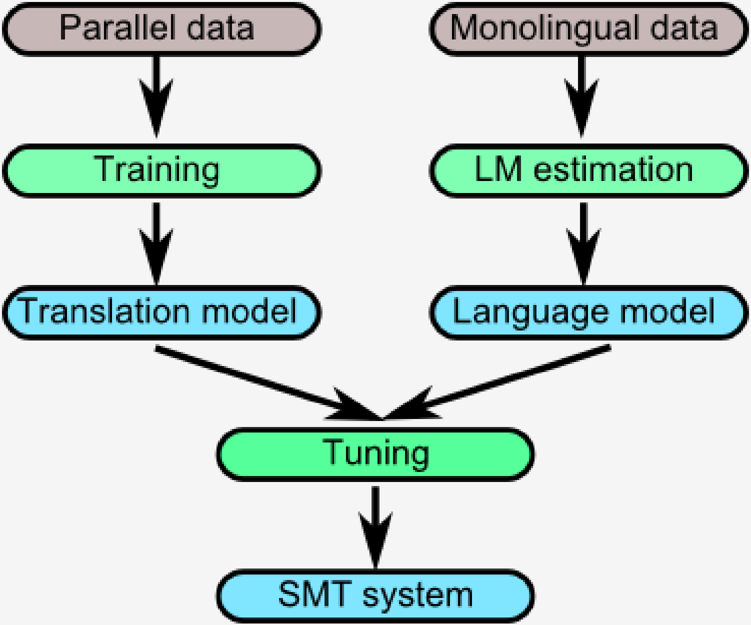
Statistical Machine Translation

-phrase based smt-

Artem Sokolov
Computerlinguistik
Universität Heidelberg
Sommersemester 2015

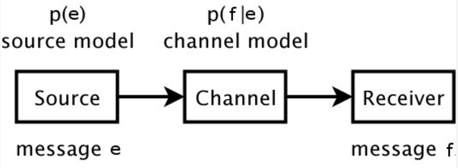
material from P. Koehn


Parallel data
Monolingual data
Training
LM estimation
Translation model
Language model
Tuning
SMT system

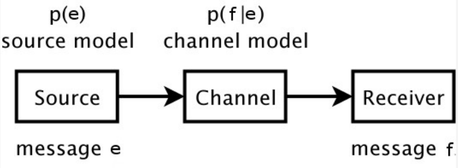
Every time I fire a linguist, the performance goes up.

Frederic Jelinek, IBM 1988

$$\arg\max_e p(e|f)$$

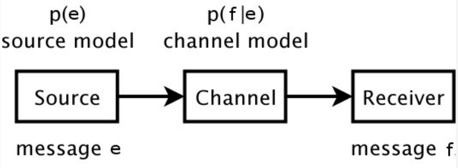

p(e)
source model
p(f|e)
channel model
Source
Channel
Receiver
message e
message f

$$\arg\max_{e} p(e|f) = \arg\max_{e} p(f|e)p(e)$$

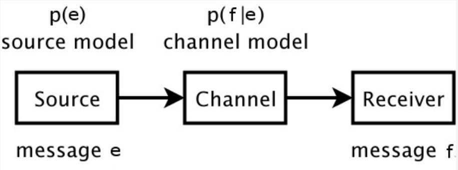


Noisy Channel Model

$$\arg\max_e p(e|f) = \arg\max_e \underbrace{p(f|e)}_{\text{transl. model}} \; \underbrace{p(e)}_{\text{lang. model}}$$

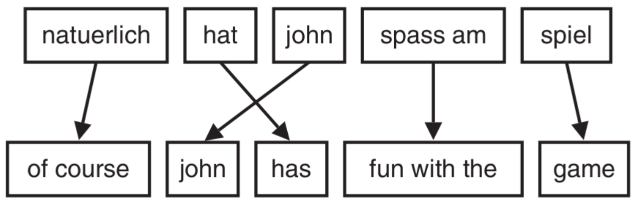


$$\arg\max_e p(e|f) = \arg\max_e \underbrace{p(f|e)}_{\text{transl. model}} \underbrace{p(e)}_{\text{lang. model}}$$



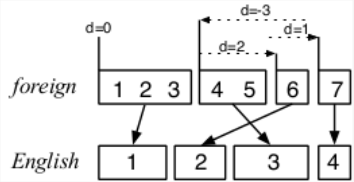
SMT was born from automatic speech recognition:

- $p(e)$ = language model
- $p(f|e)$ = acoustic model
- however, SMT must deal with word reordering!



1. input is segmented into phrases
   (not necessarily linguistically motivated)
2. translated one-to-one into phrases in Engish
3. possibly reordered

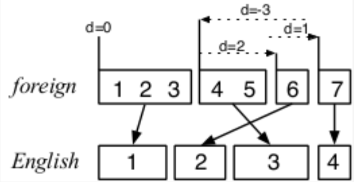
$2 + 3 \rightarrow$ would become the translation model from the noisy channel model



$$d(\mathsf{start}_i - \mathsf{end}_{i-1} - 1)$$

start$_i$ – position of the **first** word of a **foreign** phrase corresponding to the **i$^{\mathbf{th}}$** **English** phrase.

end$_{i-1}$ – position of the **last** word of a **foreign** phrase corresponding to the **previous English** phrase.



$$d(\text{start}_i - \text{end}_{i-1} - 1)$$

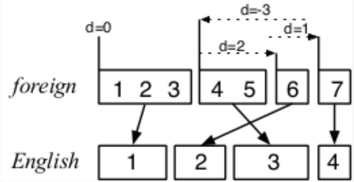
start_i – position of the **first** word of a **foreign** phrase corresponding to the **i$^{\text{th}}$ English** phrase.

end_{i-1} – position of the **last** word of a **foreign** phrase corresponding to the **previous English** phrase.

phrase	translates	movement	calculation
2	6	skip over 4–5	
3	4–5	move back over 4–6	

Example (phrase 2):

$d(\text{start}_i - \text{end}_{i-1} - 1) = \qquad\qquad =$



$$d(\text{start}_i - \text{end}_{i-1} - 1)$$

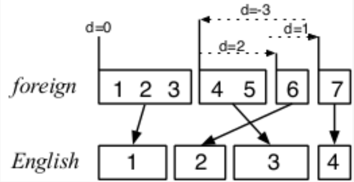
start_i – position of the **first** word of a **foreign** phrase corresponding to the **i$^{\text{th}}$** **English** phrase.

end_{i-1} – position of the **last** word of a **foreign** phrase corresponding to the **previous English** phrase.

phrase	translates	movement	calculation
2	6	skip over 4–5	
3	4–5	move back over 4–6	

Example (phrase 2):

$d(\text{start}_i - \text{end}_{i-1} - 1) = d(\text{start}_2 - \text{end}_1 - 1) =$



$$d(\text{start}_i - \text{end}_{i-1} - 1)$$

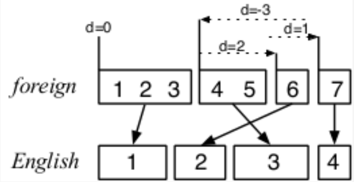
start_i – position of the **first** word of a **foreign** phrase corresponding to the **i**$^{\textbf{th}}$ **English** phrase.

end_{i-1} – position of the **last** word of a **foreign** phrase corresponding to the **previous English** phrase.

phrase	translates	movement	calculation
2	6	skip over 4–5	6-3-1=2
3	4–5	move back over 4–6	

Example (phrase 2):

$d(\text{start}_i - \text{end}_{i-1} - 1) = d(\text{start}_2 - \text{end}_1 - 1) = d(6 - 3 - 1) = 2$



$$d(\text{start}_i - \text{end}_{i-1} - 1)$$

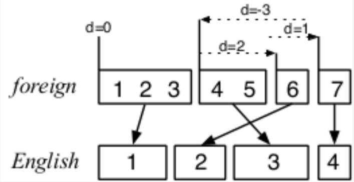
start_i – position of the **first** word of a **foreign** phrase corresponding to the **i^th^** **English** phrase.

end_{i-1} – position of the **last** word of a **foreign** phrase corresponding to the **previous English** phrase.

phrase	translates	movement	calculation
2	6	skip over 4–5	6-3-1=2
3	4–5	move back over 4–6	

Example (phrase 3):

$d(\text{start}_i - \text{end}_{i-1} - 1) =$ $\qquad\qquad =$



$$d(\text{start}_i - \text{end}_{i-1} - 1)$$

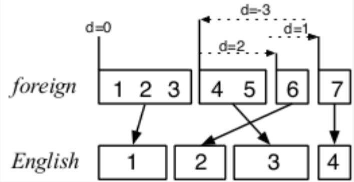
start_i – position of the **first** word of a **foreign** phrase corresponding to the **i^th^** **English** phrase.

end_{i-1} – position of the **last** word of a **foreign** phrase corresponding to the **previous English** phrase.

phrase	translates	movement	calculation
2	6	skip over 4–5	6-3-1=2
3	4–5	move back over 4–6	

Example (phrase 3):

$d(\text{start}_i - \text{end}_{i-1} - 1) = d(\text{start}_3 - \text{end}_2 - 1) =$



$$d(\mathsf{start}_i - \mathsf{end}_{i-1} - 1)$$

start_i – position of the **first** word of a **foreign** phrase corresponding to the **i$^{\text{th}}$ English** phrase.

end_{i-1} – position of the **last** word of a **foreign** phrase corresponding to the **previous English** phrase.

phrase	**translates**	**movement**	**calculation**
2	6	skip over 4–5	6-3-1=2
3	4–5	move back over 4–6	4-6-1=-3

Example (phrase 3):

$d(\mathsf{start}_i - \mathsf{end}_{i-1} - 1) = d(\mathsf{start}_3 - \mathsf{end}_2 - 1) = d(4 - 6 - 1) = -3$

$$\arg\max_e p(e|f) = \arg\max_e p(f|e)p(e)$$

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

$$\arg\max_e p(e|f) = \arg\max_e p(f|e)p(e)$$

$$\arg\max_e \prod_i^I \phi(\bar{f}_i|\bar{e_i})^{\lambda\phi}$$
$$\cdot\, d(\mathsf{start}_i - \mathsf{end}_{i-1} - 1)^{\lambda d}$$
$$\cdot \prod_{j=1}^{|e|} p_{\mathsf{LM}}(e_j|e_1,\ldots,e_{j-1})^{\lambda_{\mathsf{LM}}}$$

$$\arg\max_e p(e|f) = \arg\max_e p(f|e)p(e)$$

$$\arg\max_e \prod_i^I \phi(\bar{f}_i|\bar{e_i})^{\lambda\phi}$$

$$\cdot\, d(\mathsf{start}_i - \mathsf{end}_{i-1} - 1)^{\lambda d}$$

$$\cdot \prod_{j=1}^{|e|} p_{\mathsf{LM}}(e_j|e_1, \ldots, e_{j-1})^{\lambda_{\mathsf{LM}}}$$

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

$$\lambda_i = \text{parameter}$$

$$h_i = \text{features}$$

$$
\begin{aligned}
&\arg\max_e \ \exp(\lambda_\phi \sum_i^I \log \phi(\bar{f}_i|\bar{e}_i) \\
&+ \lambda_d \sum_i^I \log d(\mathsf{start}_i - \mathsf{end}_{i-1} - 1) \\
&+ \lambda_{\mathsf{LM}} \sum_{j=1}^{|e|} \log p_{\mathsf{LM}}(e_j|e_1, \ldots, e_{j-1}))
\end{aligned}
$$

- Word count:
  $\mathsf{wc}(e) = \log |e|^{\omega}, \begin{array}{l}\omega{<}1 \text{ prefers fewer words} \\ \omega{>}1 \text{ prefers more words}\end{array}$

  Corrects the bias of the language model towards short translations.
- Phrase count:
  $\mathsf{pc}(e) = \log |I|^{\rho}, \begin{array}{c} I{=}\text{number of phrases} \\ \rho{<}1 \text{ prefers fewer phrases, i.e., longer phrase} \\ \rho{>}1 \text{ prefers shorter phrases, i.e., more phrases}\end{array}$
  Fine-tunes fine or coarse phrase segmentation (trade-off)

  Choosing more linguistically motivated boundaries was not shown to be especially helpful.

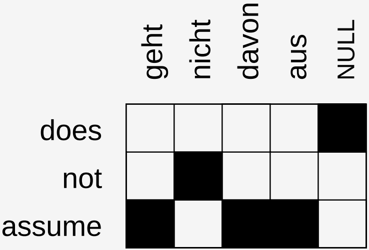
- Multiple language models
- Multiple translation models:
  e.g., src-trg and trg-src translation models
- Bidirectional alignment probabilities

- Lexically weighted phrase translation probabilities:

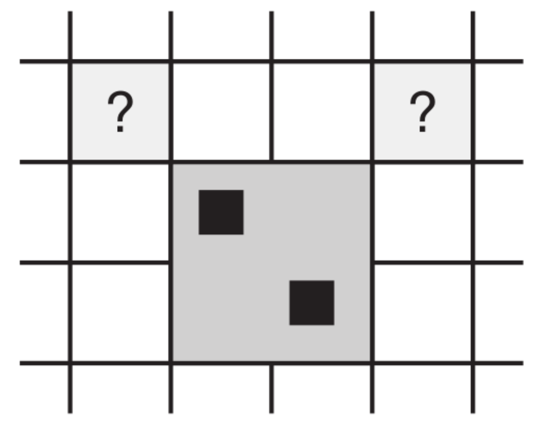
$$\text{lex}(\bar{e}|\bar{f}, a) = \prod_{i=1}^{|\bar{e}|} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall (i,j) \in a} w(e_i|f_j) \tag{1}$$

IBM1-style: $e_i \in \bar{e}$ is generated independently by an aligned $f_j \in \bar{f}$ with the word translation probability $w(e_i|f_j)$ or average if multiple alignment is possible

Again, we can use both $\text{lex}(\bar{e}|\bar{f})$ and $\text{lex}(\bar{f}|\bar{e})$.



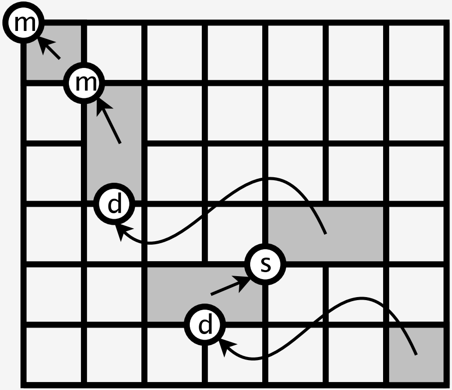
$$\textbf{Example:}\ \ \mathrm{lex}(\bar{e}|\bar{f}, a) = w(\text{does}|\text{NULL}) \cdot w(\text{not}|\text{nicht}) \cdot \frac{1}{3}\big(w(\text{assume}|\text{geht}) + w(\text{assume}|\text{davon}) + w(\text{assume}|\text{aus})\big)$$



$$\text{orientation} \in \{\text{monotone}, \text{swap}, \text{discontinuous}\}$$

During phrase extraction from alignment, check:

- if a word alignment point to the top left exists ⇒ **monotone** (m)
- elsif a word alignment point to the top right exists ⇒ **swap** (s)
- else ⇒ **discontinuous** (d)



Estimation of lexical reordering probability:

unsmoothed estimate:

$$\hat{p}(\text{orientation}|\bar{e}, \bar{f}) = \frac{\text{count}(\text{orientation}, \bar{e}, \bar{f})}{\sum_o \text{count}(\text{orientation}, \bar{e}, \bar{f})}$$

smoothed estimate:

$$\hat{p}(\text{orientation}|\bar{e}, \bar{f}) = \frac{\lambda p(\text{orientation}) + \text{count}(\text{orientation}, \bar{e}, \bar{f})}{\lambda \cdot 1 + \sum_o \text{count}(\text{orientation}, \bar{e}, \bar{f})}$$

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

$\Rightarrow$ linear interpolation with unlexicalized orientation model.