Statistical Machine Translation

May 25th, 2014

UNIVERSITÄT DES SAARLANDES

Josef van Genabith
DFKI GmbH
Josef.van_Genabith@dfki.de

Language Technology II

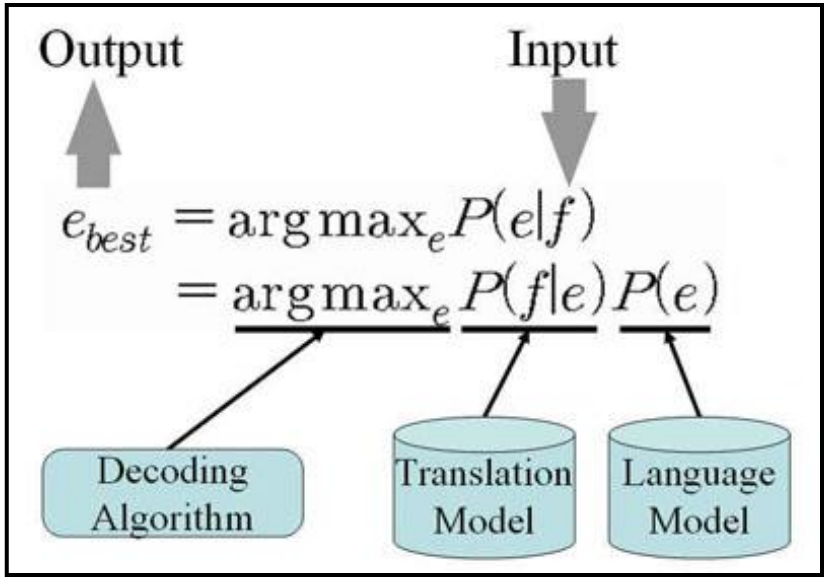
SS 2014

Based on Kevin Knight's 1999
A Statistical MT Tutorial Work Book

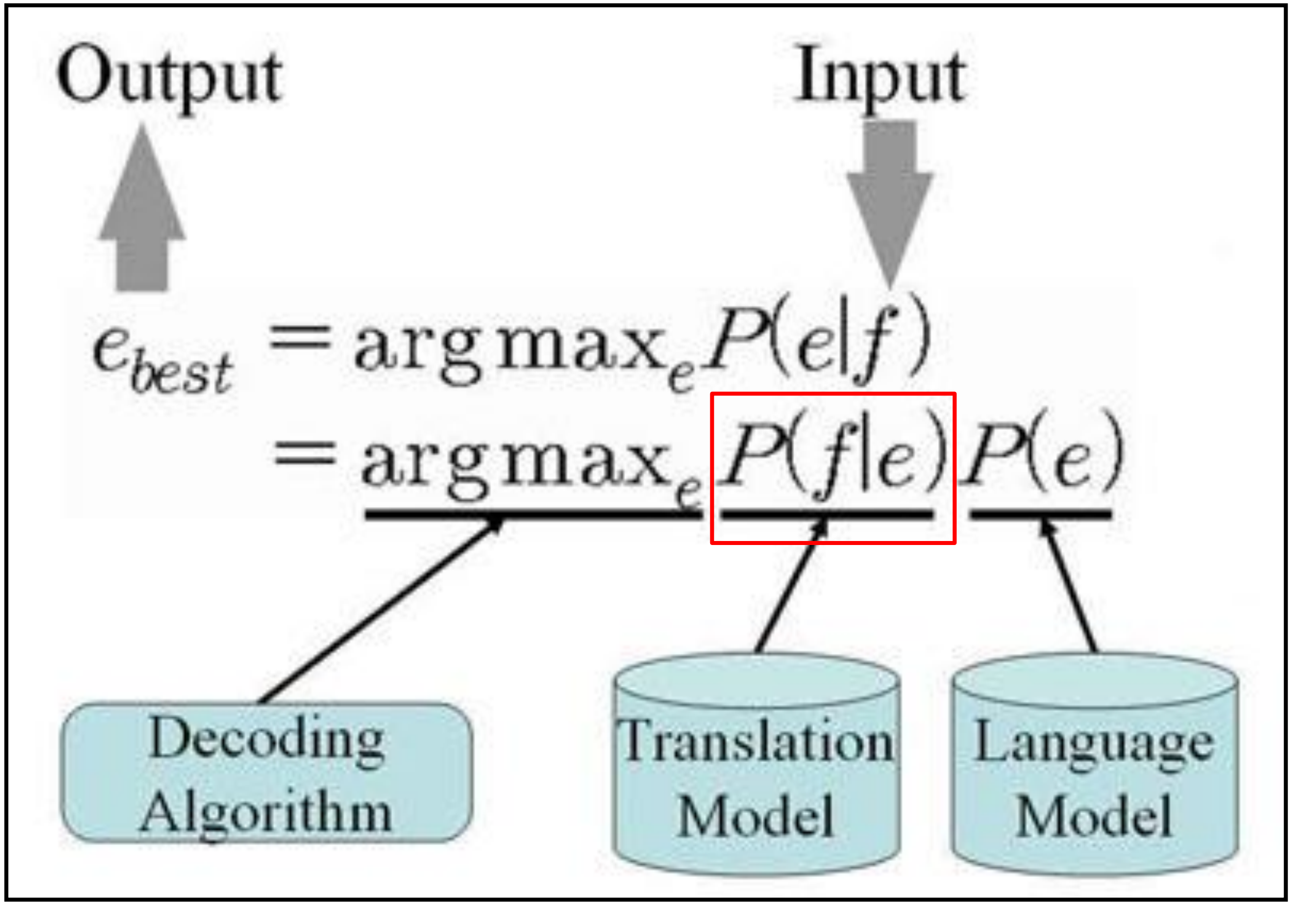
Overview

UNIVERSITÄT DES SAARLANDES

- Introduction: the basic idea
- IBM models: the noisy channel, Model 3, EM
- Phrase-Based SMT



Translation Modelling



Today: Model 3 $P(f|e)$ in full glory:

$$P(a,f|e) = \binom{m-\varphi_0}{\varphi_0} \times {p_0}^{(m-2\varphi_0)} \times {p_1}^{\varphi_0}$$

$$\times \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right)$$

$$\times \prod_{j:a_j \neq 0}^{m} d\left(j \middle| a_j, l, m\right) \times \prod_{i=0}^{l} \varphi_i! \times \frac{1}{\varphi_0!}$$

Recall that

$$P(f|e) = \sum_{a} P(a,f|e) \quad \text{and} \quad P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$$

Translation Modelling

- Remember that translating f to e we reason backwards
- We observe f
- We want to know what e is (most) likely to be uttered and likely to have been translated into f

$$\hat{e} = \arg\max_{e} P(f|e) \times P(e)$$

- Story: replace words in e by f (French) words and scramble them around
- “What kind of a crackpot story is that?” (Kevin Knight, 1999)
- IBM Model 3 ☺

What happens in translation?

- What happens in translation?
- Actually a lot ….

- EN: `Mary did not slap the green witch`
- ES: `Mary no daba una botefada a la bruja verde`

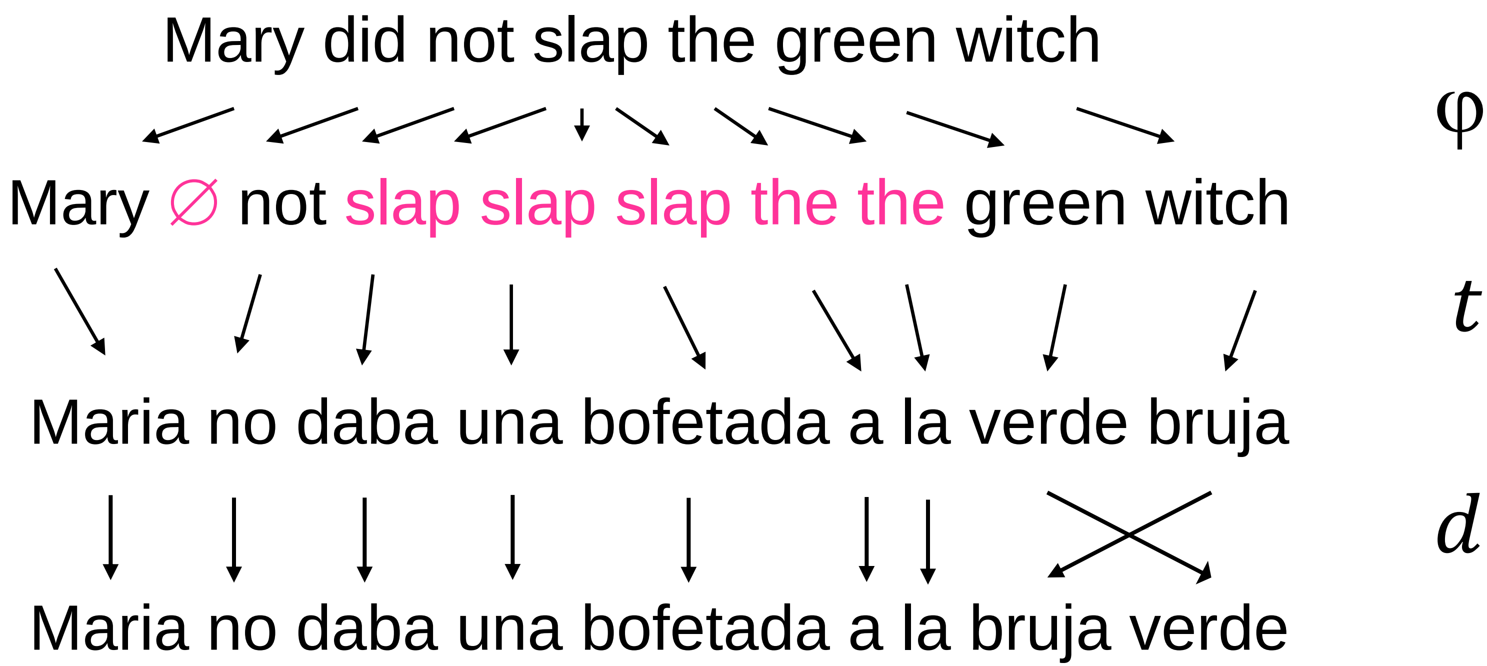
- But from a purely external point of view
  - Source words get replaced by target words
  - Words in target are moved around (“reordered”)
  - Source and target need not be equally long ….
- So minimally that is what we need to model …

Model parameters

1. For each word e_i in an English sentence $i = (1 \;... l)$, we choose a fertility φ_i. The choice of fertility is dependent solely on the English word in question, nothing else.
2. For each word e_i, we generate φ_i French words: $t(f|e)$. The choice of French word is dependent solely on the English word that generates it. It is not dependent on the English context around the English word. It is not dependent on other French words that have been generated from this or any other English word.
3. All those French words are permuted: $d(\pi_f|\pi_e, l, m)$. Each French word is assigned an absolute target "position slot." For example, one word may be assigned position 3, and another word may be assigned position 2 -- the latter word would then precede the former in the final French sentence. The choice of position for a French word is dependent solely on the absolute position of the English word that generates it.

Translation as String Rewriting

UNIVERSITÄT DES SAARLANDES



Model parameters

UNIVERSITÄT DES SAARLANDES

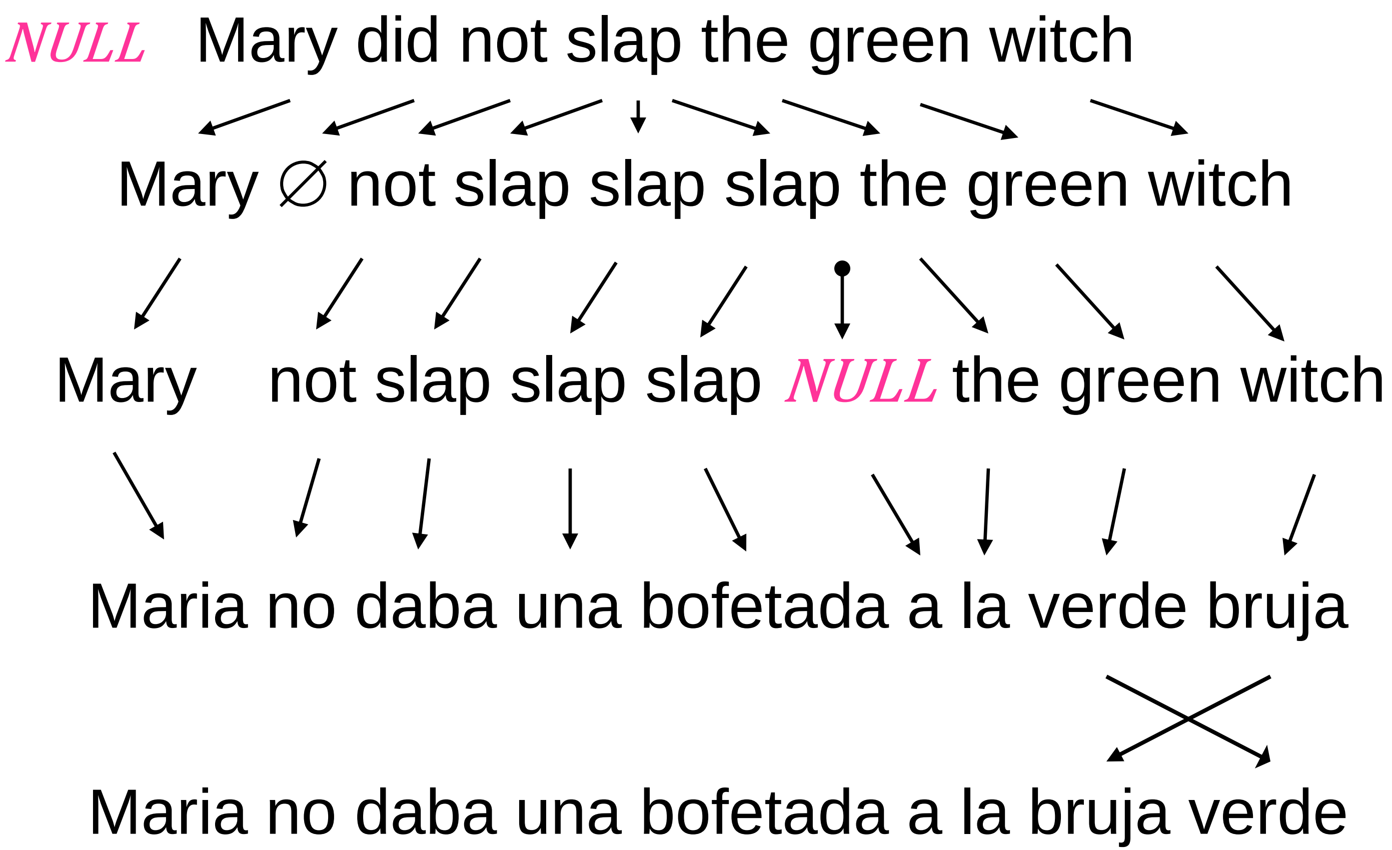
- We would like to learn the Parameters for fertility, (word) translation and distortion from data
- The parameters look like this
  - $n(3|slap)$
  - $t(maison|house)$
  - $d(5|2,4,6)$
- And they have probabilities associated with them

NULL

UNIVERSITÄT DES SAARLANDES

- One more twist: spurious words
- E.g. function words can appear in target that do not have correspondences in source
- Pretend that every English sentence has NULL word in position 0 and can generate spurious words in target: $t(a|NULL)$
- Longer sentences are more likely to have more spurious words, therefore:
- $NULL$ doesn't have fertility distribution n but a probability p_1 with which it can generate a spurious word after each properly generated word, how many decided by φ_0
- $p_0 = 1 - p_1$ is probability of not tossing in spurious word

NULL



NULL Mary did not slap the green witch
Mary ∅ not slap slap slap the green witch
Mary not slap slap slap NULL the green witch
Maria no daba una bofetada a la verde bruja
Maria no daba una bofetada a la bruja verde

Model 3

1. For each English word e_i indexed by $i = 1,2, \dots, l$ choose fertility φ_i with probability $n(\varphi_i | e_i)$.
2. Choose the number φ_0 of "spurious" French words to be generated from $e_0 = NULL$, using probability p_1 and the sum of fertilities from step 1.
3. Let m be the sum of fertilities for all words, including $NULL$.
4. For each $i = 1,2, \dots, l$ and each $k = 1,2, \dots, \varphi_i$ choose a French word $\tau_{i,k}$ with probability $t(\tau_{i,k} | e_i)$.
5. For each each $i = 1,2, \dots, l$ and each $k = 1,2, \dots, \varphi_i$ choose target French position $\pi_{i,k}$ with probability $d(\pi_{i,k} | \ i, l, m)$.
6. For each $k = 1,2, \dots, \varphi_0$ choose a position $\pi_{0,k}$ from the φ_0 - $k + 1$ remaining vacant positions in $1,2, \dots, m$ for a total probability of $1/\varphi_0$!.
7. Output the French sentence with words $\tau_{i,k}$ in positions $\pi_{i,k}$ $(0 \leq i \leq l, 1 \leq k \leq \varphi_i)$.

Model 3

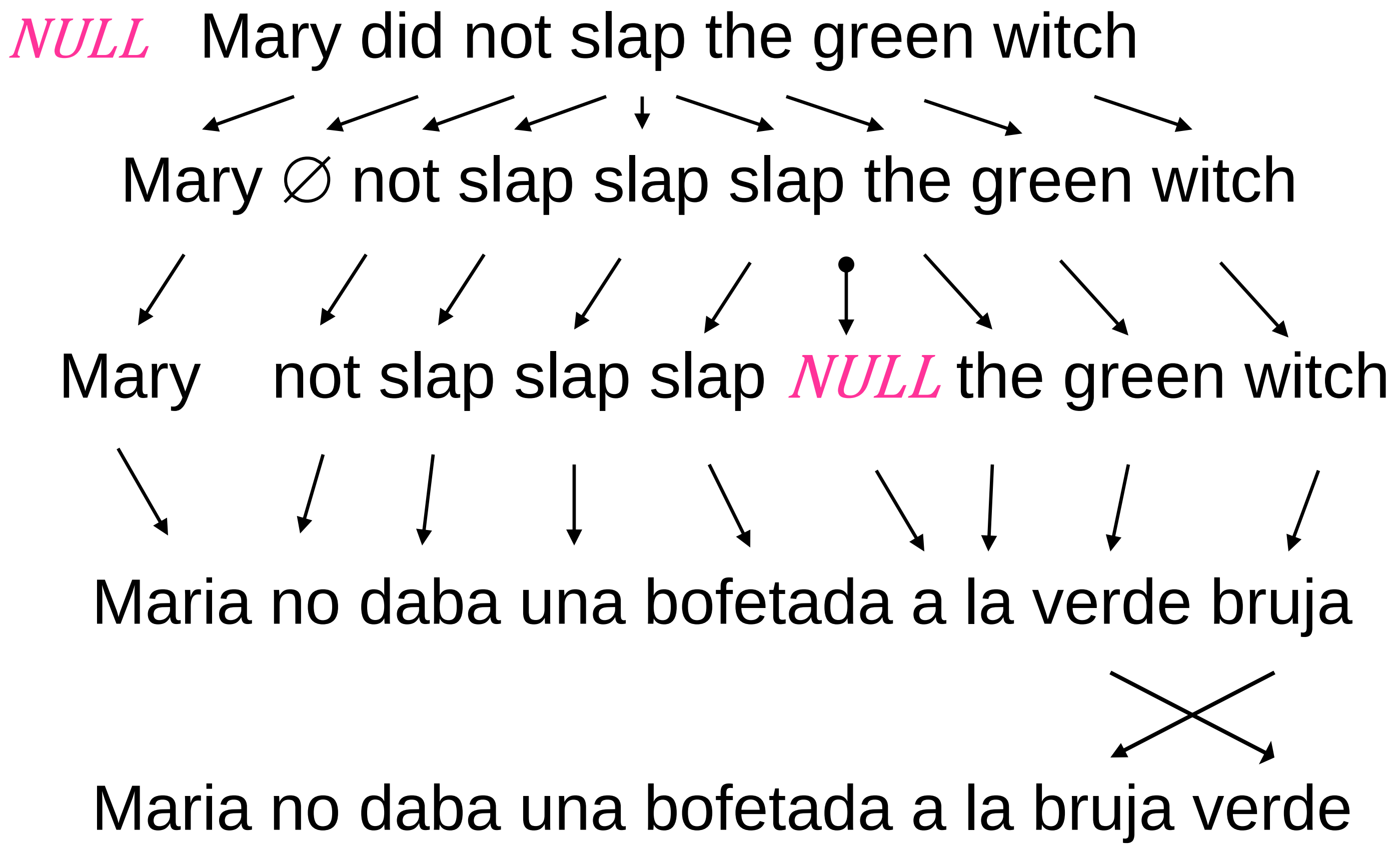
Model 3 has four types of parameters

n, t, p and d

Need to think about two things:

- How to get parameter values from data
- Once we have those, how to compute $P(f|e)$ for any sentences e and f

Model 3 as String Rewriting



Model 3 as String Rewriting

UNIVERSITÄT DES SAARLANDES

NULL Mary did not slap the green witch

Mary ∅ not slap slap slap the green witch

Mary not slap slap slap *NULL* the green witch

Maria no daba una bofetada a la verde bruja

Maria no daba una bofetada a la bruja verde

Model 3 as String Rewriting

UNIVERSITÄT DES SAARLANDES

NULL Mary did not slap the green witch

Mary ∅ not slap slap slap the green witch

Mary not slap slap slap *NULL* the green witch

Maria no daba una bofetada a la verde bruja

Maria no daba una bofetada a la bruja verde

- If we had a million English – French translations
- + their step by step rewrites
- We could easily estimate parameter
- Use MLE: just count and divide

Model 3 as String Rewriting

UNIVERSITÄT DES SAARLANDES

NULL Mary did not slap the green witch

Mary ∅ not slap slap slap the green witch

Mary not slap slap slap *NULL* the green witch

Maria no daba una bofetada a la verde bruja

Maria no daba una bofetada a la bruja verde

- If did occurred 15,000 times
- and did → ∅ occurred 2000 times
- Then $n(0|did) = 2/15$

Model 3 as String Rewriting

UNIVERSITÄT DES SAARLANDES

Exercise.
Take 10,000 English sentences and rewrite them into French, storing all intermediate strings. No, make that a million English sentences! Ha, ha, just kidding. Don't do this exercise.

Kevin Knight, A Statistical MT Tutorial Workbook, 1999, p.14

Word-for-Word Alignments

Our generative model in terms of string rewriting:

```
NULL  And  the  program  has  been  implemented
           the  program  has  been  implemented implemented implemented
           Le programme  a    ete        mis           en         application
           Le programme  a    ete        mis           en         application
```

A simple data-structure that captures (most) of this: alignments

```
NULL  And  the  program  has  been  implemented
  |     |    |       |      |     |        |
             |       |      |     |     +-+---+
             |       |      |     |     | |   |
              Le programme   a   ete   mis en application
```

Word-for-Word Alignments

```
NULL  And  the  program  has  been  implemented
  |     |    |      |      |    |         |
             |      |      |    |       +-+---+
             |      |      |    |       | |   |
            Le programme   a   ete     mis en application
```

Word alignments that correspond best to Model 3:

- Every French word connected to exactly one English word (incl. NULL)
- So we never have 2 (or more) English generate one French
- [2, 3, 4, 5, 6, 6, 6]

Word-for-Word Alignments and MLE Parameter Estimation: n and t

UNIVERSITÄT DES SAARLANDES

```
NULL  And  the  program  has  been  implemented
  |     |    |       |      |    |         |
             |       |      |    |      +-+----+
             |       |      |    |      | |    |
              Le programme   a    ete   mis en application
```

If we had a million of these we could estimate Model 3 parameters (MLE):

$$n(0|the) = count(the \rightarrow \emptyset)/count(the)$$
$$t(la|the) = count(the \rightarrow la)/count(the)$$

$$n(0|the) = c(the \rightarrow \emptyset)/c(the)$$
$$t(la|the) = c(the \rightarrow \ la)/c(the)$$

$$n(0|the) = \#(the \rightarrow \emptyset)/\#(the)$$
$$t(la|the) = \#(the \rightarrow \ la)/\#(the)$$

Word-for-Word Alignments and MLE Parameter Estimation: d and p

- $d(5|2,4,6)$

- English word 2 in French position 5, where English sentence is 4 words long and French 6

- How to estimate probability distribution over $d(j|2,4,6)$?

$$d(5|2,4,6) = \frac{\#d(5|2,4,6)}{\sum_{j=1}^{6} \#d(j|2,4,6)}$$

Word-for-Word Alignments and MLE Parameter Estimation: d and p

UNIVERSITÄT DES SAARLANDES

- p_1 is probability for tossing in “spurious” NULL generated word after each properly generated word
- In aligned data: M words generated by NULL. Then M spurious words will be generated in $N - M$ cases:

$$p_1 = \frac{M}{N - M}$$

- If we had large aligned data, we could estimate our parameters
- Unfortunately we don’t have such data
- Bootstrapping
- Learning with/from incomplete data: we have the translations but not the alignments a

Word-for-Word Alignments and MLE Parameter Estimation:

UNIVERSITÄT DES SAARLANDES

- If we had large aligned data, we could estimate our parameters
- Unfortunately we don't have such data

- If we had the parameters we could estimate alignments
- Unfortunately we don't have such data

- Bootstrapping
- Learning with/from incomplete data: we have the translations but not the alignments a

A Chicken and Egg Problem

- If we had alignments, we could estimate model parameters (such as translation probabilities, fertilities etc.)
- If we had model parameters, we could estimate alignments
- We don't want to/can't spend a lot of money to manually align 100s of thousands (or millions) of sentences of bi-text
- Need a way of estimating model parameters from incomplete data
- The thing we don't have is called a "hidden" variable (a "latent" variable, unobserved …)
- In our case this is the alignment a, a is a latent variable

- Expectation Maximisation

Expectation Maximisation (EM)

UNIVERSITÄT DES SAARLANDES

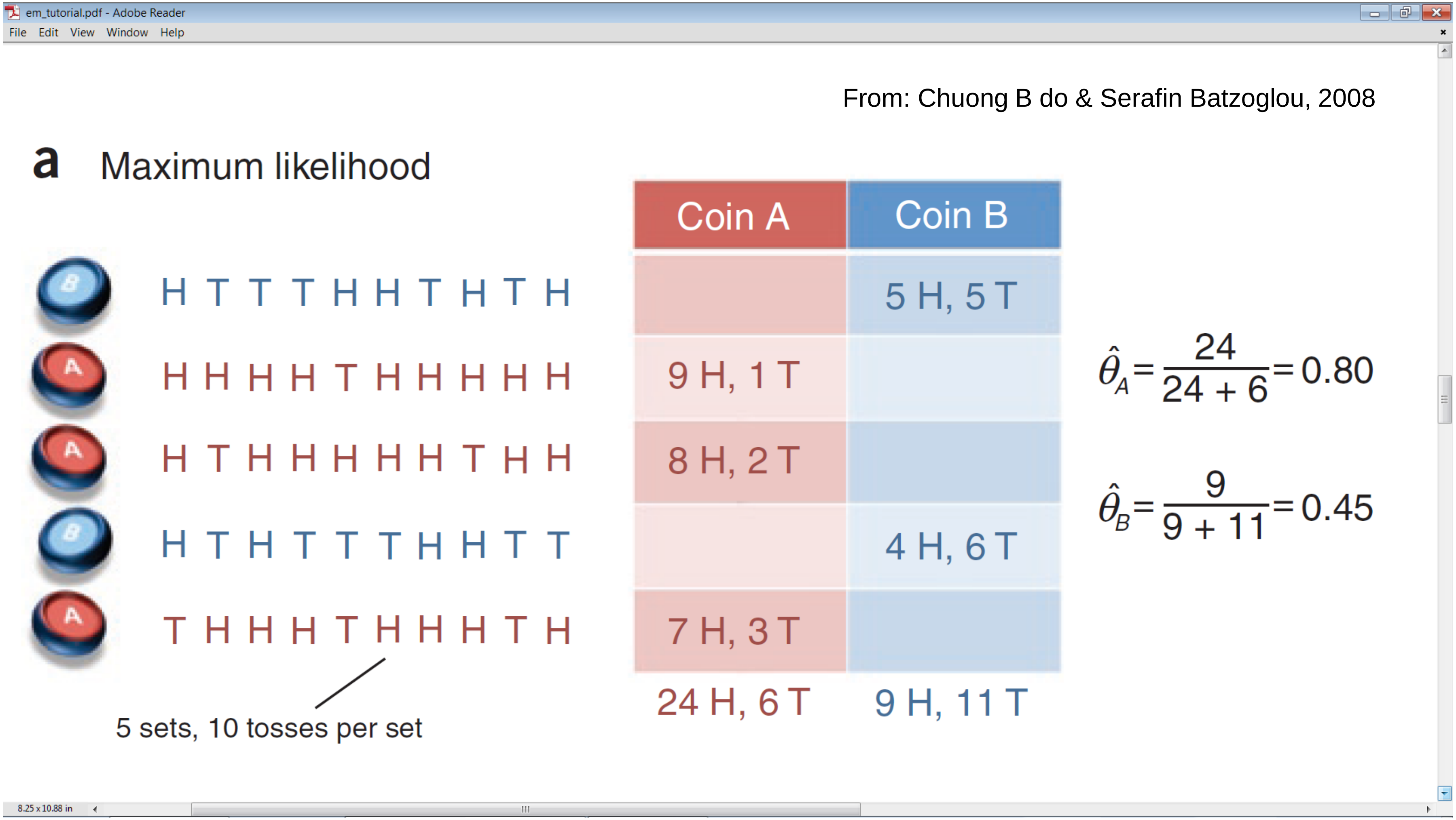
- Alignment a, a is a latent variable
- Incomplete data
- Learning from incomplete data

- Up to know, we always have learned from complete data
- Maximum Likelihood Estimation (MLE)
- Maximises the likelihood of the data

- Now parts of the data missing:
- Expectation Maximisation (EM)

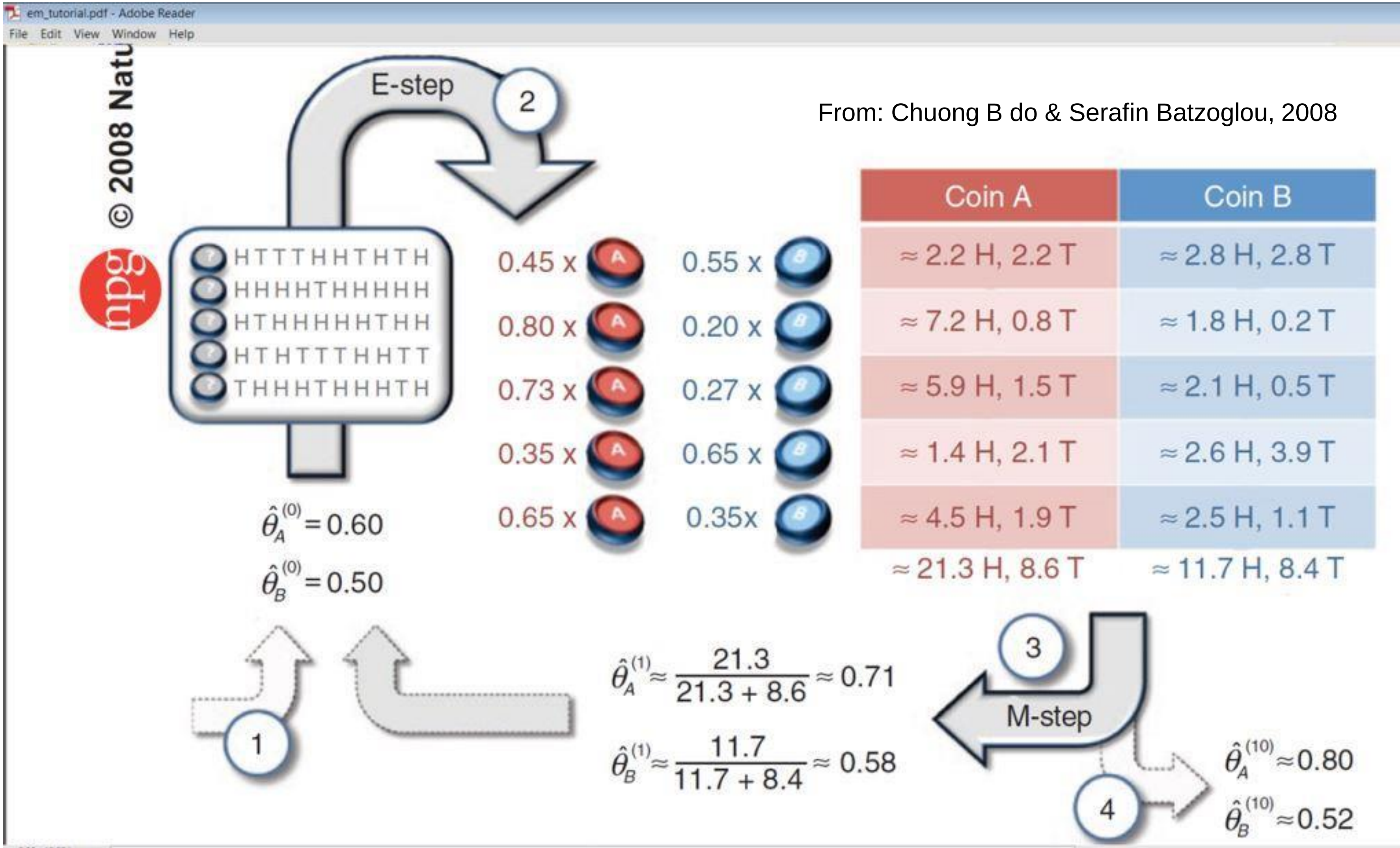
Expectation Maximisation (EM)

UNIVERSITÄT DES SAARLANDES



Expectation Maximisation (EM)

UNIVERSITÄT DES SAARLANDES



From: Chuong B do & Serafin Batzoglou, 2008

Expectation Maximisation (EM)

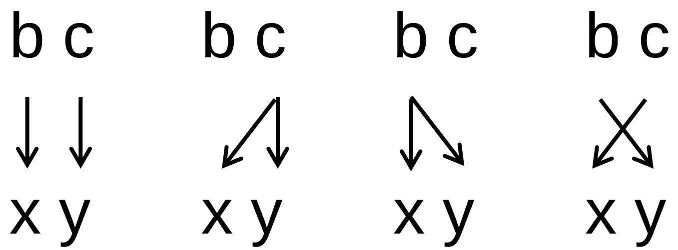
- EM is a bit like magic ☺
- Kind of reduces incomplete data setting to complete data
- Converges
- But not perfect:
- Only local maximum
- A few other constraints
- But very common:
  - e.g. Baum-Welsh for estimating HMMs
  - Many others

Translation model $P(f|e)$

- In estimating IBM Model-3 $P(f|e)$ parameters the latent variable is the alignment a
- Given no further information/knowledge what is our best guess about a?
- Best here means least bias …

- As starting point
- We have to assume that given a sentence pair a can align any word with any other word
- That is many alignments …

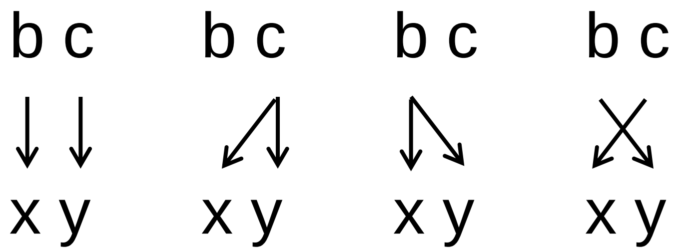
Many alignments/All alignments



To handle parameter estimation if you assume multiple/all alignments you could use fractional counts

$$t(x|b) = \frac{\#(b \to x)}{\#b}$$

Many alignments/All alignments



w_1	w_2	w_3	w_4
0.3	0.2	0.4	0.1

- Alignments may also have weights w associated with them
- Some more important than others …
- These weights are then reflected in the counts for estimating parameters:

$$n(1|b) = \frac{0.3 + 0.1}{0.3 + 0.1 + 0.2 + 0.4}$$

Probability of alignment a

- Weights are just one step away from probabilities
- Probability of alignment a given e and f:

$$P(a|e,f) = \frac{P(a,f|e)}{P(f|e)}$$

- What makes one a better than another?
- If e.g. many words aligned are likely translations of each other
- i.e. have high t parameter values

Probability of alignment a

$$P(a|e,f) = \frac{P(a,f|e)}{P(f|e)}$$

- Need to compute $P(a,f|e)$ and $P(f|e)$
- $P(a,f|e)$ … Model 3: the generative story gives you f and a
- In a way a is a summary of the choices in Model 3
- $P(f|e)$ … given an e, (many) alignments a may give you same f

$$P(f|e) = \sum_{a} P(a,f|e)$$

- Both $P(a,f|e)$ and $P(f|e)$ reduced to $P(a,f|e)$

$P(a, f|e)$

- $P(a, f|e)$ is a product of a bunch of smaller probabilities (parameters)
- For each source word e_i choosing fertility n, a translation t and a target position d:

$$P(a, f|e) = \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right) \times \prod_{j=1}^{m} d(j|a_j, l, m)$$

- l is the length of the English Sentence
- m is the length of the French Sentence
- In case you forgot: we are translating “backwards” $t\left(f_j \middle| e_{a_j}\right)$ because of Noisy Channel Model …

$P(a,f|e)$ a few Refinements

UNIVERSITÄT DES SAARLANDES

$$P(a,f|e) = \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j\middle|e_{a_j}\right) \times \prod_{j=1}^{m} d(j|a_j,l,m)$$

- d should only apply to French words generated by real English words, and not by $NULL$: $\prod_{j:a_j\neq 0}^{m} d(j|a_j,l,m)$
- Need to include costs for φ_0 "spurious" $NULL$ generated French words: there are $m - \varphi_0$ non-spurious French words, hence $\binom{m-\varphi_0}{\varphi_0}$ ways/positions of generating "spurious" words
- What about the costs for this? For φ_0 "spurious" words: ${p_1}^{\varphi_0}$ For the $(m - \varphi_0 - \varphi_0)$ don't add spurious: ${p_0}^{(m-2\varphi_0)}$

$P(a, f|e)$ a few Refinements

$$P(a, f|e) = \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right) \times \prod_{j=1}^{m} d(j|a_j, l, m)$$

- We have no costs for permuting spurious French words into their target positions: d should only apply to French words generated by real English words, and not by $NULL$: $\prod_{j:a_j \neq 0}^{m} d(j|a_j, l, m)$
- Once we have generated φ_0 spurious words we have $\varphi_0!$ ways of permuting them, each of them with a probability of $\frac{1}{\varphi_0!}$

$P(a,f|e)$ a few Refinements

$$P(a,f|e) = \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right) \times \prod_{j=1}^{m} d(j|a_j,l,m)$$

- There is a final problem: the alignment loses a bit of info about how e can get turned into f by the generative process:
- If English x is connected to both French z and y, a doesn't tell us whether they were generated in that order, or as y followed by z and then permuted … similarly for when x is connected with three (or more) French words
- We add a factor

$$\prod_{i=0}^{l} \varphi_i\,!$$

$P(a,f|e)$ in full glory:

$$P(a,f|e) = \binom{m-\varphi_0}{\varphi_0} \times {p_0}^{(m-2\varphi_0)} \times {p_1}^{\varphi_0}$$

$$\times \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j\middle|e_{a_j}\right)$$

$$\times \prod_{j:a_j \neq 0}^{m} d\left(j\middle|a_j,l,m\right) \times \prod_{i=0}^{l} \varphi_i! \times \frac{1}{\varphi_0!}$$

$P(a, f|e)$ in full glory:

$$P(a, f|e) = \binom{m - \varphi_0}{\varphi_0} \times p_0^{(m-2\varphi_0)} \times p_1^{\varphi_0} \times \prod_{i=1}^{l} n(\varphi_i|e_i) \times \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right) \times \prod_{j:a_j \neq 0}^{m} d\left(j \middle| a_j, l, m\right) \times \prod_{i=0}^{l} \varphi_i! \times \frac{1}{\varphi_0!}$$

Recall that

$$P(f|e) = \sum_{a} P(a, f|e) \quad \text{and} \quad P(a|e, f) = \frac{P(a, f|e)}{\sum_a P(a, f|e)}$$

Chicken and Eggs

- If we have parameters we can compute alignments
- If we have alignments we can compute parameters

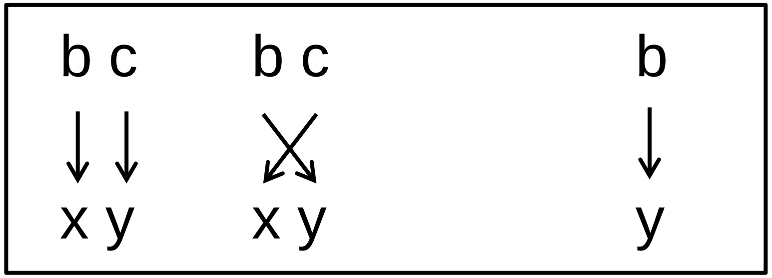
$$P(f|e) = \sum_a P(a, f|e) \quad \text{and} \quad P(a|e, f) = \frac{P(a, f|e)}{\sum_a P(a, f|e)}$$

Expectation Maximisation to the Rescue

UNIVERSITÄT DES SAARLANDES

- Start with uniform parameter values
  - Let French vocab = $40{,}000$ words
  - Then $t(f|e) = \frac{1}{40{,}000}$ for each e
  - $d(4|4{,}10{,}10) = \frac{1}{10}$
  - Pick random value for p_1, say 0.15
- With this we can compute alignment probabilities a for each pair of sentences
- Collect fractional counts, normalise => better parameter values => better alignment probabilities => revised parameter values => and so on …

A two Sentence Example

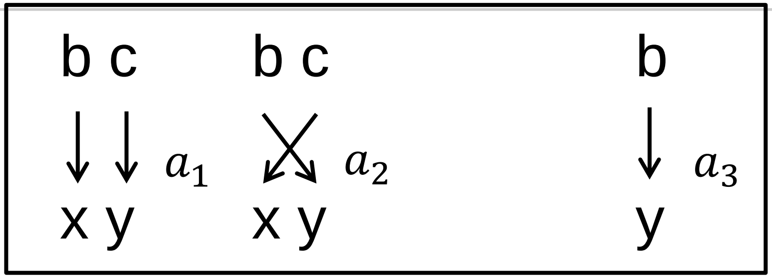


- Simplifications: no $NULL$, $\varphi = 1$ (always), no d
- Only t impacts on a
- So $P(a, f|e)$ reduces to

$$P(a, f|e) = \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right)$$

- ~ IBM Model 1 (except IBM Model 1 has $NULL$)

A two Sentence Example



Here goes EM … (Round 1)

- Step 1: uniform parameters

$$t(x|b) = t(y|b) = t(x|c) = t(y|c) = \frac{1}{2}$$

$$P(a, f|e) = \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right)$$

- Step 2: compute $P(a, f|e)$ for all alignments

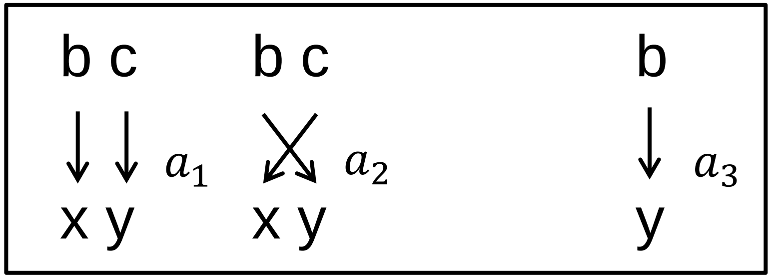
$$P(a_1, f|e) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} \quad P(a_2, f|e) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4} \quad P(a_3, f|e) = \frac{1}{2}$$

$$P(a|e, f) = \frac{P(a, f|e)}{\sum_a P(a, f|e)}$$

- Step 3: normalise $P(a, f|e)$ to get $P(a|f, e)$

$$P(a_1|f, e) = \frac{\frac{1}{4}}{\frac{2}{4}} = \frac{1}{2} \quad P(a_2|f, e) = \frac{\frac{1}{4}}{\frac{2}{4}} = \frac{1}{2} \quad P(a_3|f, e) = \frac{\frac{1}{2}}{\frac{1}{2}} = 1$$

A two Sentence Example



Here goes EM … (Round 1)

- Step 4: collect fractional counts (weighted by alignment probabilities from Step 3)

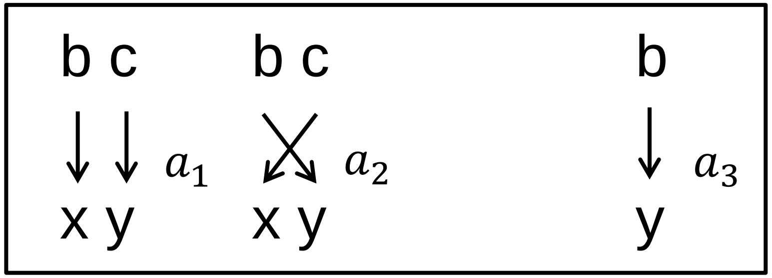
$$\#t(x|b) = \frac{1}{2} \quad \#t(y|b) = \frac{1}{2} + 1 = \frac{3}{2} \quad \#t(x|c) = \frac{1}{2} \quad \#t(y|c) = \frac{1}{2}$$

- Step 5: normalise fractional counts to get revised parameters t

$$t(x|b) = \frac{\frac{1}{2}}{\frac{4}{2}} = \frac{1}{4} \quad t(y|b) = \frac{\frac{3}{2}}{\frac{4}{2}} = \frac{3}{4} \quad t(x|c) = \frac{1}{2}/1 = \frac{1}{2} \quad t(y|c) = \frac{1}{2}/1 = \frac{1}{2}$$

Normalised by sum of factional counts from Step 4 …

A two Sentence Example



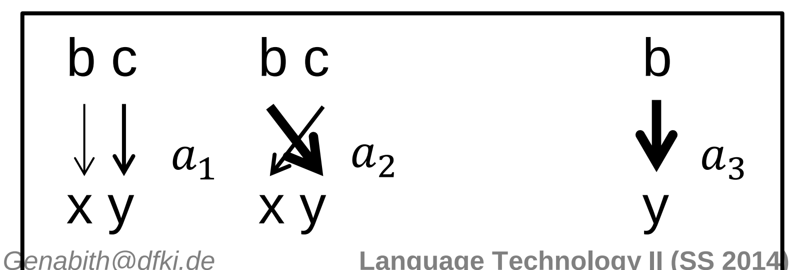
Here goes EM … (Round 1)

- Step 4: collect fractional counts (weighted by alignment probabilities from Step 3)

$$\#t(x|b) = \frac{1}{2} \quad \#t(y|b) = \frac{1}{2} + 1 = \frac{3}{2} \quad \#t(x|c) = \frac{1}{2} \quad \#t(y|c) = \frac{1}{2}$$

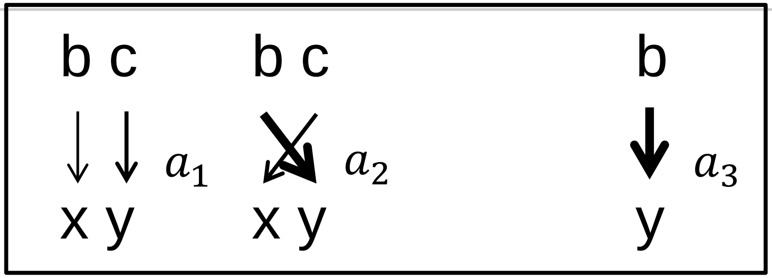
- Step 5: normalise fractional counts to get revised parameters t

$$t(x|b) = \frac{\frac{1}{2}}{\frac{4}{2}} = \frac{1}{4} \quad t(y|b) = \frac{\frac{3}{2}}{\frac{4}{2}} = \frac{3}{4} \quad t(x|c) = \frac{1}{2}/1 = \frac{1}{2} \quad t(y|c) = \frac{1}{2}/1 = \frac{1}{2}$$



A two Sentence Example

UNIVERSITÄT DES SAARLANDES



Here goes EM … (Round 2)

$$P(a,f|e) = \prod_{j=1}^{m} t\left(f_j \middle| e_{a_j}\right)$$

- Repeat Step 2: compute $P(a,f|e)$ for all alignments

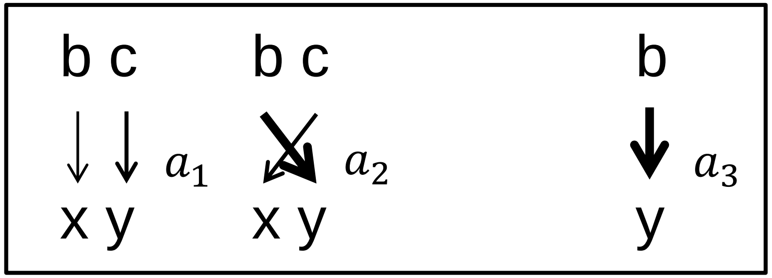
$$P(a_1,f|e) = \frac{1}{4} \times \frac{1}{2} = \frac{1}{8} \quad P(a_2,f|e) = \frac{3}{4} \times \frac{1}{2} = \frac{3}{8} \quad P(a_3,f|e) = \frac{3}{4}$$

$$P(a|e,f) = \frac{P(a,f|e)}{\sum_a P(a,f|e)}$$

- Repeat Step 3: normalise $P(a,f|e)$ to get $P(a|f,e)$

$$P(a_1|f,e) = \frac{\frac{1}{8}}{\frac{4}{8}} = \frac{1}{4} \quad P(a_2|f,e) = \frac{\frac{3}{8}}{\frac{4}{8}} = \frac{3}{4} \quad P(a_3|f,e) = \frac{\frac{3}{4}}{\frac{3}{4}} = 1$$

A two Sentence Example



Here goes EM … (Round 2)

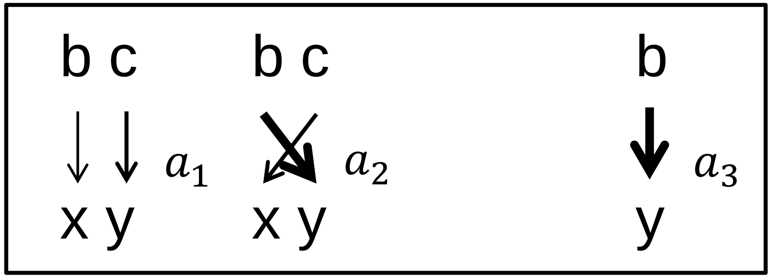
- Repeat Step 4: collect fractional counts (weighted by alignment probabilities from Repeat Step 3):

$$\#t(x|b) = \frac{1}{4} \quad \#t(y|b) = \frac{3}{4} + 1 = \frac{7}{4} \quad \#t(x|c) = \frac{3}{4} \quad \#t(y|c) = \frac{1}{4}$$

- Repeat Step 5: normalise fractional counts to get revised parameters t

$$t(x|b) = \frac{\frac{1}{4}}{\frac{8}{4}} = \frac{1}{8} \quad t(y|b) = \frac{\frac{7}{4}}{\frac{8}{4}} = \frac{7}{8} \quad t(x|c) = \frac{3}{4}/1 = \frac{3}{4} \quad t(y|c) = \frac{1}{4}/1 = \frac{1}{4}$$

A two Sentence Example



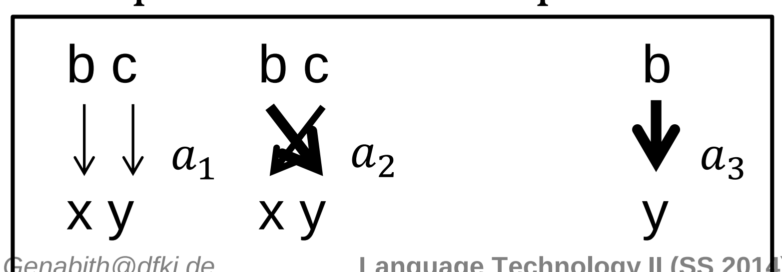
Here goes EM … (Round 2)

- Repeat Step 4: collect fractional counts (weighted by alignment probabilities from Step 3):

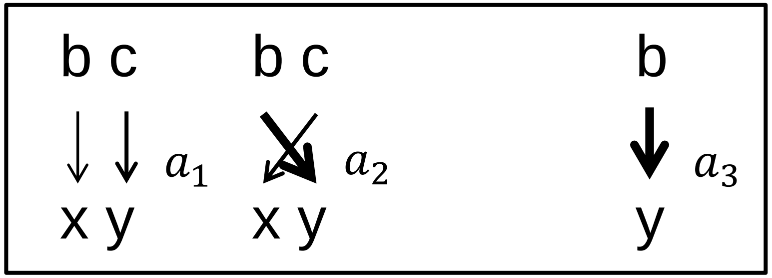
$$\#t(x|b) = \frac{1}{4} \quad \#t(y|b) = \frac{3}{4} + 1 = \frac{7}{4} \quad \#t(x|c) = \frac{3}{4} \quad \#t(y|c) = \frac{1}{4}$$

- Repeat Step 5: normalise fractional counts to get revised parameters t

$$t(x|b) = \frac{\frac{1}{4}}{\frac{8}{4}} = \frac{1}{4} \quad t(y|b) = \frac{\frac{7}{4}}{\frac{8}{4}} = \frac{7}{8} \quad t(x|c) = \frac{3}{4}/1 = \frac{3}{4} \quad t(y|c) = \frac{1}{4}/1 = \frac{1}{4}$$



A two Sentence Example



Here goes EM …

- Repeating Steps 2 – 5 many times:

$$t(x|b) = 0.0001 \quad t(y|b) = 0.9999 \quad t(x|c) = 0.9999 \quad t(y|c) = 0.0001$$