Word Alignment Models

Word Alignments

“As English not all languages words in the same order put. Hmmmmmm.» - Yoda

Word alignment task

Given a corpus of **(e, f)** sentence pairs:

- English, source: $e = (e_1, e_2, \ldots e_I)$
- Foreign, target: $f = (f_1, f_2, \ldots f_J)$

Predict:

- Alignments **a** between **e** and **f**:

e: The appetite comes with eating.

a?

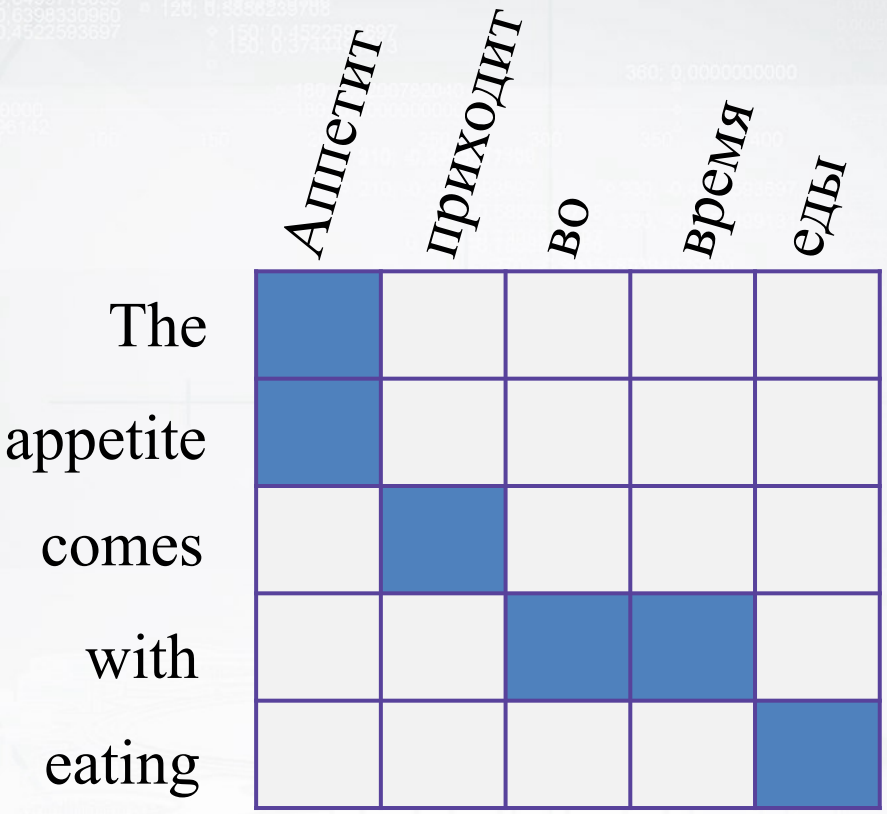
f: Аппетит приходит во время еды.

Recap: Bayes' rule

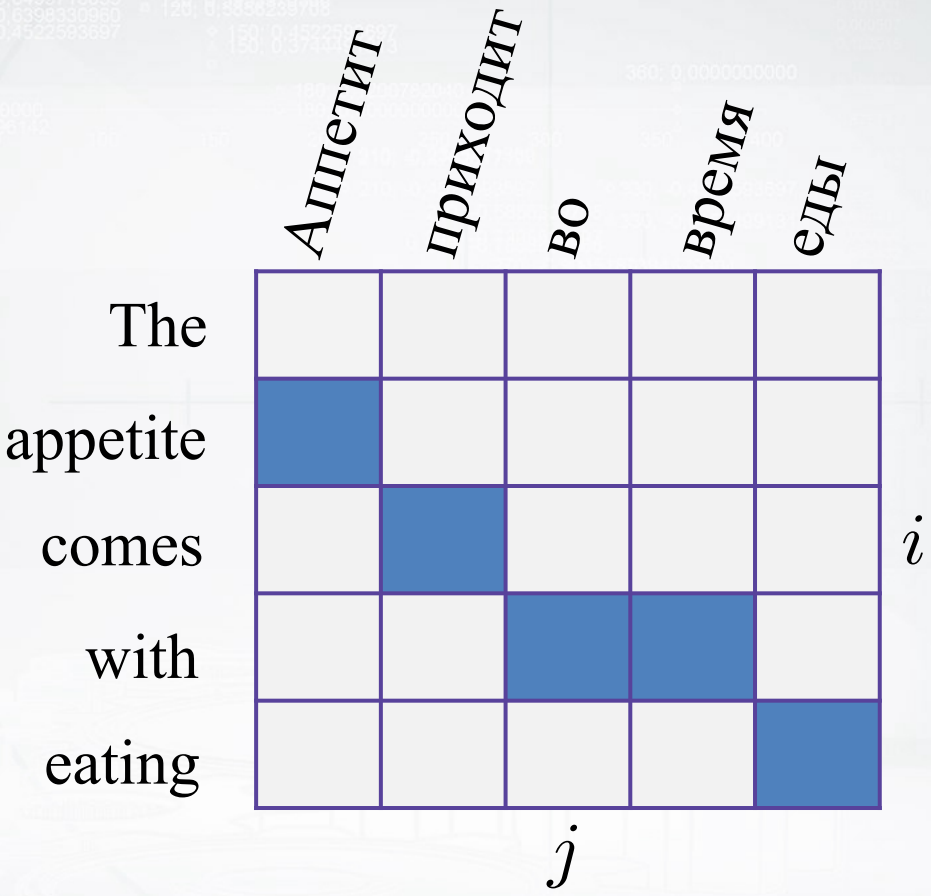
$$e^* = \underset{e \in E}{\operatorname{argmax}} \underbrace{p(e)}_{\text{Language model}} \underbrace{p(f|e)}_{\text{Translation model}}$$

- $p(e)$ models the *fluency* of the translation
- $p(f|e)$ models the *adequacy* of the translation
- argmax is the search problem implemented by a *decoder*

Word alignment matrix

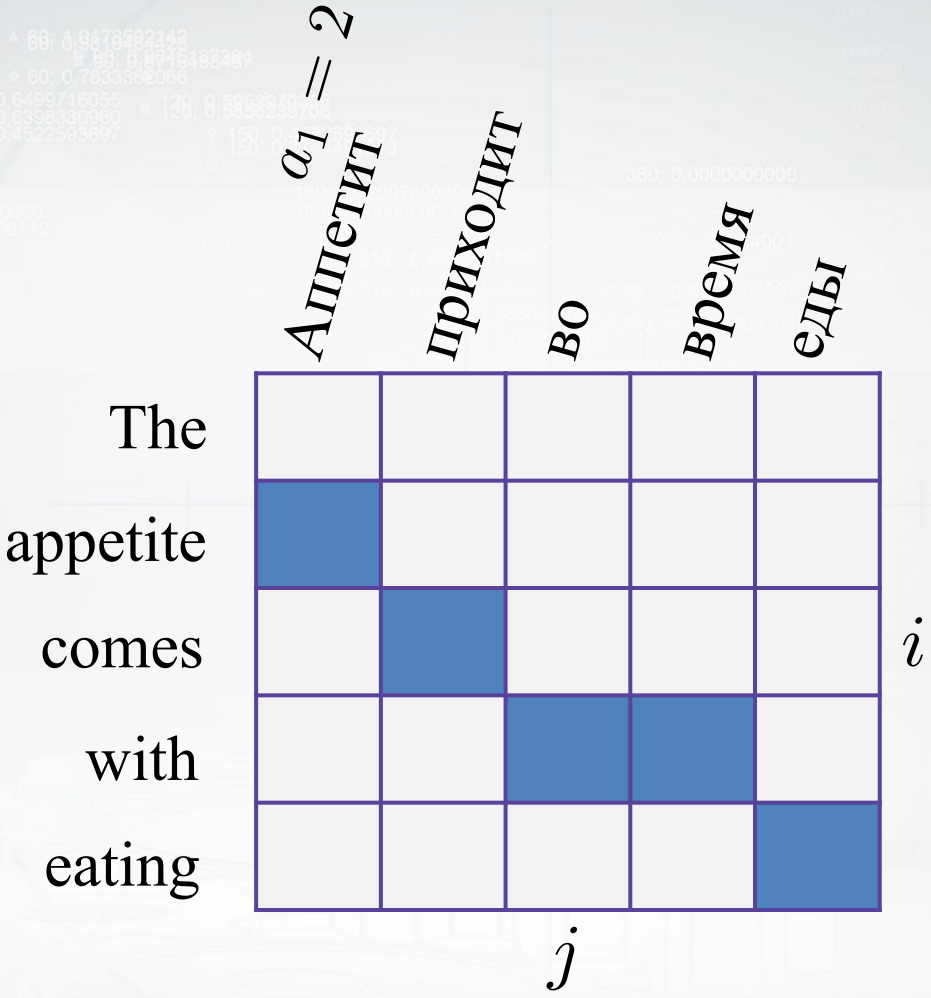


Word alignment matrix



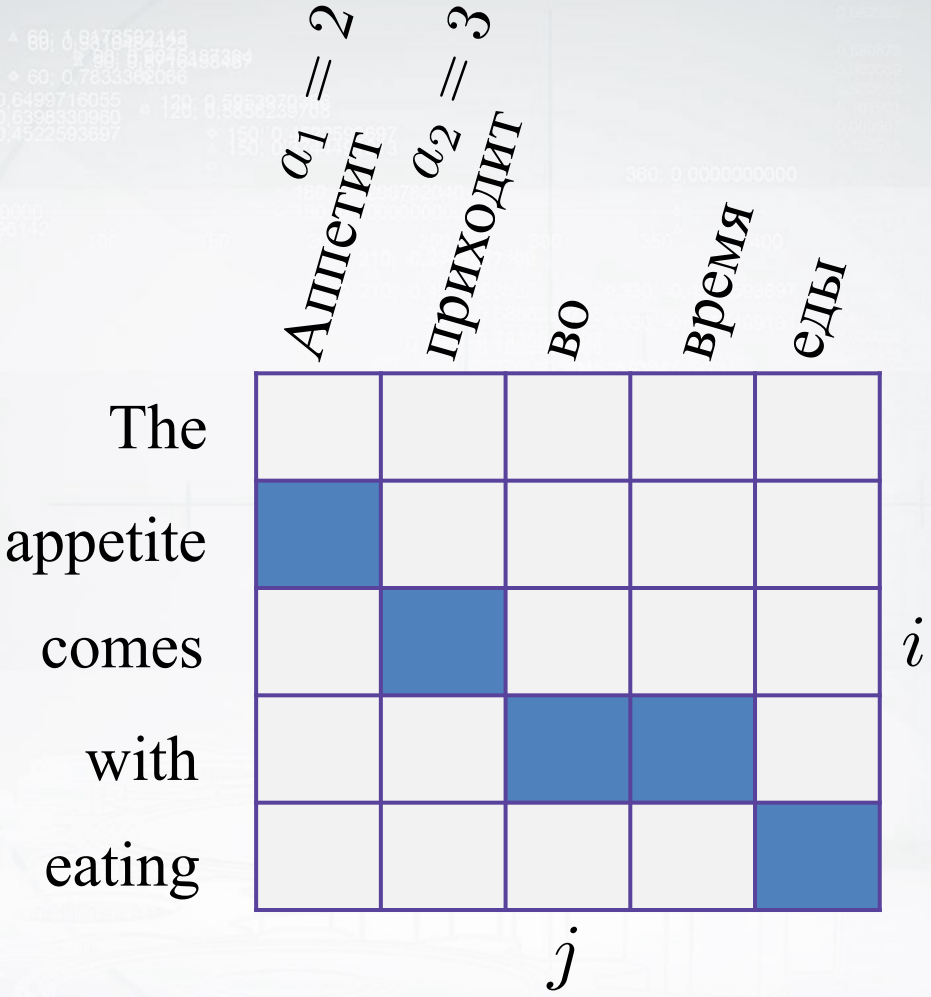
Each target word is allowed to have only one source!

Word alignment matrix



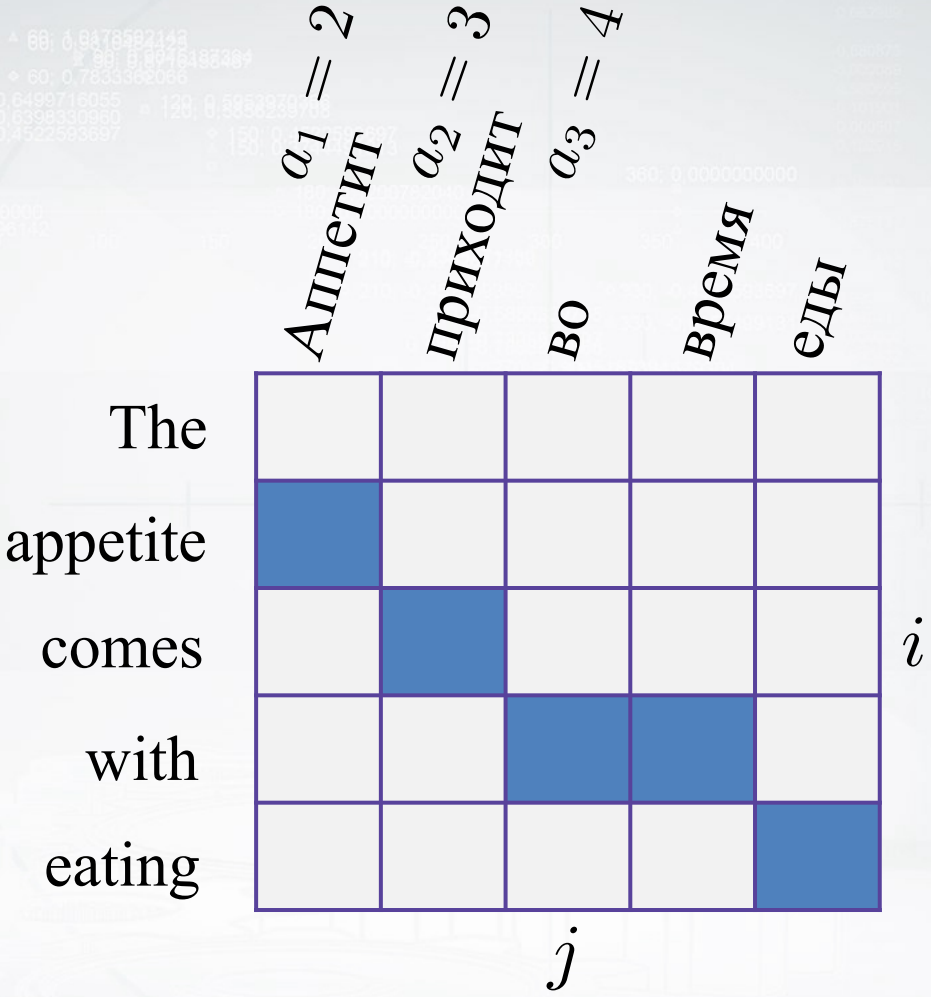
Each target word is allowed to have only one source!

Word alignment matrix



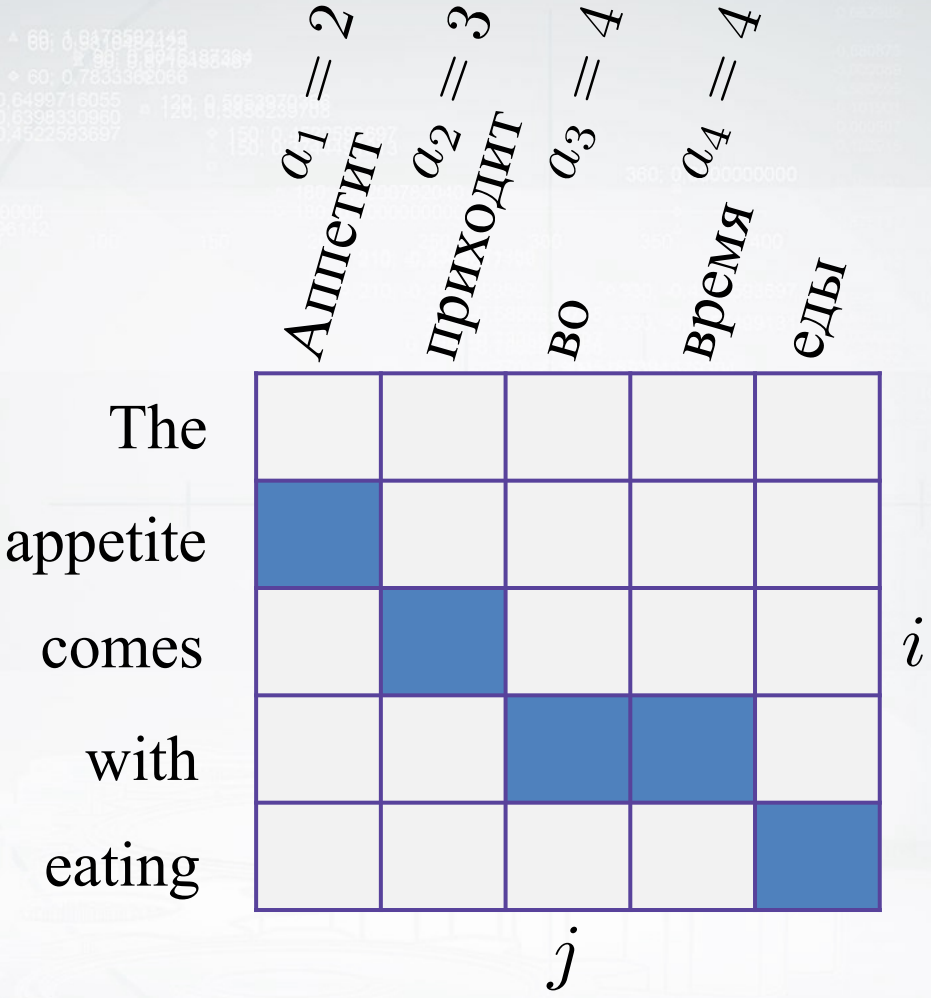
Each target word is allowed to have only one source!

Word alignment matrix



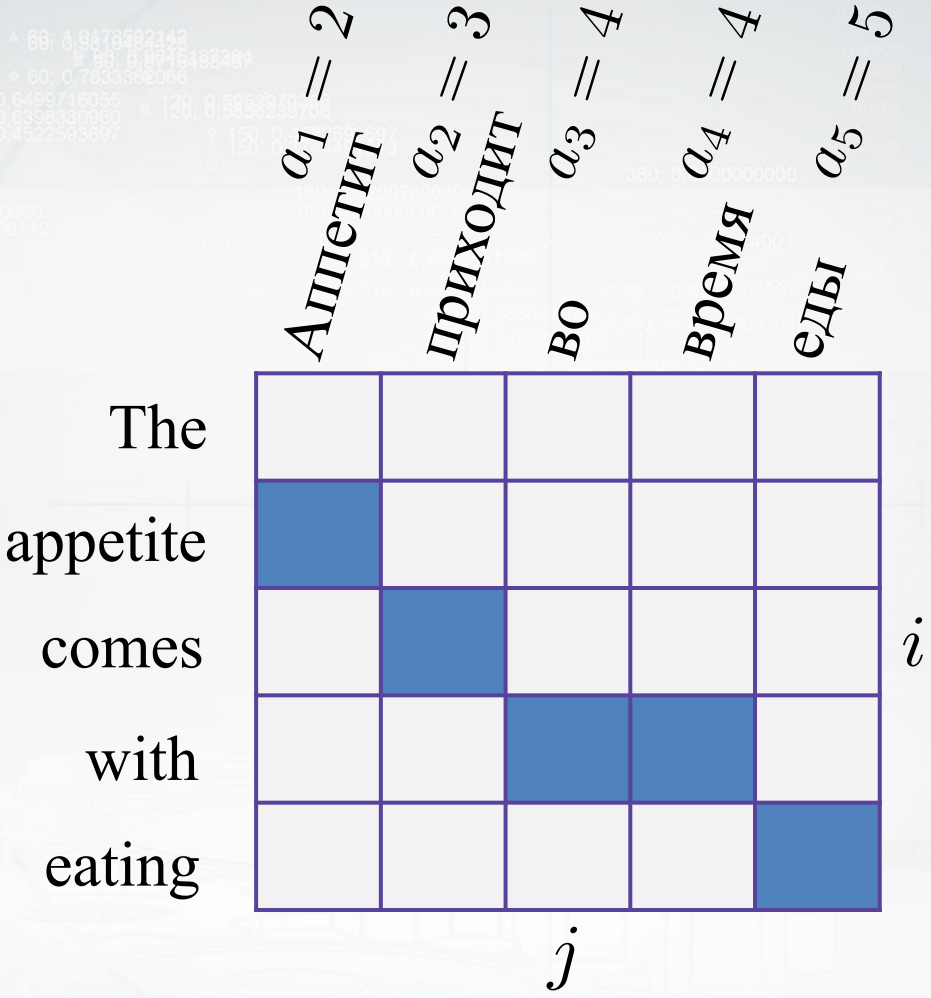
Each target word is allowed to have only one source!

Word alignment matrix



Each target word is allowed to have only one source!

Word alignment matrix



Each target word is allowed to have only one source!

Sketch of learning algorithm

1. **Probabilistic model (generative story)**

   Given **e**, model the generation of **f**:

$$p(f, a|e, \Theta) = ?$$

The most creative step:

- How do we parametrize the model?
- Is it too complicated or too unrealistic?

Sketch of learning algorithm

1. **Probabilistic model (generative story)**

   Given **e**, model the generation of **f**:

$$p(f, a|e, \Theta) = ?$$

observable variables

The most creative step:

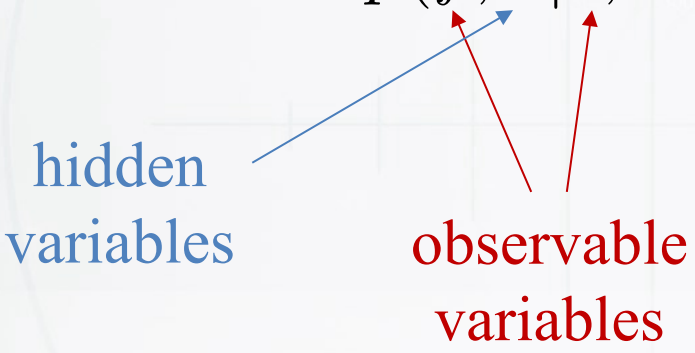
- How do we parametrize the model?
- Is it too complicated or too unrealistic?

Sketch of learning algorithm

1. **Probabilistic model (generative story)**

   Given **e**, model the generation of **f**:

$$p(f, a|e, \Theta) = ?$$



The most creative step:

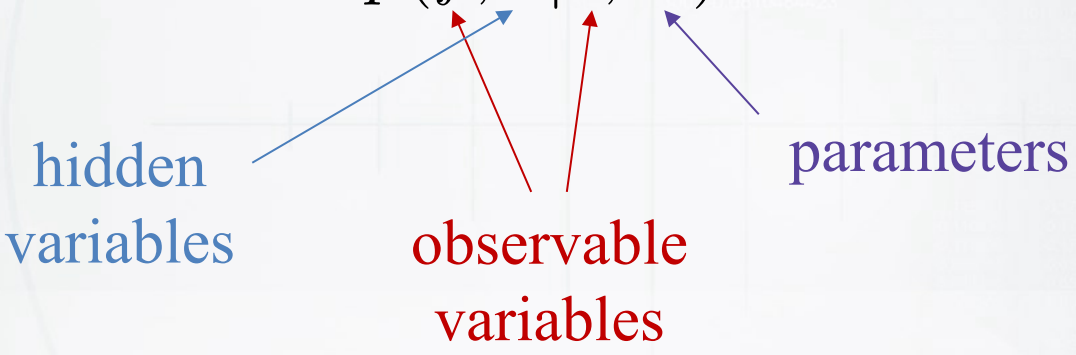
- How do we parametrize the model?
- Is it too complicated or too unrealistic?

Sketch of learning algorithm

1. **Probabilistic model (generative story)**

   Given **e**, model the generation of **f**:

$$p(f, a|e, \Theta) = ?$$



The most creative step:

- How do we parametrize the model?
- Is it too complicated or too unrealistic?

Sketch of learning algorithm

2. Likelihood maximization for the incomplete data:

$$p(f|e,\Theta) = \sum_{a} p(f,a|e,\Theta) \rightarrow \max_{\Theta}$$

Sketch of learning algorithm

2. Likelihood maximization for the incomplete data:

$$p(f|e,\Theta) = \sum_{a} p(f,a|e,\Theta) \rightarrow \max_{\Theta}$$

3. EM-algorithm to the rescue!

Iterative process:

- E-step: estimates posterior probabilities for alignments
- M-step: updates Θ – parameters of the model

Generative story

$$p(f, a|e) = p(J|e)$$

1. Choose the length of the foreign sentence

Generative story

$$p(f, a|e) = p(J|e) \prod_{j=1}^{J} p(a_j|a_1^{j-1}, f_1^{j-1}, J, e) \times$$

1. Choose the length of the foreign sentence
2. Choose an alignment for each word (given lots of things)

Generative story

$$p(f, a|e) = p(J|e) \prod_{j=1}^{J} p(a_j|a_1^{j-1}, f_1^{j-1}, J, e) \times$$
$$\times p(f_j|a_j, a_1^{j-1}, f_1^{j-1}, J, e)$$

1. Choose the length of the foreign sentence
2. Choose an alignment for each word (given lots of things)
3. Choose the word (given lots of things)

IBM model 1

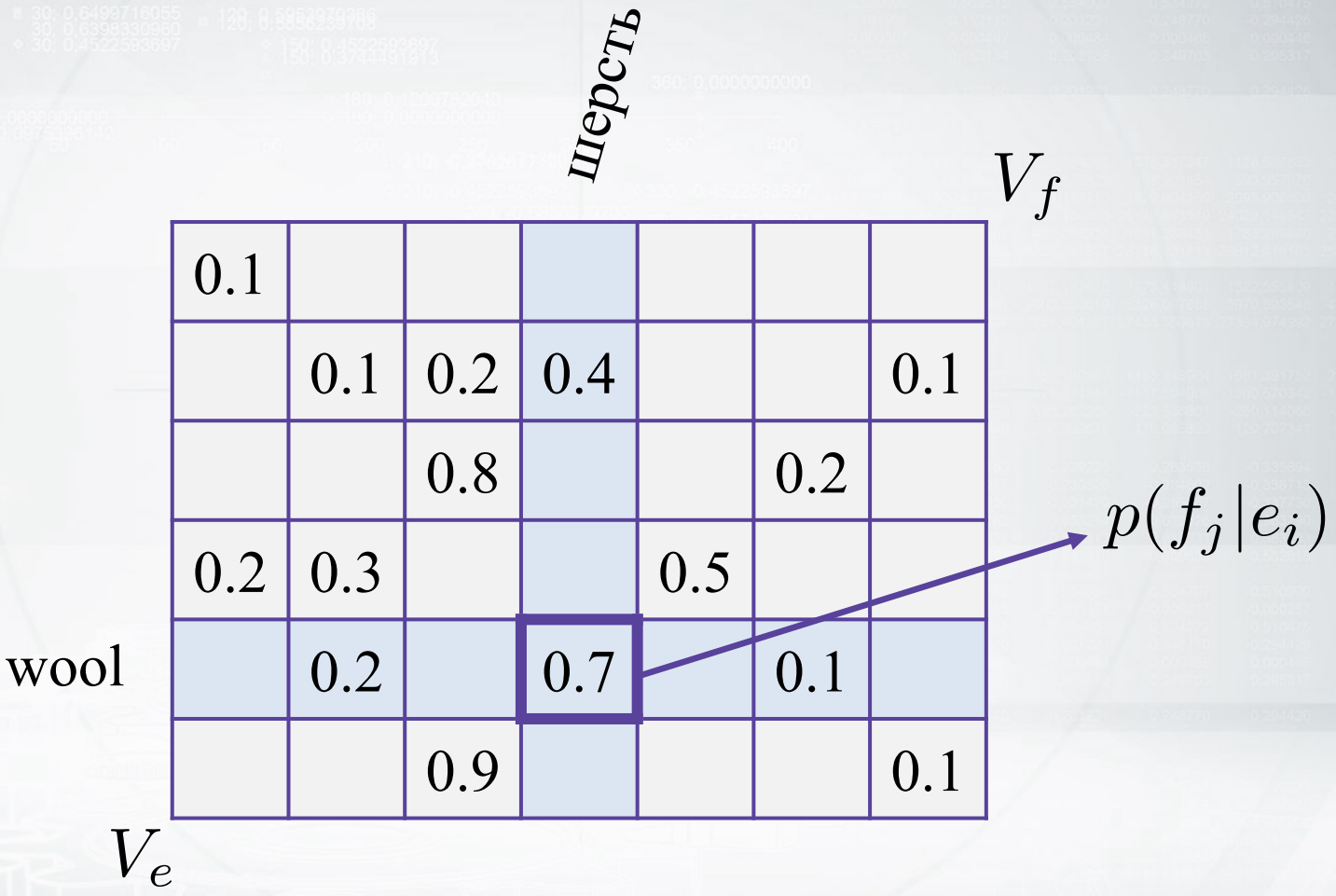
$$p(f, a|e) = p(J|e) \prod_{j=1}^{J} p(a_j) p(f_j|a_j, e)$$

Uniform prior
ε

Translation table
$t(f_j|e_{a_j})$

+ The model is simple and has not too many parameters
– The alignment prior does not depend on word positions

Translation table



IBM model 2

$$p(f, a|e) = p(J|e) \prod_{j=1}^{J} p(a_j|j, I, J) p(f_j|a_j, e)$$

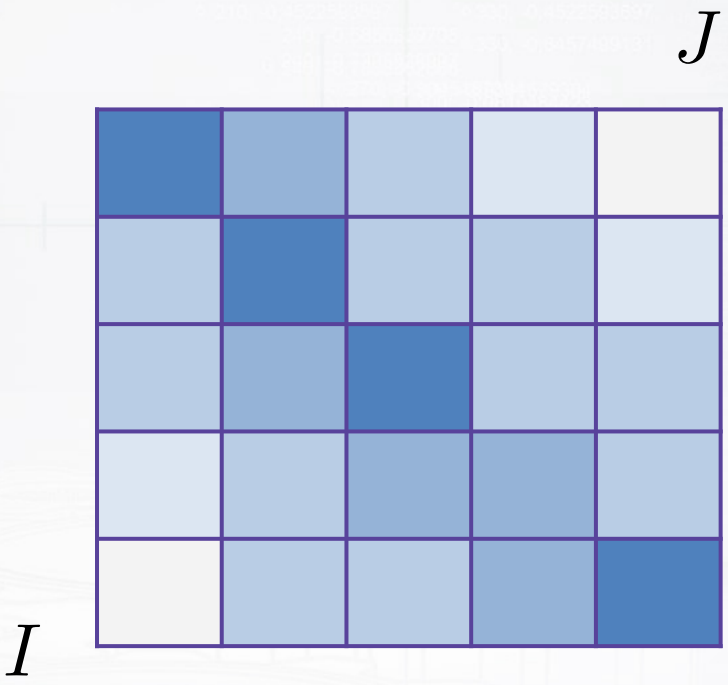
Position-based prior $d(a_j|j, I, J)$

Translation table $t(f_j|e_{a_j})$

+ The alignments depend on position-based prior
– Quite a lot of parameters for the alignments

Position-based prior

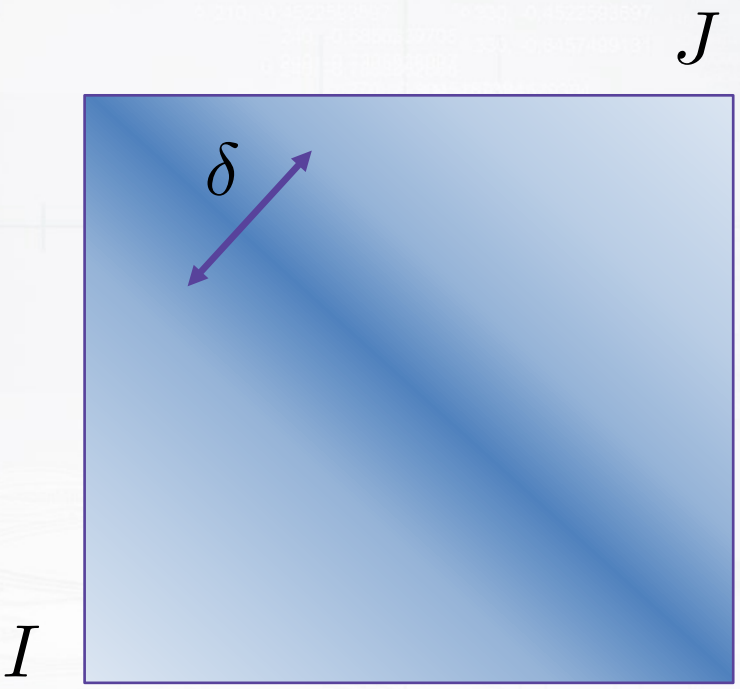
- For each pair of the **lengths** of the sentences:
  - $I \times J$ matrix of probabilities



Dyer et al. A Simple, Fast, and Effective Reparameterization of IBM Model 2, 2013

Re-parametrization, Dyer et. al 2013

- If we know, it's going to be diagonal – let's model it diagonal!
- Much less parameters, easier to train on small data



Dyer et al. A Simple, Fast, and Effective Reparameterization of IBM Model 2, 2013

HMM for the prior

$$p(f, a|e) = \prod_{j=1}^{J} p(a_j|a_{j-1}, I, J)p(f_j|a_j, e)$$

Transition probabilities
$d(a_j|a_{j-1}, I, J)$

Translation table
$t(f_j|e_{a_j})$

e: All cats are grey in the dark.

f: В темноте все кошки серы.

Resume

- IBM models – first working systems of MT
- Lot's of problems with models 1 and 2:
  - How to deal with *spurious words*
  - How to control *fertility*
  - ....
- How to do many-to-many alignments?
  - Phrase-based machine translation