

SMT models: Word-Based Models. Phrase-Based Models. Decoding.

1. Word-Based Models

In word-based translation, the fundamental unit of translation is a word in some natural language. Typically, the number of words in translated sentences are different, because of compound words, morphology, and idioms. The ratio of the lengths of sequences of translated words is called fertility, which tells how many foreign words each native word produces. Necessarily it is assumed by information theory that each covers the same concept. In practice this is not really true. For example, the English word corner can be translated in Spanish by either rincón or esquina, depending on whether it is to mean its internal or external angle.

Simple word-based translation cannot translate between languages with different fertility. Word-based translation systems can relatively simply be made to cope with high fertility, such that they could map a single word to multiple words, but not the other way about. For example, if we were translating from English to French, each word in English could produce any number of French words—sometimes none at all. But there's no way to group two English words producing a single French word.

An example of a word-based translation system is the freely available GIZA++ package (GPLed), which includes the training program for IBM models and HMM model and Model 6.

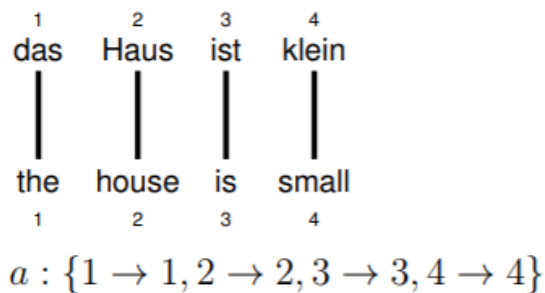
The word-based translation is not widely used today; phrase-based systems are more common. Most phrase-based system are still using GIZA++ to align the corpus. The alignments are used to extract phrases or deduce syntax rules. And matching words in bitext is still a problem actively discussed in the community. Because of the predominance of GIZA++, there are now several distributed implementations of it online. [1]

f – a foreign language, *e* – English

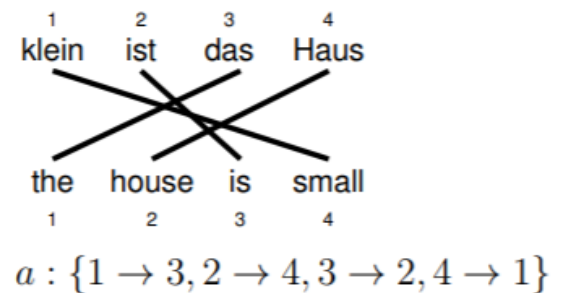
1.1 Alignment function

Mapping an English target word at position j to a German source word at position i with a function $a: j \rightarrow i$. Function a must be fully defined on the English (target) side. The reason for this is the noisy channel model, where the output sentence is the code that gets transmitted and distorted, so we must account for every e . [2]

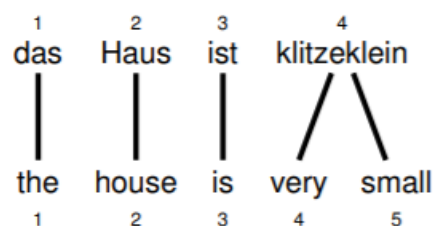
Example 1: Alignment



Example 2: Reordering

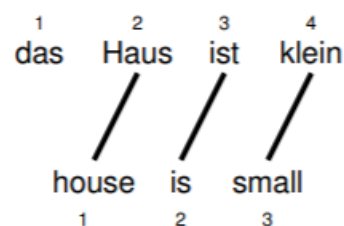


Example 3: One-to-Many Translation



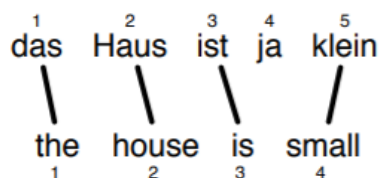
$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\}$$

Example 4: Dropping Words



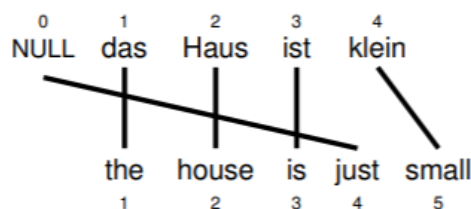
$$a : \{1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4\}$$

Example 5: Dropping Words



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 5\}$$

Example 6: Inserting Words



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 0, 5 \rightarrow 4\}$$

1.2 Basics

Take now a simplified look at a parallel corpus (parallelism on sentence level) *with given alignments* and imagine having observed the following alignments possibilities for the word *Haus*:

Translation of Haus	Count
house	8,000
building	1,600
home	200
household	150
shell	50
total	10,000

We want to estimate the lexical translation probabilities from corpus statistics, i.e. the probability of foreign word f being translated as English translation e :

$$p_f : e \rightarrow p_f(e).$$

It should be a probability function with usual properties of a probability distribution: $0 \leq p_f(e) \leq 1, \sum_e p_f(e) = 1, \forall f$.

1.2.1 Maximum Likelihood Estimation

How do we estimate $p_f(e)$ for $e = house$ and $f = Haus$?

$$p_{Haus}(house) \equiv p(house|Haus) = \frac{\text{count}(Haus \rightarrow house)}{\text{count}(Haus \rightarrow .)} = \frac{8,000}{10,000} = 0.8$$

For all translations of *Haus*, we get

$$p_f(e) = \begin{cases} 0.8 & \text{if } e = house, \\ 0.16 & \text{if } e = building, \\ 0.02 & \text{if } e = home, \\ 0.015 & \text{if } e = household, \\ 0.005 & \text{if } e = shell. \end{cases}$$

Estimation based on ratios of counts is called ‘maximum likelihood estimation’. [2]

1.3 IBM Model 1

IBM Models in general:

Generative models, which break up the translation process into smaller steps and achieve better statistics with simpler models.

IBM Model 1 uses only lexical translation. Ignores any position information (order), resulting in translating multisets of words into multisets of words.

Translation probability

- for a foreign sentence $f = (f_1, \dots, f_{l_f})$ of length l_f
- to an English sentence $e = (e_1, \dots, e_{l_e})$ of length l_e
- translation probability $t(e|f) \equiv p(e|f)$ (*t-tables*)
- with an alignment of each English word e_j to a foreign word f_i according to the alignment function $a : j \rightarrow i$

$$p(e, a|f) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \quad (1)$$

$\prod_{j=1}^{l_e} t(e_j | f_{a(j)})$ is the product over the lexical translation probabilities for all l_e generated target words. We use the product, since we assume that the lexical translation probabilities are independent.

ϵ is a normalization constant, s.t. $\sum_{e,a} p(e, a|f) = 1$. or a distribution of lengths $\epsilon(l_e|l_f)$.

$(l_f + 1)^{l_e}$ is the number of alignments of $l_f + \text{NULL}$ input words with l_e output words: the uniform probabilities over alignments.

Can also be defined the reverse direction: $p(f, a|e)$ (original IBM1).

Generative story for IBM translation Model 1:

1. pick a length l_e for e according to distribution $\epsilon(l_e|l_f)$.
2. for each $j = 1, \dots, l_e$ choose a value for a_j from $0, 1, \dots, l_f$ according to uniform distribution.
3. for each $j = 1, \dots, l_e$ choose a output word e_j according to $t(e_j | f_{a_j})$. [2]

Example:

das		Haus		ist		klein	
e	$t(e f)$	e	$t(e f)$	e	$t(e f)$	e	$t(e f)$
the	0.7	house	0.8	is	0.8	small	0.4
that	0.15	building	0.16	's	0.16	little	0.4
which	0.075	home	0.02	exists	0.02	short	0.1
who	0.05	household	0.015	has	0.015	minor	0.06
this	0.025	shell	0.005	are	0.005	petty	0.04

$$\begin{aligned}
 p(e, a|f) &= \frac{\epsilon}{5^4} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein}) \\
 &= \frac{\epsilon}{5^4} \times 0.7 \times 0.8 \times 0.8 \times 0.4 = 0.0029\epsilon
 \end{aligned}$$

1.4 Learning Lexical Translation Models

We would like to estimate the lexical translation probabilities $t(e/f)$ (and $t(f/e)$) from a corpus of parallel translations.

Problem: We don't have the alignments, only parallel sentences (i.e., sentences in source language, paired with sentences that are translations in target language).

Chicken-and-egg problem caused by incomplete data:

<i>machine translation</i>	<i>machine learning</i>
If we had alignments, we could estimate $t(e/f)$ by relative frequency count.	If we had complete data, we could estimate the model by Maximum Likelihood Estimation.
If we had the model $t(e/f)$, we could assign most probable alignments.	If we had the model, we could complete our data by most probable predictions.

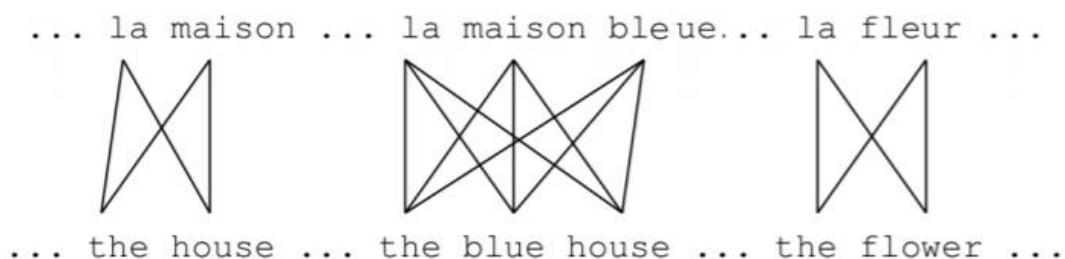
1.4.1 EM Algorithm

EM (Expectation Maximization) in a nutshell:

1. Initialize model parameters, e.g., uniform.
2. Assign probabilities to missing data.
3. Estimate model parameters from completed/manufactured/expected data.
4. Iterate step 2 - 3 until convergence. [2]

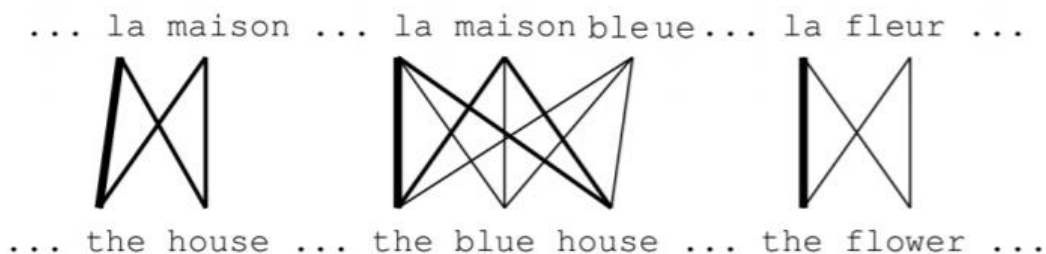
Initial step:

All alignments are equally likely. The Model learns that, e.g., "la" is often aligned with "the".



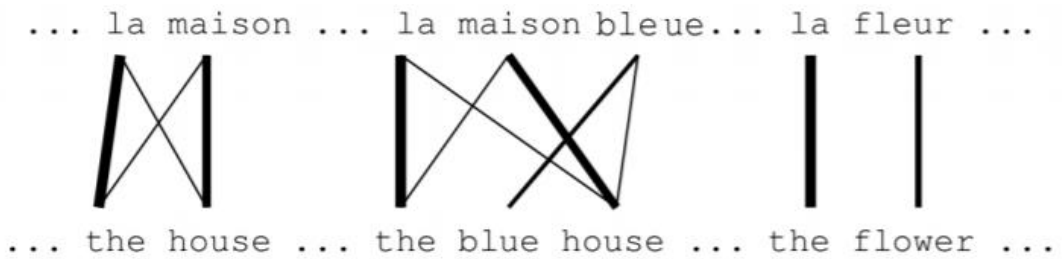
After one iteration:

Alignments, e.g., between "la" and "the" are more likely.



After another iteration:

It becomes apparent that alignments, e.g., between "fleur" and "flower" are more likely (pigeon hole principle).



Convergence:
Inherent hidden structure revealed by EM.



$$p(la/the) = 0.453$$

$$p(le/the) = 0.334$$

$$p(maison/house) = 0.876$$

$$p(bleue/blue) = 0.563$$

2. Phrase-Based Models

In phrase-based translation, the aim is to reduce the restrictions of word-based translation by translating whole sequences of words, where the lengths may differ. The sequences of words are called blocks or phrases, but typically are not linguistic phrases, but phrasemes found using statistical methods from corpora. It has been shown that restricting the phrases to linguistic phrases (syntactically motivated groups of words, see syntactic categories) decreases the quality of translation.

The chosen phrases are further mapped one-to-one based on a phrase translation table and may be reordered. This table can be learnt based on word-alignment, or directly from a parallel corpus. The second model is trained using the expectation maximization algorithm, similarly to the word-based IBM model. [1]

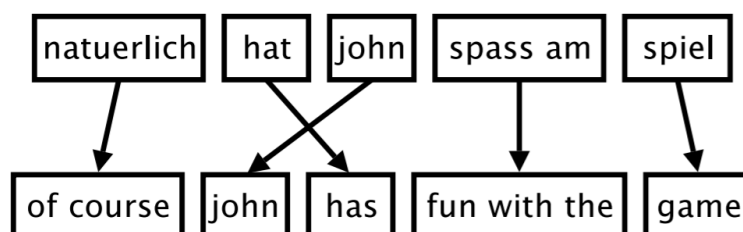
Word-Based Models translate words as atomic units.

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.



- 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*

<i>Translation</i>	<i>Probability $\varphi(\bar{e} \bar{f})$</i>
of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05

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

<i>English</i>	<i>$\varphi(\bar{e} \bar{f})$</i>	<i>English</i>	<i>$\varphi(\bar{e} \bar{f})$</i>
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

- Bayes rule

$$e_{best} = \operatorname{argmax}_e p(e|f) = \operatorname{argmax}_e p(f|e) p_{LM}(e)$$

- translation model $p(e|f)$
- language model $p_{LM}(e)$

- Decomposition of the translation model

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^I \phi(\bar{f}_i|\bar{e}_i) d(\operatorname{start}_i - \operatorname{end}_{i-1} - 1)$$

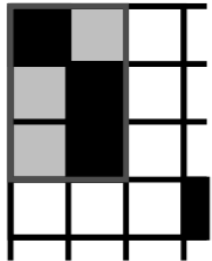
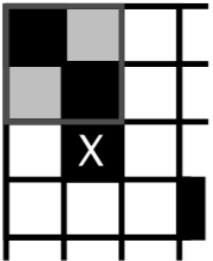
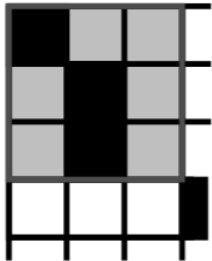
- phrase translation probability φ

Extracting Phrase Pairs

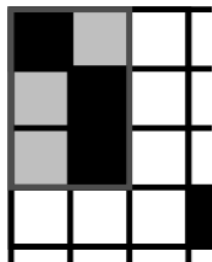
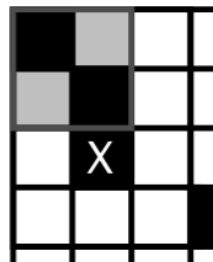
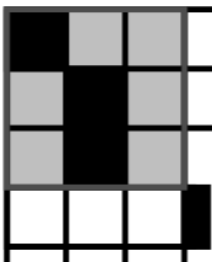
	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■	■	■				
that		■	■	■	■	■	■			
he							■			
will										■
stay										■
in								■		
the								■		
house									■	

Extract phrase pair consistent with word alignment: assumes that / geht davon aus , dass.

Consistent

		
consistent	inconsistent	consistent
ok	violated	ok
	one alignment point outside	unaligned word is fine

All words of the phrase pair have to align to each other.

		
consistent	inconsistent	consistent

Phrase pair (\bar{e}, \bar{f}) consistent with an alignment A , if all words f_1, \dots, f_n in \bar{f} that have alignment points in A have these with words e_1, \dots, e_n in \bar{e} and vice versa:

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

Phrase Pair Extraction

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■						
that						■				
he							■			
will										■
stay										■
in								■		
the								■		
house									■	

Smallest phrase pairs:

- michael — michael
- assumes — geht davon aus / geht davon aus ,
- that — dass / , dass
- he — er
- will stay — bleibt
- in the — im
- house — haus

unaligned words (here: German comma) lead to multiple translations.

Larger Phrase Pairs

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael	■									
assumes		■	■	■						
that						■				
he							■			
will										■
stay										■
in								■		
the								■		
house									■	

michael assumes — michael geht davon aus / michael geht davon aus ,
 assumes that — geht davon aus , dass ; assumes that he — geht davon aus , dass er
 that he — dass er / , dass er ; in the house — im haus
 michael assumes that — michael geht davon aus , dass
 michael assumes that he — michael geht davon aus , dass er
 michael assumes that he will stay in the house — michael geht davon aus , dass er im haus bleibt
 assumes that he will stay in the house — geht davon aus , dass er im haus bleibt
 that he will stay in the house — dass er im haus bleibt ; dass er im haus bleibt ,
 he will stay in the house — er im haus bleibt ; will stay in the house — im haus bleibt

Scoring Phrase Translations

- Phrase pair extraction: collect all phrase pairs from the data
- Phrase pair scoring: assign probabilities to phrase translations
- Score by relative frequency:

$$\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{e}, \bar{f})}{\sum_{\bar{f}_i} \text{count}(\bar{e}, \bar{f}_i)}$$

Size of the Phrase Table

- Phrase translation table typically bigger than corpus
 ... even with limits on phrase lengths (e.g., max 7 words)
 → Too big to store in memory?
- Solution for training
 – extract to disk, sort, construct for one source phrase at a time
- Solutions for decoding
 – on-disk data structures with index for quick look-ups
 – suffix arrays to create phrase pairs on demand

Weighted Model

- Described standard model consists of three sub-models
 – phrase translation model $\phi(\bar{f}|\bar{e})$
 – reordering model d
 – language model $p_{LM}(e)$

$$e_{best} = \underset{e}{\operatorname{argmax}} \prod_{i=1}^I \phi(\bar{f}_i | \bar{e}_i) d(\operatorname{start}_i - \operatorname{end}_{i-1} - 1) \prod_{i=1}^{|e|} p_{LM}(e_i | e_1 \dots e_{i-1})$$

- Some sub-models may be more important than others
- Add weights $\lambda_\phi, \lambda_d, \lambda_{LM}$

$$e_{best} = \underset{e}{\operatorname{argmax}} \prod_{i=1}^I \phi(\bar{f}_i | \bar{e}_i)^{\lambda_\phi} d(\operatorname{start}_i - \operatorname{end}_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|e|} p_{LM}(e_i | e_1 \dots e_{i-1})^{\lambda_{LM}}$$

Log-Linear Model

- Such a weighted model is a log-linear model:

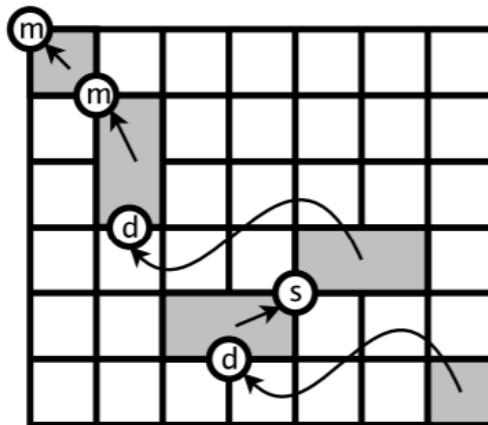
$$p(x) = \exp \sum_{i=1}^n \lambda_i h_i(x)$$

- Our feature functions
 - number of feature function $n = 3$
 - random variable $x = (e, f, \operatorname{start}, \operatorname{end})$
 - feature function $h_1 = \log \phi$
 - feature function $h_2 = \log d$
 - feature function $h_3 = \log p_{LM}$

Weighted Model as Log-Linear Model

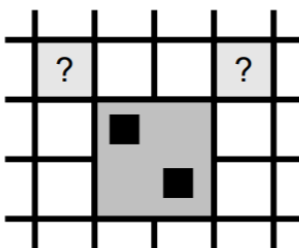
$$p(e, a|f) = \exp \left(\lambda_\phi \sum_{i=1}^I \log \phi(\bar{f}_i | \bar{e}_i) + \lambda_d \sum_{i=1}^I \log d(a_i - b_{i-1} - 1) + \lambda_{LM} \sum_{i=1}^{|e|} \log p_{LM}(e_i | e_1 \dots e_{i-1}) \right)$$

Lexicalized Reordering



- Distance-based reordering model is weak
 - learn reordering preference for each phrase pair
- Three orientations types: (m) monotone, (s) swap, (d) discontinuous
 - $\text{orientation} \in \{m, s, d\}$
 - $p_o(\text{orientation} | \bar{f}, \bar{e})$

Learning Lexicalized Reordering



- Collect orientation information during phrase pair extraction
 - if word alignment point to the top left exists → *monotone*
 - if a word alignment point to the top right exists → *swap*
 - if neither a word alignment point to top left nor to the top right exists → neither monotone nor swap → *discontinuous*

- Estimation by relative frequency

$$p_o(\text{orientation}) = \frac{\sum_{\bar{f}} \sum_{\bar{e}} \text{count}(\text{orientation}, \bar{e}, \bar{f})}{\sum_o \sum_{\bar{f}} \sum_{\bar{e}} \text{count}(o, \bar{e}, \bar{f})}$$

- Smoothing with unlexicalized orientation model $p(\text{orientation})$ to avoid zero probabilities for unseen orientations

$$p_o(\text{orientation} | \bar{f}, \bar{e}) = \frac{\sigma p(\text{orientation}) + \text{count}(\text{orientation}, \bar{e}, \bar{f})}{\sigma + \sum_o \text{count}(o, \bar{e}, \bar{f})}$$

EM Training of the Phrase Model

- We presented a heuristic set-up to build phrase translation table (word alignment, phrase extraction, phrase scoring)
- Alternative: align phrase pairs directly with EM algorithm
 - initialization: uniform model, all $\phi(\bar{e} | \bar{f})$ are the same
 - expectation step:
 - * estimate likelihood of all possible phrase alignments for all sentence pairs
 - maximization step:
 - * collect counts for phrase pairs $(\bar{e} | \bar{f})$, weighted by alignment probability
 - * update phrase translation probabilities $p(\bar{e} | \bar{f})$
- However: method easily overfits (learns very large phrase pairs, spanning entire sentences). [3]

3. Decoding

- We have a mathematical model for translation

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

- Task of decoding: find the translation e_{best} with highest probability

$$e_{best} = \text{argmax}_e p(e | f)$$

- Two types of error
 - the most probable translation is bad → fix the model

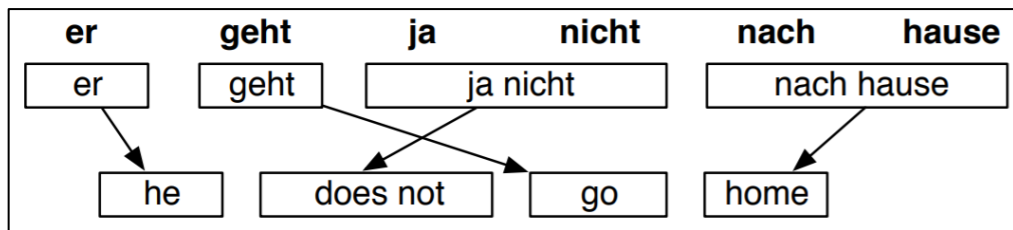
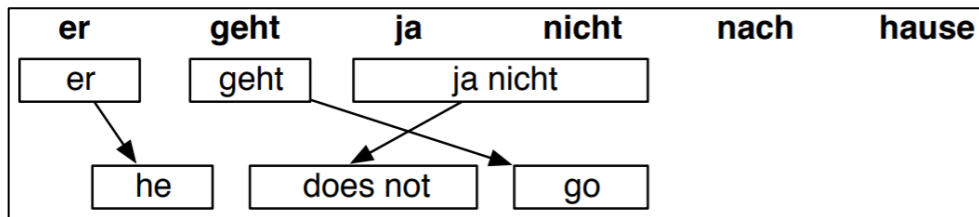
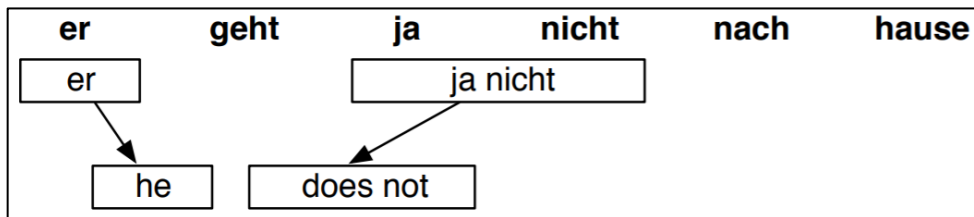
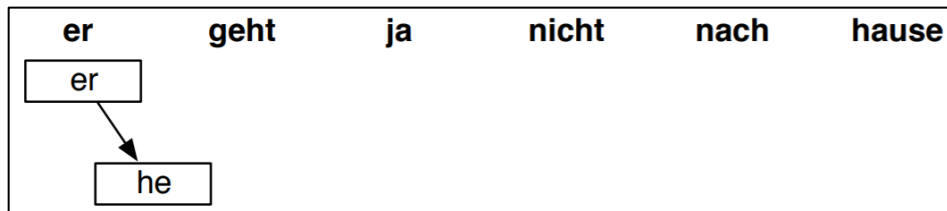
- search does not find the most probably translation → fix the search
- Decoding is evaluated by search error, not quality of translations (although these are often correlated)

Translation Process

- Task: translate this sentence from German into English

er geht ja nicht nach hause

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



Computing Translation Probability

- Probabilistic model for phrase-based translation:

$$e_{best} = \operatorname{argmax}_e \prod_{i=1}^I \phi(\bar{f}_i | \bar{e}_i) d(\operatorname{start}_i - \operatorname{end}_{i-1} - 1) p_{LM}(e)$$

- Score is computed incrementally for each partial hypothesis
- Components

Phrase translation Picking phrase \bar{f} to be translated as a phrase \bar{e}_i

→ 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

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

Language model For n-gram model, need to keep track of last $n - 1$ words

→ compute score $p_{LM}(w_i | w_{i-(n-1)}, \dots, w_{i-1})$ for added words w_i

Translation Options

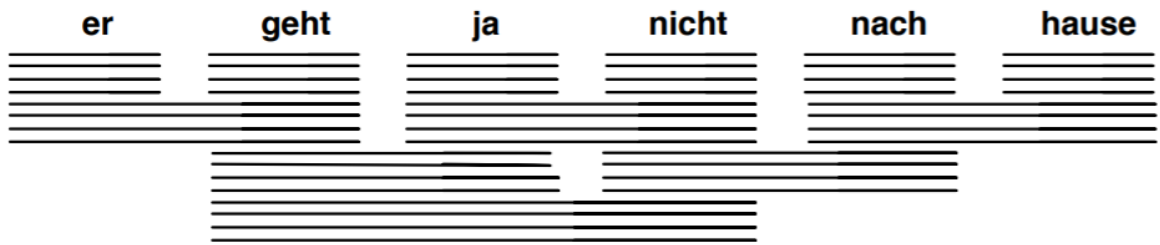
er	geht	ja	nicht	nach	hause
he	is	yes	not	after	house
it	are	is	do not	to	home
, it	goes	, of course	does not	according to	chamber
, he	go	,	is not	in	at home
it is		not		home	
he will be		is not		under house	
it goes		does not		return home	
he goes		do not		do not	
	is		to		
	are		following		
	is after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

- 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

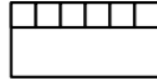
er	geht	ja	nicht	nach	hause
he	is	yes	not	after	house
it	are	is	do not	to	home
, it	goes	, of course	does not	according to	chamber
, he	go	,	is not	in	at home
it is		not		home	
he will be		is not		under house	
it goes		does not		return home	
he goes		do not		do not	
	is		to		
	are		following		
	is after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

- 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.

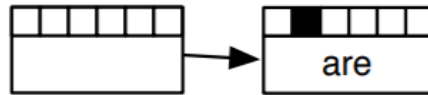
Decoding: Precompute Translation Options



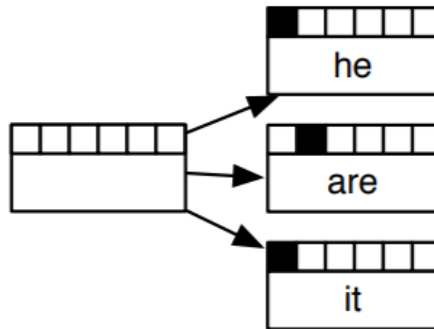
consult phrase translation table for all input phrases



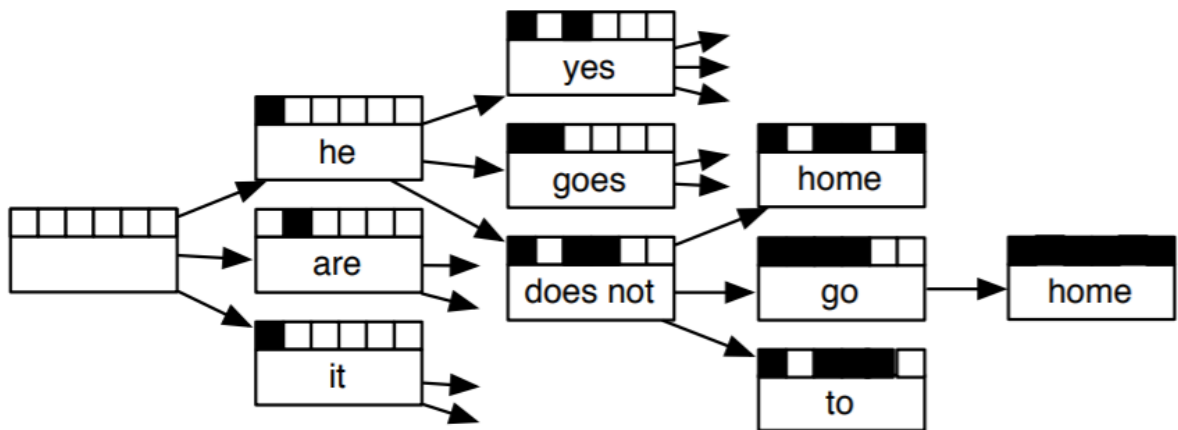
initial hypothesis: no input words covered, no output produced



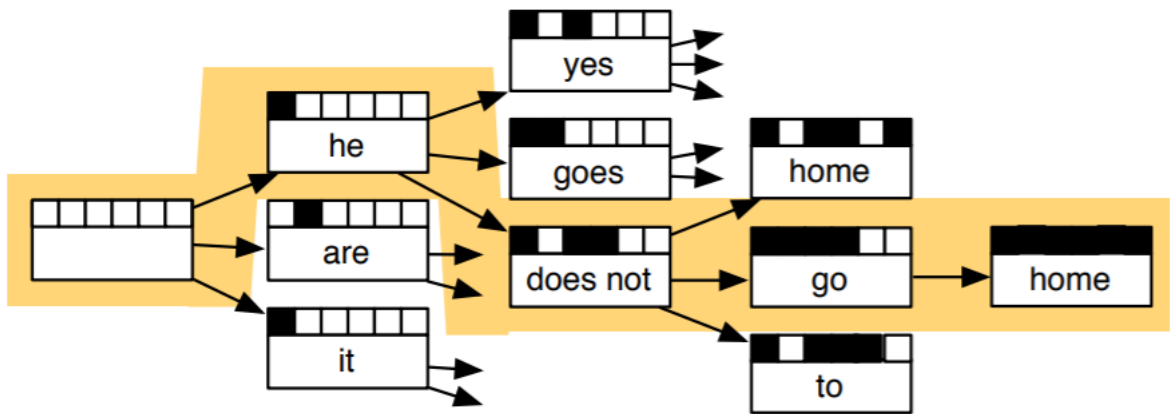
pick any translation option, create new hypothesis



create hypotheses for all other translation options



also create hypotheses from created partial hypothesis



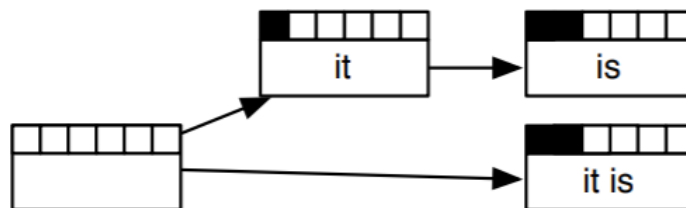
backtrack from highest scoring complete hypothesis

Computational Complexity

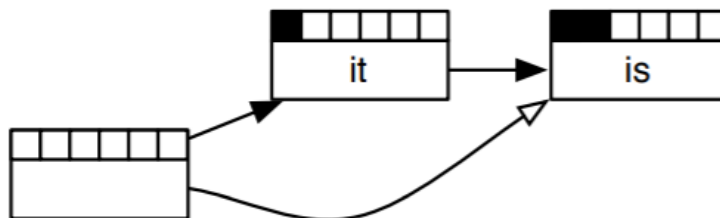
- The suggested process creates exponential number of hypothesis
- Machine translation decoding is NP-complete
- Reduction of search space:
 - recombination (risk-free)
 - pruning (risky)

Recombination

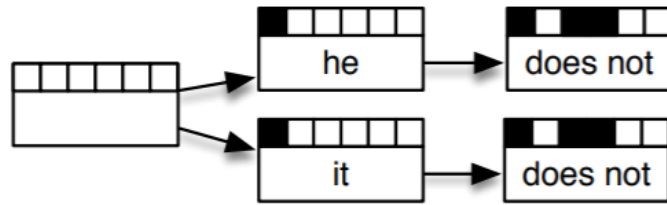
- Two hypothesis paths lead to two matching hypotheses
 - same number of foreign words translated
 - same English words in the output
 - different scores



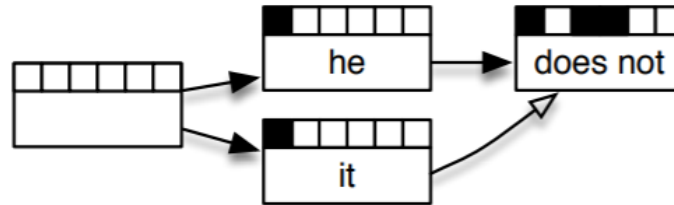
- Worse hypothesis is dropped



- 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



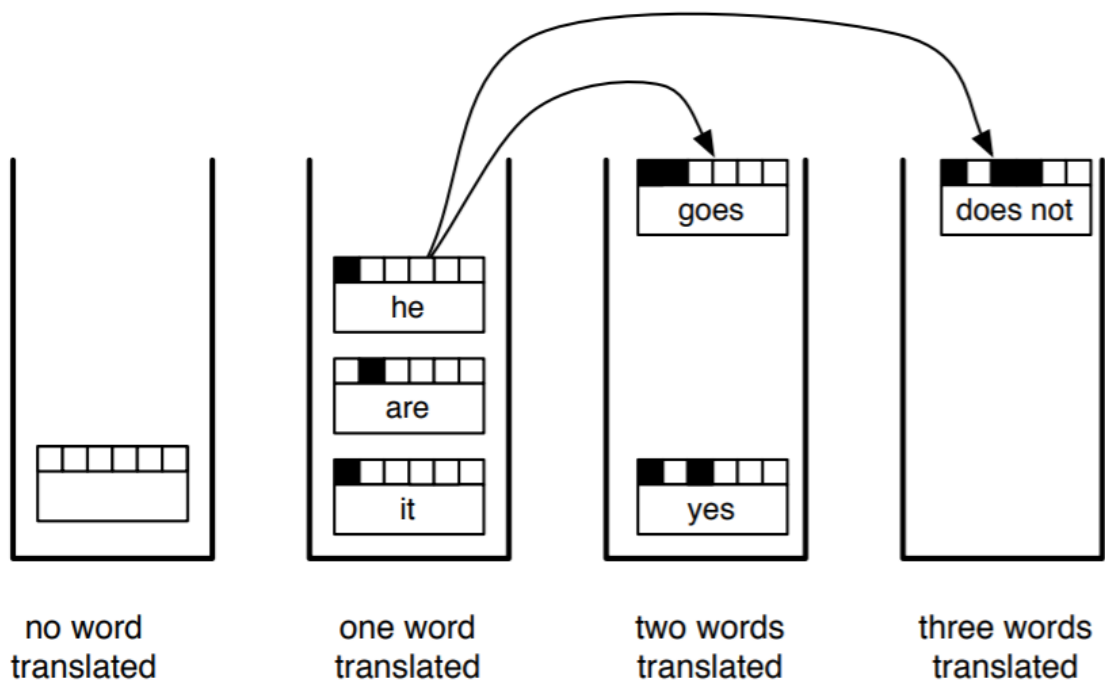
Restrictions on Recombination

- *Translation model*: Phrase translation independent from each other
→ no restriction to hypothesis recombination
- *Language model*: Last $n-1$ words used as history in n -gram language model
→ 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
→ 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

- 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 $\alpha \times \text{best score}$ ($\alpha < 1$)
- 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

Other Decoding Algorithms

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

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. [4]

References

1. Statistical machine translation. URL:
[https://en.wikipedia.org/wiki/Statistical_machine_translation#:~:text=Statistical%20machine%20translation%20\(SMT\)%20is,analysis%20of%20bilingual%20text%20corpora](https://en.wikipedia.org/wiki/Statistical_machine_translation#:~:text=Statistical%20machine%20translation%20(SMT)%20is,analysis%20of%20bilingual%20text%20corpora).
2. Statistical Machine Translation. Word-based models. Available:
<http://statmt.org/book/slides/04-word-based-models.pdf>
3. Statistical Machine Translation. Phrase-based models. Available:
<http://www.statmt.org/book/slides/05-phrase-based-models.pdf>
4. Statistical Machine Translation. Decoding. Available: <http://statmt.org/book/slides/06-decoding.pdf>