Statistical Machine Translation: Higher IBM Models (Part II.5)

Jakub Waszczuk

Heinrich Heine Universität Düsseldorf

Winter Semester 2018/19

Outline

IBM 3, 4 and 5: Generative Story

Generative story, adapted from [Schoenemann, 2013]

1. For $i = 1, 2, \ldots, n$, decide on the number ϕ_i of English words aligned to f_i (fertility). Choose with probability $P(\phi_i \mid e_i)$.
2. Choose the number ϕ_0 of unaligned words in the (still unknown) foreign sequence. Choose with probability $P(\phi_0 \mid \sum_{i=1}^{n} \phi_i)$. Since each English word belongs to exactly one foreign position (including 0), the Engish sequence is now known to be of length

$$m = \sum_{i=0}^{n} \phi_i \tag{1}$$

3. For each $i = 0, 1, \ldots, n$ and $k = 1, \ldots, \phi_i$ decide on:
   (a) The identity $e_{i,k}$ of the next English word aligned to f_i. Choose with probability $P(e_{i,k} \mid f_i)$.
   (b) The position $d_{i,k}$ of the just generated English word $e_{i,j}$ with probability

$$P(d_{i,k} \mid J) \tag{2}$$

   where J represents information generated so far and differs in the individual IBM models.

Note: the models IBM 3, 4, and 5 differ only in how the step 3(b) is implemented.

IBM 4

Preliminaries

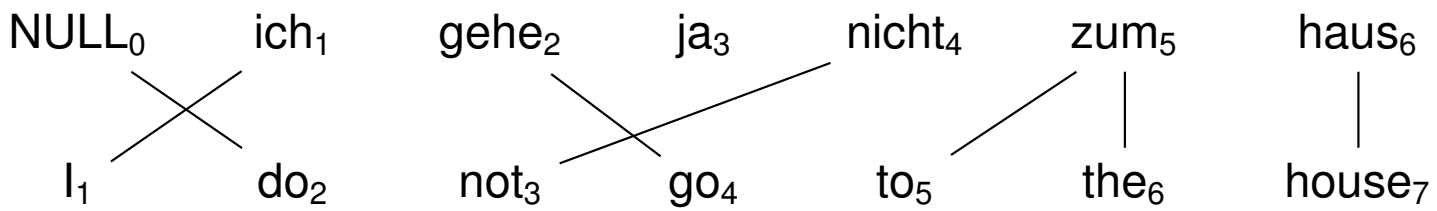
- Given input position i, we define $\triangleleft i$ as the closest preceding position $j > 0$ that has aligned output words ($\triangleleft i := 0$ if no such positions):

$$\triangleleft i = \max \{j : 0 < j < i, \phi_j > 0\} \cup \{0\} \tag{3}$$

- We also define the *center position* $\odot_i$ as the rounded average of the output positions aligned to input position i:

$$\odot_i = \begin{cases} \lceil \text{avg}\{d_{i,k} : 1 < k \leq \phi_i\} \rceil & \text{if } i > 0 \wedge \phi_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Example



- $\triangleleft 1 = 0$
- $\triangleleft 2 = 1$
- $\triangleleft 3 = 2$
- $\triangleleft 4 = 2$
- $\odot_1 = 1$
- $\odot_5 = 6$

IBM 4

Distortion

- For a given position i in the input sentence, the corresponding output positions are generated in an ascending order (i.e., for $1 < k \leq \phi_i$ we have $d_{i,k} > d_{i,k-1}$) $\Longrightarrow$
  - One-to-one correspondence between (well-formed) distortions and alignments
  - Factor $\prod_{i=1}^{n} \phi_i!$ no longer required
- The distortion probability for the words aligned to NULL ($i = 0$, $k \geq 1$):

$$P(d_{0,k} = j \mid J) = \frac{1}{m} \tag{5}$$

- The distortion probability for the first aligned word ($i > 0$, $k = 1$):

$$P(d_{i,1} = j \mid J) = P_{=1}(j \mid \odot_{\triangleleft i}) \tag{6}$$

$$= P_{=1}(j - \odot_{\triangleleft i}) \tag{7}$$

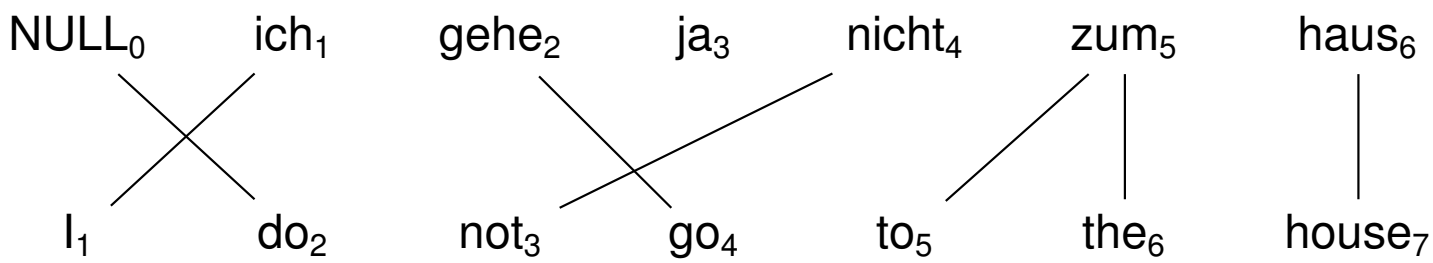
- The distortion probability for the subsequent aligned words ($i > 0$, $k > 1$):

$$P(d_{i,k} = j \mid J) = P_{>1}(j \mid d_{i,k-1}) \tag{8}$$

$$= P_{>1}(j - d_{i,k-1}) \tag{9}$$

IBM 4: Example

Target alignment



Positioning process

f_i	$d_{i,k}$	I_1	do_2	not_3	go_4	to_5	the_6	$house_7$	j	$\odot_{\triangleleft i}$	$P(d_{i,k} \mid J)$
$NULL_0$	$d_{0,1}$		x						2	-	1/7
ich_1	$d_{1,1}$	x							1	0	$P_{=1}(1)$
$gehe_2$	$d_{2,1}$				x				4	1	$P_{=1}(3)$
$nicht_4$	$d_{4,1}$			x					3	4	$P_{=1}(-1)$
zum_5	$d_{5,1}$					x			5	3	$P_{=1}(2)$
	$d_{5,2}$						x		6	-	$P_{>1}(1)$
$haus_6$	$d_{6,1}$							x	7	6	$P_{=1}(1)$

IBM 4

Deficiency

IBM 4 is deficient:

- Not only can it place two words on the same output position
- It can also place the words outside of the boundaries of the output sentence

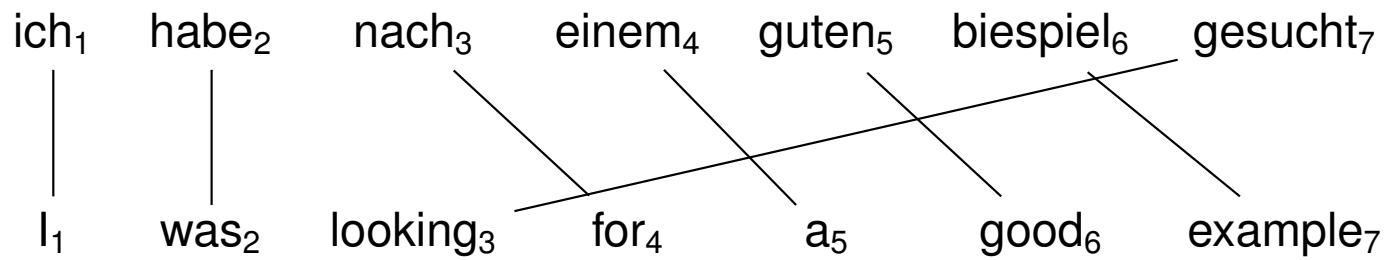
Example

f_i	$d_{i,k}$	I_1	do_2	not_3	go_4	to_5	6	$house_7$	the_8	j	$\odot_{\triangleleft i}$	$P(d_{i,k} \mid J)$
$NULL_0$	$d_{0,1}$		x							2	-	1/7
ich_1	$d_{1,1}$	x								1	0	$P_{=1}(1)$
$gehe_2$	$d_{2,1}$				x					4	1	$P_{=1}(3)$
$nicht_4$	$d_{4,1}$			x						3	4	$P_{=1}(-1)$
zum_5	$d_{5,1}$					x				8	3	$P_{=1}(2)$
	$d_{5,2}$								x	6	-	$P_{>1}(3)$
$haus_6$	$d_{6,1}$							x		7	6	$P_{=1}(-1)$

- Jumping from *to*$_5$ to *the*$_8$ is possible provided that $P_{=1}(3) > 0$

IBM 4 vs IBM 3

Example



Distortion probabilities factored in depending on the model:

- IBM 3 ($P(k \mid k)$ optimal):

$$P(1 \mid 1) \times P(2 \mid 2) \times P(3 \mid 7) \times P(4 \mid 3) \times P(5 \mid 4) \times P(6 \mid 5) \times P(7 \mid 6)$$

  Penalty incurred 5 times: everything apart from $P(1 \mid 1)$ and $P(2 \mid 2)$

- IBM 4 ($P(1)$ optimal):

$$P(1) \times P(1) \times P(2) \times P(1) \times P(1) \times P(1) \times P(-4)$$

  Penalty incurred twice: $P(2)$ and $P(-4)$

IBM 5

Positioning Strategy in IBM 5

We have to pick the position $d_{i,k}$ to place the next English word $e_{i,k}$ aligned to f_i.

- We say that an output position $i \in \{1, \ldots, m\}$ is an *open slot* if no words have been placed on this position so far
- We define v_j as the number of remaining open slots up to the output position j
- We define $v_{\max}$ as the total number of remaining open slots
- We pick one of the remaning slots and move on

Note: v_j and $v_{\max}$ are specific to a particular $d_{i,k}$

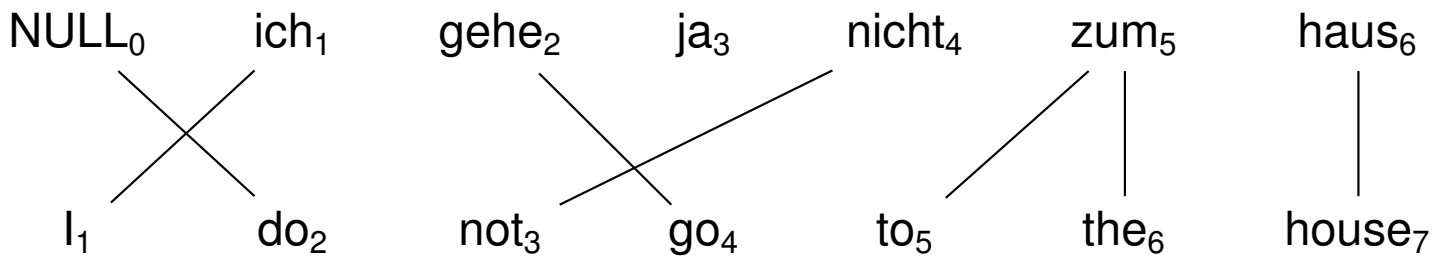
Deficiency

Since we only consider the open slots, it is not possible to place two words on a single output position twice (the first time a word is placed on this position, it becomes closed).

Thanks to that, IBM 5 is **not deficient**.

IBM 5: Example

Target alignment

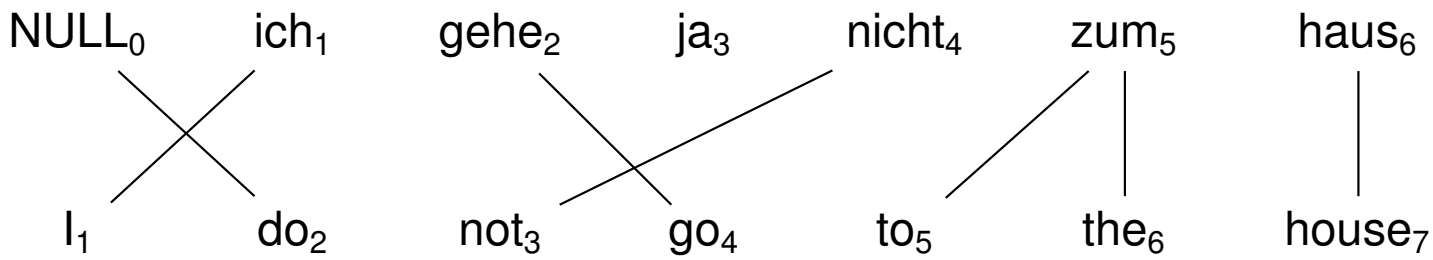


Positioning process

f_i	$d_{i,k}$	v_1 I$_1$	v_2 do$_2$	v_3 not$_3$	v_4 go$_4$	v_5 to$_5$	v_6 the$_6$	v_7 house$_7$	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-		

IBM 5: Example

Target alignment

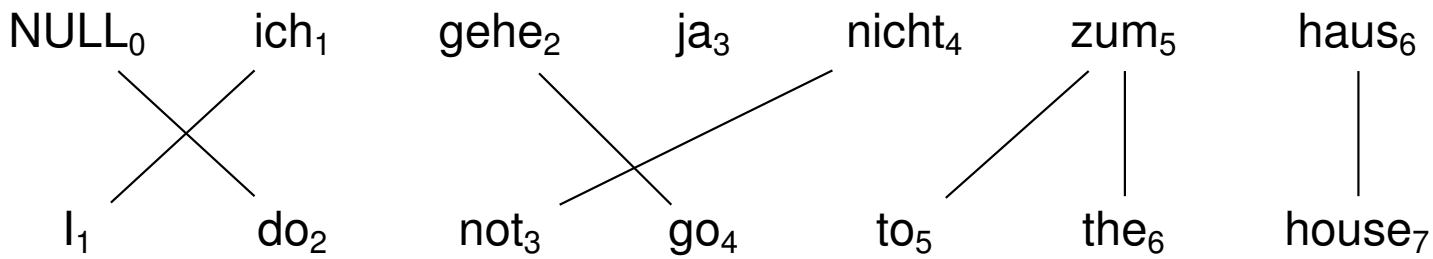


Positioning process

f_i	$d_{i,k}$	v_1 I$_1$	v_2 do$_2$	v_3 not$_3$	v_4 go$_4$	v_5 to$_5$	v_6 the$_6$	v_7 house$_7$	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0		

IBM 5: Example

Target alignment

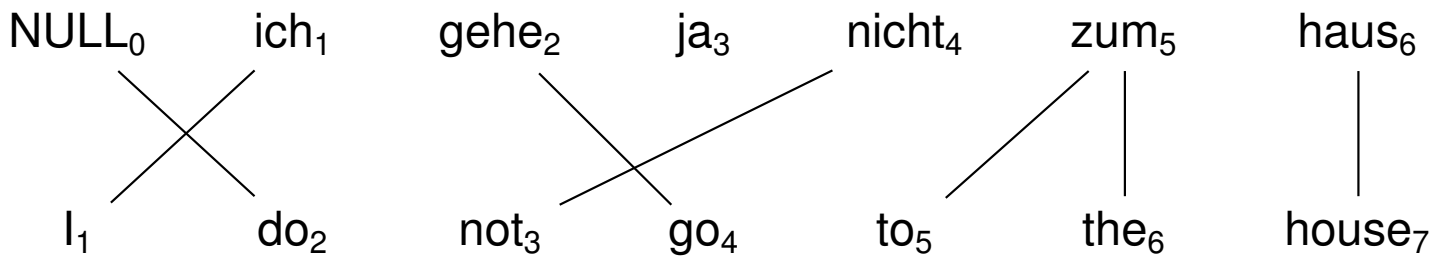


Positioning process

f_i	$d_{i,k}$	v_1 I$_1$	v_2 do$_2$	v_3 not$_3$	v_4 go$_4$	v_5 to$_5$	v_6 the$_6$	v_7 house$_7$	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1
gehe$_2$	$d_{2,1}$	0	0	1	2	3	4	5	4	1		

IBM 5: Example

Target alignment

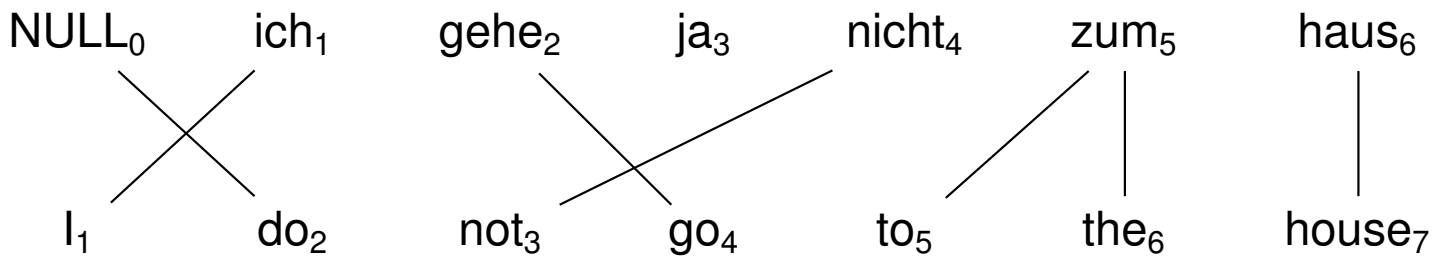


Positioning process

f_i	$d_{i,k}$	v_1 I$_1$	v_2 do$_2$	v_3 not$_3$	v_4 go$_4$	v_5 to$_5$	v_6 the$_6$	v_7 house$_7$	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1
gehe$_2$	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2
nicht$_4$	$d_{4,1}$	0	0	1	1	2	3	4	3	4		

IBM 5: Example

Target alignment

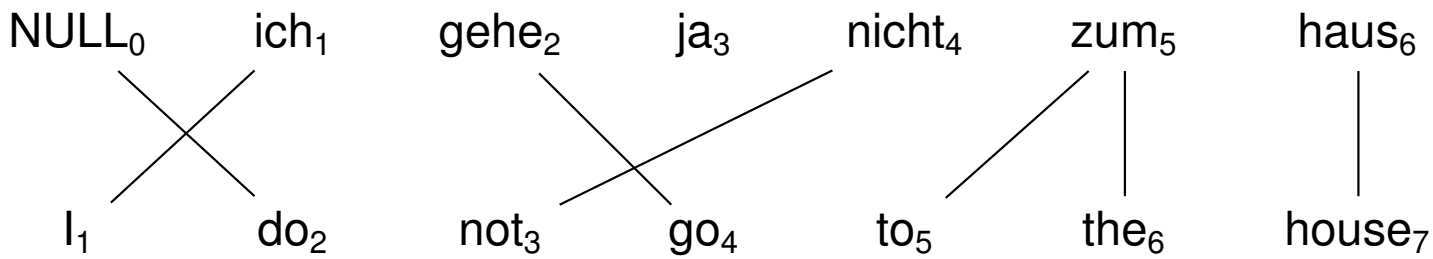


Positioning process

f_i	$d_{i,k}$	v_1 I$_1$	v_2 do$_2$	v_3 not$_3$	v_4 go$_4$	v_5 to$_5$	v_6 the$_6$	v_7 house$_7$	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1
gehe$_2$	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2
nicht$_4$	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1
zum$_5$	$d_{5,1}$	0	0	0	0	1	2	3	5	3		

IBM 5: Example

Target alignment

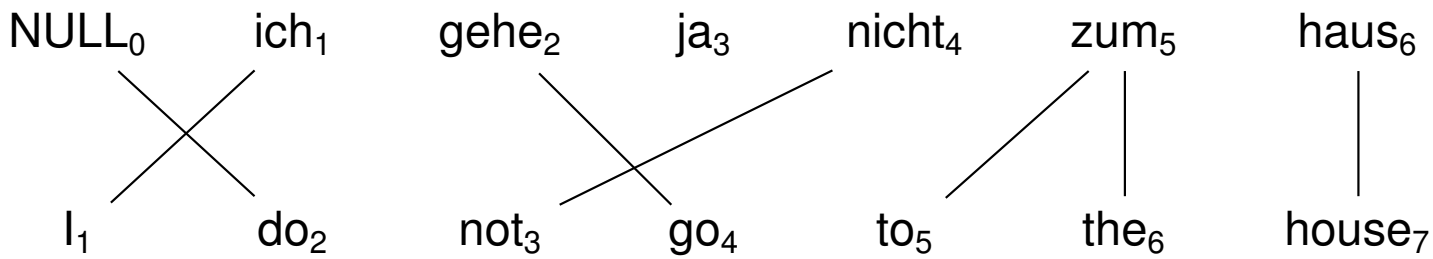


Positioning process

f_i	$d_{i,k}$	v_1 I$_1$	v_2 do$_2$	v_3 not$_3$	v_4 go$_4$	v_5 to$_5$	v_6 the$_6$	v_7 house$_7$	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1
gehe$_2$	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2
nicht$_4$	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1
zum$_5$	$d_{5,1}$	0	0	0	0	1	2	3	5	3	2	1
	$d_{5,2}$	0	0	0	0	0	1	2	6	-		

IBM 5: Example

Target alignment

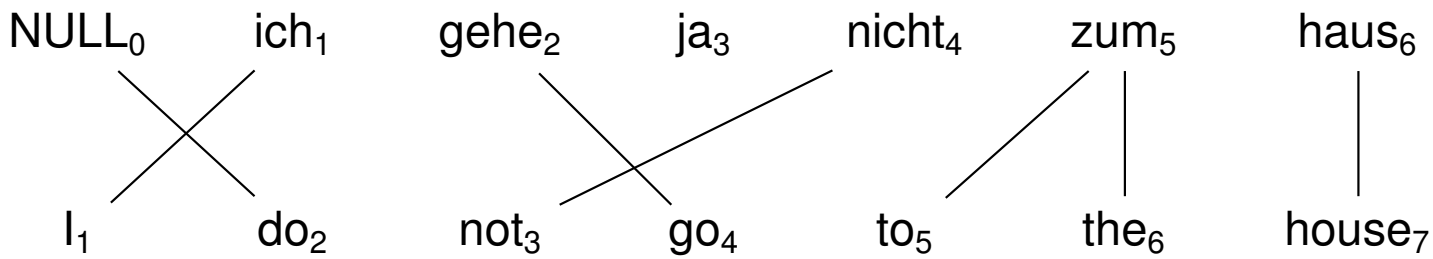


Positioning process

f_i	$d_{i,k}$	v_1 I$_1$	v_2 do$_2$	v_3 not$_3$	v_4 go$_4$	v_5 to$_5$	v_6 the$_6$	v_7 house$_7$	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1
gehe$_2$	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2
nicht$_4$	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1
zum$_5$	$d_{5,1}$	0	0	0	0	1	2	3	5	3	2	1
	$d_{5,2}$	0	0	0	0	0	1	2	6	-	2	1
haus$_6$	$d_{6,1}$	0	0	0	0	0	0	1	7	6		

IBM 5: Example

Target alignment



Positioning process

f_i	$d_{i,k}$	v_1 I$_1$	v_2 do$_2$	v_3 not$_3$	v_4 go$_4$	v_5 to$_5$	v_6 the$_6$	v_7 house$_7$	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1
gehe$_2$	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2
nicht$_4$	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1
zum$_5$	$d_{5,1}$	0	0	0	0	1	2	3	5	3	2	1
	$d_{5,2}$	0	0	0	0	0	1	2	6	-	2	1
haus$_6$	$d_{6,1}$	0	0	0	0	0	0	1	7	6	1	1

IBM 5

Distortion

- For a given position i in the input sentence, the corresponding output positions are generated in an ascending order (i.e., for $1 < k \leq \phi_i$ we have $d_{i,k} > d_{i,k-1}$)
- The distortion probability for the words aligned to NULL ($i = 0$, $k \geq 1$):

$$P(d_{0,k} = j \mid J) = \frac{1}{v_{\max}} \tag{10}$$

- The distortion probability for the first aligned word ($i > 0$, $k = 1$):

$$P(d_{i,1} = j \mid J) = P_{=1}(v_j \mid v_{\odot_{\lhd i}}, v_{\max}) \tag{11}$$

- The distortion probability for the subsequent aligned words ($i > 0$, $k > 1$):

$$P(d_{i,k} = j \mid J) = P_{>1}(v_j \mid v_{d_{i,k-1}}, v_{\max}) \tag{12}$$

IBM 5 vs IBM 4

In theory

Cost to pay for non-deficiency: more distortion parameters in IBM 5. For instance:

$$P_{>1}(1 \mid 1, 10) \neq P_{>1}(1 \mid 1, 11)$$

In practice

- IBM4 outperfoms IBM5 [Och and Ney, 2003]
- IBM5 outperfoms IBM4 [Schoenemann, 2013]

Training IBM 4 & 5 (Similar as in IBM 3)

Viterbi alignment

- Initialization: alignment obtained with a lower IBM model (e.g. IBM 3)
- Optimization: hill climbing (approximate)

Training

- Initialization: start with parameters obtained with lower IBM models
- Counting: for a paraticular parameter (e.g., $P(\phi \mid f)$) and a pair $(\boldsymbol{e}, \boldsymbol{f})$
  - We search for a representative set of alingments $A \subset A(m, n)$
  - Find a (close to) Viterbi alingment $\hat{a}$ and put it in A
  - Find alignments **neighboring** $\hat{a}$ and put them to A as well
  - The expected count of f having fertility ϕ is

$$\mathbb{E}[C(f, \phi; \boldsymbol{e}, \boldsymbol{f})] = \sum_{a \in A} P(a \mid \boldsymbol{e}, \boldsymbol{f}) \cdot C(f, \phi; a, \boldsymbol{e}, \boldsymbol{f}) \tag{13}$$

  where $C(f, \phi; a, \boldsymbol{e}, \boldsymbol{f})$ can be calculated directly (no hidden variables).

Neighboring Alignments

Formally

Neighboring alignments are typically defined as alignments that differ by a *move* or a *swap*:

Move

Two alignemnts a_1 and a_2 differ by a **move** if they differ only in the alginemnt for one output word on position i:

$$\exists_i : a_1(i) \neq a_2(i), \quad \forall_{i' \neq i} : a_1(i') = a_2(i') \tag{14}$$

Swap

Two alignemnts a_1 and a_2 differ by a **swap** if they agree in the alignments for all words, except for two, for which the alignment points are switched:

$$\begin{aligned} &\exists_{i_1, i_2} : i_1 \neq i_2, \\ &\quad a_1(i_1) = a_2(i_2), a_1(i_2) = a_2(i_1), a_2(i_2) \neq a_2(i_1), \\ &\quad \forall_{i' \neq i_1, i_2} : a_1(i') = a_2(i') \end{aligned} \tag{15}$$

References I

Och, F. J. and Ney, H. (2003).

A systematic comparison of various statistical alignment models.
Computational linguistics, 29(1):19–51.

Schoenemann, T. (2013).

Training nondeficient variants of ibm-3 and ibm-4 for word alignment.
In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22–31. Association for Computational Linguistics.