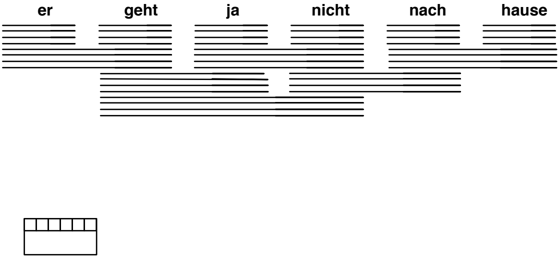
- The machine translation decoder does not know the right answer
  - picking the right translation options
  - arranging them in the right order

→ Search problem solved by heuristic beam search

Decoding: Precompute Translation Options

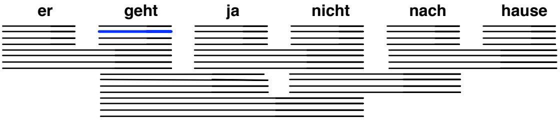


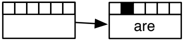
Decoding: Start with Initial Hypothesis



initial (empty) hypothesis: no input words covered, no output produced

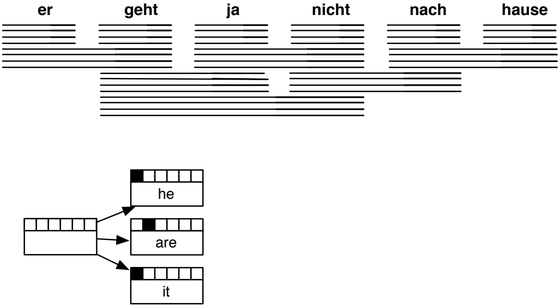
Decoding: Hypothesis Expansion





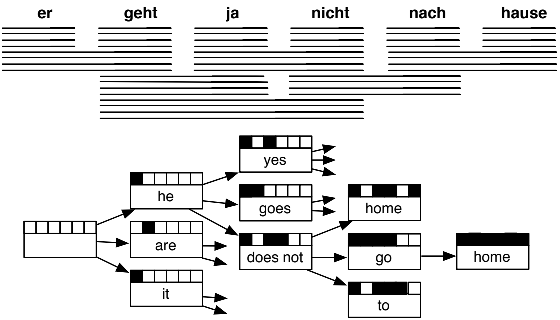
pick any translation option, create new hypothesis

Decoding: Hypothesis Expansion



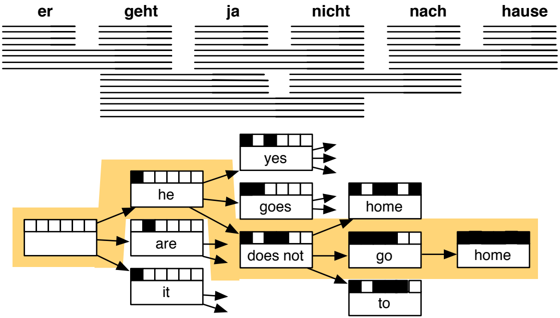
create hypotheses for all other translation options

Decoding: Hypothesis Expansion



also create hypotheses from created partial hypothesis

Decoding: Find Best Path



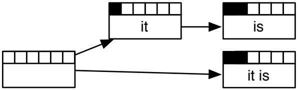
backtrack from highest scoring complete hypothesis

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)

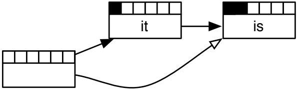
Hypothesis Recombination

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



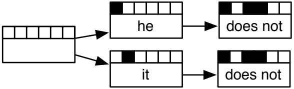
Hypothesis Recombination

- We drop the worse hypothesis as it can never be part of the best overall path.



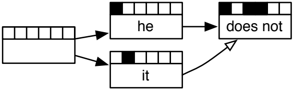
Hypothesis Recombination

- 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



Hypothesis Recombination

- Again, we drop the worse hypothesis as it can never be part of the best overall path.



Hypothesis Recombination

- We do not erase the worse hypothesis.
- We keep them for extracting the second, third, etc best hypotheses.

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 the 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)

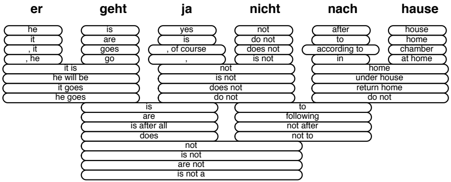
- Pruning:
  - remove bad hypotheses early
  - preferably before the completion of the path

Stack decoding

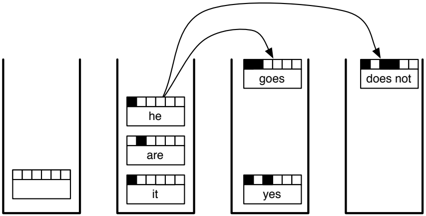
- put comparable hypothesis into stacks
  (hypotheses that have translated same number of input words)

- limit number of hypotheses in each stack
  (if it gets too large, we prune the worst hypothesis in the stack)

Stacks

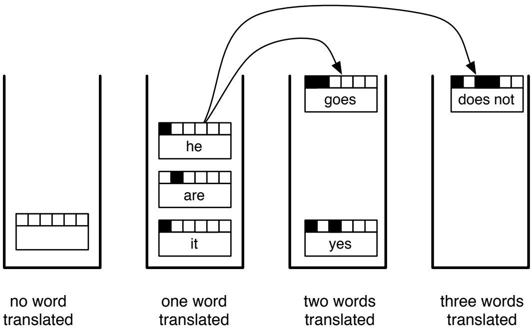


er
geht
ja
nicht
nach
hause
he
it
, it
, he
is
are
goes
go
yes
is
, of course
,
not
do not
does not
is not
after
to
according to
in
house
home
chamber
at home
it is
he will be
it goes
he goes
not
is not
does not
do not
home
under house
return home
do not
is
are
is after all
does
to
following
not after
not to
not
is not
are not
is not a



he
are
it
goes
yes
does not
no word translated
one word translated
two words translated
three words translated

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.
    - The size k has direct relation to decoding speed.
    - Quadratic cost instead of exponential cost, but finding the best translation is not guaranteed.

Pruning

- Pruning strategies
  - Threshold pruning:
    - Keep hypothesis with score $\alpha \times$ best score ($\alpha < 1$).
    - Cost is less predictable, but also roughly quadratic.

- Approaches can combine both pruning strategies.

Pruning

- 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

- The input can be processed in any order.
- Reordering can be a simple (French/English) or a hard (German/English) task.

- Limiting reordering: a maximum of d words can be skipped.

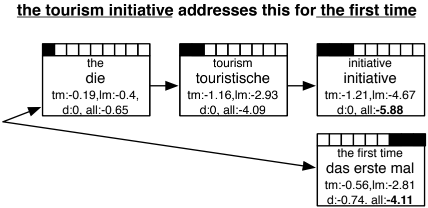
Reordering Limits

- Limiting reordering to maximum reordering distance
- Typical reordering distance 5–8 words
  - depending on language pair
  - larger reordering limit hurts translation quality
- It reduces complexity to linear, the complexity does not grows with sentence length.

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

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

Translating the Easy Part First?



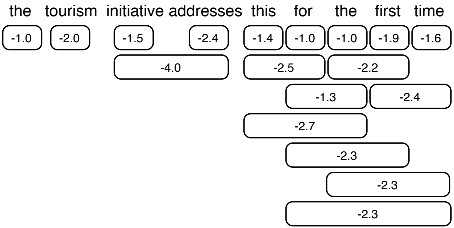
both hypotheses translate 3 words

worse hypothesis has better score

Estimating Future Cost

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

Cost Estimates from Translation Options



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

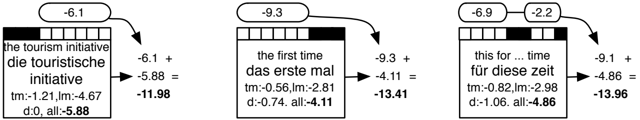
Cost Estimates for all Spans

- 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 $\rightarrow$ total cost -11.98
  - middle hypothesis starts with easiest part: the first time
    score: -4.11, future cost: -9.3 $\rightarrow$ total cost -13.41
  - right hypothesis picks easy parts: this for ... time
    score: -4.86, future cost: -9.1 $\rightarrow$ total cost -13.96

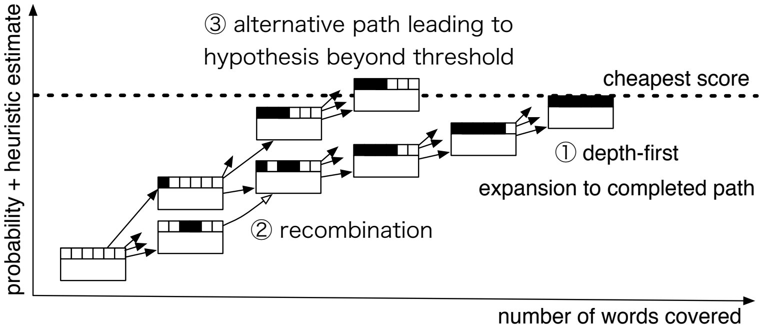
Other Decoding Algorithms

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

A* Search

- Pruning the search space that is risk free
- It needs an *admissible* estimated cost heuristic that never overestimates cost
- Usually, it ignores reordering costs and relies on phrase translation costs.

A* Search



- Translation agenda: create hypothesis with lowest score + heuristic cost
- Done, when complete hypothesis created

A* Search - disadvantages

- No guarantee that it finishes in polynomial time.
- We need an admissible cost: a future cost that is never an underestimate, an estimated cost that is never an overestimate.

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

Greedy Hill-Climbing

- Advantages:
  - We always have a full translation to output.
  - We can stop any time if we are happy with the output or have time constraints.

- Disadvantages:
  - It covers small search spaces.
  - We can get stuck in local optima.

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

Suggested reading

- Statistical Machine Translation, Philipp Koehn (chapter 6).