Statistical Machine Translation: Higher IBM Models (Part II)

Jakub Waszczuk

Heinrich Heine Universität Düsseldorf

Winter Semester 2018/19

Outline

Outline

IBM 3: Theory

Starting point:

$$\lambda_1 = \frac{\left(\frac{(k+1)N}{n+N} - 1\right)}{\left(\frac{Nk}{n} - 1\right)} \tag{1}$$

Simplifications:

IBM 3: Theory

Starting point:

$$\lambda_1 = \frac{\left(\frac{(k+1)N}{n+N} - 1\right)}{\left(\frac{Nk}{n} - 1\right)} \tag{1}$$

Simplifications:

$$\text{(numerator:)} \quad \frac{(k+1)N}{n+N} - 1 = \frac{kN+N}{n+N} - \frac{n+N}{n+N} = \frac{kN+N-n-N}{n+N} = \frac{kN-n}{n+N}$$

IBM 3: Theory

Starting point:

$$\lambda_1 = \frac{\left(\frac{(k+1)N}{n+N} - 1\right)}{\left(\frac{Nk}{n} - 1\right)} \tag{1}$$

Simplifications:

$$\text{(numerator:)} \quad \frac{(k+1)N}{n+N} - 1 = \frac{kN+N}{n+N} - \frac{n+N}{n+N} = \frac{kN+N-n-N}{n+N} = \frac{kN-n}{n+N}$$

$$\text{(denominator:)} \quad \frac{Nk}{n} - 1 = \frac{Nk}{n} - \frac{n}{n} = \frac{Nk-n}{n}$$

IBM 3: Theory

Starting point:

$$\lambda_1 = \frac{\left(\frac{(k+1)N}{n+N} - 1\right)}{\left(\frac{Nk}{n} - 1\right)} \tag{1}$$

Simplifications:

$$\text{(numerator:)} \quad \frac{(k+1)N}{n+N} - 1 = \frac{kN+N}{n+N} - \frac{n+N}{n+N} = \frac{kN+N-n-N}{n+N} = \frac{kN-n}{n+N}$$

$$\text{(denominator:)} \quad \frac{Nk}{n} - 1 = \frac{Nk}{n} - \frac{n}{n} = \frac{Nk-n}{n}$$

Getting back to Eq. 1:

$$\lambda_1 = \frac{\left(\frac{kN-n}{n+N}\right)}{\left(\frac{Nk-n}{n}\right)} = \frac{kN-n}{n+N} \times \frac{n}{Nk-n} = \frac{n}{n+N}$$

IBM 3: Theory

Bigram

$$\lambda_1 = \frac{C(w_{i-1})}{C(w_{i-1}) + |V|} \qquad \lambda_2 = 1 - \frac{C(w_{i-1})}{C(w_{i-1}) + |V|}$$

Trigram

$$\lambda_1 = \frac{C(w_{i-2}, w_{i-1})}{C(w_{i-2}, w_{i-1}) + |V|} \qquad \lambda_2 = 1 - \frac{C(w_{i-2}, w_{i-1})}{C(w_{i-2}, w_{i-1}) + |V|}$$

Interpretation

In the practical exercise (small traing set), $|V| = 1764$.

Bigram:

- Max $C(w)$: 2609
- Average $C(w)$: 1.48

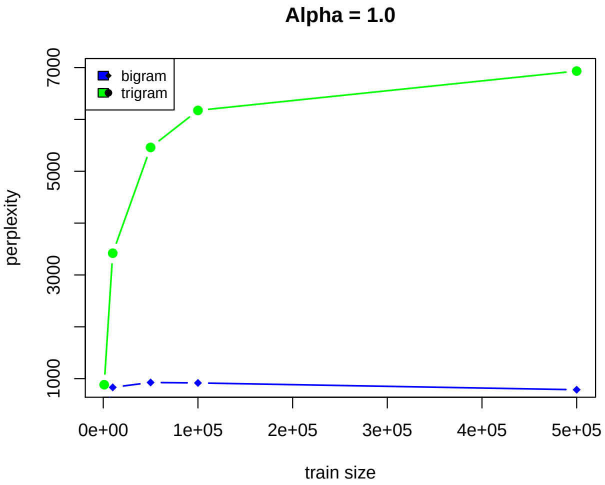
Trigram:

- Max $C(w, w')$: 1000
- Average $C(w, w')$: 0.093

Since, on average, $C(w) \ll |V|$, P_{ML} much less important than the uniform distribution estimation in bigram. In trigram, it's even worse ($C(w, w') \ll C(w) \ll |V|$)...

IBM 3: Practice

Perplexing perplexity



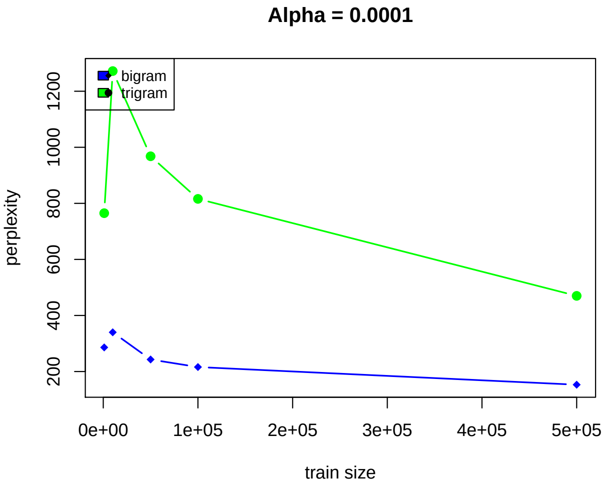
IBM 3: Practice

Add-α smoothing

$$\hat{P}_\alpha(w_i \mid w_{i-1}) = \frac{C(w_{i-1} w_i) + \alpha}{C(w_{i-1}) + \alpha|V|}$$

IBM 3: Practice

Perplexing perplexity



IBM 3: Practice

Take-home messages

- Add-1 is actually a rather poor smoothing technique
- Add-α is better (but there are still better ones)
- Given enough data and good smoothing technique, trigram may (should) outperform the bigram model

Outline

IBM 3

Setting

In what follows, we assume that the following are given:

- input sentence $\boldsymbol{f}$
- output sentence $\boldsymbol{e}$
- a corresonding alignment $a \in A(m, n)$
- and a tableau $t \in \mathcal{T}_{\boldsymbol{e},\boldsymbol{f}}(a)$

Properties (introduced last time)

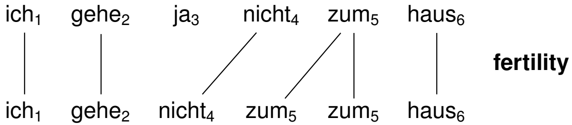
In IBM 3:

- NULL insertion allows to control the number of output words aligned with NULL
- All the tableaux $t \in \mathcal{T}_{\boldsymbol{e},\boldsymbol{f}}(a)$ have the same probability $P(t \mid \boldsymbol{f})$
- Calculating $P(a, \boldsymbol{e} \mid \boldsymbol{f})$ can be performed efficiently

We are going to see some clarifications concerning these properties.

IBM 3: Fertility

Example



The probability of the fertility in the example above is:

$$P(1 \mid \text{ich}) \cdot P(1 \mid \text{gehe}) \cdot P(0 \mid \text{ja}) \cdot P(1 \mid \text{nicht}) \cdot P(2 \mid \text{zum}) \cdot P(1 \mid \text{haus})$$

In general

Let $\boldsymbol{\phi}$ be the vector of fertilities assigned to the individual input words. Then:

$$P(\boldsymbol{\phi} \mid \mathbf{f}) = \prod_{j=1}^{n} P(\phi_j \mid f_j) \tag{2}$$

IBM 3: NULL insertion

Example

ich_1 gehe_2 nicht_4 zum_5 zum_5 haus_6

NULL insertion

ich_1 NULL_0 gehe_2 nicht_4 zum_5 zum_5 haus_6

The probability of NULL insertion in example above is:

$$p_0^1 \cdot (1 - p_0)^5$$

where p_0 is the sole parameter of the NULL insertion step.

In general

Let ϕ_0 be the number of inserted NULL words. Then:

$$P(\phi_0 \mid \boldsymbol{\phi}) = p_0^{\phi_0} \times (1 - p_0)^{(\sum_{j=1}^{n} \phi_j) - \phi_0} \tag{3}$$

$$= p_0^{\phi_0} \times (1 - p_0)^{m - 2\phi_0} \tag{4}$$

IBM 3: Lexical Translation

Example

ich_1	NULL_0	gehe_2	nicht_4	zum_5	zum_5	haus_6	
							lexical translation
I_1	do_0	go_2	not_4	to_5	the_5	house_6	
							distortion
I_1	do_2	not_3	go_4	to_5	the_6	house_7	

The probability of the lexical translation step in the example above is:

$$P(\text{I} \mid \text{ich}) \cdot P(\text{do} \mid \text{NULL}) \cdot P(\text{go} \mid \text{gehe}) \cdot P(\text{not} \mid \text{nicht}) \cdot$$
$$P(\text{to} \mid \text{zum}) \cdot P(\text{the} \mid \text{zum}) \cdot P(\text{house} \mid \text{haus})$$

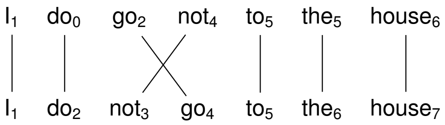
In general

In general, the probability of the lexical translation step is:

$$\prod_{i=1}^{m} P(e_i \mid f_{a(i)}) \tag{5}$$

IBM 3: Distortion

Example



distortion

The distortion probability for the example above is:

$$P(1 \mid 1, 7, 6) \cdot P(2 \mid 0, 7, 6) \cdot P(3 \mid 4, 7, 6) \cdot P(4 \mid 2, 7, 6) \cdot$$
$$P(5 \mid 5, 7, 6) \cdot P(6 \mid 5, 7, 6) \cdot P(7 \mid 6, 7, 6)$$

In general

Let $\boldsymbol{l} = (l_1, \ldots, l_m)$ be a vector of indices assigned to the individual tokens after the lexical translation step and π be a distortion function (permutation). Then:

$$P(\pi \mid \boldsymbol{l}) = \prod_{i=1}^{m} P(i \mid l_{\pi(i)}, m, n) = \prod_{i=1}^{m} P(i \mid a(i), m, n) \tag{6}$$

IBM 3: Tableau

Tableau probability

$$P(t \mid \boldsymbol{f}) = p_0^{\phi_0} \times (1 - p_0)^{m-2\phi_0} \times \prod_{j=1}^{n} P(\phi_j \mid f_j) \tag{7}$$

$$\times \prod_{i=1}^{m} P(e_i \mid f_{a(i)}) \cdot P(i \mid a(i), m, n)$$

Observation

The probability $P(t \mid \boldsymbol{f})$ (cf. Eq. 7) does not depend on the particular t, just on the properties of the underlying alignment a. Hence, $P(t \mid \boldsymbol{f})$ is the same for each $t \in \mathcal{T}_{\boldsymbol{e},\boldsymbol{f}}(a)$.

IBM 3: Alignment

Proposition (see complementary material)

Given $\boldsymbol{e}$, $\boldsymbol{f}$, and $a \in A(m,n)$, there are

$$\binom{m-\phi_0}{\phi_0} \times \prod_{j=1}^{n} \phi_j! \tag{8}$$

different tableaux $t \in \mathcal{T}_{\boldsymbol{e},\boldsymbol{f}}(a)$ corresponding to alignment a.

Alignment probability

The overall probability of alignment a and output $\boldsymbol{e}$ given input $\boldsymbol{f}$ is:

$$P(a, \boldsymbol{e} \mid \boldsymbol{f}) = \sum_{t \in \mathcal{T}_{\boldsymbol{e},\boldsymbol{f}}(a)} P(t \mid \boldsymbol{f}) = P(\bar{t} \mid \boldsymbol{f}) \times \binom{m-\phi_0}{\phi_0} \times \prod_{j=1}^{n} \phi_j!$$

where $\bar{t} \in \mathcal{T}_{\boldsymbol{e},\boldsymbol{f}}(a)$ is chosen arbitrarily.

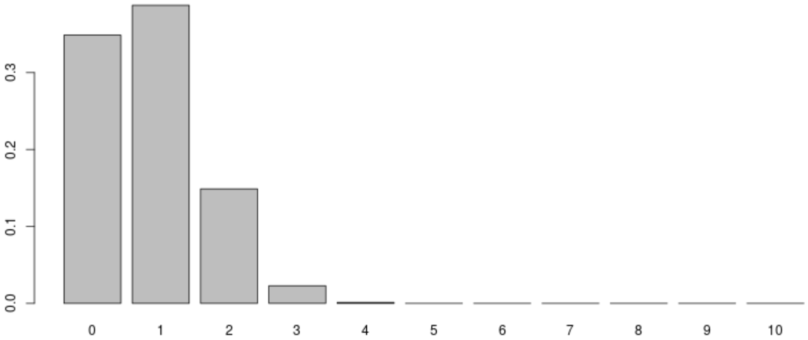
IBM 3: Sentence Length

Example

$P(a, \boldsymbol{e} \mid \boldsymbol{f})$ depends on ϕ_0 and p_0 as follows:

$$P(a, \boldsymbol{e} \mid \boldsymbol{f}) = \binom{m - \phi_0}{\phi_0} \times p_0^{\phi_0} \times (1 - p_0)^{m - 2\phi_0} \times \ldots$$

In case of $m = 10$ and $p_0 = 0.2$, this gives the following distribution for ϕ_0:



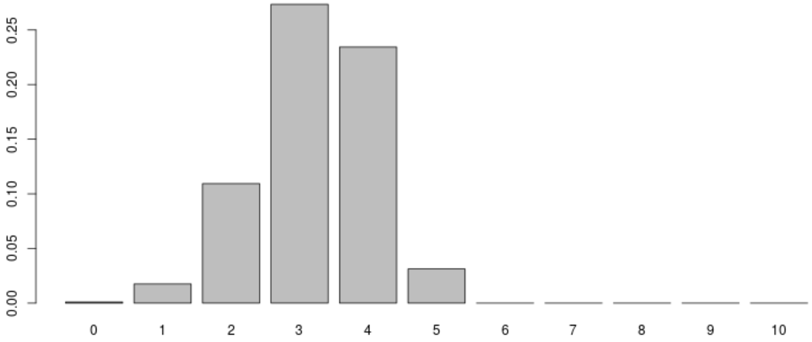
IBM 3: Sentence Length

Example

$P(a, \boldsymbol{e} \mid \boldsymbol{f})$ depends on ϕ_0 and p_0 as follows:

$$P(a, \boldsymbol{e} \mid \boldsymbol{f}) = \binom{m - \phi_0}{\phi_0} \times p_0^{\phi_0} \times (1 - p_0)^{m - 2\phi_0} \times \ldots$$

In case of $m = 10$ and $p_0 = 0.5$, this gives the following distribution for ϕ_0:



IBM 3: Architecture

Generative story

IBM 3 follows the following generative story:

$$P(t \mid \boldsymbol{f}) = P(\mathit{fer}(t) \mid \boldsymbol{f}) \times P(\mathit{null}(t) \mid \mathit{fer}(t)) \times P(\mathit{lex}(t) \mid \mathit{null}(t)) \times P(\mathit{dist}(t) \mid \mathit{lex}(t))$$

How do we know that this is a sound architecture?

IBM 3: Architecture

Generative story

IBM 3 follows the following generative story:

$$P(t \mid \mathbf{f}) = P(\mathit{fer}(t) \mid \mathbf{f}) \times P(\mathit{null}(t) \mid \mathit{fer}(t)) \times P(\mathit{lex}(t) \mid \mathit{null}(t)) \times P(\mathit{dist}(t) \mid \mathit{lex}(t))$$

How do we know that this is a sound architecture?

Proposition (see complementary material)

Let:

- X, Y, Z be three random variables
- $P(X \mid Y)$, $P(Y \mid Z)$ be two conditional probability functions

Then, $P(X, Y \mid Z) = P(X \mid Y) \times P(Y \mid Z)$ is a conditional probability function as well.

Conclusion

Provided that all the IBM 3 steps are sound, IBM 3 is sound as a whole.

IBM 3: Deficiency

Distortion again

Recall that, given $\boldsymbol{l} = (l_1, \dots, l_m)$ – a vector of indices assigned to the individual tokens after the lexical translation step – distortion (reordering, permutation) π is modeled according to:

$$P(\pi \mid \boldsymbol{l}) = \prod_{i=1}^{m} P(i \mid l_{\pi(i)}, m, n)$$

Deficiency

The distortion module of IBM 3 is **deficient** – it can assign non-zero probabilities to invalid distortions and, hence, some amount of probability mass can be wasted. Formally:

$$\sum_{\pi \in S_m} P(\pi \mid \boldsymbol{l}) < 1 \tag{9}$$

where $S_m \subset A(m, n)$ is the set of all permutations over $\{1, 2, \dots, m\}$.

Proof sketch (see complementary material)

It can be shown that $\sum_{\pi \in A(m,n)} P(\pi \mid \boldsymbol{l}) = 1$. If there are some non-permutation distortions $\pi \in A(m, n) \setminus S_m$ such that $P(\pi \mid \boldsymbol{l}) > 0$, then it must follow that $\sum_{\pi \in S_m} P(\pi \mid \boldsymbol{l}) < 1$.

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 $P(\phi_i \mid e_i)$.

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

$$m = \sum_{i=0}^{n} \phi_i \tag{10}$$

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

$$m = \sum_{i=0}^{n} \phi_i \tag{10}$$

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 $P(e_{i,k} \mid f_i)$.

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

$$m = \sum_{i=0}^{n} \phi_i \tag{10}$$

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

$$P(d_{i,k} \mid J) \tag{11}$$

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

IBM 3, 4 and 5: Generative Story

Generative story, adapted from [Schoenemann, 2013]

1. For $i = 1, 2, \dots, n$, 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

$$m = \sum_{i=0}^{n} \phi_i \tag{10}$$

3. For each $i = 0, 1, \dots, n$ and $k = 1, \dots, \phi_i$ 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

$$P(d_{i,k} \mid J) \tag{11}$$

   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 3: Alternative View

IBM 3

In IBM 3, the position $d_{i,k}$ of the k-th English word corresponding to the i-th input word (see point 3(b)) depends only on i and the sentence lengths:

$$P(d_{i,k} \mid J) = P(d_{i,k} \mid i, m, n) \tag{12}$$

Deficiency again

- The alternative generative story allows a different point of view on IBM 3's deficiency
- The position $d_{i,k}$ is chosen completely independently from the positions chosen before
- Hence, it is possible to choose the same position several times (while, clearly, there should be only one word occupying one output position)

IBM 4

Preliminaries

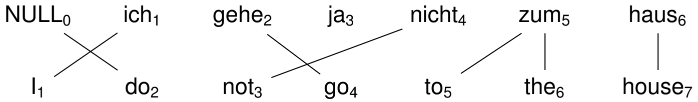
- Given input position i, we define $\lhd i$ as the closest preceding position $j > 0$ that has aligned output words ($\lhd i := 0$ if no such positions):

$$\lhd i = \max \{j \colon 0 < j < i, \phi_j > 0\} \cup \{0\} \tag{13}$$

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

$$\odot_i = \begin{cases} \lceil \text{avg}\{d_{i,k} : 1 < k \leq \phi_i\} \rceil & \text{if } i > 0 \land \phi_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Example



- $\lhd 1 =$
- $\lhd 2 =$
- $\lhd 3 =$
- $\lhd 4 =$
- $\odot_1 =$
- $\odot_5 =$

IBM 4

Preliminaries

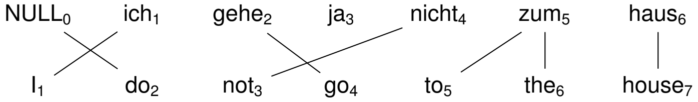
- Given input position i, we define $\triangleleft i$ as the closest preceding position $j > 0$ that has aligned output words ($\triangleleft i := 0$ if no such positions):

$$\triangleleft i = \max \{j \colon 0 < j < i, \phi_j > 0\} \cup \{0\} \tag{13}$$

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

$$\odot_i = \begin{cases} \lceil \text{avg}\{d_{i,k} : 1 < k \leq \phi_i\} \rceil & \text{if } i > 0 \land \phi_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Example



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

IBM 4

Preliminaries

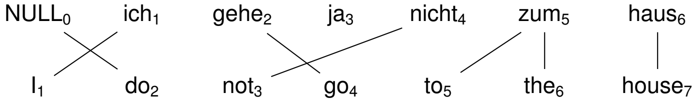
- Given input position i, we define $\lhd i$ as the closest preceding position $j > 0$ that has aligned output words ($\lhd i := 0$ if no such positions):

$$\lhd i = \max\,\{j \colon 0 < j < i, \phi_j > 0\} \cup \{0\} \tag{13}$$

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

$$\odot_i = \begin{cases} \lceil \mathrm{avg}\{d_{i,k} : 1 < k \leq \phi_i\} \rceil & \text{if } i > 0 \wedge \phi_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Example



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

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 ($i = 0$, $k \geq 1$):

$$P(d_{0,k} = j \mid J) = \frac{1}{m} \tag{15}$$

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

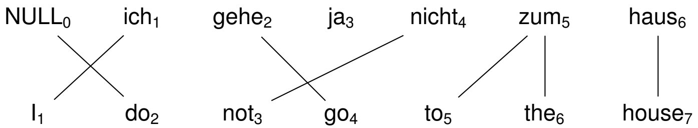
$$\begin{aligned} P(d_{i,1} = j \mid J) &= P_{=1}(j \mid \odot_{\triangleleft i}) && (16) \\ &= P_{=1}(j - \odot_{\triangleleft i}) && (17) \end{aligned}$$

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

$$\begin{aligned} P(d_{i,k} = j \mid J) &= P_{>1}(j \mid d_{i,k-1}) && (18) \\ &= P_{>1}(j - d_{i,k-1}) && (19) \end{aligned}$$

IBM 4: Example

Target alignment

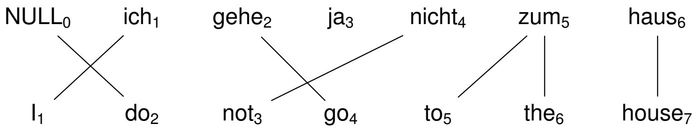


Positioning process

f_i	$d_{i,k}$	I_1	do_2	not_3	go_4	to_5	the_6	house_7	j	$\odot_{\lhd i}$	$P(d_{i,k} \mid J)$
NULL_0	$d_{0,1}$										

IBM 4: Example

Target alignment

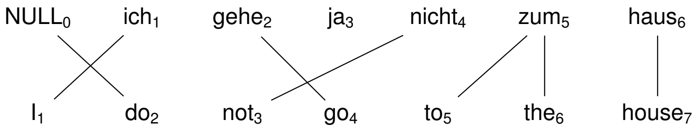


Positioning process

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	$d_{1,1}$										

IBM 4: Example

Target alignment

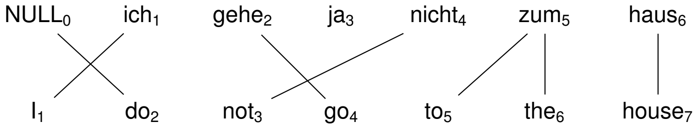


Positioning process

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	$P_{=1}(1)$
$gehe_2$	$d_{2,1}$										

IBM 4: Example

Target alignment

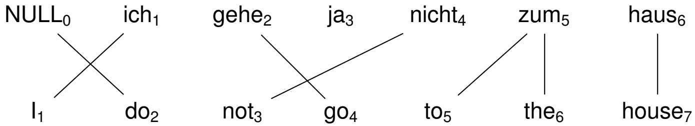


Positioning process

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$_4$	$d_{4,1}$										

IBM 4: Example

Target alignment

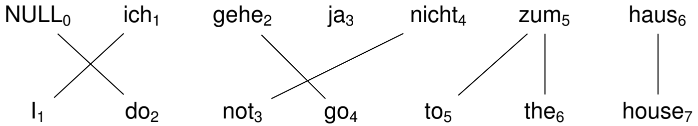


Positioning process

f_i	$d_{i,k}$	I$_1$	do$_2$	not$_3$	go$_4$	to$_5$	the$_6$	house$_7$	j	$\odot_{\lhd i}$	$P(d_{i,k} \mid J)$
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$_5$	$d_{5,1}$										

IBM 4: Example

Target alignment

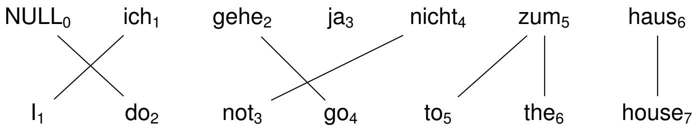


Positioning process

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	$P_{=1}(2)$
	$d_{5,2}$										

IBM 4: Example

Target alignment

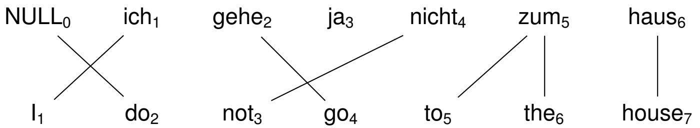


Positioning process

f_i	$d_{i,k}$	I$_1$	do$_2$	not$_3$	go$_4$	to$_5$	the$_6$	house$_7$	j	$\odot_{\lhd i}$	$P(d_{i,k} \mid J)$
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$_6$	$d_{6,1}$										

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	$\odot_{\lhd i}$	$P(d_{i,k} \mid J)$
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

- **Question**: Is IBM 4 non-deficient? I.e., does it guarantee that only well-formed alignments get generated?

IBM 4

Deficiency

- **Question**: Is IBM 4 non-deficient? I.e., does it guarantee that only well-formed alignments get generated?
- **Answer**: IBM 4 is deficient as well:
  - 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

IBM 4

Deficiency

- **Question**: Is IBM 4 non-deficient? I.e., does it guarantee that only well-formed alignments get generated?
- **Answer**: IBM 4 is deficient as well:
  - 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

Advantages

Nevertheless, IBM 4 presents clear advantages in comparison with IBM 3:

- IBM 4 captures *1-order relations* between placements of adjacent words
  - This is good for modeling the intuition that word order is typically preserved
- *Relatve* distortions are more linguistically motivated than absolute ones

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 $d_{i,k}$

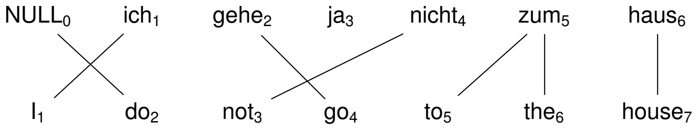
Deficiency

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

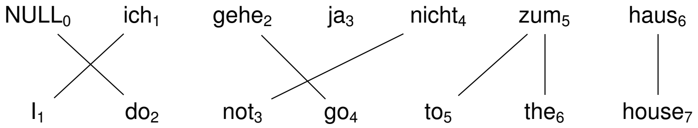


Positioning process

f_i	$d_{i,k}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	j	$\odot_{\lhd i}$	v_{max}	v_j
		I_1	do_2	not_3	go_4	to_5	the_6	house_7				
NULL_0	$d_{0,1}$	1	2	3	4	5	6	7	2	-		

IBM 5: Example

Target alignment

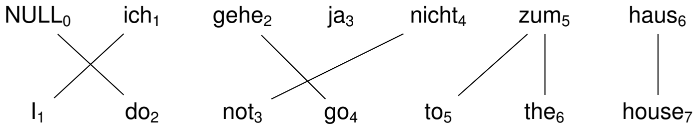


Positioning process

f_i	$d_{i,k}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	j	$\odot_{\lhd i}$	$v_{\max}$	v_j
		I$_1$	do$_2$	not$_3$	go$_4$	to$_5$	the$_6$	house$_7$				
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

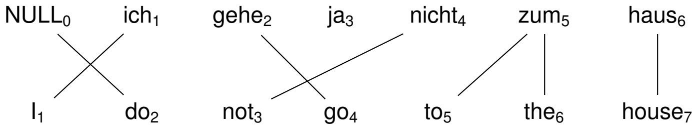


Positioning process

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	$\odot_{\lhd i}$	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

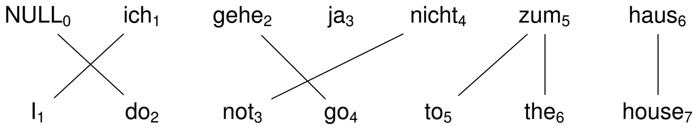


Positioning process

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

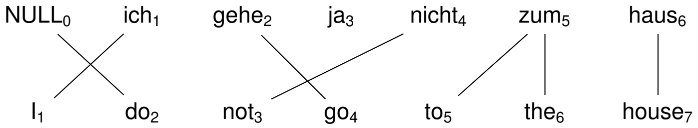


Positioning process

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

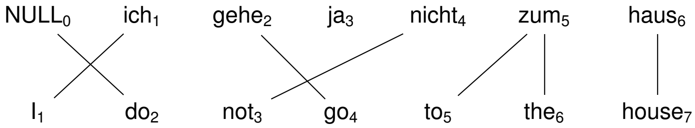


Positioning process

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

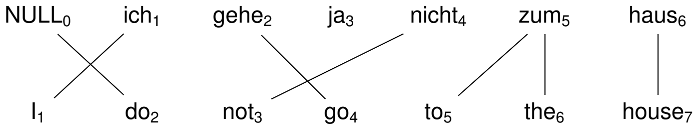


Positioning process

f_i	$d_{i,k}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	j	$\odot_{\lhd i}$	$v_{\max}$	v_j
		I_1	do_2	not_3	go_4	to_5	the_6	house_7				
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	v_2	v_3	v_4	v_5	v_6	v_7	j	$\odot_{\lhd i}$	v_{max}	v_j
		I$_1$	do$_2$	not$_3$	go$_4$	to$_5$	the$_6$	house$_7$				
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 ($i = 0$, $k \geq 1$):

$$P(d_{0,k} = j \mid J) = \frac{1}{v_{\max}} \tag{20}$$

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

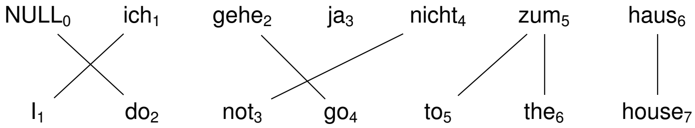
$$P(d_{i,1} = j \mid J) = P_{=1}(v_j \mid v_{\odot_{\triangleleft i}}, v_{\max}) \tag{21}$$

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

$$P(d_{i,k} = j \mid J) = P_{>1}(v_j \mid v_{d_{i,k-1}}, v_{\max}) \tag{22}$$

IBM 5

Target alignment

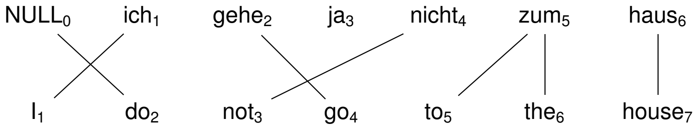


Positioning process

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	$v_{\odot_{\triangleleft i}}$	$P(d_{i,k} \mid J)$
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2		

IBM 5

Target alignment

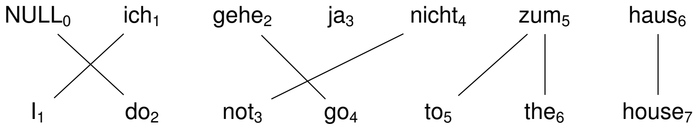


Positioning process

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	$\odot_{\lhd i}$	$v_{\max}$	v_j	$v_{\odot_{\lhd i}}$	$P(d_{i,k} \mid J)$
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2	-	1/7
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1		

IBM 5

Target alignment

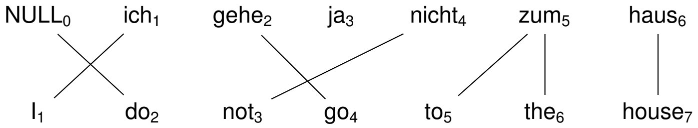


Positioning process

f_i	$d_{i,k}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	j	$\odot_{\triangleleft i}$	$v_{\max}$	v_j	$v_{\odot_{\triangleleft i}}$	$P(d_{i,k} \mid J)$
		I_1	do_2	not_3	go_4	to_5	the_6	house_7						
NULL_0	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2	-	1/7
ich_1	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1	0	$P_{=1}(1 \mid 0, 6)$
gehe_2	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2		

IBM 5

Target alignment

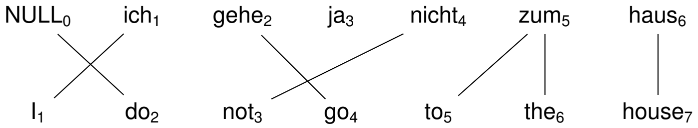


Positioning process

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	$\odot_{\lhd i}$	$v_{\max}$	v_j	$v_{\odot_{\lhd i}}$	$P(d_{i,k} \mid J)$
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2	-	1/7
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1	0	$P_{=1}(1 \mid 0, 6)$
gehe$_2$	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2	0	$P_{=1}(2 \mid 0, 5)$
nicht$_4$	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1		

IBM 5

Target alignment

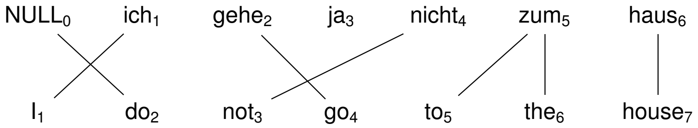


Positioning process

f_i	$d_{i,k}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	j	$\odot_{\lhd i}$	$v_{\max}$	v_j	$v_{\odot_{\lhd i}}$	$P(d_{i,k} \mid J)$
		I_1	do_2	not_3	go_4	to_5	the_6	house_7						
NULL_0	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2	-	1/7
ich_1	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1	0	$P_{=1}(1 \mid 0, 6)$
gehe_2	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2	0	$P_{=1}(2 \mid 0, 5)$
nicht_4	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1	1	$P_{=1}(1 \mid 1, 4)$
zum_5	$d_{5,1}$	0	0	0	0	1	2	3	5	3	2	1		

IBM 5

Target alignment

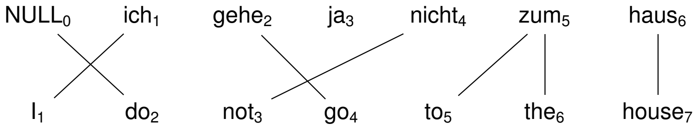


Positioning process

f_i	$d_{i,k}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	j	$\odot_{\lhd i}$	$v_{\max}$	v_j	$v_{\odot_{\lhd i}}$	$P(d_{i,k} \mid J)$
		I_1	do_2	not_3	go_4	to_5	the_6	house_7						
NULL_0	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2	-	1/7
ich_1	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1	0	$P_{=1}(1 \mid 0, 6)$
gehe_2	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2	0	$P_{=1}(2 \mid 0, 5)$
nicht_4	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1	1	$P_{=1}(1 \mid 1, 4)$
zum_5	$d_{5,1}$	0	0	0	0	1	2	3	5	3	2	1	0	$P_{=1}(1 \mid 0, 2)$
	$d_{5,2}$	0	0	0	0	0	1	2	6	-	2	1		

IBM 5

Target alignment

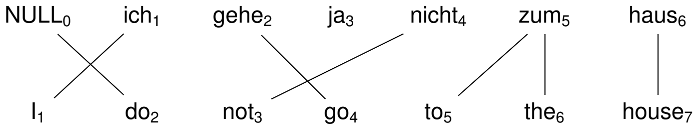


Positioning process

f_i	$d_{i,k}$	v_1	v_2	v_3	v_4	v_5	v_6	v_7	j	$\odot_{\lhd i}$	$v_{\max}$	v_j	$v_{\odot_{\lhd i}}$	$P(d_{i,k} \mid J)$
		I_1	do_2	not_3	go_4	to_5	the_6	house_7						
NULL_0	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2	-	1/7
ich_1	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1	0	$P_{=1}(1 \mid 0, 6)$
gehe_2	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2	0	$P_{=1}(2 \mid 0, 5)$
nicht_4	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1	1	$P_{=1}(1 \mid 1, 4)$
zum_5	$d_{5,1}$	0	0	0	0	1	2	3	5	3	2	1	0	$P_{=1}(1 \mid 0, 2)$
	$d_{5,2}$	0	0	0	0	0	1	2	6	-	2	1	-	$P_{>1}(1 \mid 1, 2)$
haus_6	$d_{6,1}$	0	0	0	0	0	0	1	7	6	1	1		

IBM 5

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	$\odot_{\lhd i}$	$v_{\max}$	v_j	$v_{\odot_{\lhd i}}$	$P(d_{i,k} \mid J)$
NULL$_0$	$d_{0,1}$	1	2	3	4	5	6	7	2	-	7	2	-	1/7
ich$_1$	$d_{1,1}$	1	1	2	3	4	5	6	1	0	6	1	0	$P_{=1}(1 \mid 0, 6)$
gehe$_2$	$d_{2,1}$	0	0	1	2	3	4	5	4	1	5	2	0	$P_{=1}(2 \mid 0, 5)$
nicht$_4$	$d_{4,1}$	0	0	1	1	2	3	4	3	4	4	1	1	$P_{=1}(1 \mid 1, 4)$
zum$_5$	$d_{5,1}$	0	0	0	0	1	2	3	5	3	2	1	0	$P_{=1}(1 \mid 0, 2)$
	$d_{5,2}$	0	0	0	0	0	1	2	6	-	2	1	-	$P_{>1}(1 \mid 1, 2)$
haus$_6$	$d_{6,1}$	0	0	0	0	0	0	1	7	6	1	1	0	$P_{=1}(1 \mid 0, 1)$

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

Viterbi

Similar as in IBM 3:

- Hill climbing (approximate)

Training

Similar as in IBM 3:

- 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 similar alignments to $\hat{a}$ and put them to A as well
  - The expected count of f having fertility ϕ is

$$\mathbb{E}[C(f, \phi; \boldsymbol{e}, \boldsymbol{f})] = \sum_{a \in A} P(a \mid \boldsymbol{e}, \boldsymbol{f}) \cdot C(f, \phi; a, \boldsymbol{e}, \boldsymbol{f}) \tag{23}$$

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

Neighboring Alignments

Formally

We define **neighboring alignments** 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:

$$\exists_i : a_1(i) \neq a_2(i), \quad \forall_{i' \neq i} : a_1(i') = a_2(i') \tag{24}$$

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:

$$\begin{aligned} \exists_{i_1, i_2} : \ & i_1 \neq i_2, \\ & a_1(i_1) = a_2(i_2), a_1(i_2) = a_2(i_1), a_2(i_2) \neq a_2(i_1), \\ & \forall_{i' \neq i_1, i_2} : a_1(i') = a_2(i') \end{aligned} \tag{25}$$

Outline

Alignment Evaluation

Alignment Evaluation

- We have a translation model, which makes predictions as to probable alignments
- We wish to tell how good those predictions are
- Why? Alignments have useful applications (also outside of the context of translation)

Alignment Applications

- Starting point for refined phrase-based statistical translation systems
- Automatic extraction of bilingual lexica and terminology from corpora
- Transfer text analysis tools (morphologic analyzers, part-of-speech taggers, parsers) from a rich-resource language to a low-resource langauge

Alignment Evaluation

How this is done?

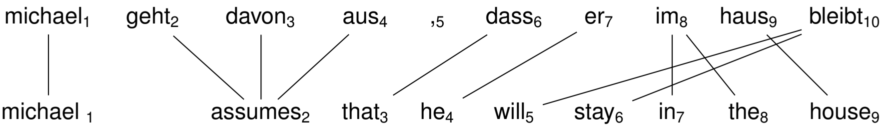
Compare:

- The predicted (Viterbi) alignments
- With gold standard alignments (made by hand)

Issue

We compare two objects with possibly different form:

- Predicted alignment *functions* ($\{1, \ldots, m\} \rightarrow \{0, \ldots, m\}$)
- Gold standard alignment *relations* ($\{1, \ldots, m\} \times \{0, \ldots, m\}$)



Alignment Evaluation

Bidirectional alignments

Given a bilingual EN-GE corpus, we can use EM to:

- Train a GE $\rightarrow$ EN translation model
- Train an EN $\rightarrow$ GE translation model

Having both translations models and a test sentence:

- Determine the best GE $\rightarrow$ EN alignment function A_1
- Determine the best EN $\rightarrow$ GE alignment function A_1

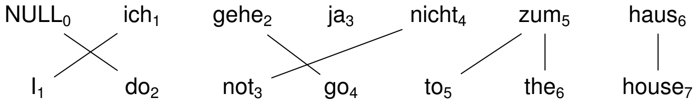
Issue 2

For a given sentence pair, which aligment – A_1 or A_2 – should we chose as the best alignment?

Alignment Evaluation

Functions as relations

Alignment functions can be represented as relations, for instance:



as:

$$\{(1,0),(2,0),(3,4),(4,2),(5,5),(6,6),(7,6)\}$$

As a result, we can perform standard set-theoretic operations on alignments (intersection, union, etc.)

Alignment Evaluation

Bidirectional alignment

Given two alignment relations:

- A_1 from GE to EN
- A_2 from EN to GE

We can define the final alignment relation as:

Sum

The set of all the alignment arcs present in either A_1 or A_2 (inversed):

$$A := A_1 \cup A_2^{-1} \tag{26}$$

Useful if we want to be sure to predict as many gold standard arcs as possible (*recall*)

Intersection

The set of all the alignment arcs present in both A_1 and A_2 (inversed):

$$A := A_1 \cap A_2^{-1} \tag{27}$$

Useful if we want to be sure to mostly predict only gold standard arcs (*precision*)

Alignment Evaluation

Gold standard

Manual alignment is not easy task (the notion of ,,correspondence“ between words is subjective), hence the gold standard often consists of:

- The set of *possible* arcs M
- The set of *sure* arcs $S \subseteq M$

Alignment error rate

$$AER(S, M, A) = 1 - \frac{|A \cap S| + |A \cap M|}{|A| + |S|} \tag{28}$$

AER returns a value between 0 (the best) and 1 (the worst).

Observation

A perfect error rate of 0 is achieved when:

- Every sure arc is predicted ($S \subseteq A$)
- Every predicted arc is possible ($A \subseteq M$)

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.