Machine Translation

CSE 447 / 517

March 3rd, 2022 (Week 9)

Logistics

- A8 due is due **tomorrow** (Friday, March 4th)

Agenda

- Beam Search
- IBM Model 1
  - EM algorithm
- IBM Model 2
- Quiz 8
- Q & A

Beam search decoding

- Core idea: On each step of decoder, keep track of the *k* most probable partial translations (which we call *hypotheses*)
  - *k* is the beam size (in practice around 5 to 10)

- A hypothesis $y_1, \ldots, y_t$ has a score which is its log probability:

$$\text{score}(y_1, \ldots, y_t) = \log P_{\text{LM}}(y_1, \ldots, y_t | x) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

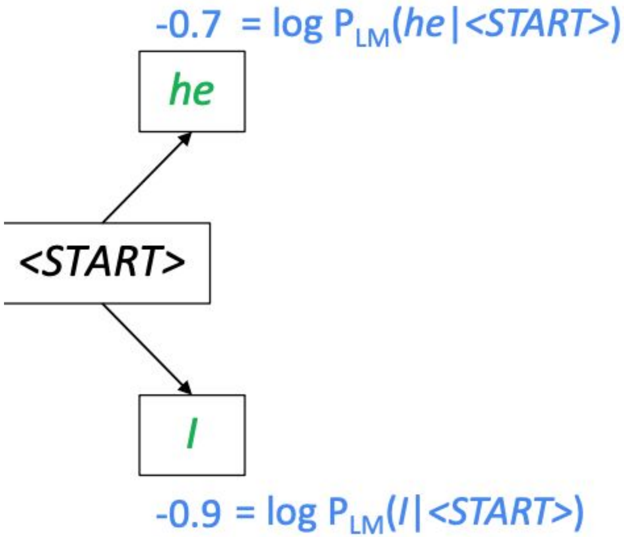
  - Scores are all negative, and higher score is better
  - We search for high-scoring hypotheses, tracking top *k* on each step

- Beam search is not guaranteed to find optimal solution
- But much more efficient than exhaustive search!

Slides from Abigail See

Beam search decoding: example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

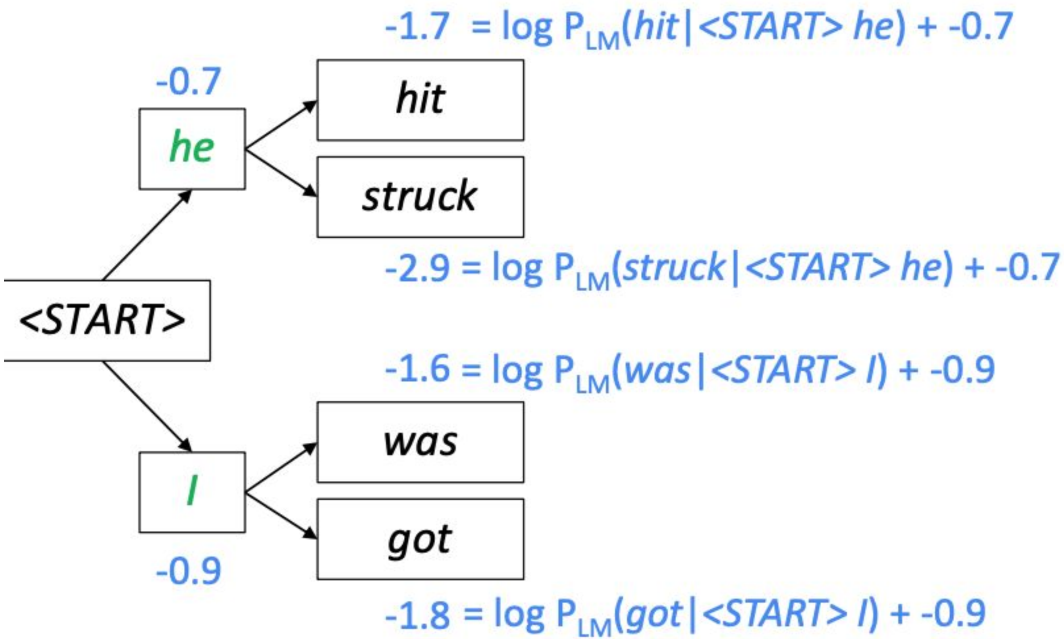


Slides from Abigail See

Beam search decoding: example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

For each of the *k* hypotheses, find top *k* next words and calculate scores

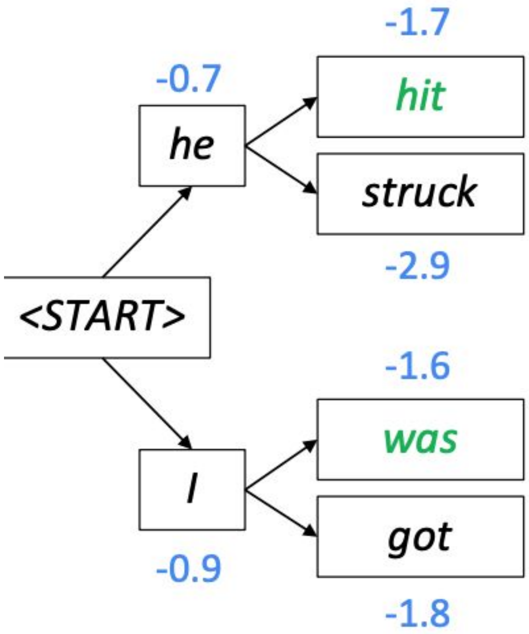


Slides from Abigail See

Beam search decoding: example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$

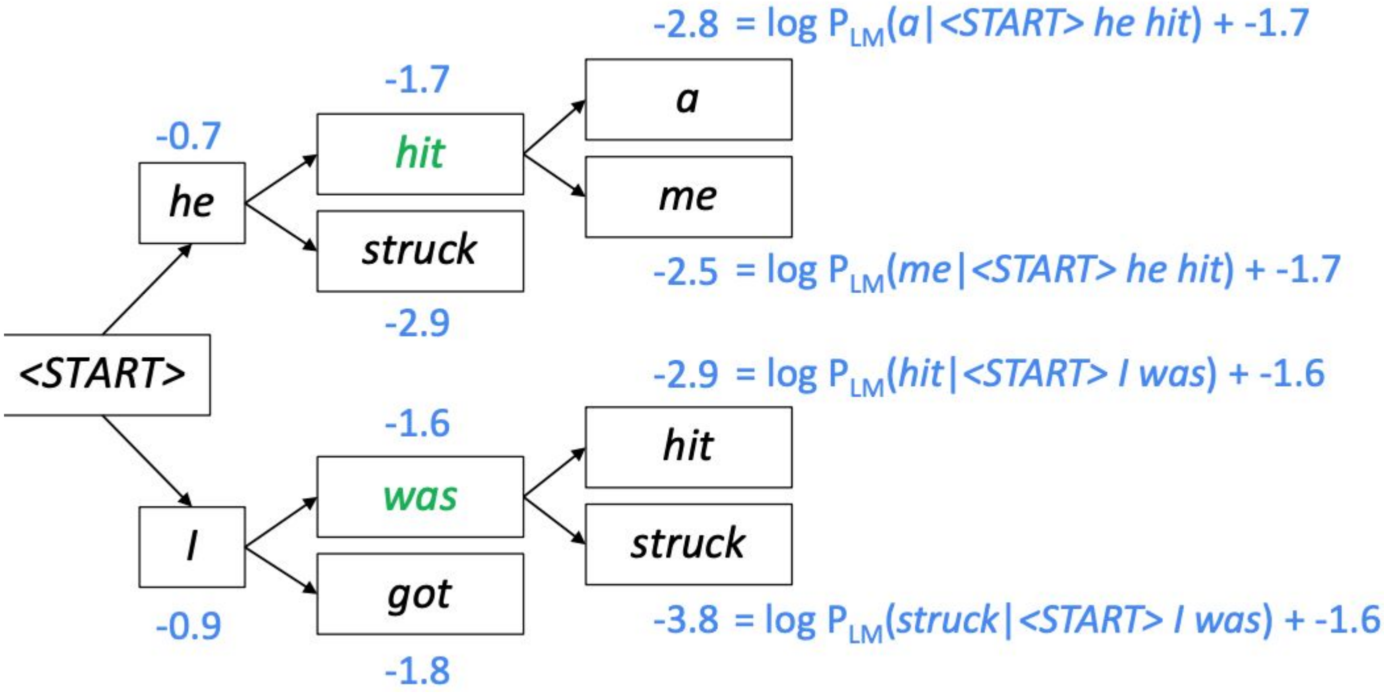
Of these k^2 hypotheses, just keep *k* with highest scores



Slides from Abigail See

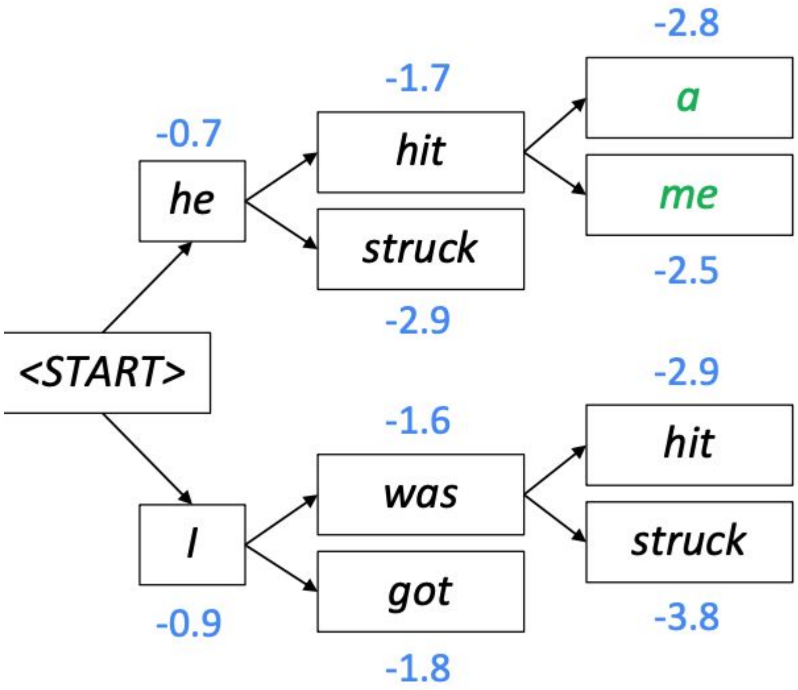
Beam search decoding: example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



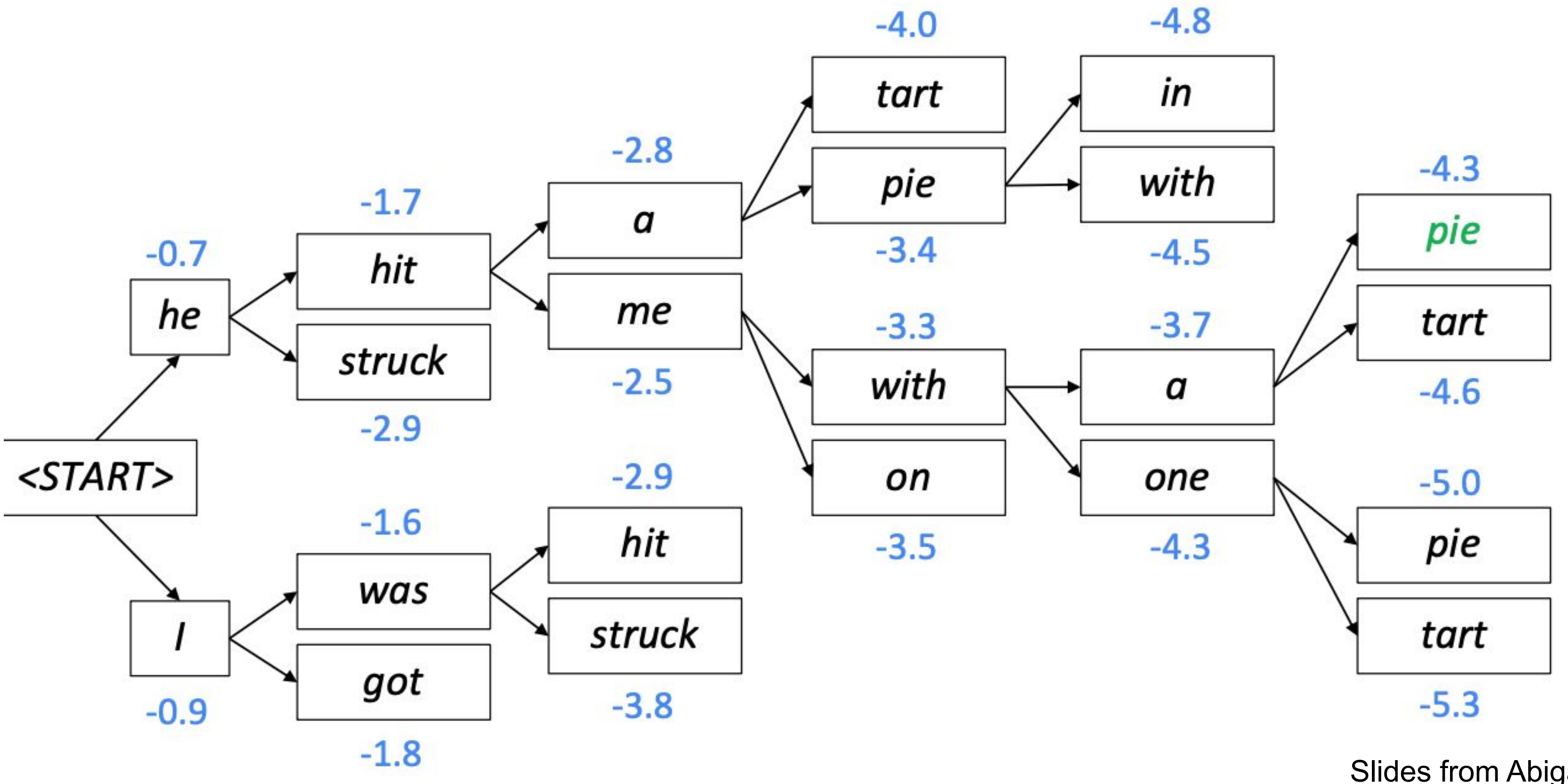
Beam search decoding: example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Beam search decoding: example

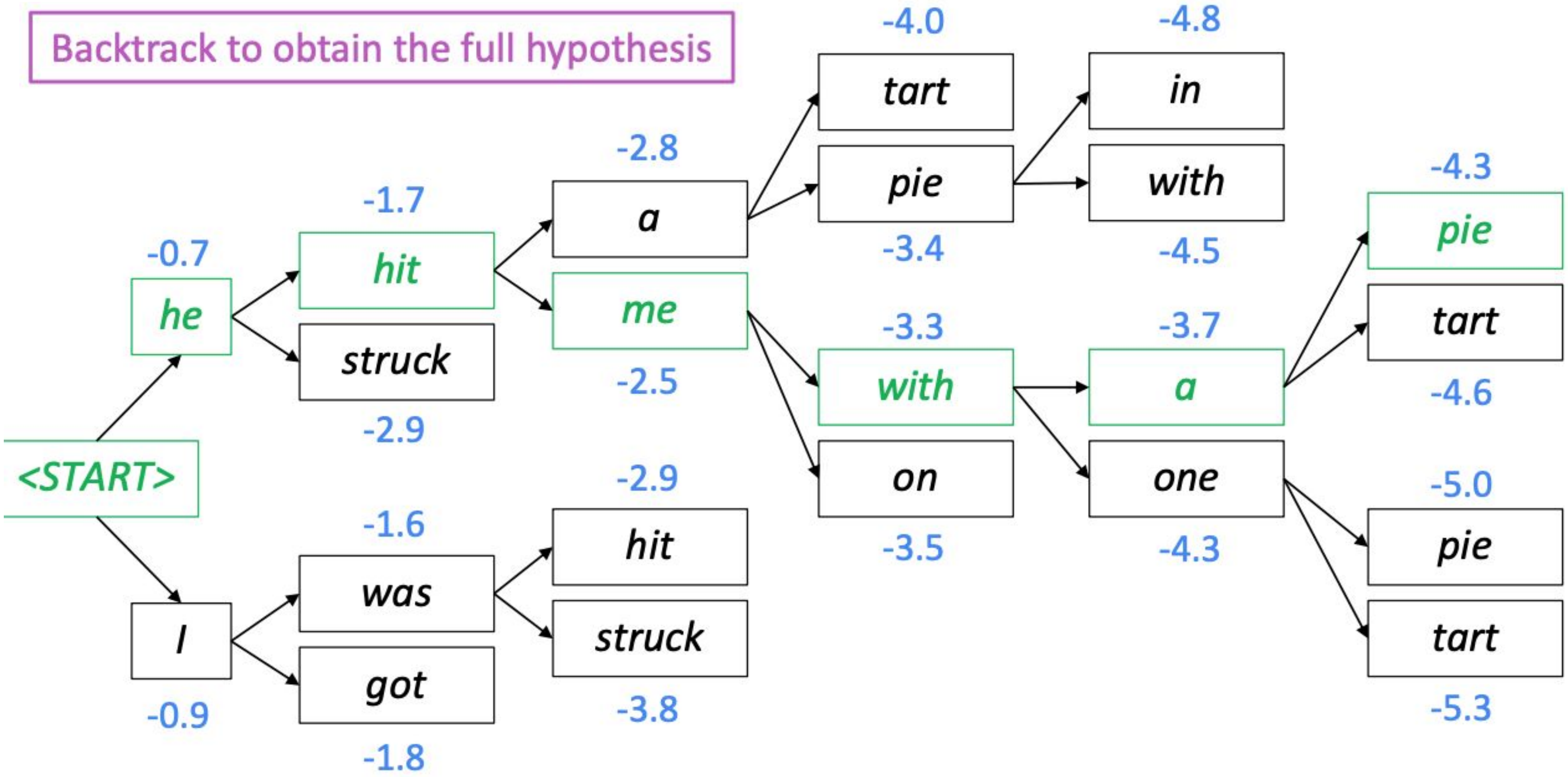
Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Slides from Abigail See

Beam search decoding: example

Beam size = k = 2. Blue numbers = $\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$



Quiz 8: FOL

Select **all** the correct translation of following sentences into FOL using the key below:

G: ... is guilty

C: ... is a criminal

L: ... loves...

Quiz 8: FOL

Not every criminal is innocent.

$$\neg\forall x\,(C(x) \Rightarrow \neg G(x))$$

$$\exists x\,(C(x) \wedge G(x))$$

Quiz 8: FOL

Not every criminal is innocent.

$$\neg\forall x \left(C(x) \Rightarrow \neg G(x)\right)$$

$$\exists x \left(C(x) \land G(x)\right)$$

They are logically equivalent – so both is correct

Quiz 8: FOL

Nobody loves anybody who loves nobody.

$$\forall x\,(\forall y \neg L\,(x,y) \Rightarrow \forall z \neg L\,(z,\ x))$$

$$\forall x\,(\forall y \neg L\,(x,y) \Rightarrow \neg \exists z L\,(z,x))$$

$$\forall x\,(\forall y \neg L\,(x,y) \Rightarrow \neg \forall z L\,(z,x))$$

Quiz 8: FOL

Nobody loves anybody who loves nobody.

$$\forall x\,(\forall y \neg L(x,y) \Rightarrow \forall z \neg L(z,\ x))$$

$$\forall x\,(\forall y \neg L(x,y) \Rightarrow \neg \exists z L(z,x))$$

$$\forall x\,(\forall y \neg L(x,y) \Rightarrow \neg \forall z L(z,x))$$

Logically equivalent, both are correct

NLP Task: Machine Translation

Mr President , Noah's ark was filled not with production factors , but with living creatures .

(***From Language X***)

Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

(***To Language Y***)

The Noisy Channel Model

Language X ⇨ ***Language Y***

We want to translate ***Language X*** into ***Language Y***.

The Noisy Channel Model

Language X → ***Language Y***

We want to translate ***Language X*** into ***Language Y***.

Source → ***Language Y*** → Channel → ***Language X***

Imagine there is a source that generates ***Language Y***. But then it is passed through some channel, and we observe ***Language X*** on the other side of the channel.

The Noisy Channel Model

Imagine there is a source that generates ***Language Y***. But then it is passed through some channel, and we observe ***Language X*** on the other side of the channel.

$$y^* = \text{argmax}_y \; p(y \mid x)$$
$$= \text{argmax}_y \; p(x \mid y) \cdot \boxed{p(y)}$$

Source model aka a LM for Language Y! This captures the fluency in the target language.

The Noisy Channel Model

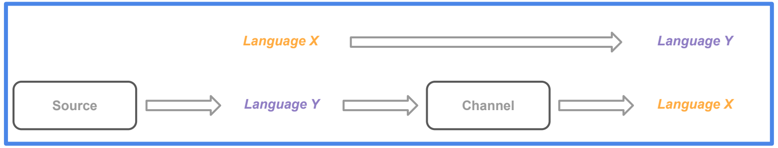
Imagine there is a source that generates ***Language Y***. But then it is passed through some channel, and we observe ***Language X*** on the other side of the channel.

$$y^* = \operatorname{argmax}_y \, p(y \mid x)$$

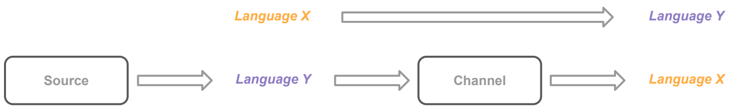
$$= \operatorname{argmax}_y \, p(x \mid y) \cdot p(y)$$

Channel model, captures the faithfulness of the translation.

The Noisy Channel Model



Refer to this when you get lost which is which!



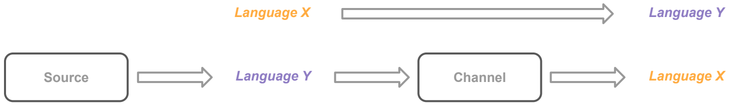
IBM Model 1 - Motivation

Mr President , Noah's ark was filled not with production factors , but with living creatures .

Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

. . . .

IBM Model 1: What is the mapping from each token in ***Language X*** to ***Language Y***?



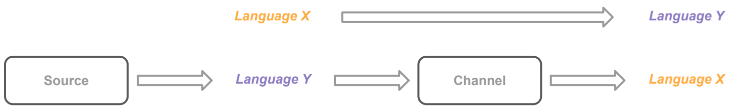
IBM Model 1 - Alignment

IBM Model 1: What is the mapping from each token in ***Language X*** to ***Language Y***?

Let **l** be the length of **y** and **m** be the length of **x**.

Latent variable $a = \langle a_1,...,a_m \rangle$, each a_i ranging over **{0,...,l}** (positions in **y**).

$$a_i = j$$



IBM Model 1 - Alignment

IBM Model 1: What is the mapping from each token in ***Language X*** to ***Language Y***?

Let **l** be the length of **y** and **m** be the length of **x**.

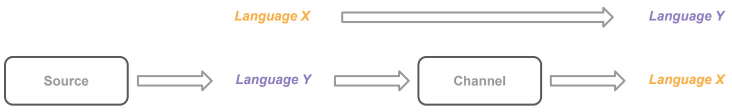
Latent variable a = $\langle a_1,\ldots,a_m \rangle$, each a_i ranging over **{0,...,l}** (positions in **y**).

$$a_i = j$$

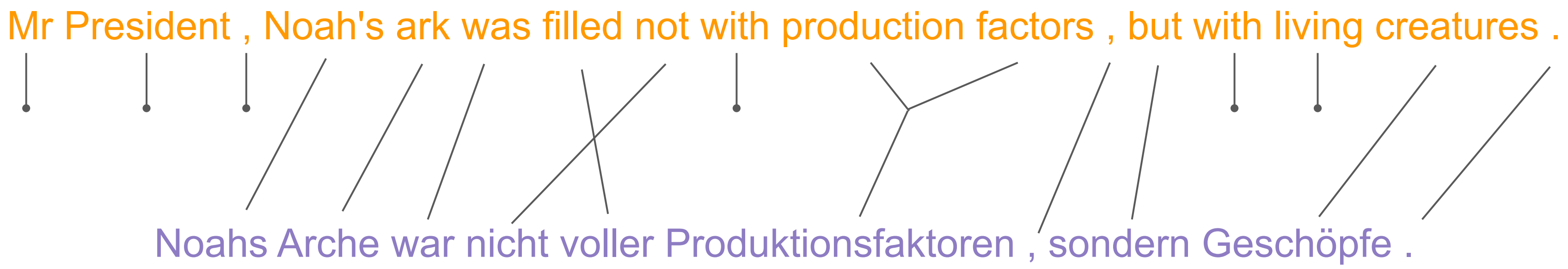
The ith token in Language X.

The jth token in Language Y.

IBM Model 1

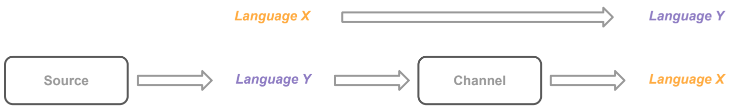


$a_i = j$

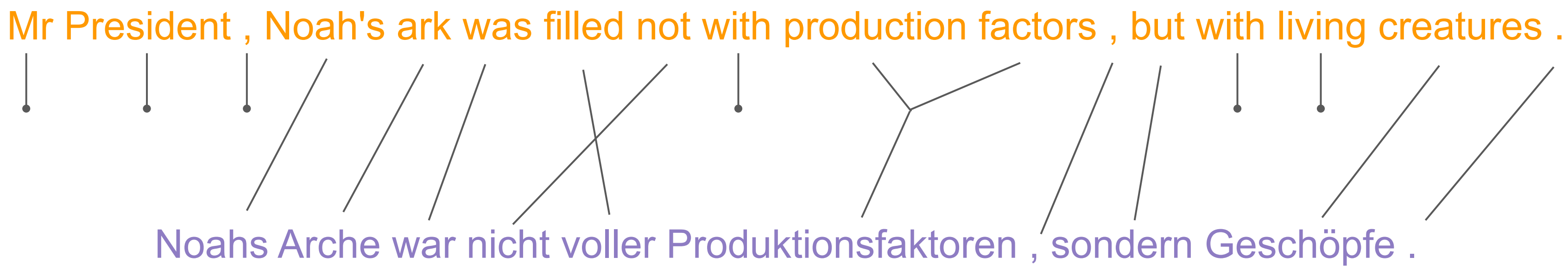


IBM Model 1: What is the mapping from each token in ***Language X*** to ***Language Y***?

IBM Model 1



$a_i = j$

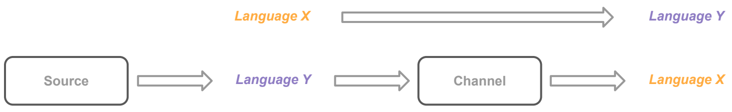


IBM Model 1: What is the mapping from each token in ***Language X*** to ***Language Y***?

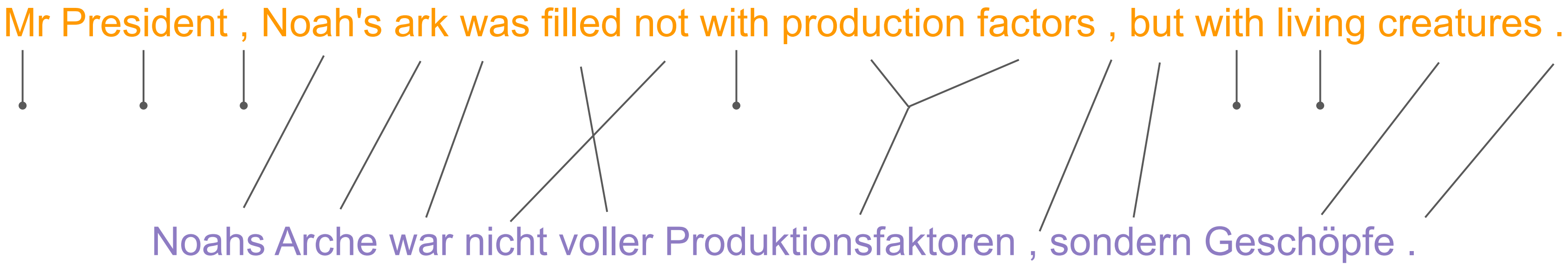
1 2 3 4

a = [0, 0, 0, 1, **???**]

IBM Model 1



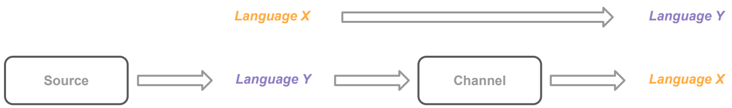
$a_i = j$



IBM Model 1: What is the mapping from each token in ***Language X*** to ***Language Y***?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	1	2	3	5	4	0	6	6	7	8	0	0	9	10

$$\mathbf{a} = [0, 0, 0, 1, 2, 3, 5, 4, 0, 6, 6, 7, 8, 0, 0, 9, 10]$$



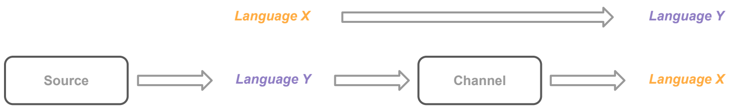
$a_i = j$

IBM Model 1

Our channel model:

$$p(\mathbf{x} \mid \mathbf{y}, m; \theta) = \sum_{\mathbf{a} \in \{0,..l\}^m} p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta)$$

IBM Model 1

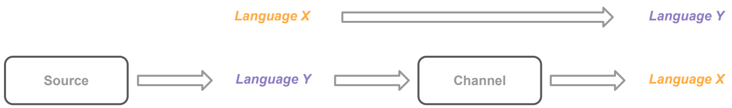


$a_i = j$

Our channel model:

$$p(\mathbf{x} \mid \mathbf{y}, m; \theta) = \sum_{\mathbf{a} \in \{0, ..l\}^m} p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta)$$

Marginalized over all possible **a** vectors.



$a_i = j$

IBM Model 1

Our channel model:

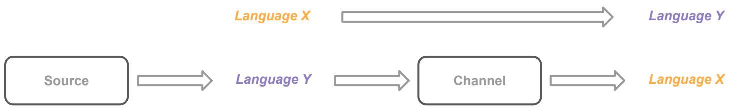
$$p(\mathbf{x} \mid \mathbf{y}, m; \theta) = \sum_{\mathbf{a} \in \{0,..l\}^m} p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta)$$

where

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} p(a_i \mid i, l, m) \cdot p(x_i \mid y_{a_i}; \theta)$$

$$= \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

IBM Model 1



$a_i = j$

Our channel model:

$$p(\mathbf{x} \mid \mathbf{y}, m; \theta) = \sum_{\mathbf{a} \in \{0,..l\}^m} p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta)$$

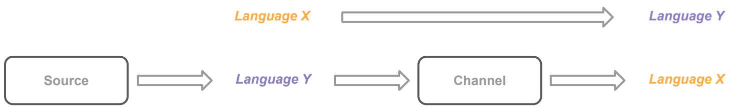
Go through every position in **x**.

where

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} p(a_i \mid i, l, m) \cdot p(x_i \mid y_{a_i}; \theta)$$

$$= \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

IBM Model 1



$a_i = j$

Our channel model:

$$p(\mathbf{x} \mid \mathbf{y}, m; \theta) = \sum_{\mathbf{a} \in \{0,..l\}^m} p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta)$$

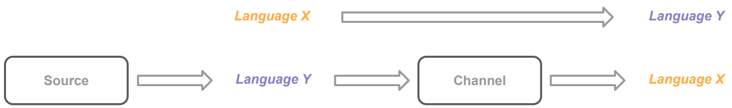
where

How likely is the current alignment *without* regard to the text?

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} p(a_i \mid i, l, m) \cdot p(x_i \mid y_{a_i}; \theta)$$

$$= \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

IBM Model 1



$a_i = j$

Our channel model:

$$p(\mathbf{x} \mid \mathbf{y}, m; \theta) = \sum_{\mathbf{a} \in \{0, ..l\}^m} p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta)$$

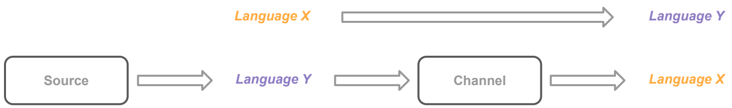
where

How likely is the current alignment *with* regard to the text?

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} p(a_i \mid i, l, m) \, p(x_i \mid y_{a_i}; \theta)$$

$$= \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

IBM Model 1



$a_i = j$

Our channel model:

$$p(\mathbf{x} \mid \mathbf{y}, m; \theta) = \sum_{\mathbf{a} \in \{0,..l\}^m} p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta)$$

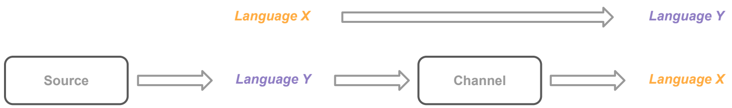
where

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} p(a_i \mid i, l, m) \cdot p(x_i \mid y_{a_i}; \theta)$$

$$= \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

Uniform distribution (all distortions modelled by ***a*** are treated the same).

IBM Model 1



$a_i = j$

Our channel model:

$$p(\mathbf{x} \mid \mathbf{y}, m; \theta) = \sum_{\mathbf{a} \in \{0, ..l\}^m} p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta)$$

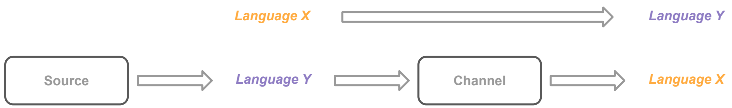
where

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} p(a_i \mid i, l, m) \cdot p(x_i \mid y_{a_i}; \theta)$$

$$= \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

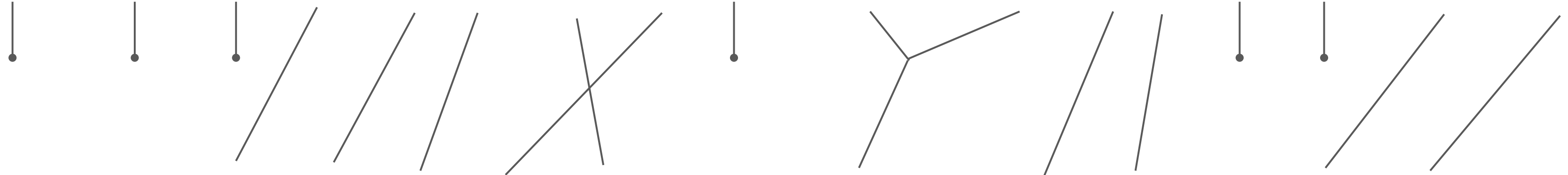
Learned parameter.

IBM Model 1



$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

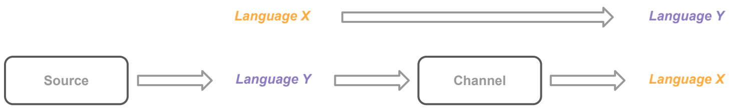
Mr President , Noah's ark was filled not with production factors , but with living creatures .



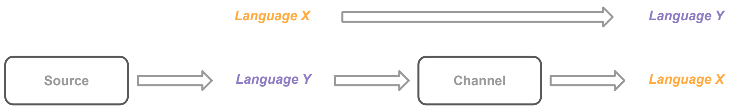
Noahs Arche war nicht voller Produktionsfaktoren , sondern Geschöpfe .

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \frac{1}{1+10} \cdot \theta_{Mr \mid null} + \ldots$$

IBM Model 1 - Learning



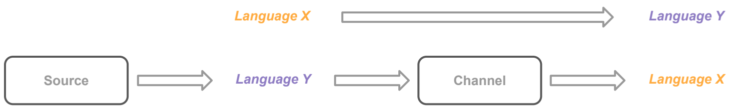
$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$



IBM Model 1 - Learning

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

How do we estimate this?

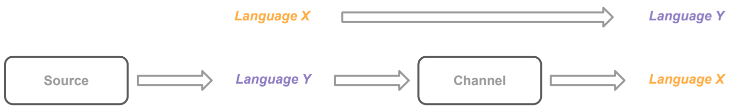


IBM Model 1 - Learning

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

The problem: we don’t know the alignments ahead of time. So we can’t apply MLE to find the parameter.

The solution: expectation maximization.



Expectation Maximization

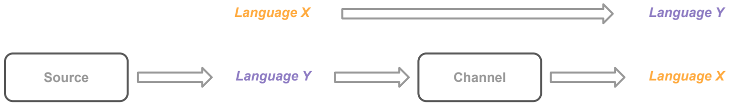
Goal: finding $\theta_{x_i|y_{a_i}}$.

Step 1: initialize $\theta_{x_i|y_{a_i}}$ with some value.

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate "soft" alignments.

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

Step 4: repeat from 2!



Expectation Maximization

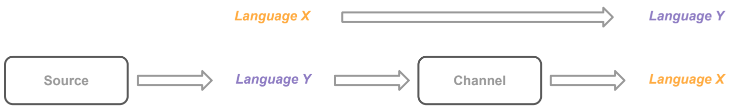
Goal: finding $\theta_{x_i|y_{a_i}}$.

Step 1: initialize $\theta_{x_i|y_{a_i}}$ with some value. 👍

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate “soft” alignments.

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

Step 4: repeat from 2!



Expectation Maximization

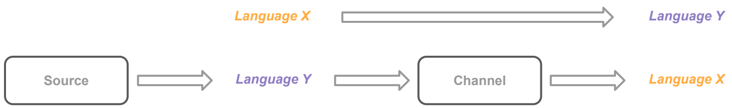
Goal: finding $\theta_{x_i|y_{a_i}}$.

Step 1: initialize $\theta_{x_i|y_{a_i}}$ with some value.

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate "soft" alignments.

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

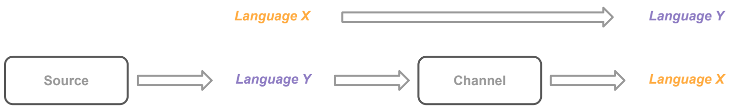
Step 4: repeat from 2!



Expectation Maximization

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate "soft" alignments: $q_i(j) = p(a_i = j; \theta)$.

$$q_i(j) \leftarrow \frac{\theta_{x_i|y_j}}{\sum_{j'=1}^{l} \theta_{x_i|y_{j'}}}$$

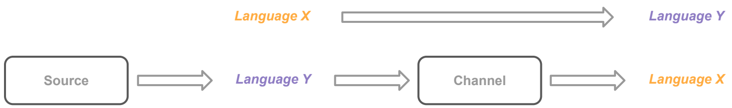


Expectation Maximization

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate "soft" alignments: $q_i(j) = p(a_i = j; \theta)$.

What is the likelihood of generating x_i given the y_j?

$$q_i(j) \leftarrow \frac{\theta_{x_i|y_j}}{\sum_{j'=1}^{l} \theta_{x_i|y_{j'}}}$$



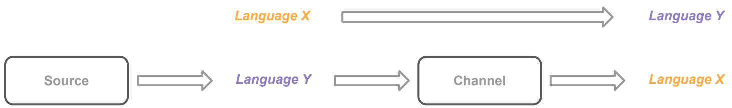
Expectation Maximization

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate “soft” alignments: $q_i(j) = p(a_i = j; \theta)$.

What is the likelihood of generating x_i given the y_j?

$$q_i(j) \leftarrow \frac{\theta_{x_i|y_j}}{\sum_{j'=1}^{l} \theta_{x_i|y_{j'}}}$$

… out of all possible y_j’ that x_i could be aligned to.

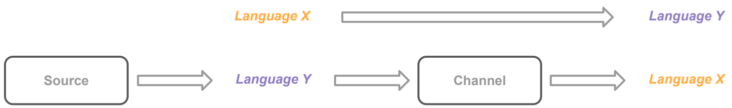


Expectation Maximization

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate “soft” alignments: $q_i(j) = p(a_i = j; \theta)$.

We want a soft assignment for each sample n.

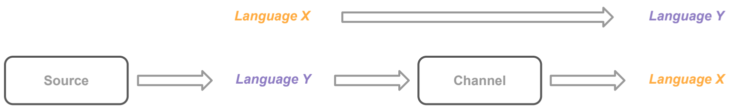
$$q_i^{(n)}(j) \leftarrow \frac{\theta_{x_i^{(n)}|y_j^{(n)}}}{\sum_{j'=1}^{l^{(n)}} \theta_{x_i^{(n)}|y_{j'}^{(n)}}}$$



Expectation Maximization

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate "soft" alignments: $q_i(j) = p(a_i = j; \boldsymbol{\theta})$.

$$q_i^{(n)}(j) \leftarrow \frac{\boldsymbol{\theta}_{x_i^{(n)}|y_j^{(n)}}}{\sum_{j'^{(n)}=1}^{I} \boldsymbol{\theta}_{x_i^{(n)}|y_{j'}^{(n)}}}$$



Expectation Maximization

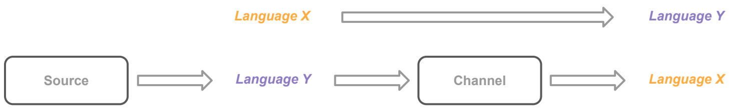
Goal: finding $\theta_{x_i|y_{a_i}}$.

Step 1: initialize $\theta_{x_i|y_{a_i}}$ with some value.

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate “soft” alignments.

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

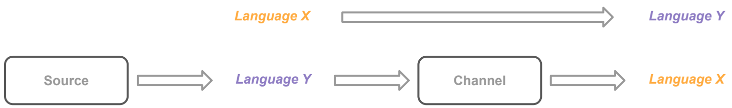
Step 4: repeat from 2!



Expectation Maximization

Step 3: estimate $\theta_{x_i | y_{a_i}}$ with MLE principle.

$$\hat{\theta}_{x|y} \leftarrow \frac{\sum_{n=1}^{N} \sum_{i:x_i^{(n)}=x} \sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}{\sum_{n=1}^{N} \sum_{i=1}^{m^{(n)}} \sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}$$

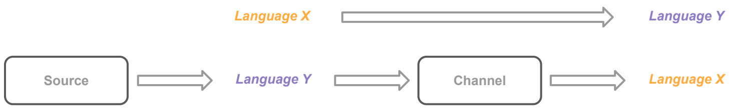


Expectation Maximization

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

Go through all samples in the dataset.

$$\hat{\theta}_{x|y} \leftarrow \frac{\sum_{n=1}^{N}\sum_{i:x_i^{(n)}=x}\sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}{\sum_{n=1}^{N}\sum_{i=1}^{m^{(n)}}\sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}$$

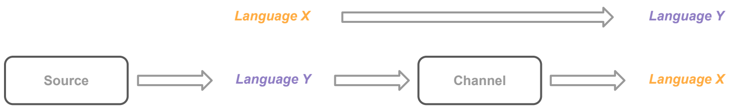


Expectation Maximization

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

Go through each position i that token x appeared.

$$\hat{\theta}_{x|y} \leftarrow \frac{\sum_{n=1}^{N} \sum_{i:x_i^{(n)}=x} \sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}{\sum_{n=1}^{N} \sum_{i=1}^{m^{(n)}} \sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}$$

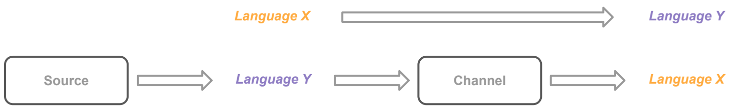


Expectation Maximization

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

How much of the probability mass is assigned to i matching j?

$$\hat{\theta}_{x|y} \leftarrow \frac{\sum_{n=1}^{N}\sum_{i:x_i^{(n)}=x}\sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}{\sum_{n=1}^{N}\sum_{i=1}^{m^{(n)}}\sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}$$

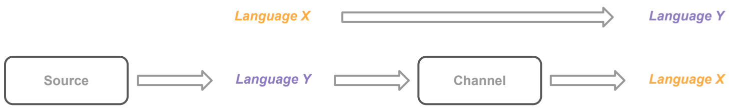


Expectation Maximization

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

$$\hat{\theta}_{x|y} \leftarrow \frac{\sum_{n=1}^{N} \sum_{i:x_i^{(n)}=x} \sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}{\sum_{n=1}^{N} \sum_{i=1}^{m^{(n)}} \sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}$$

Go through all samples in the dataset.

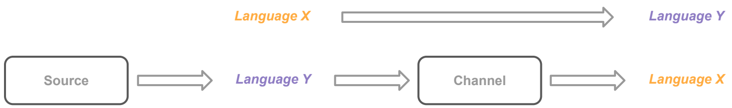


Expectation Maximization

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

$$\hat{\theta}_{x|y} \leftarrow \frac{\sum_{n=1}^{N}\sum_{i:x_i^{(n)}=x}\sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}{\sum_{n=1}^{N}\sum_{i=1}^{m^{(n)}}\sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}$$

Go through every position i.

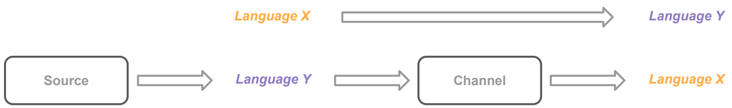


Expectation Maximization

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

$$\hat{\theta}_{x|y} \leftarrow \frac{\sum_{n=1}^{N} \sum_{i:x_i^{(n)}=x} \sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}{\sum_{n=1}^{N} \sum_{i=1}^{m^{(n)}} \sum_{j:y_j^{(n)}=y} q_i^{(n)}(j)}$$

How much probability mass is assigned to the word x matching y?



Expectation Maximization

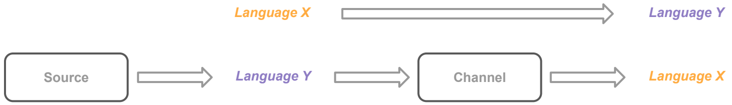
Goal: finding $\theta_{x_i|y_{a_i}}$.

Step 1: initialize $\theta_{x_i|y_{a_i}}$ with some value.

Step 2: use $\theta_{x_i|y_{a_i}}$ to estimate “soft” alignment.

Step 3: estimate $\theta_{x_i|y_{a_i}}$ with MLE principle.

Step 4: repeat from 2!

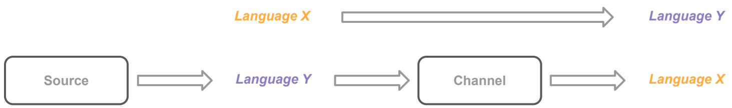


IBM Model 2

Recall IBM Model 1:

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} p(a_i \mid i, l, m) \cdot p(x_i \mid y_{a_i}; \theta)$$

$$= \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$



IBM Model 2

Recall IBM Model 1:

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} p(a_i \mid i, l, m) \cdot p(x_i \mid y_{a_i}; \theta)$$

$$= \prod_{i=0}^{m} \frac{1}{l+1} \cdot \theta_{x_i \mid y_{a_i}}$$

IBM Model 2: removed uniform distortion assumption.

$$p(\mathbf{x}, \mathbf{a} \mid \mathbf{y}, m; \theta) = \prod_{i=0}^{m} = \boxed{\theta^{distortion}_{a_i \mid i, l, m}} \cdot \theta^{translation}_{x_i \mid y_{a_i}}$$

Q & A