USC Viterbi
School of Engineering
Department of Computer Science

CSCI 544
Applied Natural Language Processing

Mohammad Rostami
USC Computer Science Department

Slides adopted from slides by Michael Collins and Philipp Koehn

Logistical Notes

- Proposals: feedback will be provided, if necessary. Otherwise, go ahead with your plan
- Status report: 11/11
- Midterm:

- A normal written exam for 80-90 minutes
- 16-20 essay questions: no multiple-choice question
- There will be different versions of the exam
- Open book but you are not allowed to type or use your phone
- Remote students: camera
- Uploading your exam: 10-15 minutes after the initial 90 minutes
- We will still collect your written papers

- Dropping the Course

IBM Models

- Key ideas in the IBM translation models:
- Alignment mappings
- Lexical word translation parameters
- Distortion parameters
- EN Algorithm is used for learning the parameters

$$p(f, a \mid e, m) = \prod_{i=1}^{m} \mathbf{q}(a_j \mid j, l, m)\mathbf{t}(f_j \mid e_{a_j})$$

- Once the parameters are trained, we can recover the most likely alignments on training examples

$$p(a \mid f, e, m) = \frac{p(f, a \mid e, m)}{\sum_{a \in \mathcal{A}} p(f, a \mid e, m)}$$

$$a^* = \arg\max_a p(a \mid f, e, m)$$

IBM Models

- Weaknesses of IBM model's alignments:

1. Noisy: not accurate
2. Many-to-One: many words in the source language can be mapped to a single word, i.e., for each source word we find one target word -> Many-to-Many
3. Non-compositional phrases are not encoded
4. Context is not considered in translation
5. Propositions may not be translated properly

Phrase Based Translation Models

- 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, e.g., red herring, hot dog
  - use of local context in translation
  - the more data, the longer phrases can be learned
- SOTA used by Google Translate and others until about 2017

Phrase Based Translation Models

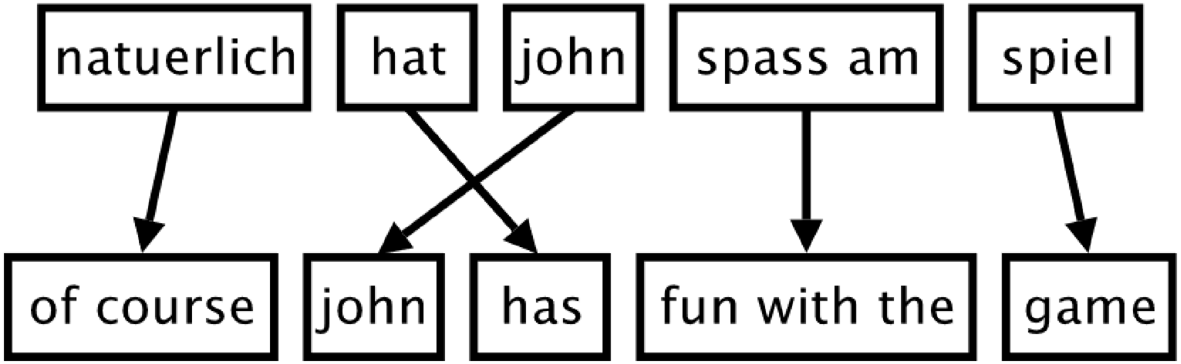
- Translation involves many phrase-based (PB) lexicons, e.g., non-compositional phrases, “ we can infer”, “United Kingdom”
- A PB lexicon pairs strings in one language with strings in another language, e.g.,

nach Kanada	$\leftrightarrow$	in Canada
zur Konferenz	$\leftrightarrow$	to the conference
Morgen	$\leftrightarrow$	tomorrow
fliege	$\leftrightarrow$	will fly
...		

- Improves upon word-to-word MT models of IBM

Phrase Based Translation

- Source language input is segmented into phrases (a phrase can be a single word)
- Each phrase is translated into a phrase in the target language
- Phrases are reordered



- Requirement: tables with phrase translations and their probabilities

- Ex: table for “natuerlich”

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

Phrase-Level Bilingual Dictionary

- Model should not be limited to linguistic phrases (noun phrases, verb phrases, prepositional phrases, ...)
- Example non-linguistic phrase pair

$$\text{spass am} \rightarrow \text{fun with the}$$

- Prior noun often helps with translation of preposition
- Experiments show that limitation to linguistic phrases hurts quality

Phrase Based Translation Model

- A sentences is broken into I phrases

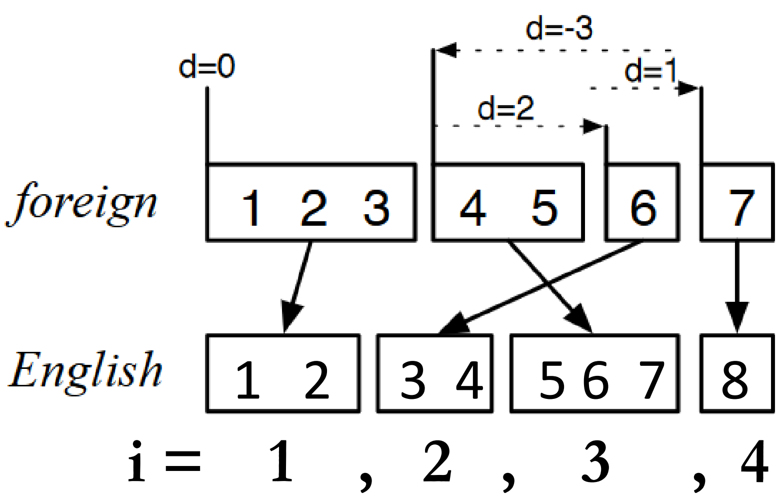
$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\, d(start_i - end_{i-1} - 1)$$

Phrase Translation Probability

Distortion: Reordering Probabilities

- Distance-based reordering:

d(starting word for the current phrase– ending word for the previous phrase- 1)



phrase	translates	movement	distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	-3
4	7	skip over 6	+1

Phrase Based Training

- Learn the model from a parallel corpus

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\; d(start_i - end_{i-1} - 1)$$

- Three stages:
  – Word alignment: using IBM models or other method as the starting point
  – Extraction of phrase pairs via extending the IBM model
  – Computing the model parameters

Building Phrase Level Alignment

- Representing alignments using matrices

English: Mary did not slap the green witch

Spanish: Maria no daba una bofetada a la bruja verde

Sp

En

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did						●			
not		●							
slap			●	●	●				
the							●		
green									●
witch								●	

IMB Model Alignment

Building Phrase Level Alignment

- Approach

1. Train a model for $p(f|e)$ using IBM 2
2. Train a model for $p(e|f)$ using IBM 2
3. Extracting phrases: take intersection of the two alignments as a starting point and use them to grow alignments on the union of the alignments

Building Phrase Level Alignment

- Example

Alignment from $p(f \mid e)$ model:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did									
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

Alignment from $p(e \mid f)$ model:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap					●				
the							●		
green									●
witch								●	

Heuristics for Growing Alignments

- Only explore alignment in union of $p(f|e)$and $p(e|f)$ alignment
- Add one alignment point at a time
- Only add alignment points which align a word that currently has no alignment
- Restrict ourselves to alignment points that are “neighbors” (adjacent or diagonal) of current alignment points (we consider other alignment points)

Building Phrase Level Alignment

- The final alignment, created by taking the intersection of the two alignments, then adding new points using the growing heuristics:

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

- Note that the alignment is no longer many-to-one: potentially multiple Spanish words can be aligned to a single English word, and vice versa.

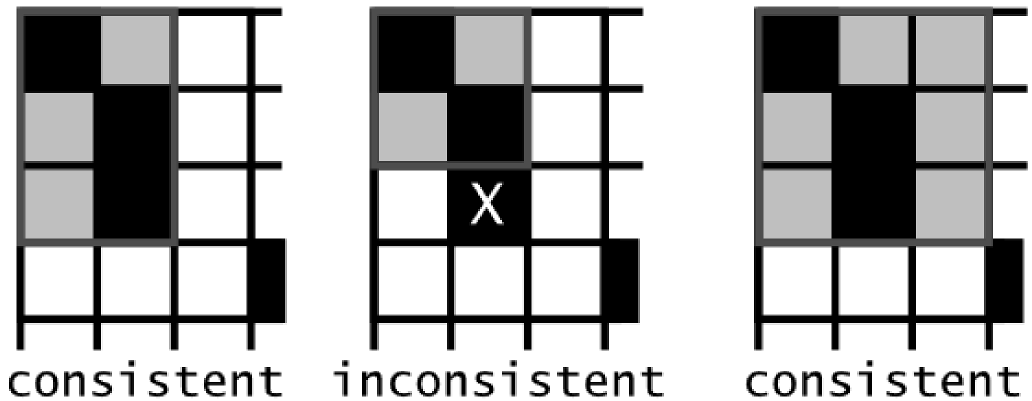
Extracting Phrase Pairs

- A phrase-pair consists of a sequence of English words, e, paired with a sequence of foreign words, f
- A phrase-pair (e,f) is consistent if: 1) there is at least one word in e aligned to a word in f; 2) there are no words in f aligned to words outside e; 3) there are no words in e aligned to words outside f, e.g., (Mary did not, Maria no) is consistent. (Mary did, Maria no) is not consistent
- We extract all consistent phrase pairs from the training example

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

Extracting Phrase Pairs

- Consistent Phrases



ok

violated
one alignment point outside

ok
unaligned word is fine

Extracting Phrase Pairs

- Ex: (Maria, Mary), (Naria no, Mary did not), (no daba una bof',did not slap), (a la bruja, not slap the)

	Maria	no	daba	una	bof'	a	la	bruja	verde
Mary	●								
did		●							
not		●							
slap			●	●	●				
the						●	●		
green									●
witch								●	

Extracting Phrase Pairs

- Scoring Phrase Translations

- Phrase pair extraction: collect all phrase pairs from the data
- Phrase pair scoring: assign probabilities to phrase translations
- Use empirical frequency

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

EM for Phrase Based MT

- Heuristic set-up to build phrase translation table: (word alignment, phrase extraction, phrase scoring)
- Align phrase pairs directly with EM algorithm

– initialization: uniform model, all probabilities are equally likely

– expectation step:

estimate likelihood of all possible phrase alignments for all sentence pairs

– maximization step:

collect counts for phrase pairs, weighted by alignment probability

update phrase translation probabilities

Phrase Lexicon Probabilities

- Real Example: Koehn, EACL 2006

- Translation table for “den Vorschlag”

English	$t(e\|f)$	**English**	$t(e\|f)$
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	...	...

Decoding in Machine Translation

- We can estimate $p(\mathbf{f}|\mathbf{e})$ using the parallel bilingual corpus
- We can estimate $p(\mathbf{e})$ using the target language corpus
- Translation procedure: given a foreign language, find a sequence in the target language such that:

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

Decoding in Machine Translation

- Challenges:

- Discrete optimization $\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\, p(\mathbf{e}|\mathbf{f})$
- Very large search space

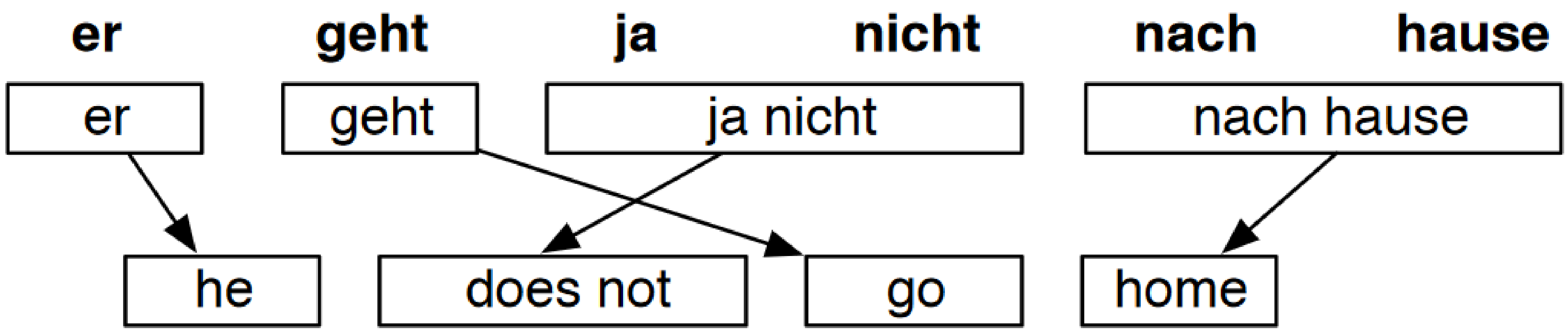
- Two types of error

- the most probable translation is bad →fix the model
- search does not find the most probably translation →fix the search process

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

Decoding Process by Human Translators

- Translate Sentence by Sentence
- Pick phrases in the sentence
- Translate the phrases
- Reorder the phrases



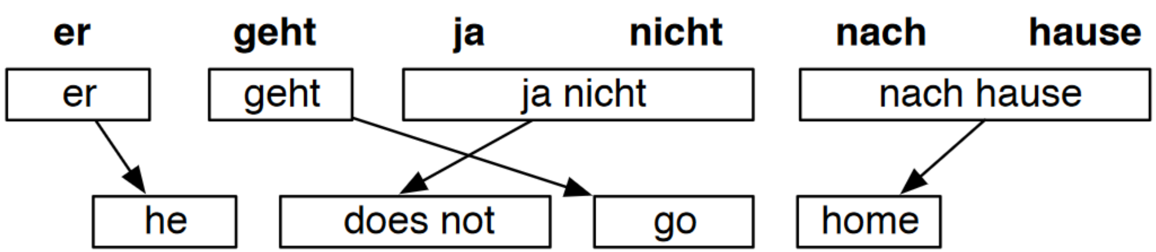
Decoding Process in MT

- Probabilistic model for phrase-based translation:

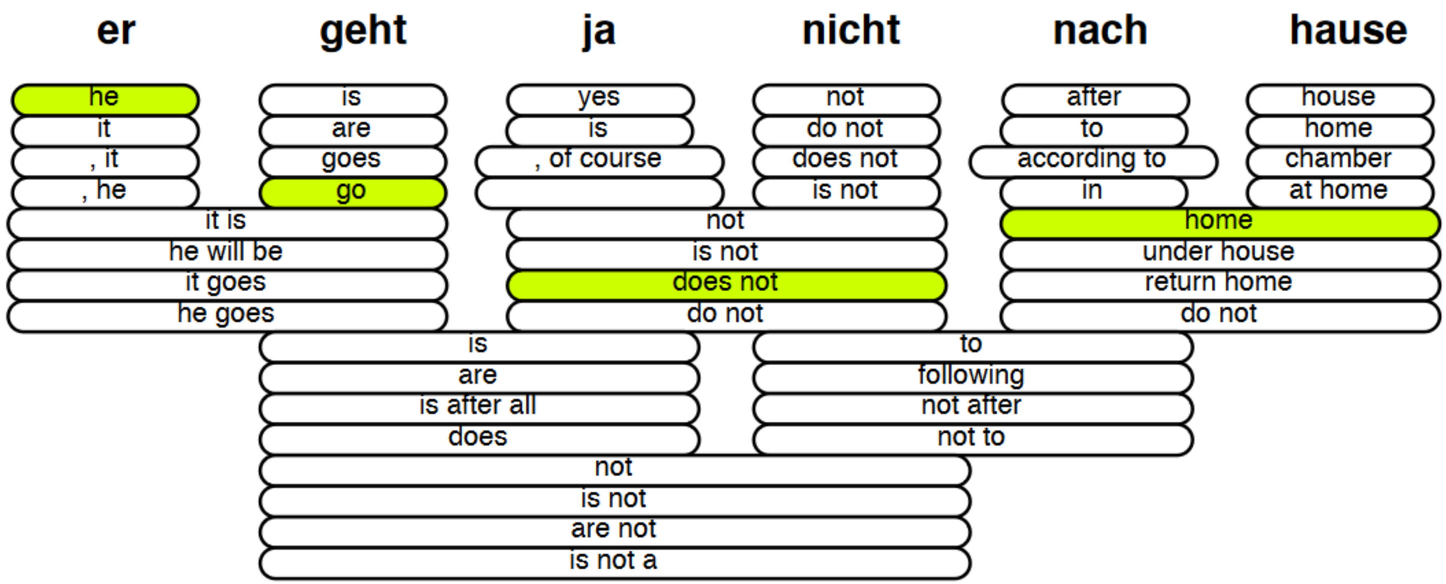
$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}} \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\; d(start_i - end_{i-1} - 1)\; p_{\text{LM}}(\mathbf{e})$$

- Generating candidates and incrementally and compute probability for each partial hypothesis
- Procedure:

- Picking phrases: translate using phrase translation tables
- Reordering: Previous phrase ended in end_{i-1} current phrase starts at
- $start_i \rightarrow \text{compute } d(start_i - end_{i-1} - 1)$
- Language model: keep track of the sequence as it is built to compute $p_{\text{LM}}(\mathbf{e})$



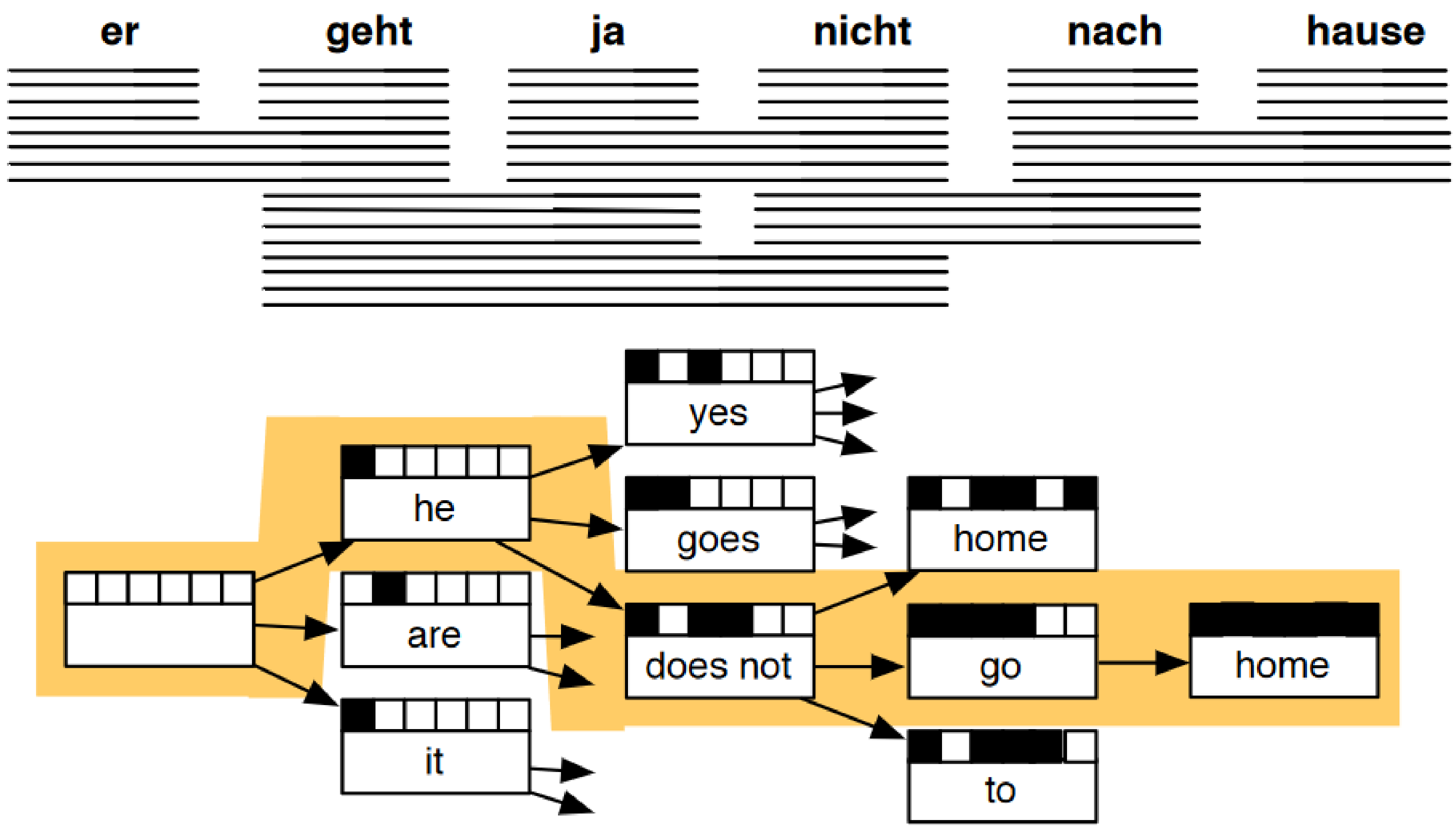
Decoding Process in MT



- Challenge: we have many translation options to choose
  - in Europarl phrase table: 2727 matching phrase pairs for this sentence
  - by pruning to the top 20 per phrase, 202 translation options remain
- MT Decoder:
  - picking the right translation options
  - arranging them in the right order
  - Search is performed using a version beam search

Decoding: Find the Best Path

- backtrack from highest scoring complete hypothesis (incomplete sentence)



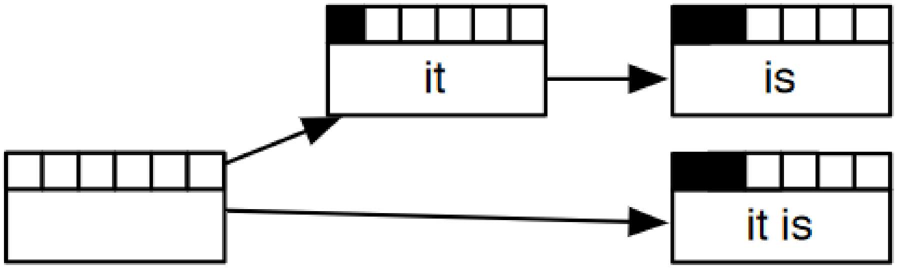
Computational Complexity

- The suggested process creates exponential number of hypothesis
- Machine translation decoding is NP-complete
- Solution: we need to reduce the search space

- Recombination
- Pruning

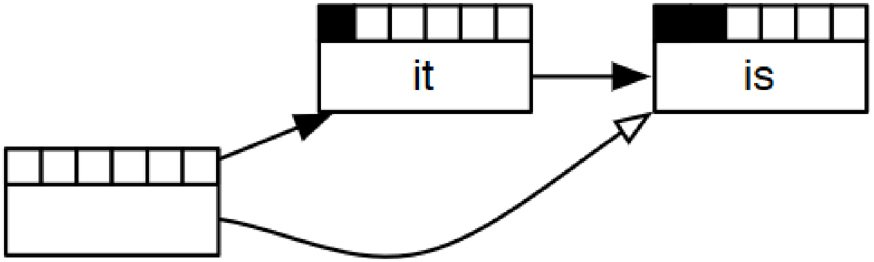
Recombination

- Two hypothesis paths lead to two matching hypotheses
- Same foreign words are translated
- Same English words at the output



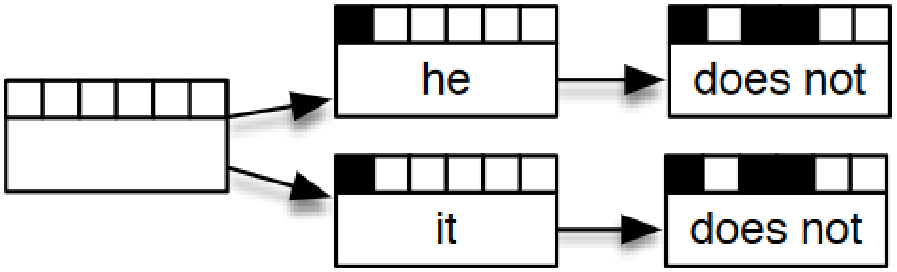
- Worse hypothesis is dropped

\- Same foreign words are translated

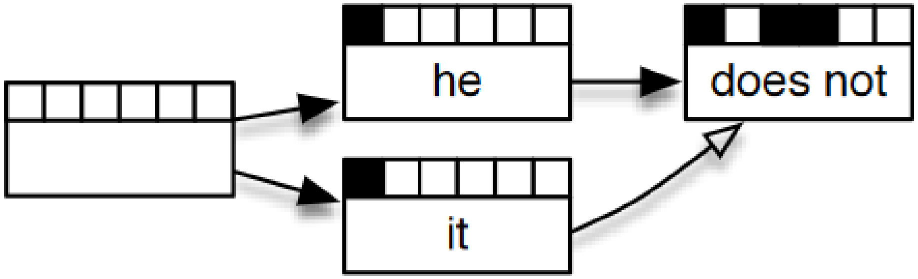


Recombination

- Two hypothesis paths lead to hypotheses indistinguishable in subsequent search
- Same foreign words are translated
- Same last two English words in output (assuming trigram language model)
- Same last foreign word translated



- Worse hypothesis is dropped



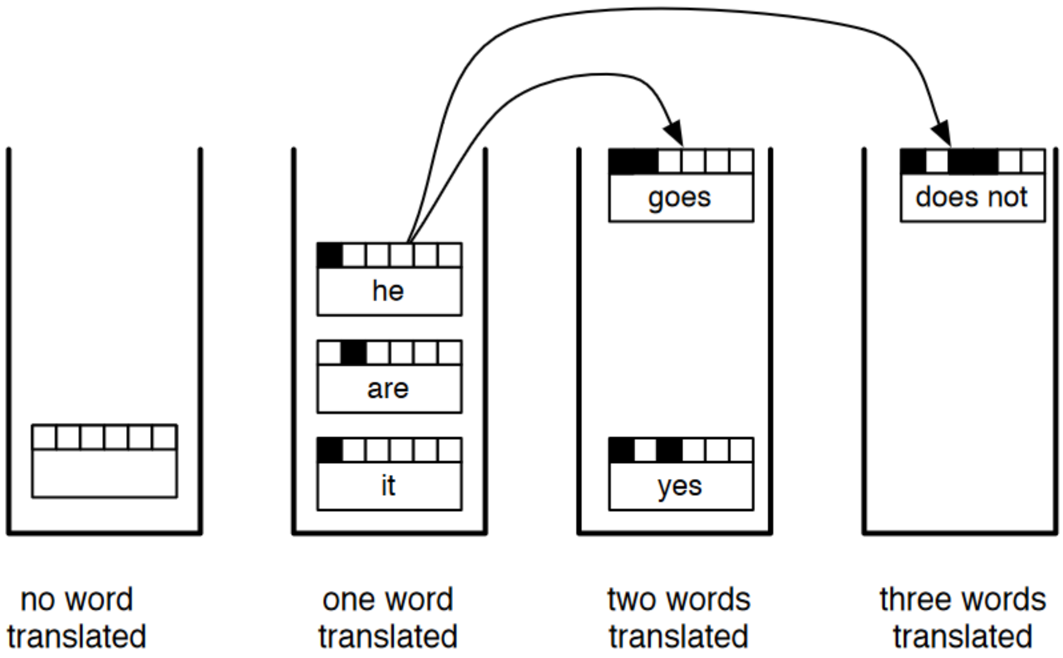
Restrictions on Recombination

- **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
- Recombination reduces search space, but not enough (we still have a NP complete problem on our hands)

Pruning

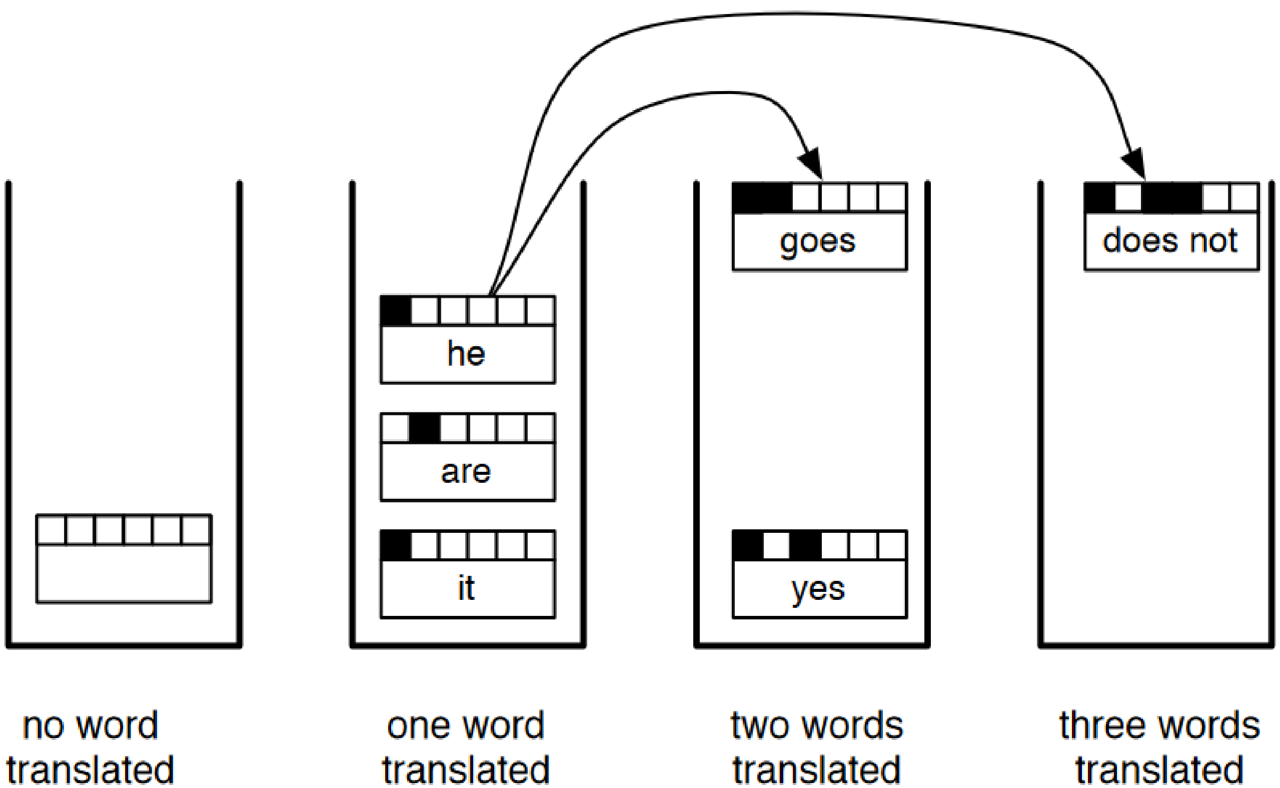
- Remove bad hypotheses early:

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



Pruning

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



Stack Decoding Algorithm

- Quadratic complexity with respect to sentence length

```
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
```