



UPPSALA  
UNIVERSITET

# Phrase-based Statistical Machine Translation

Sara Stymne

2021-09-08

Some slides from Philipp Koehn



# Lecture outline

- What is phrase-based SMT?
- Phrase-based modeling
- Training
- Log-linear models and features
- Decoding



# Word-based vs Phrase-based SMT

- Word-based models translate **words** as atomic units
- Phrase-based models translate **phrases** as atomic units



# Word-based vs Phrase-based SMT

- Word-based models translate **words** as atomic units
- Phrase-based models translate **phrases** as atomic units
  - A phrase is a continuous sequence of words
  - Not necessarily a linguistic phrase



# Word-based SMT: Generative Model

Bakom huset hittade polisen en stor mängd narkotika .

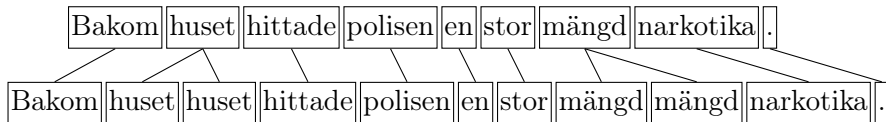


# Word-based SMT: Generative Model

Bakom huset hittade polisen en stor mängd narkotika .



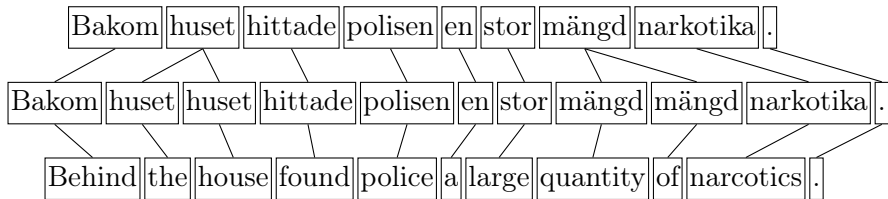
# Word-based SMT: Generative Model



## 1 Fertility



# Word-based SMT: Generative Model

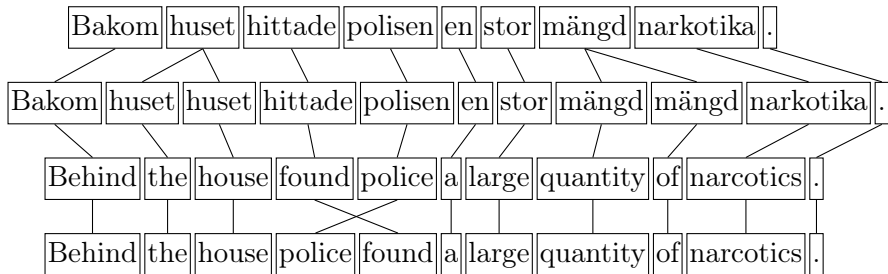


- 1 Fertility
- 2 Word translation





# Word-based SMT: Generative Model



- 1 Fertility
- 2 Word translation
- 3 Output ordering



# Phrase-based SMT: Generative Model

Bakom huset hittade polisen en stor mängd narkotika .



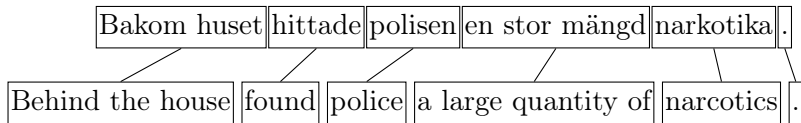
# Phrase-based SMT: Generative Model

Bakom huset hittade polisen en stor mängd narkotika.

## 1 Phrase segmentation



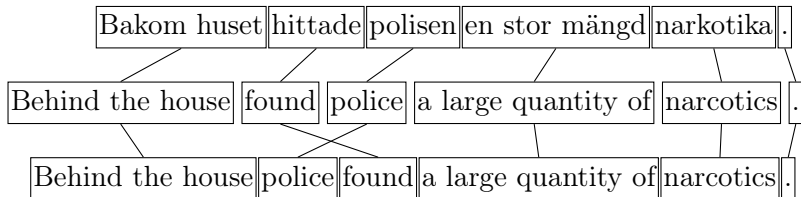
# Phrase-based SMT: Generative Model



- 1 Phrase segmentation
- 2 Phrase translation



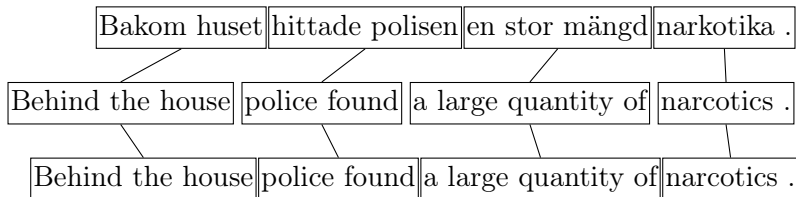
# Phrase-based SMT: Generative Model



- 1 Phrase segmentation
- 2 Phrase translation
- 3 Output ordering



# Phrase-based SMT: Alternative segmentation



- 1 Phrase segmentation
- 2 Phrase translation
- 3 Output ordering



# Advantages of Phrase-based SMT

- Translating phrases helps to reduce translation ambiguities
- Phrases of arbitrary length: sometimes an entire (short) sentence might be covered by a phrase
- Simpler model: no more need to explicitly model the concepts of fertility, insertion and deletion of words



# Advantages of Phrase-based SMT

- Phrase translation can handle:
  - Non-compositional phrases  
kick the bucket – ins Gras beissen (lit: into grass bite)
  - Compounds  
myggmedel – mosquito repellent
  - Phrasal verbs  
koppla av – relax
- Use of local context:
  - Local word order  
affaires extérieures – external affairs
  - Local agreement issues  
**ett rött** block – a red block  
**en röd** kon – a red cone





# Phrase translation table

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

Translation	Probability $\phi(\bar{t} \bar{s})$
of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05



# Real example

UPPSALA  
UNIVERSITET

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t} \bar{s})$	English	$\phi(\bar{t} \bar{s})$
the	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	this announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	a message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	were told	0.000229
the message	0.012816	the back and	2.917e-05



# Real example

UPPSALA  
UNIVERSITET

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t} \bar{s})$	English	$\phi(\bar{t} \bar{s})$
the	0.226415	the news	0.012816
<b>told</b>	0.169811	the report	0.008544
<b>announcement</b>	0.075472	the information	0.008544
<b>message</b>	0.056604	the back	0.004272
<b>news</b>	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	this announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	a message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	were told	0.000229
the message	0.012816	the back and	2.917e-05

- Lexical variation (announcement, message, news, told, ...)



# Real example

UPPSALA  
UNIVERSITET

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t} \bar{s})$	English	$\phi(\bar{t} \bar{s})$
the	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
<b>information</b>	0.037736	the death	0.004272
<b>informed</b>	0.037736	this announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	a message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	were told	0.000229
the message	0.012816	the back and	2.917e-05

- Lexical variation (announcement, message, news, told, ...)
- Morphological variation (information, informed)



# Real example

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t} \bar{s})$	English	$\phi(\bar{t} \bar{s})$
the	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	<b>this</b> announcement	0.002848
learnt	0.037736	this news	0.002136
peace of mind by ensuring	0.027778	<b>a</b> message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	<b>were</b> told	0.000229
<b>the</b> message	0.012816	the back and	2.917e-05

- Lexical variation (announcement, message, news, told, ...)
- Morphological variation (information, informed)
- Included function words (the, a, were, this)



# Real example

- Phrase translations for **begreppet** learned from the Europarl corpus

English	$\phi(\bar{t} \bar{s})$	English	$\phi(\bar{t} \bar{s})$
<b>the</b>	0.226415	the news	0.012816
told	0.169811	the report	0.008544
announcement	0.075472	the information	0.008544
message	0.056604	the back	0.004272
news	0.056604	the suspension	0.004272
information	0.037736	the death	0.004272
informed	0.037736	this announcement	0.002848
learnt	0.037736	this news	0.002136
<b>peace of mind by ensuring</b>	0.027778	a message	0.001539
insight	0.018868	his answer	0.000356
the announcement	0.017088	were told	0.000229
the message	0.012816	<b>the back and</b>	2.917e-05

- Lexical variation (announcement, message, news, told, ...)
- Morphological variation (information, informed)
- Included function words (the, a, were, this)
- Noise (the, the back and, piece of ...)



- Model is not limited to linguistic phrases  
(noun phrases, verb phrases, prepositional phrases)
- Example of useful non-linguistic phrases:
  - det finns – there is/are
  - put off – skjuta upp



- Model is not limited to linguistic phrases  
(noun phrases, verb phrases, prepositional phrases)
- Example of useful non-linguistic phrases:
  - det finns – there is/are
  - put off – skjuta upp
- Experiments have shown that limitation to only linguistic phrases hurts quality





# Probabilistic model

- Bayes rule

$$\begin{aligned} t_{best} &= \arg \max_t p(t|s) \\ &= \arg \max_t p(s|t)p_{LM}(t) \end{aligned} \tag{1}$$

- translation model:  $p(s|t)$
- language model:  $p_{LM}(t)$



# Probabilistic model

- Bayes rule

$$\begin{aligned} t_{best} &= \arg \max_t p(t|s) \\ &= \arg \max_t p(s|t)p_{LM}(t) \end{aligned} \tag{1}$$

- translation model:  $p(s|t)$

- language model:  $p_{LM}(t)$

- Decomposition of translation model

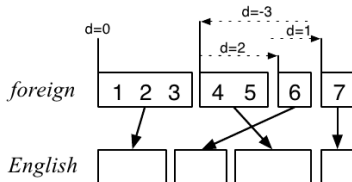
$$p(\bar{s}_1^I | \bar{t}_1^I) = \prod_{i=1}^I \phi(\bar{s}_i | \bar{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1)$$

- phrase translation model:  $\phi$

- reordering probability:  $d$



# Distance-based reordering



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

- distance =  $\text{start}_i - \text{end}_{i-1} - 1$
- Scoring function:  $d(x) = \alpha^{|x|}$  – exponential with distance



# Learning a Phrase Translation Table

- Task: learn the model from a parallel corpus
- Three stages:
  - Word alignment
  - Extraction of phrase pairs
  - Scoring of phrase pairs



# Word alignment

	nyss	hade	jag	precis	tappat	bort	glassen
a	■						
moment	■						
ago	■						
I			■				
had		■					
just				■			
lost					■	■	
my							
ice							■
cream							■



# Obtaining a word alignment

- Using some word alignment software and models, often:
  - GIZA++
  - IBM1 – HMM – IBM3 – IBM4



# Obtaining a word alignment

- Using some word alignment software and models, often:
  - GIZA++
  - IBM1 – HMM – IBM3 – IBM4
- Such models are directional
  - Gives 1–N links
  - Does not give M–1 or M–N links



# Obtaining a word alignment

- Using some word alignment software and models, often:
  - GIZA++
  - IBM1 – HMM – IBM3 – IBM4
- Such models are directional
  - Gives 1–N links
  - Does not give M–1 or M–N links
- We want all types of links!
- Solution: symmetrize directional alignments





# Directional word alignment

	nyss	hade	jag	precis	tappat	bort	glassen
a	■						
moment	■						
ago	■						
I			■				
had		■					
just				■			
lost					■		
my							
ice							■
cream							■

■ En-Sv (M-1)



# Directional word alignment

	nyss	hade	jag	precis	tappat	bort	glassen
a							
moment	■						
ago							
I			■				
had		■					
just				■			
lost					■	■	
my							
ice							
cream							■


■ En-Sv (M-1)


■ Sv-En (1-N)



# Word alignment – symmetrization

	nyss	hade	jag	precis	tappat	bort	glassen
a	En-Sv (M-1)						
moment	Both						
ago	En-Sv (M-1)						
I			Both				
had		Both					
just				Both			
lost					Both	Sv-En (1-N)	
my							
ice							En-Sv (M-1)
cream							Both

 En-Sv (M-1)

 Sv-En (1-N)

 Both



# Word alignment – intersection

	nyss	hade	jag	precis	tappat	bort	glassen
a							
moment	■						
ago							
I			■				
had		■					
just				■			
lost					■		
my							
ice							
cream							■

Intersection



# Word alignment – union

	nyss	hade	jag	precis	tappat	bort	glassen
a	■						
moment	■						
ago	■						
I			■				
had		■					
just				■			
lost					■	■	
my							
ice							■
cream							■

Union



# Word alignment symmetrization

- Intersection: too few links
- Union: too many links

English–Swedish alignment (Holmqvist, 2008)

	Precision	Recall
Intersection	90	75
Union	60	91



# Word alignment symmetrization

- Intersection: too few links
- Union: too many links
- Use heuristics: grow-diag-final-and

English–Swedish alignment (Holmqvist, 2008)

	Precision	Recall
Intersection	90	75
Union	60	91



# Word alignment symmetrization

- Intersection: too few links
- Union: too many links
- Use heuristics: grow-diag-final-and

English–Swedish alignment (Holmqvist, 2008)

	Precision	Recall
Intersection	90	75
Union	60	91
grow-diag-final-and	70	88





# Extracting phrase pairs

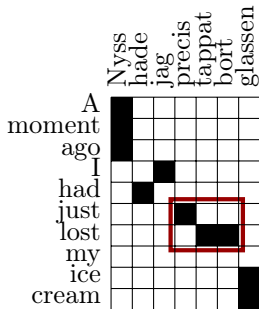
- Extract phrase pairs that are consistent with word alignments

	Nyss	hade	jag	precis	tappat	bort	glassen
A	■						
moment	■						
ago	■						
I			■				
had		■					
just				■			
lost					■	■	
my					■	■	
ice							■
cream							■



# Extracting phrase pairs

- Extract phrase pairs that are consistent with word alignments

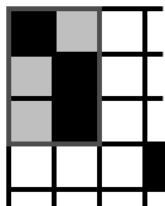


just lost-precis tappat bort



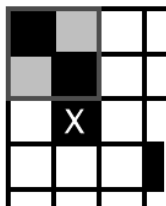
# Consistent phrase pairs

All words of the phrase pairs have to align to each other



consistent

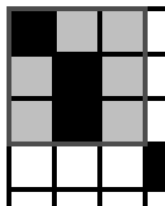
**ok**



inconsistent

**violated**

one alignment  
point outside



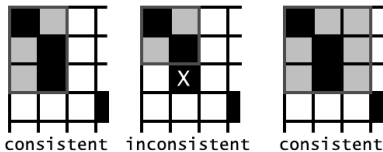
consistent

**ok**

unaligned  
word is fine



# Phrase extraction definition



A phrase pair  $(\bar{t}, \bar{s})$  is consistent with an alignment  $A$ , if all words  $s_1, \dots, s_m$  in  $\bar{s}$  that have alignment points in  $A$  have these with words  $t_1, \dots, t_n$  in  $\bar{t}$  and vice versa and at least one word in  $\bar{t}$  is aligned to at least one word in  $\bar{s}$



# Phrases extracted

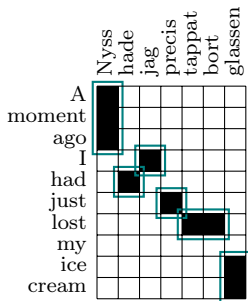
UPPSALA  
UNIVERSITET

	Nyss	hade	jag	precis	tappat	bort	glassen
A	■						
moment	■						
ago	■						
I			■				
had		■					
just				■			
lost					■	■	
my							
ice							■
cream							■



# Phrases extracted

a moment ago–nyss, I–jag, had–hade, just–precis  
lost–tappat bort, ice cream–glassen

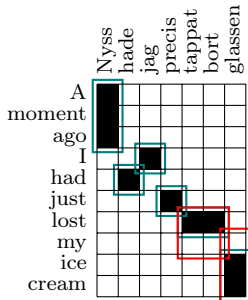




# Phrases extracted

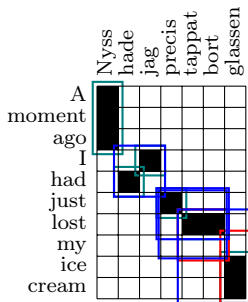
a moment ago–nyss, I–jag, had–hade, just–precis  
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen





# Phrases extracted



a moment ago–nyss, I–jag, had–hade, just–precis  
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

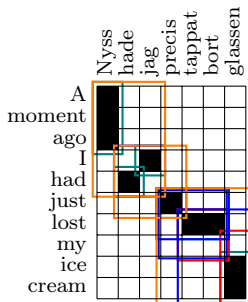
I had–hade jag, lost my ice cream–tappat bort glassen  
just lost–precis tappat bort, just lost my–precis tappat bort





# Phrases extracted

UPPSALA  
UNIVERSITET



a moment ago–nyss, I–jag, had–hade, just–precis  
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

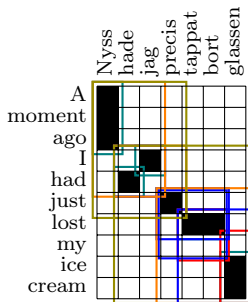
I had–hade jag, lost my ice cream–tappat bort glassen  
just lost–precis tappat bort, just lost my–precis tappat bort

a moment ago I had–nyss hade jag, I had just–hade jag precis  
just lost my ice cream–precis tappat bort glassen



# Phrases extracted

UPPSALA  
UNIVERSITET



a moment ago–nyss, I–jag, had–hade, just–precis  
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

I had–hade jag, lost my ice cream–tappat bort glassen  
just lost–precis tappat bort, just lost my–precis tappat bort

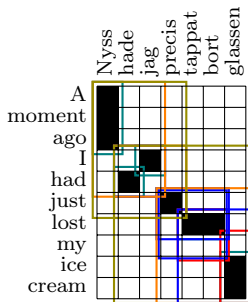
a moment ago I had–nyss hade jag, I had just–hade jag precis  
just lost my ice cream–precis tappat bort glassen

a moment ago I had just–nyss hade jag precis  
I had just lost my ice cream–hade jag precis tappat bort glassen



# Phrases extracted

UPPSALA  
UNIVERSITET



a moment ago–nyss, I–jag, had–hade, just–precis  
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

I had–hade jag, lost my ice cream–tappat bort glassen  
just lost–precis tappat bort, just lost my–precis tappat bort

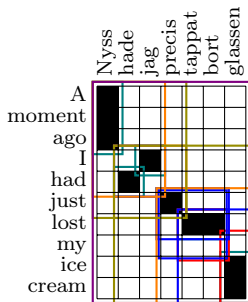
a moment ago I had–nyss hade jag, I had just–hade jag precis  
just lost my ice cream–precis tappat bort glassen

a moment ago I had just–nyss hade jag precis  
I had just lost my ice cream–hade jag precis tappat bort glassen

...



# Phrases extracted



a moment ago–nyss, I–jag, had–hade, just–precis  
lost–tappat bort, ice cream–glassen

lost my–tappat bort, my ice cream–glassen

I had–hade jag, lost my ice cream–tappat bort glassen  
just lost–precis tappat bort, just lost my–precis tappat bort

a moment ago I had–nyss hade jag, I had just–hade jag precis  
just lost my ice cream–precis tappat bort glassen

a moment ago I had just–nyss hade jag precis  
I had just lost my ice cream–hade jag precis tappat bort glassen

...

a moment ago I had just lost my ice cream–  
nyss hade jag precis tappat bort glassen



# Scoring phrase translations

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



# 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:

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



## Size of the phrase table

- Phrase translation table typically much bigger than corpus
- Limit the length of phrase pairs (often to 7 tokens)



# Size of the phrase table

- Phrase translation table typically much bigger than corpus
- Limit the length of phrase pairs (often to 7 tokens)
- Too big to store in memory?
  - Store on disk
  - Use smart data structures





# Size of the phrase table

- Phrase translation table typically much bigger than corpus
- Limit the length of phrase pairs (often to 7 tokens)
- Too big to store in memory?
  - Store on disk
  - Use smart data structures
- Prune phrase table – i.e., remove non-useful phrase pairs
  - Limit translation options for each phrase (often to 20–30)
  - Prune table based on statistics, such as  $\chi^2$



# Weighted models

- Described model consists of three sub-models:
  - Phrase translation models  $\phi(\bar{s}|\bar{t})$
  - Reordering model  $d$
  - Language model  $p_{LM}(t)$

$$t_{best} = \arg \max_t \prod_{i=1}^I \phi(\bar{s}_i|\bar{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1) \prod_{i=1}^{|t|} p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1})$$



# Weighted models

- Described model consists of three sub-models:
  - Phrase translation models  $\phi(\bar{s}|\bar{t})$
  - Reordering model  $d$
  - Language model  $p_{LM}(t)$

$$t_{best} = \arg \max_t \prod_{i=1}^I \phi(\bar{s}_i|\bar{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1) \prod_{i=1}^{|t|} p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1})$$

- Some sub-models may be more important than others



# Weighted models

- Described model consists of three sub-models:
  - Phrase translation models  $\phi(\bar{s}|\bar{t})$
  - Reordering model  $d$
  - Language model  $p_{LM}(t)$

$$t_{best} = \arg \max_t \prod_{i=1}^I \phi(\bar{s}_i|\bar{t}_i) d(\text{start}_i - \text{end}_{i-1} - 1) \prod_{i=1}^{|t|} p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1})$$

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



# Weighted models

- Described model consists of three sub-models:
  - Phrase translation models  $\phi(\bar{s}|\bar{t})$
  - Reordering model  $d$
  - Language model  $p_{LM}(t)$

$$t_{best} = \arg \max_t \prod_{i=1}^I \phi(\bar{s}_i|\bar{t}_i)^{\lambda_\phi} d(\text{start}_i - \text{end}_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|t|} p_{LM}(t_i|t_{i-(n-1)} \dots t_{i-1})^{\lambda_{LM}}$$

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



# Log-linear models

- Such a weighted model can be expressed as a log-linear model:

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

- Our feature functions:
  - three feature functions  $n = 3$
  - random variable  $x = (s, t, \text{start}, \text{end})$
  - feature function  $h_1 = \log \phi$
  - feature function  $h_2 = \log d$
  - feature function  $h_3 = \log p_{LM}$



# Weighted model as a log-linear model

$$p(t, a|s) = \exp(\lambda_\phi \sum_{i=1}^I \log \phi(\bar{s}_i | \bar{t}_i) + \\ \lambda_d \sum_{i=1}^I \log d(\text{start}_i - \text{end}_{i-1} - 1) + \\ \lambda_{LM} \sum_{i=1}^{|t|} \log p_{LM}(t_i | t_{i-(n-1)} \dots t_{i-1}))$$



# More feature functions

$$t^* = \arg \max_t \sum_i \lambda_i h_i(s, t)$$

- Easy and useful to add more feature functions





# More feature functions

$$t^* = \arg \max_t \sum_i \lambda_i h_i(s, t)$$

- Easy and useful to add more feature functions
  - Bidirectional alignment probabilities  $\phi(\bar{s}|\bar{t})$  and  $\phi(\bar{t}|\bar{s})$



# More feature functions

$$t^* = \arg \max_t \sum_i \lambda_i h_i(s, t)$$

- Easy and useful to add more feature functions
  - Bidirectional alignment probabilities  $\phi(\bar{s}|\bar{t})$  and  $\phi(\bar{t}|\bar{s})$
  - Lexical weighting of phrase pairs:



# More feature functions

$$t^* = \arg \max_t \sum_i \lambda_i h_i(s, t)$$

- Easy and useful to add more feature functions
  - Bidirectional alignment probabilities  $\phi(\bar{s}|\bar{t})$  and  $\phi(\bar{t}|\bar{s})$
  - Lexical weighting of phrase pairs:

$$\text{lex}(\bar{t}|\bar{s}, a) = \prod_{i=1}^{\text{length}(\bar{t})} \frac{1}{|\{j|(i, j) \in a\}|} \sum_{\forall (i, j) \in a} w(t_i|s_j)$$



# More feature functions

$$t^* = \arg \max_t \sum_i \lambda_i h_i(s, t)$$

- Easy and useful to add more feature functions
  - Bidirectional alignment probabilities  $\phi(\bar{s}|\bar{t})$  and  $\phi(\bar{t}|\bar{s})$
  - Lexical weighting of phrase pairs:

$$\text{lex}(\bar{t}|\bar{s}, a) = \prod_{i=1}^{\text{length}(\bar{t})} \frac{1}{|\{j|(i, j) \in a\}|} \sum_{\forall (i, j) \in a} w(t_i|s_j)$$

- Useful since rare phrase pairs have unreliable probability estimates



## More feature functions 2

- Language model has a bias towards short translations
  - word count:  $wc(t)$



## More feature functions 2

- Language model has a bias towards short translations
  - word count:  $wc(t)$
- We may prefer finer or coarser segmentations
  - phrase count:  $pc(t)$



## More feature functions 2

- Language model has a bias towards short translations
  - word count:  $wc(t)$
- We may prefer finer or coarser segmentations
  - phrase count:  $pc(t)$
- Lexicalized reordering models



## More feature functions 2

- Language model has a bias towards short translations
  - word count:  $wc(t)$
- We may prefer finer or coarser segmentations
  - phrase count:  $pc(t)$
- Lexicalized reordering models
- Multiple language models





# Tuning – Optimizing feature weights

- How do we learn the best weights  $\lambda_i$ ?
- Optimize the weights on a small corpus
- Called **Tuning**



# Tuning

- 1 Translate a development set using some initial  $\lambda_i$  and output a  $n$ -best list
- 2 Score the  $n$ -best list using some MT metric
- 3 Optimize  $\lambda_i$  so that translations with high metric scores get a high rank in the  $n$ -best list
- 4 Re-translate the development set with optimized  $\lambda_i$
- 5 Repeat step 2–4 until
  - No weight changes more than some small threshold
  - There are no new translations as a result of re-translating
  - You're fed up (after a maximum number of iterations)



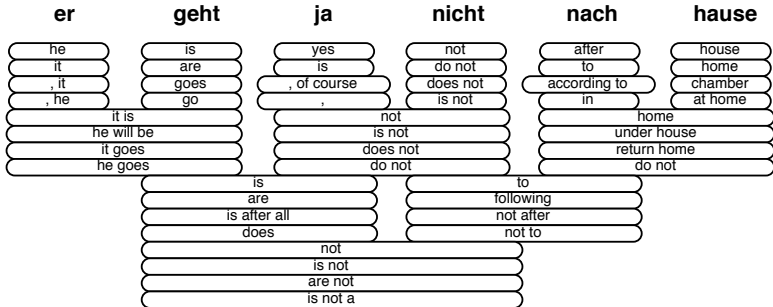
# Decoding

- Decoding is the process of using all these models and weights to actually perform translation
- Find the best translation among all possible translations

$$t^* = \arg \max_t \sum_i \lambda_i h_i(s, t)$$



# Translation Options





# Decoding by Hypothesis Expansion



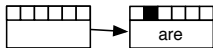


# Decoding by Hypothesis Expansion



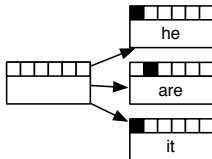


# Decoding by Hypothesis Expansion





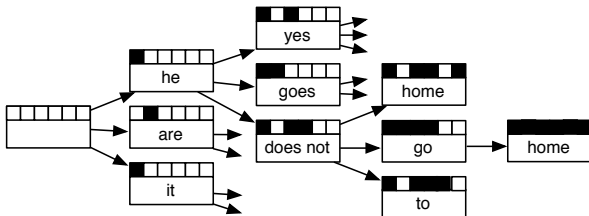
# Decoding by Hypothesis Expansion







# Decoding by Hypothesis Expansion





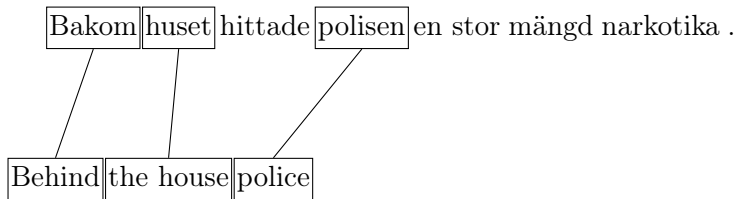
# Decoding complexity

Naively, in a sentence of  $N$  words with  $T$  translation options for each phrase, we can have

- $O(2^N)$  phrase segmentations,
- $O(T^N)$  sets of phrase translations, and
- $O(N!)$  word reordering permutations.

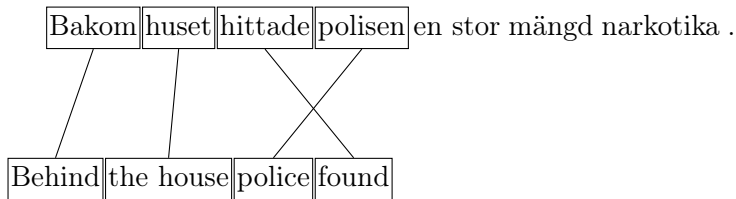


# Exploiting Model Locality



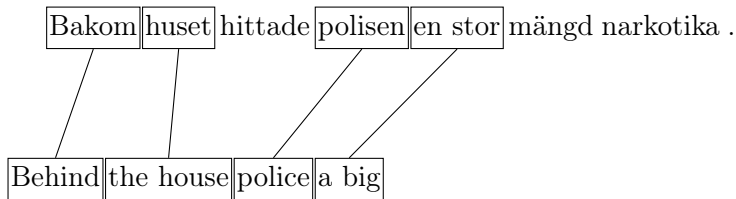


# Exploiting Model Locality



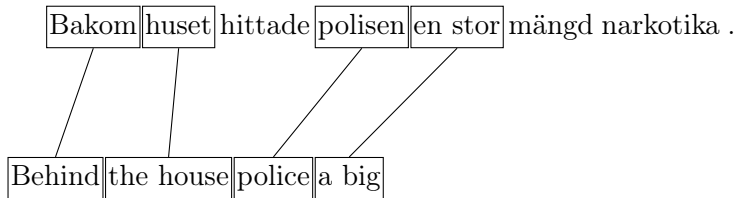


# Exploiting Model Locality





# Exploiting Model Locality

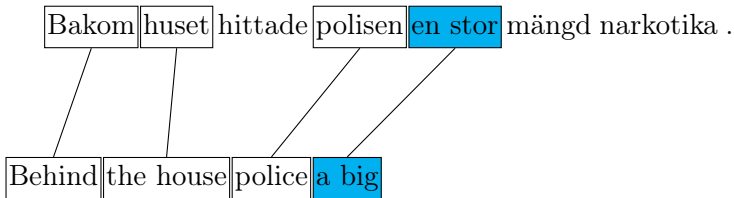


To score a new hypothesis, we need:

- the score of the previous hypothesis



# Exploiting Model Locality

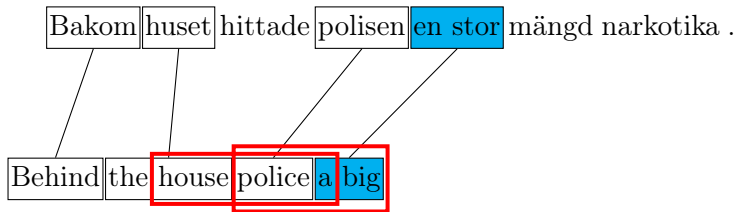


To score a new hypothesis, we need:

- the score of the previous hypothesis
- the translation model score



# Exploiting Model Locality



To score a new hypothesis, we need:

- the score of the previous hypothesis
- the translation model score
- the new language model scores





# Hypothesis recombination

- The translation model only looks at the current phrase.
- The  $n$ -gram model only looks at a window of  $n$  words.
- The choices the decoder makes are independent of everything beyond this window!
- The decoder never reconsiders its choices once they've moved out of the  $n$ -gram history.



# Hypothesis recombination

Suppose we have these hypotheses with the same coverage, and we use a trigram language model:

After the house police      Score = -12.5

Behind the house police      Score = -11.2

, the house police      Score = -22.0



# Hypothesis recombination

Suppose we have these hypotheses with the same coverage, and we use a trigram language model:

~~After the house police~~      ~~Score = -12.5~~

Behind the house police      Score = -11.2

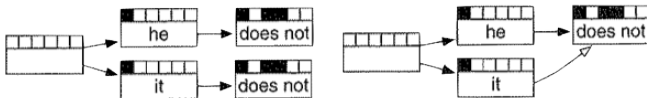
~~, the house police~~      ~~Score = -22.0~~

- We already know the winner!
- We can discard the competing hypotheses.



# Hypothesis recombination

- Hypothesis recombination combines branches in the search graph:



- It's a form of dynamic programming.
- Recombination reduces the search space substantially...
- ...it preserves search optimality...
- ...but decoding is still exponential!

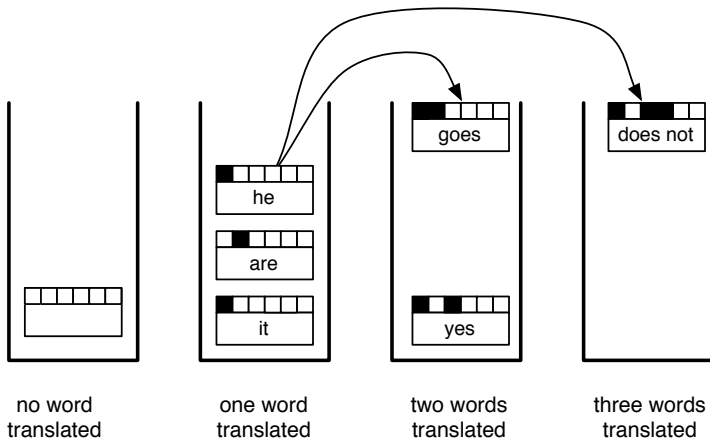


# Pruning

- To make decoding really efficient, we expand only hypotheses that look promising.
- Bad hypotheses should be *pruned* early to avoid wasting time on them.
- Pruning compromises search optimality!



# Stack decoding

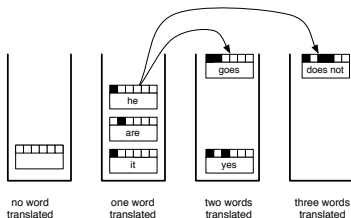




# Stack decoding algorithm

UPPSALA  
UNIVERSITET

- 1: AddToStack( $s_0, h_0$ )
- 2: **for**  $i = 0 \dots N - 1$  **do**
- 3:     **for all**  $h \in s_i$  **do**
- 4:         **for all**  $t \in T$  **do**
- 5:             **if** Applicable( $h, t$ ) **then**
- 6:                  $h' \leftarrow \text{Expand}(h, t)$
- 7:                  $j \leftarrow \text{WordsCovered}(h) + \text{WordsCovered}(t)$
- 8:                 AddToStack( $s_j, h'$ )
- 9:             **end if**
- 10:         **end for**
- 11:     **end for**
- 12: **end for**
- 13: **return** best hypothesis on stack  $s_N$





## AddToStack( $s, h$ )

- 1: **for all**  $h' \in s$  **do**
- 2:   **if**  $\text{Recombinable}(h, h')$  **then**
- 3:     add higher-scoring of  $h, h'$  to stack  $s$ , discard other
- 4:     **return**
- 5:   **end if**
- 6: **end for**
- 7: add  $h$  to stack  $s$
- 8: **if** stack too large **then**
- 9:   prune stack
- 10: **end if**





# How to prune

## Histogram pruning

Keep no more than  $S$  hypotheses per stack.

*Parameter:* Stack size  $S$

## Threshold pruning

Discard hypotheses whose score is very low compared to that of the best hypothesis on the stack  $h^*$ :

$$\text{Score}(h) < \eta \cdot \text{Score}(h^*)$$

*Parameter:* Threshold  $\eta$



# Beam search: Complexity

- For each of the  $N$  words in the input sentence,
- expand  $S$  hypotheses
- by considering  $T$  translation options each:

$$O(S \cdot N \cdot T)$$

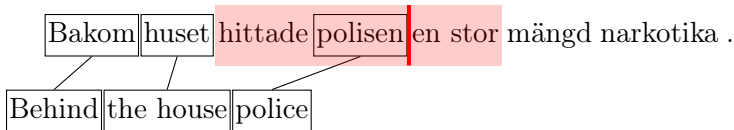
The number of translation options is linear in the sentence length:

$$O(S \cdot N^2)$$



# Distortion limit

- When translating between closely related languages, most reorderings are local...
- ...and anyhow, we haven't got any reasonable models for long-range reordering!
- If we impose a limit on reordering, the number of translation options to consider at each step is bounded by a constant.





# Distortion limit

- When translating between closely related languages, most reorderings are local...
- ...and anyhow, we haven't got any reasonable models for long-range reordering!
- If we impose a limit on reordering, the number of translation options to consider at each step is bounded by a constant.

The number of hypotheses expanded by a beam search decoder with limited reordering is linear in the stack size and the input size:

$$O(S \cdot N)$$



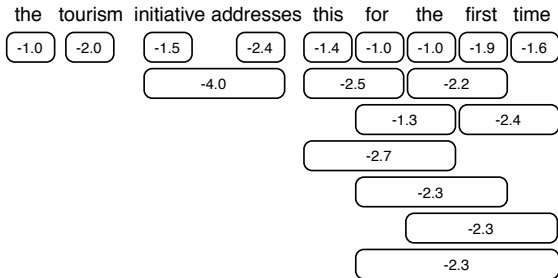
# Incremental scoring and cherry picking

- The path that looks cheapest necessarily incurs a much higher cost later.
- Pruning may discard better options before this is recognised.
- To make scores more comparable, we should take into account unavoidable future costs.
- Compare hypotheses based on current score + future score.



# Future cost estimation

- Calculating the future cost exactly would amount to full decoding!
- Cheaper approximations can be computed by making additional independence assumptions.
  - Assume independence between models.
  - Ignore LM history across phrase boundaries.





# DP Beam Search Decoding: Evaluation

- DP beam search is by far the most popular search algorithm for phrase-based SMT.
- It combines high speed with reasonable accuracy by exploiting the constraints of the standard models.
- It works well with very local models.
  - Sentence-internal long-range dependencies increase search errors by inhibiting recombination.
  - No cross-sentence dependencies on the target side.
- Current state of the art in SMT: Good local fluency, but serious problems with long-range reordering and discourse-level phenomena.



- SMT toolkit
- Free, open source
- Implements several models:
  - Phrase-based
  - Hierarchical
  - Syntax-based
- Decoding
- Training pipeline
  - Training translation models
  - Training language models
  - Optimizing feature weights





# Moses training pipeline

- 1 Prepare data
- 2 Run GIZA to create one-way alignments
- 3 Symmetrize alignment
- 4 Calculate lexical translation probabilities
- 5 Extract phrases
- 6 Score phrases
- 7 Train reordering model
- 8 (Train generation model)
- 9 Create configuration file



# Moses training pipeline

- Preprocess data (tokenization, casing, et.c.)
  
- 1 Prepare data
- 2 Run GIZA to create one-way alignments
- 3 Symmetrize alignment
- 4 Calculate lexical translation probabilities
- 5 Extract phrases
- 6 Score phrases
- 7 Train reordering model
- 8 (Train generation model)
- 9 Create configuration file
  
- Tune the system



# Coming up

- This week:
  - Assignment 2: Moses
- Coming weeks: NMT with Gongbo