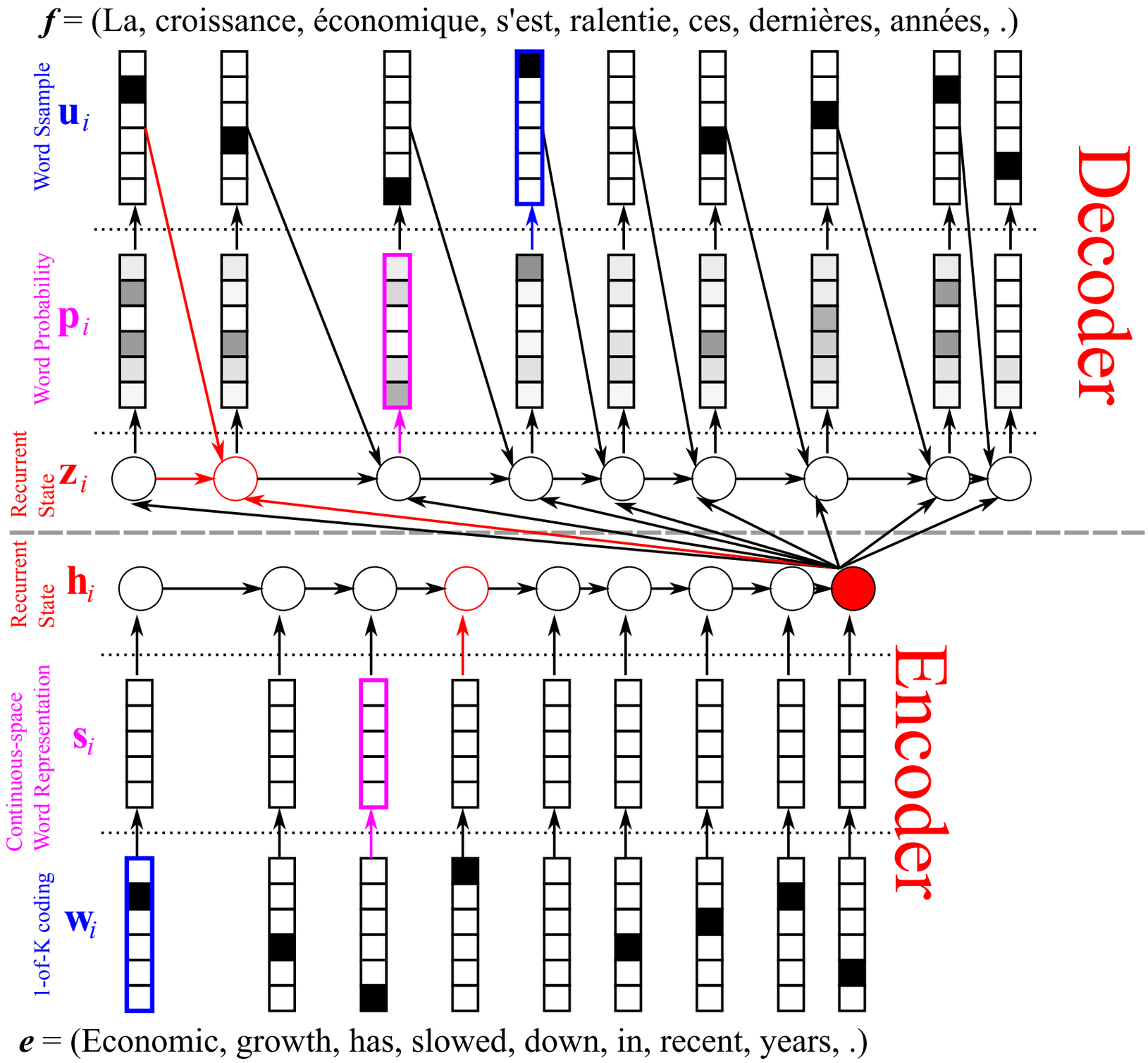
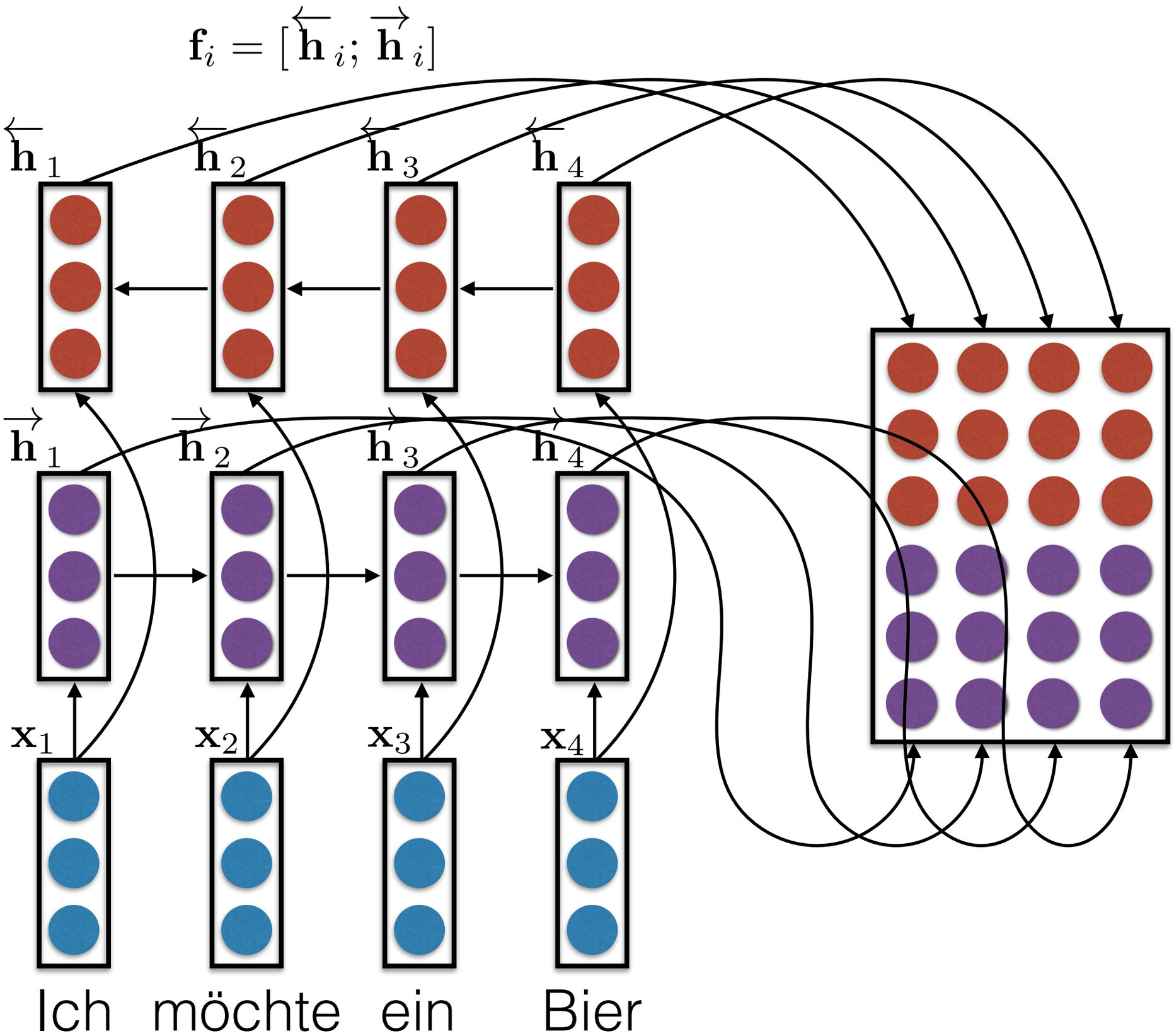
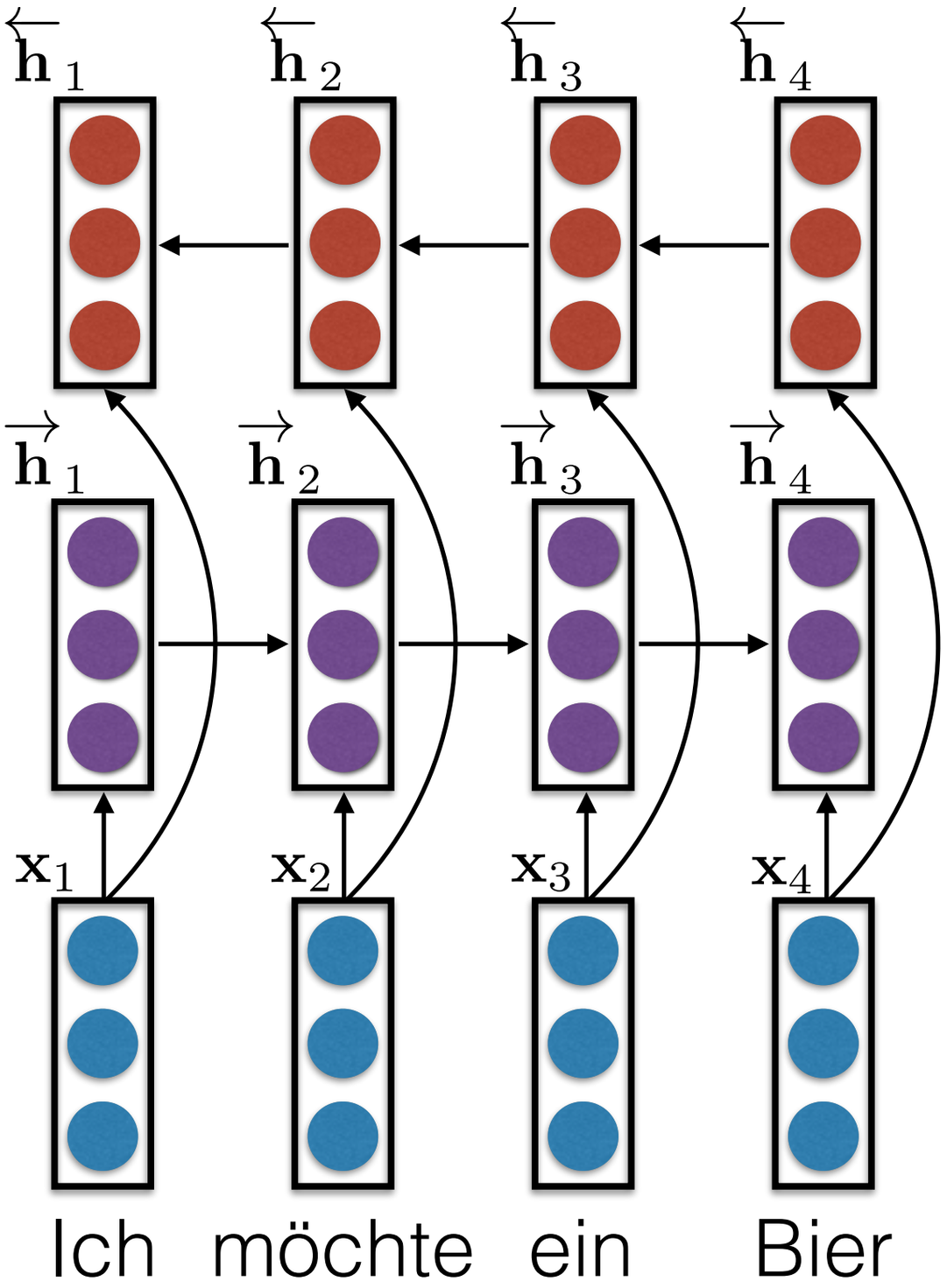

Aire
ATTENTION
ACHTUNG
2T
1.8m
6ft
Cas Siar Anois
TURN BACK NOW
JETZT WENDEN

$$p(\mathbf{f}|\mathbf{e}) = \prod_{i=1}^{|\mathbf{f}|} p(f_i|f_{i-1}, ..., f_1, \mathbf{e})$$

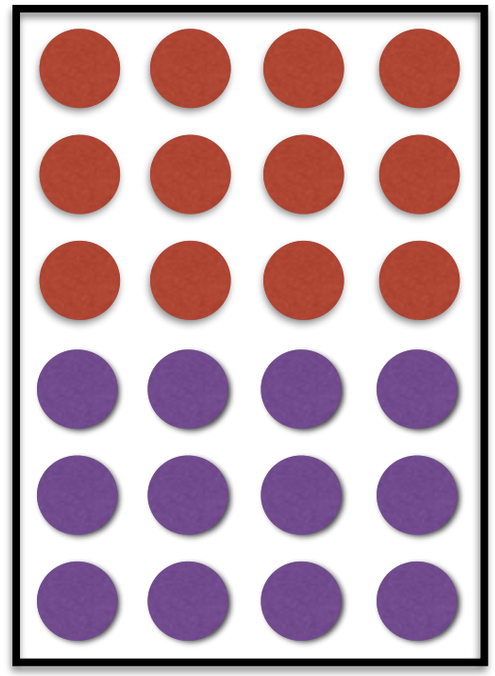



$\mathbf{f}_i = [\overleftarrow{\mathbf{h}}_i; \overrightarrow{\mathbf{h}}_i]$
$\overleftarrow{\mathbf{h}}_1$
$\overleftarrow{\mathbf{h}}_2$
$\overleftarrow{\mathbf{h}}_3$
$\overleftarrow{\mathbf{h}}_4$
$\overrightarrow{\mathbf{h}}_1$
$\overrightarrow{\mathbf{h}}_2$
$\overrightarrow{\mathbf{h}}_3$
$\overrightarrow{\mathbf{h}}_4$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_4$
Ich
möchte
ein
Bier

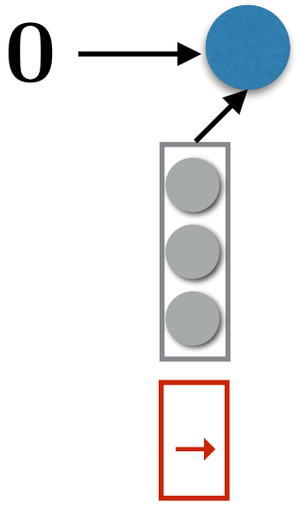
$$\mathbf{f}_i = [\overleftarrow{\mathbf{h}}_i; \overrightarrow{\mathbf{h}}_i]$$

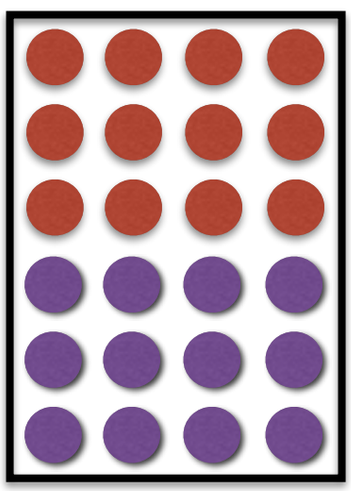


$$\mathbf{F} \in \mathbb{R}^{2n \times |\boldsymbol{f}|}$$

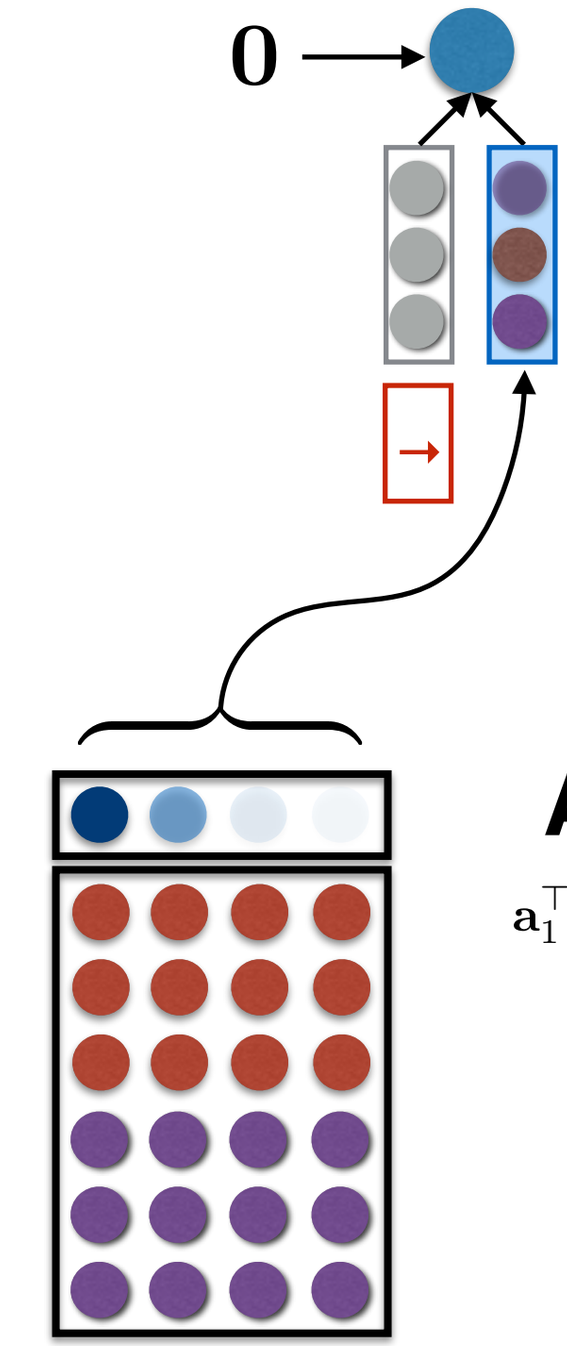


Ich möchte ein Bier

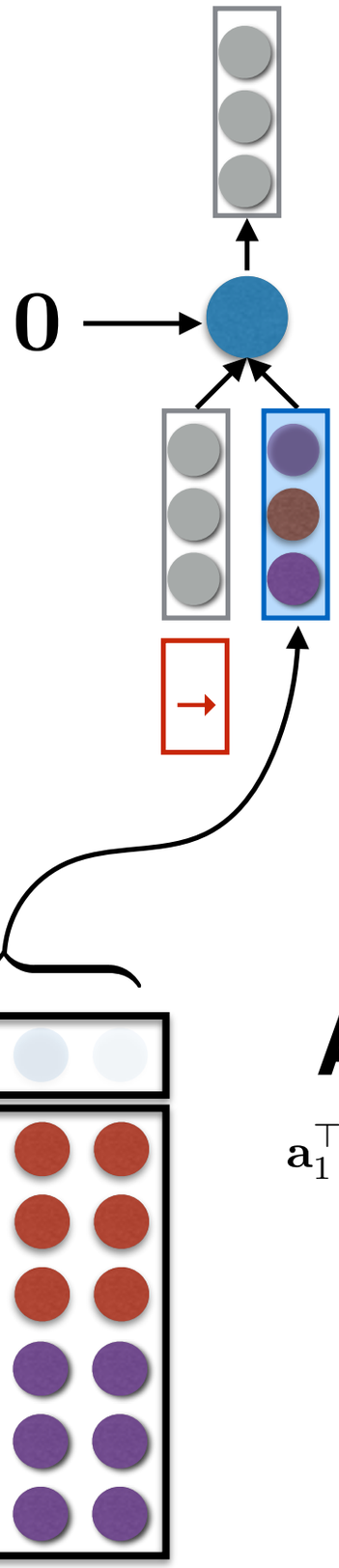




Ich möchte ein Bier

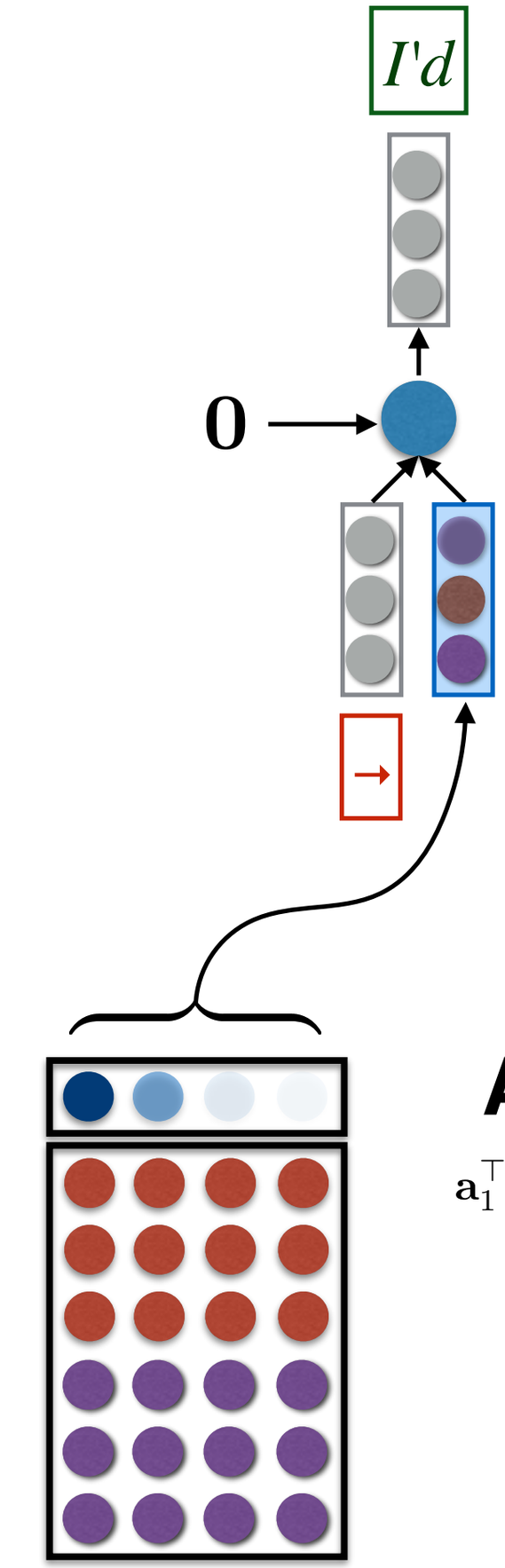


Attention history:



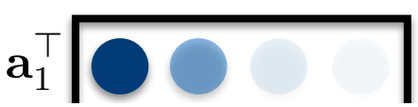
Attention history:

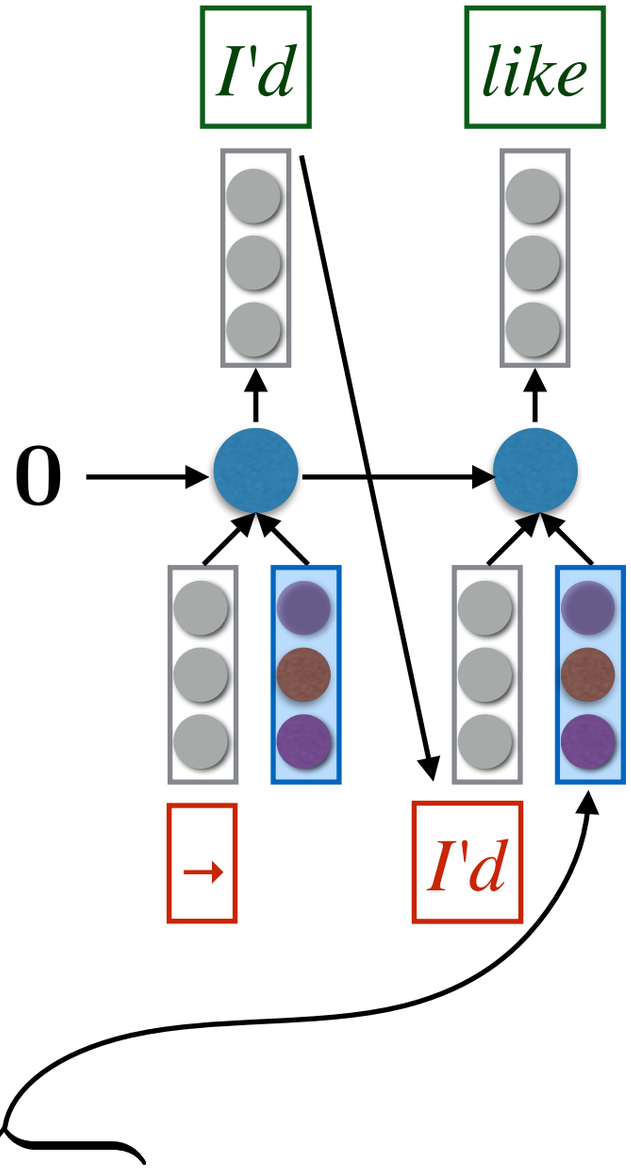
$\mathbf{a}_1^\top$

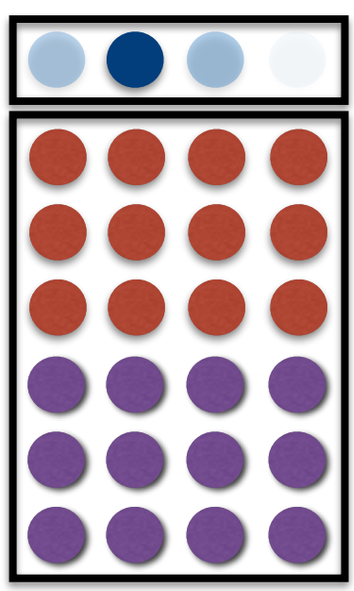


Attention history:

$\mathbf{a}_1^\top$

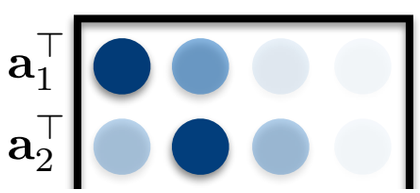


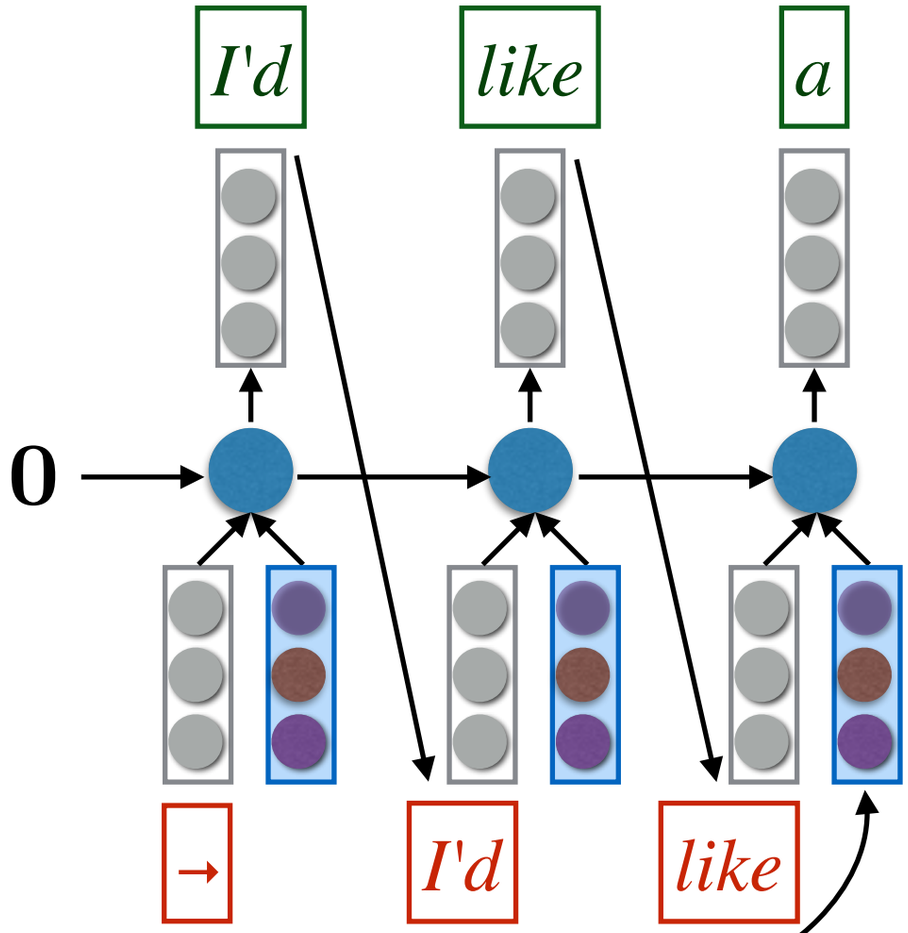


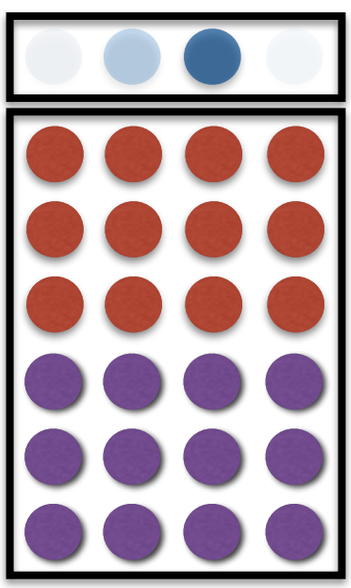


Ich möchte ein Bier

Attention history:

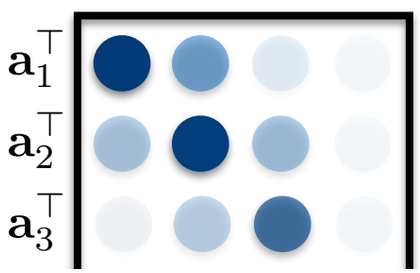


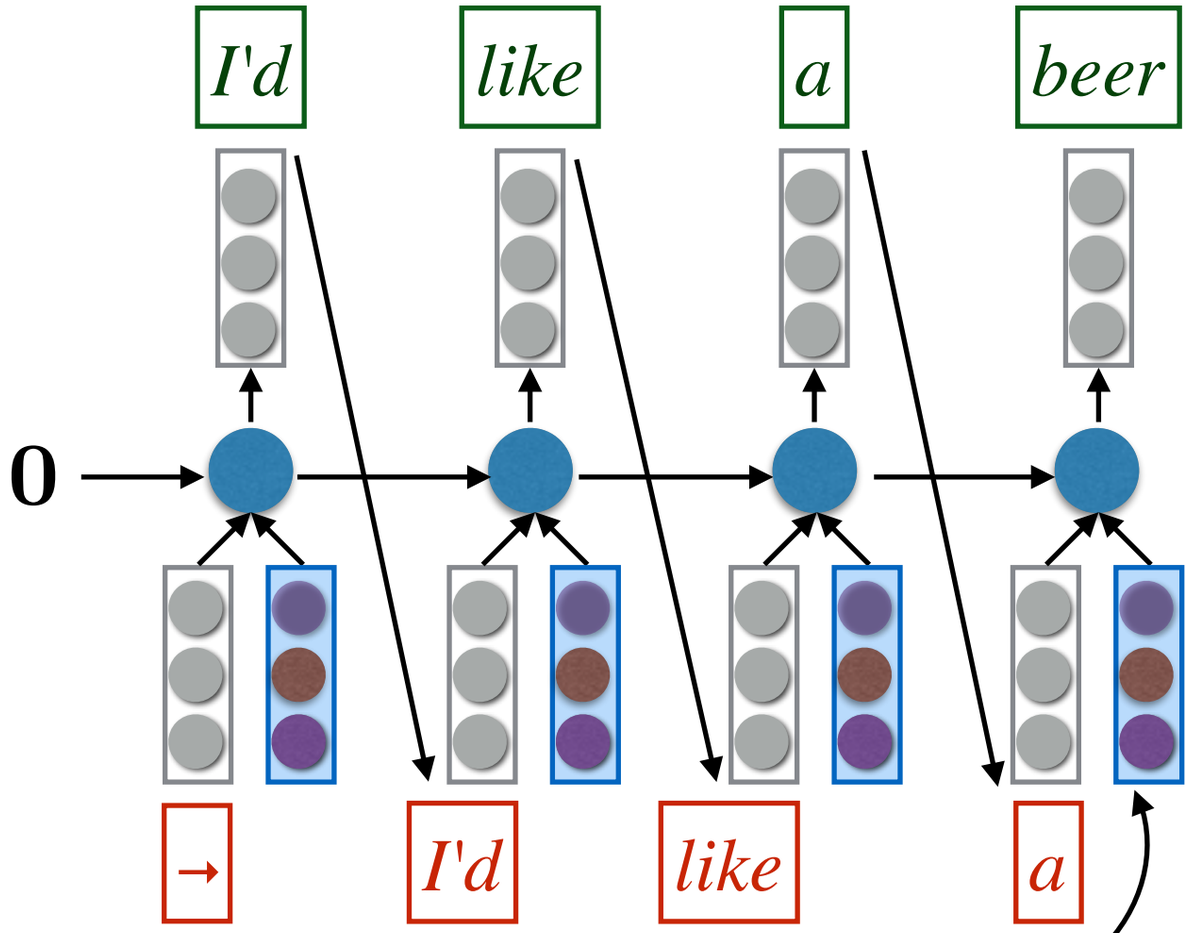


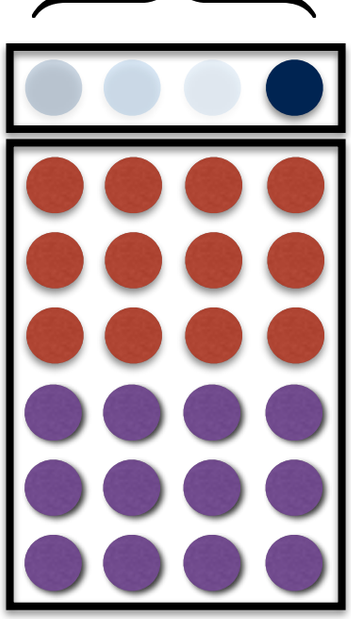


Ich möchte ein Bier

Attention history:

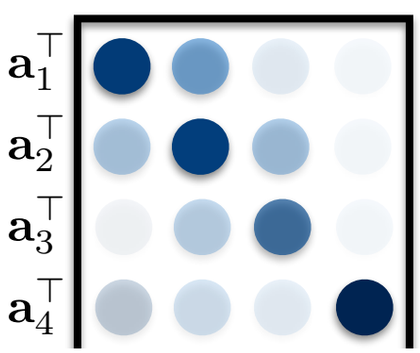


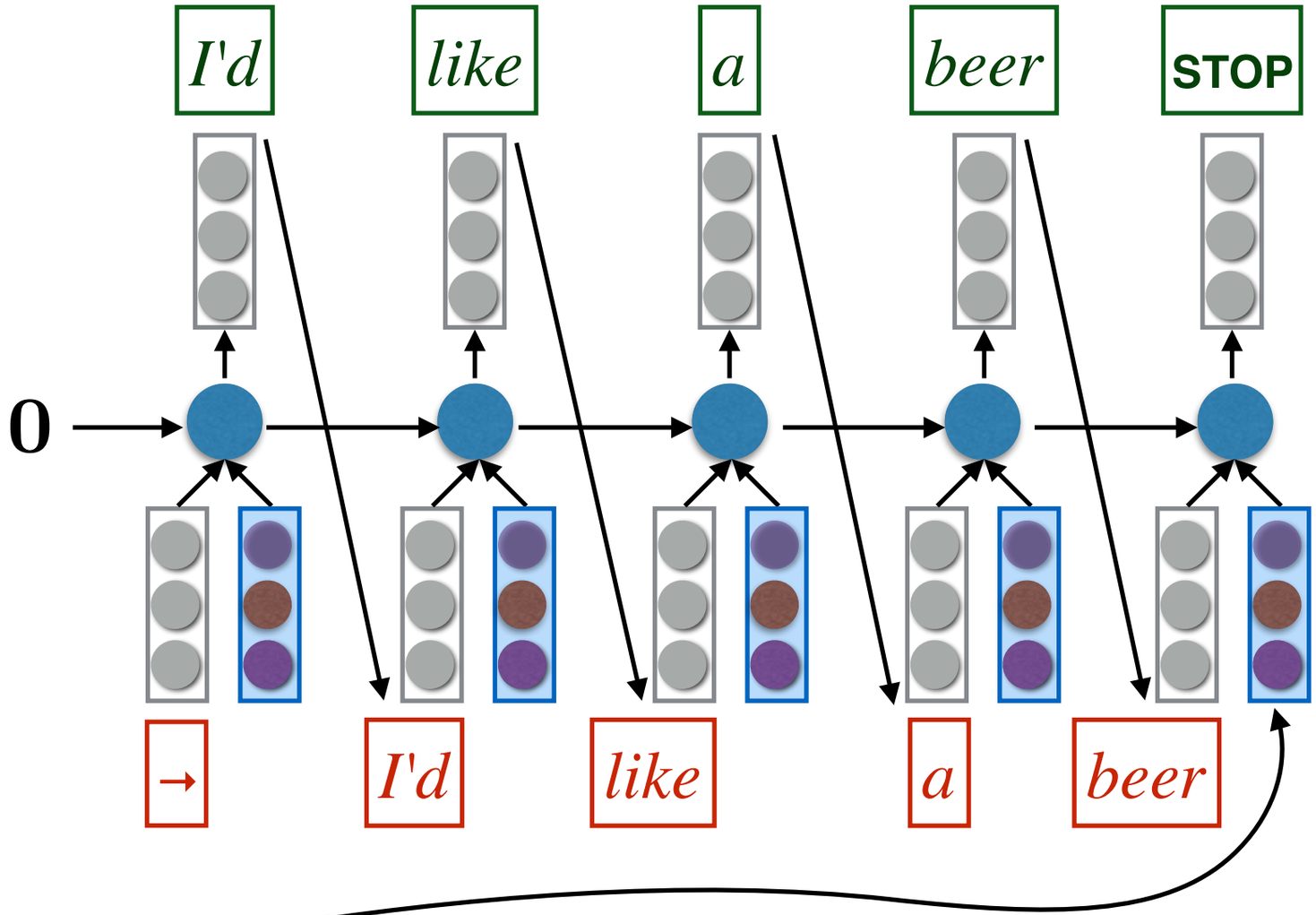


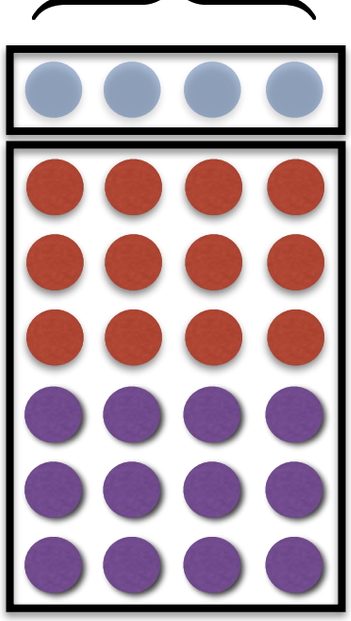


Ich möchte ein Bier

Attention history:

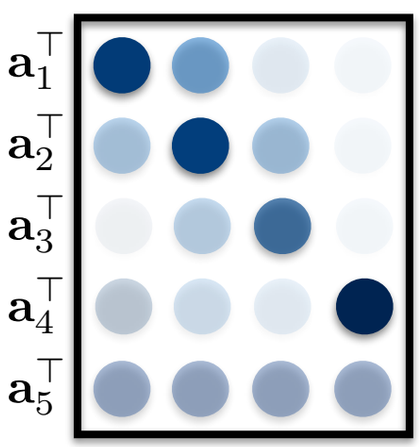


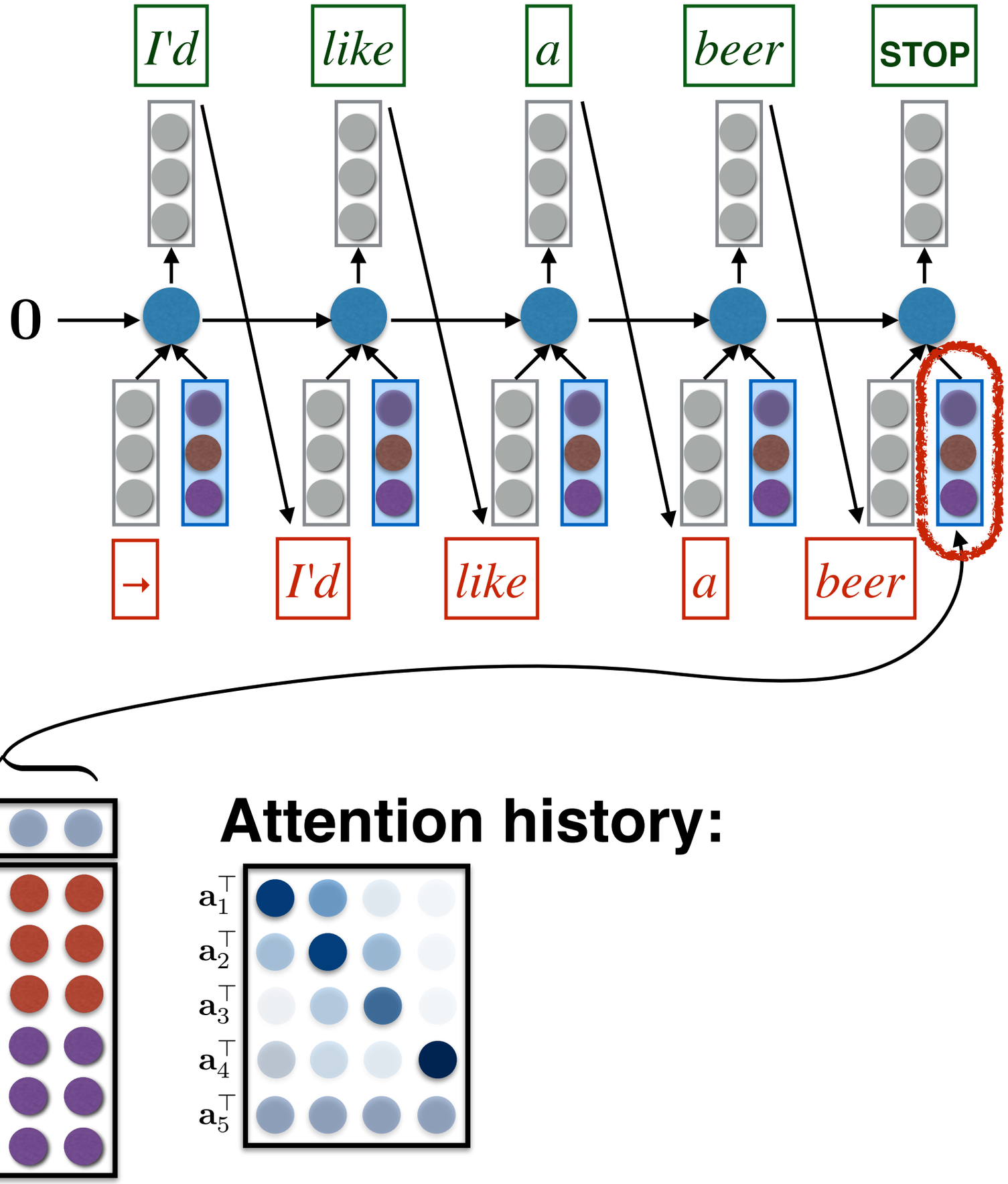




Ich möchte ein Bier

Attention history:

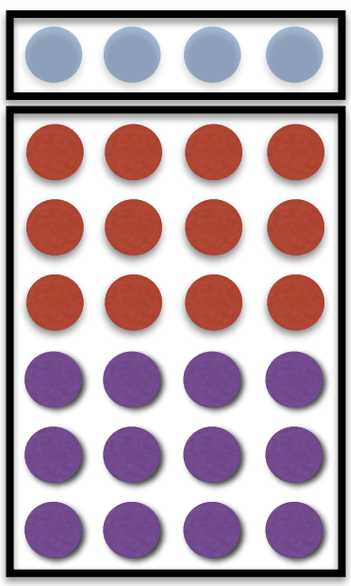




How do we compute this?

What does it represent?

What should it represent (if anything)?



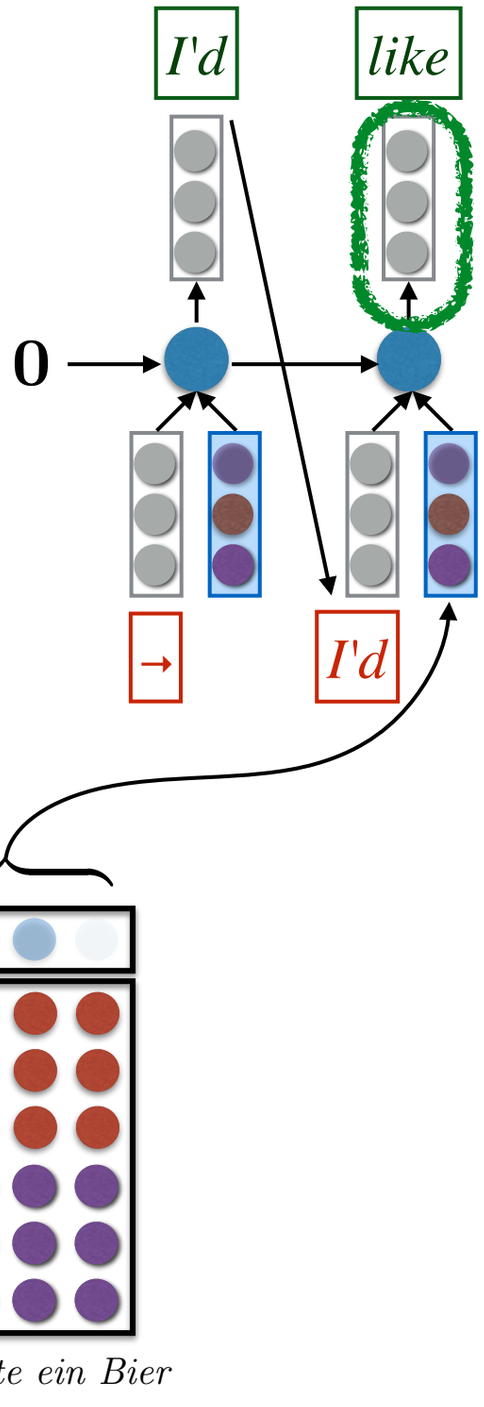
Ich möchte ein Bier

Attention history:

Computing Attention

- At each time step (one time step = one output word), we want to be able to "attend" to different words in the source sentence
  - We need a weight for every word: this is an $|\boldsymbol{f}|$-length vector $\mathbf{a}_t$
  - Here is a simplified version of Bahdanau et al.'s solution
    - Use an RNN to predict model output, call the hidden states $\mathbf{s}_t$
      ($\mathbf{s}_t$ has a fixed dimensionality, call it *m*)
    - At time *t* compute the ***expected input embedding*** $\mathbf{r}_t = \mathbf{V}\mathbf{s}_{t-1}$
      ($\mathbf{V}$ is a learned parameter)
    - Take the dot product with every column in the source matrix to compute the ***attention energy***. $\mathbf{u}_t = \mathbf{F}^\top \mathbf{r}_t$ (called $\mathbf{e}_t$ in the paper)
      (Since **F** has $|\boldsymbol{f}|$ columns, $\mathbf{u}_t$ has $|\boldsymbol{f}|$ rows)
    - Exponentiate and normalize to 1: $\mathbf{a}_t = \mathrm{softmax}(\mathbf{u}_t)$
      (called $\boldsymbol{\alpha}_t$ in the paper)
    - Finally, the ***input source vector*** for time *t* is $\mathbf{c}_t = \mathbf{F}\mathbf{a}_t$

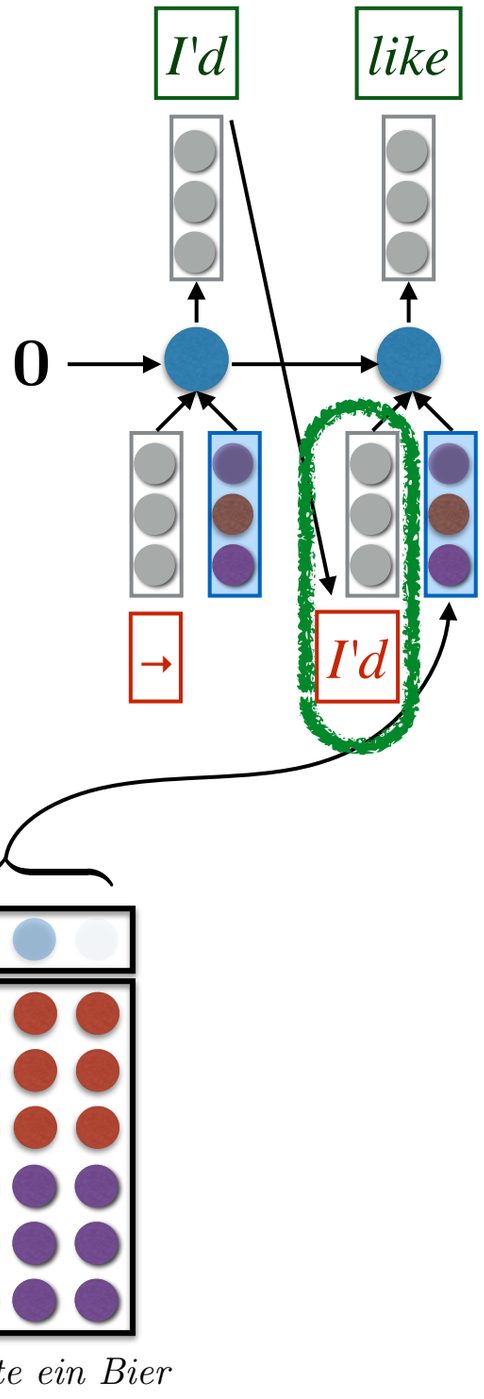
Attention v1



probability distribution

$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

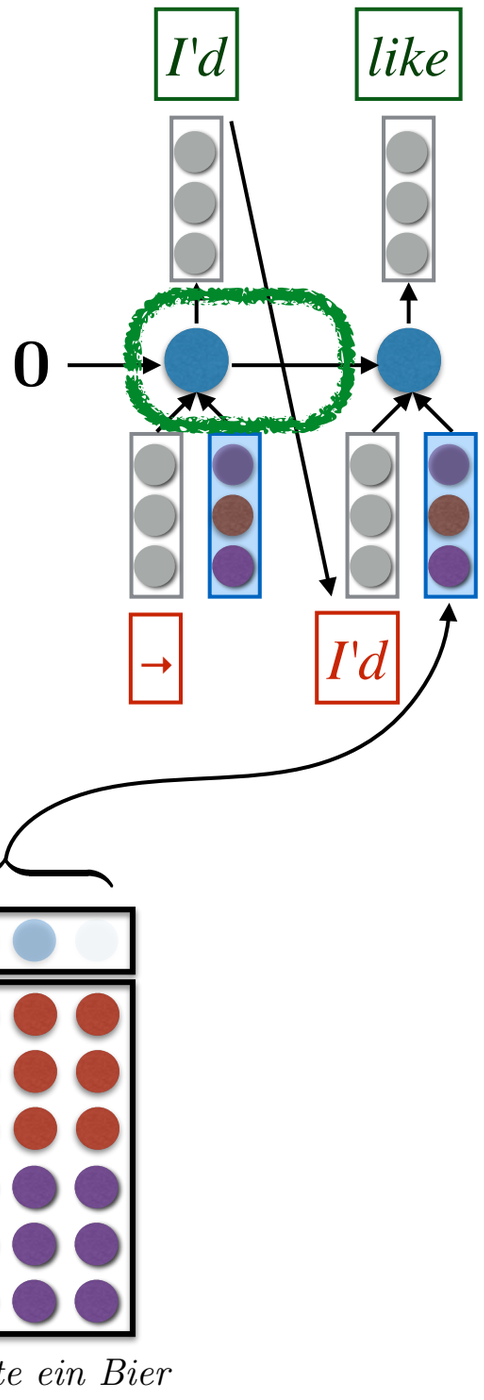
Attention v1



previous word

$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1}, s_i, c_i)$$

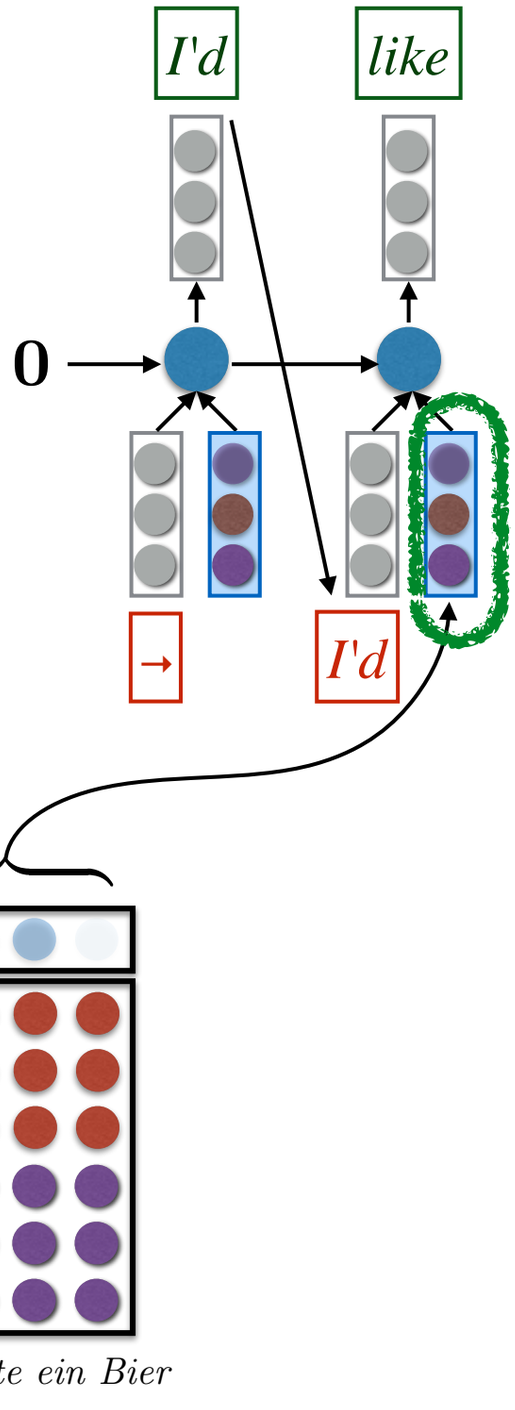
Attention v1



$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1}, s_i, c_i)$$

previous state (a function of entire history)

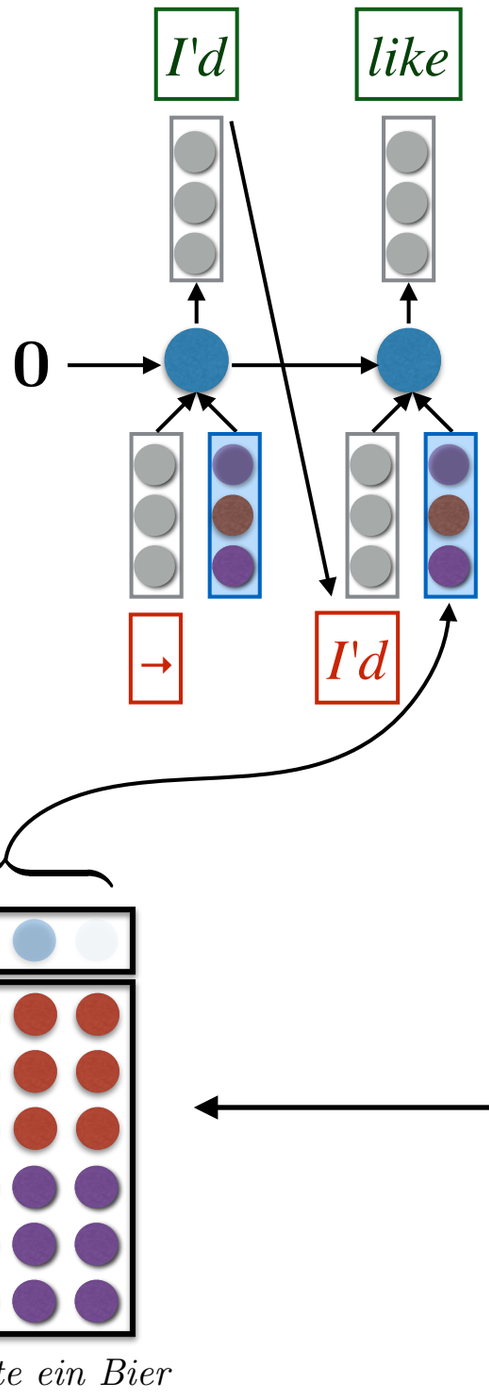
Attention v1



$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1}, s_i, c_i)$$

context

Attention v1

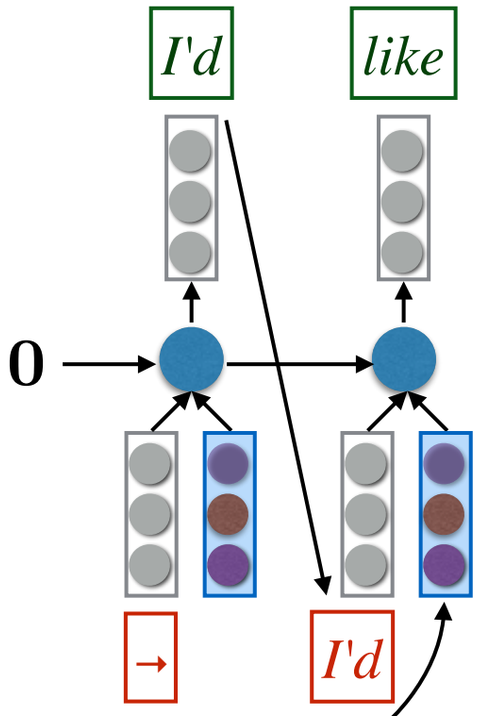


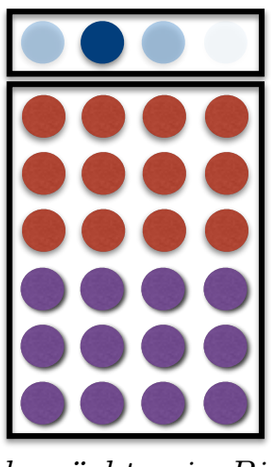
$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

$$c_i = \sum_{j=1}^{|\mathbf{e}|} \alpha_{ij} h_j$$

input word representation

Attention v1





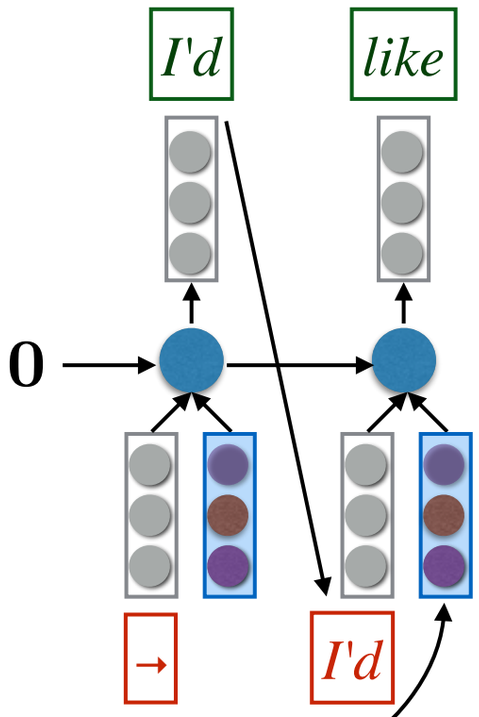
Ich möchte ein Bier

$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

$$c_i = \sum_{j=1}^{|\mathbf{e}|} \alpha_{ij} h_j$$

attention:
distribution over input word positions

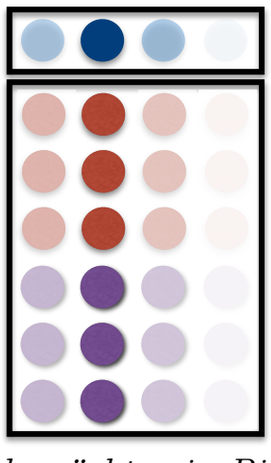
Attention v1



$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

$$c_i = \sum_{j=1}^{|\mathbf{e}|} \alpha_{ij} h_j$$

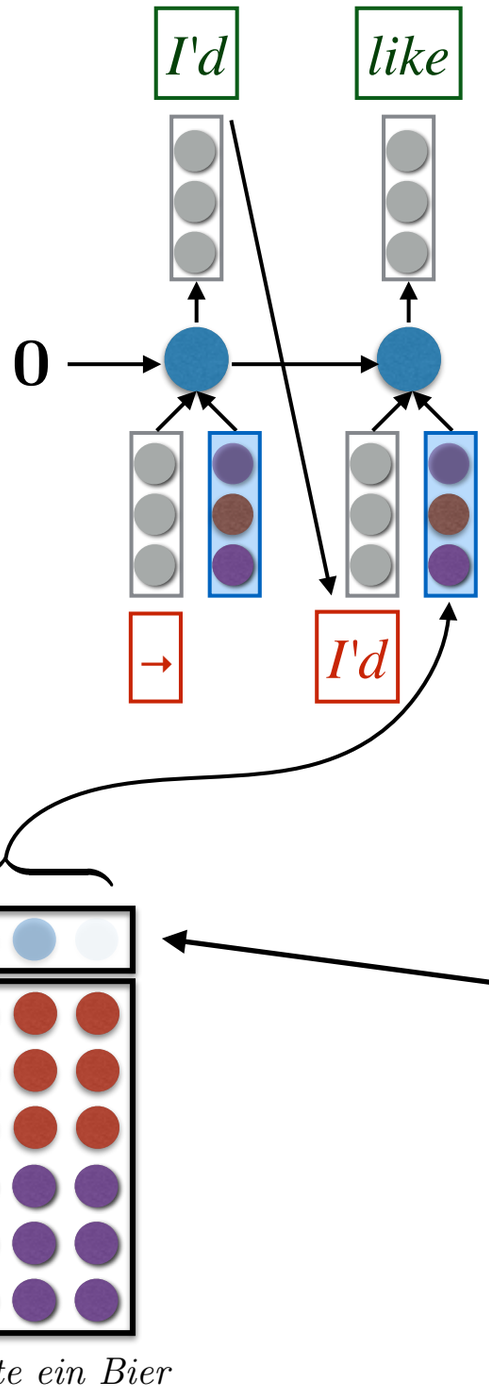
attention:
distribution over input word positions



c is an interpolation of the input vectors

Ich möchte ein Bier

Attention v1



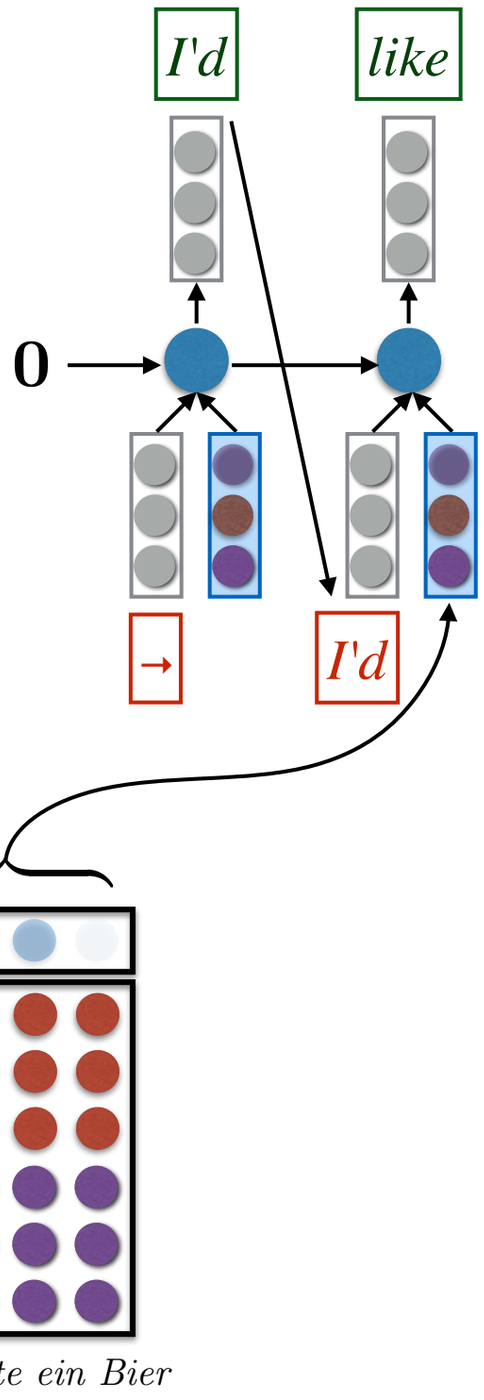
$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

$$c_i = \sum_{j=1}^{|\mathbf{e}|} \alpha_{ij} h_j$$

distribution over input word positions

Alpha is not a latent variable; g is deterministic given history

Attention v1

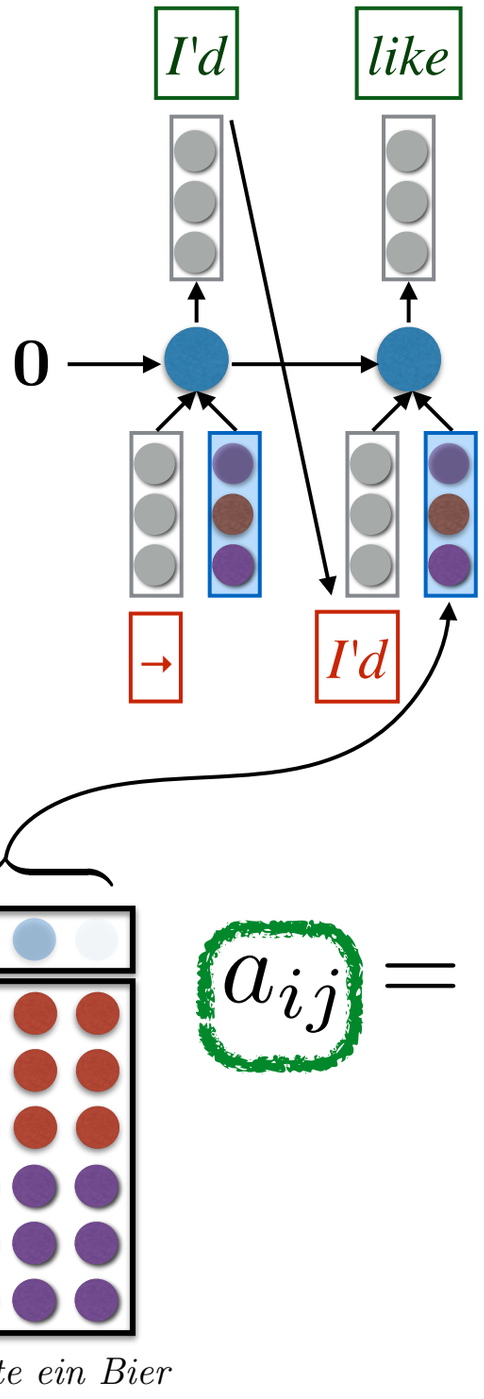


$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

$$c_i = \sum_{j=1}^{|\mathbf{e}|} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(a_{ij})}{\sum_{k=1}^{|\mathbf{e}|} \exp(a_{ik})}$$

Attention v1



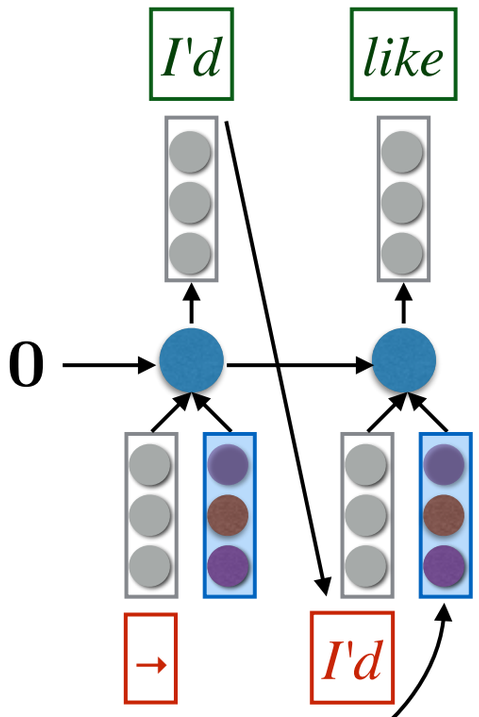
$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

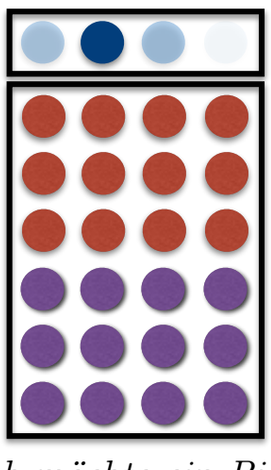
$$c_i = \sum_{j=1}^{|\mathbf{e}|} \alpha_{ij} h_j$$

$$a_{ij} = a(s_{i-1},h_j)$$

$$\alpha_{ij} = \frac{\exp(a_{ij})}{\sum_{k=1}^{|\mathbf{e}|}\exp(a_{ik})}$$

Attention v1





Ich möchte ein Bier

$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

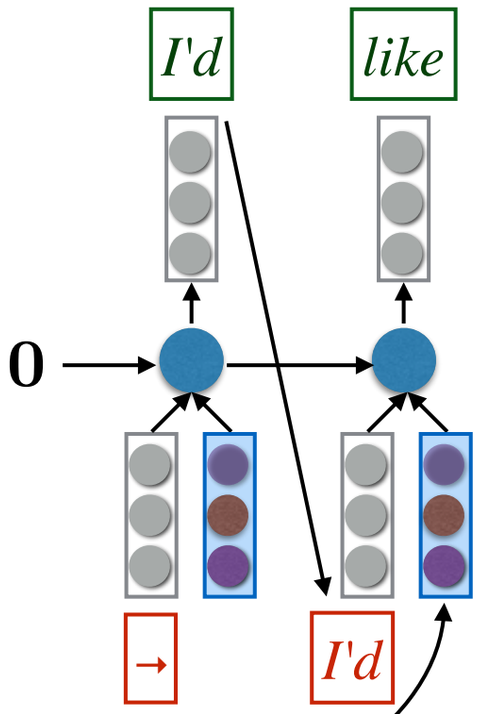
$$c_i = \sum_{j=1}^{|\mathbf{e}|} \alpha_{ij} h_j$$

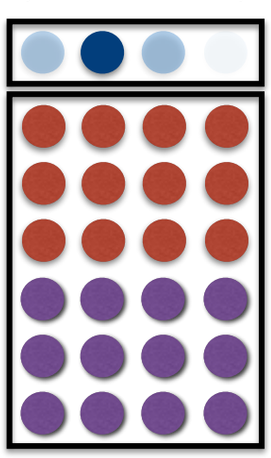
$$a_{ij} = a(s_{i-1},h_j)$$

Can be *any* function

$$\alpha_{ij} = \frac{\exp(a_{ij})}{\sum_{k=1}^{|\mathbf{e}|}\exp(a_{ik})}$$

Attention v2





Ich möchte ein Bier

$$a_{ij} = a(s_{i-1}, h_j)$$

Can be *any* function

$$p(f_i|f_{i-1}, ..., f_1, \mathbf{e}) = g(f_{i-1}, s_i, c_i)$$

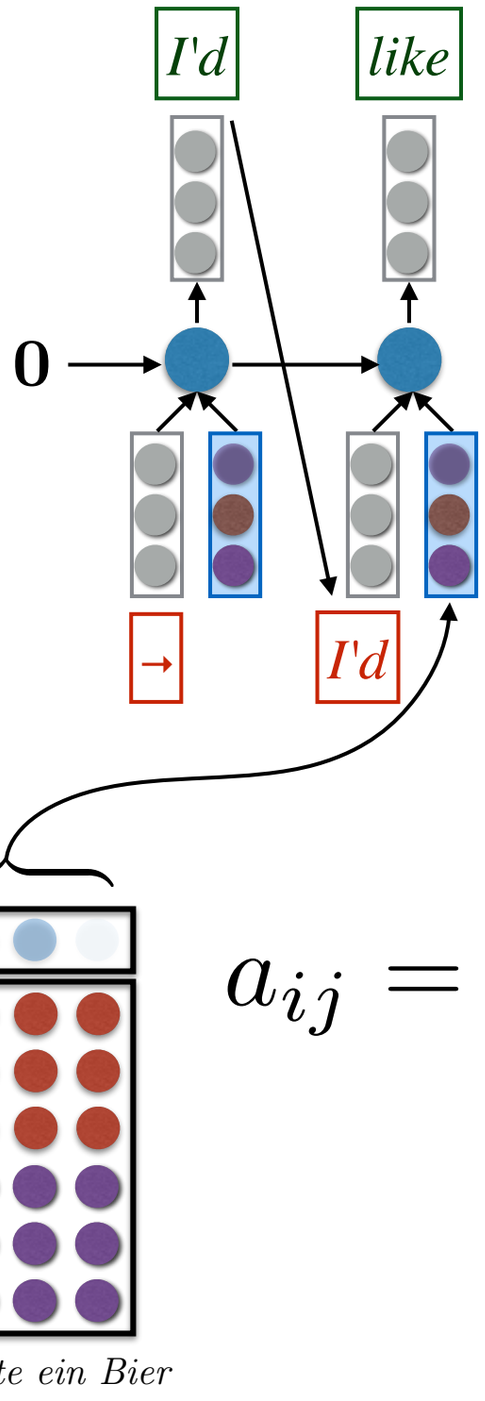
In Bahdanau et al.,
this is a feedforward network:
a is a function of *s* and *h*

What is good or bad about this?

What alternative functions can you design?

Why?

Attention v2



$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

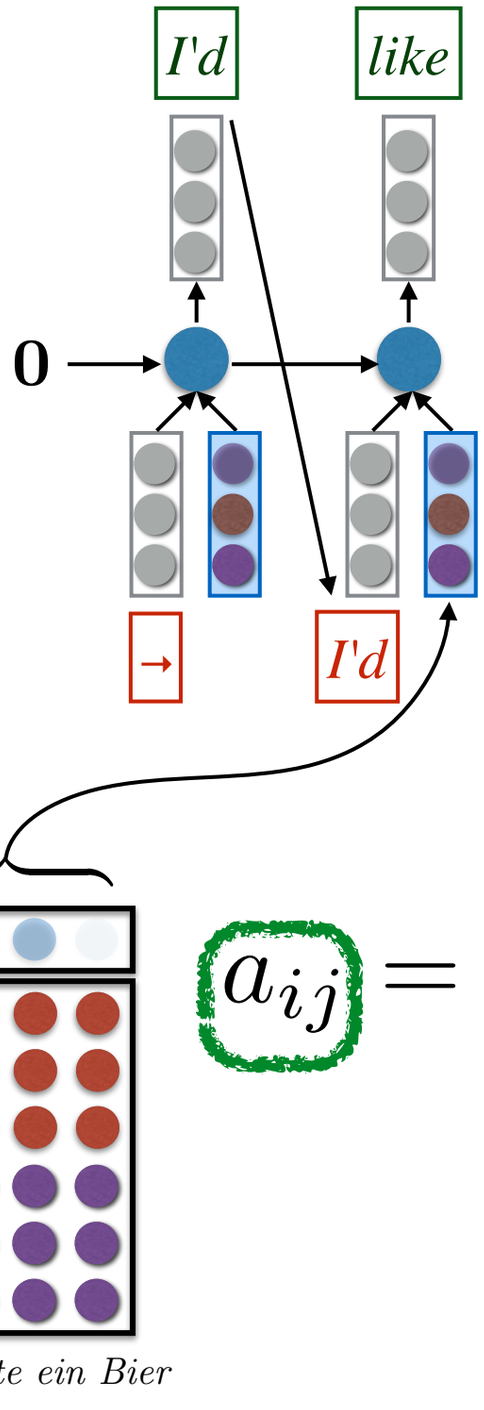
$$a_{ij} = a(s_{i-1},h_j)$$

Luong et al. variants

$$a_{ij} = s_{i-1} \cdot h_j$$

essentially, vector similarity

Attention v2



$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

$$a_{ij} = a(s_{i-1},h_j)$$

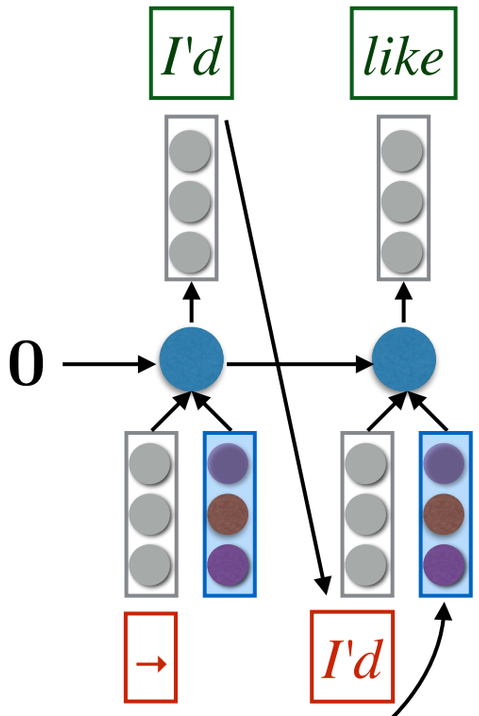
Luong et al. variants

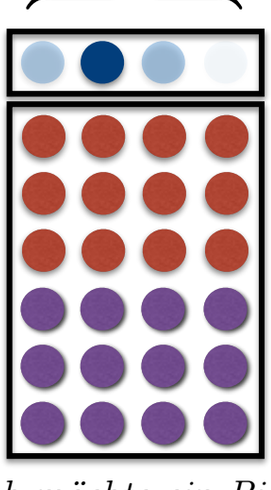
$$a_{ij} = s_{i-1} \cdot h_j$$

$$a_{ij} = s_{i-1}^\top W_a h_j$$

essentially, vector similarity after transform

Attention v2





Ich möchte ein Bier

$$\boxed{a_{ij}} = a(s_{i-1}, h_j)$$

$$p(f_i|f_{i-1}, ..., f_1, \mathbf{e}) = g(f_{i-1}, s_i, c_i)$$

Luong et al.
variants

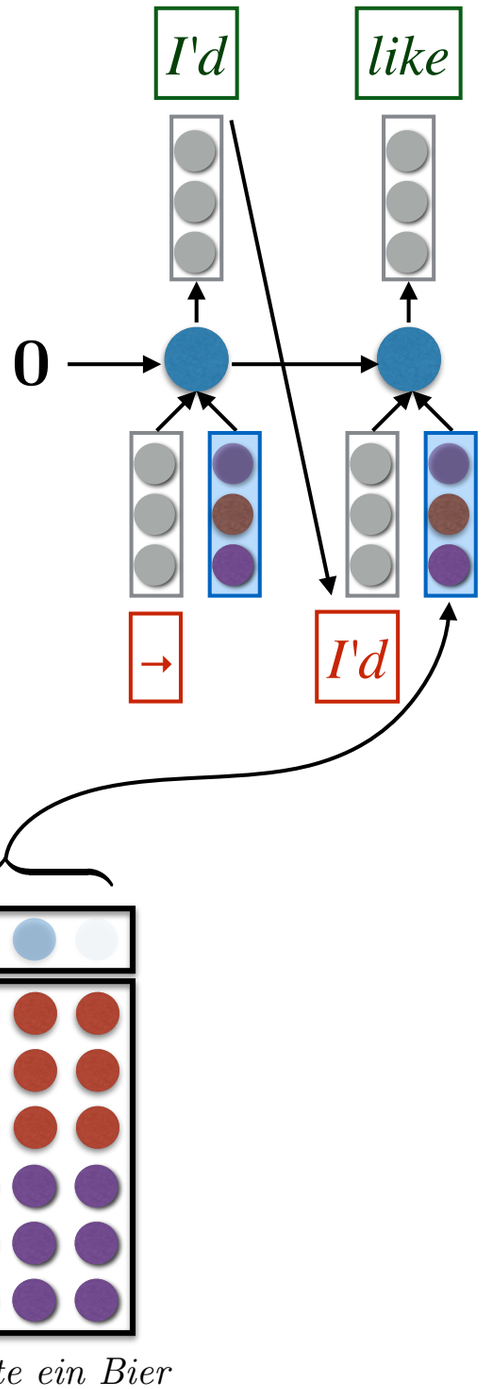
$$a_{ij} = s_{i-1} \cdot h_j$$

$$a_{ij} = s_{i-1}^\top W_a h_j$$

$$\boxed{a_{ij}} = W_a[s_{i-1}; h_j]$$

transform concatentation

Attention v2



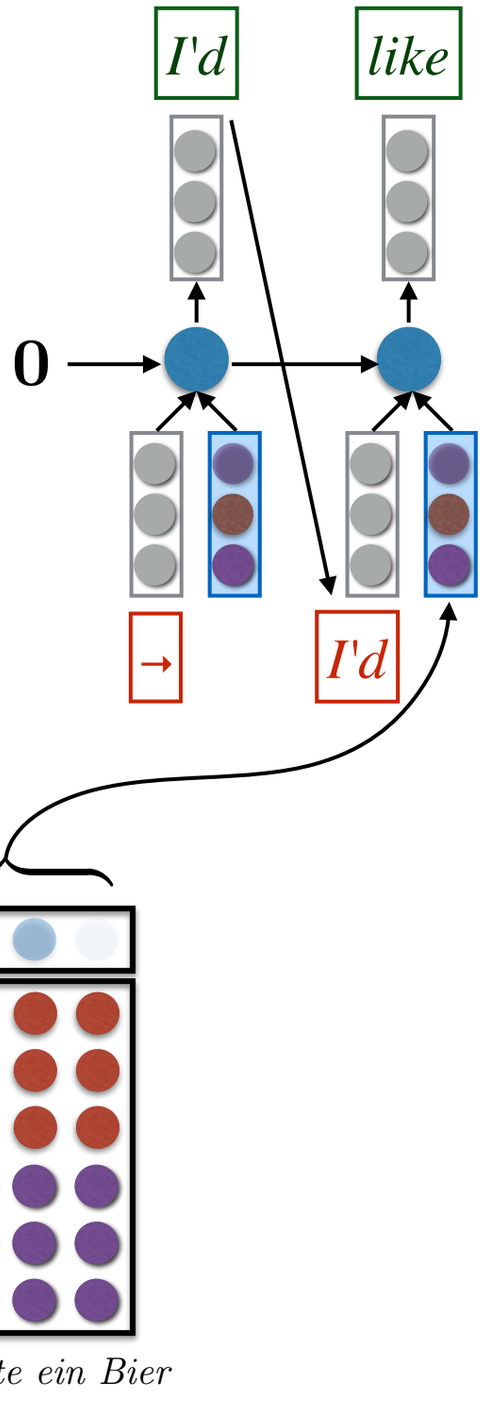
$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

$$c_i = \sum_{j=1}^{|\mathbf{e}|} \alpha_{ij} h_j$$

Soft attention

Do we need to attend to all source words?

Attention v2



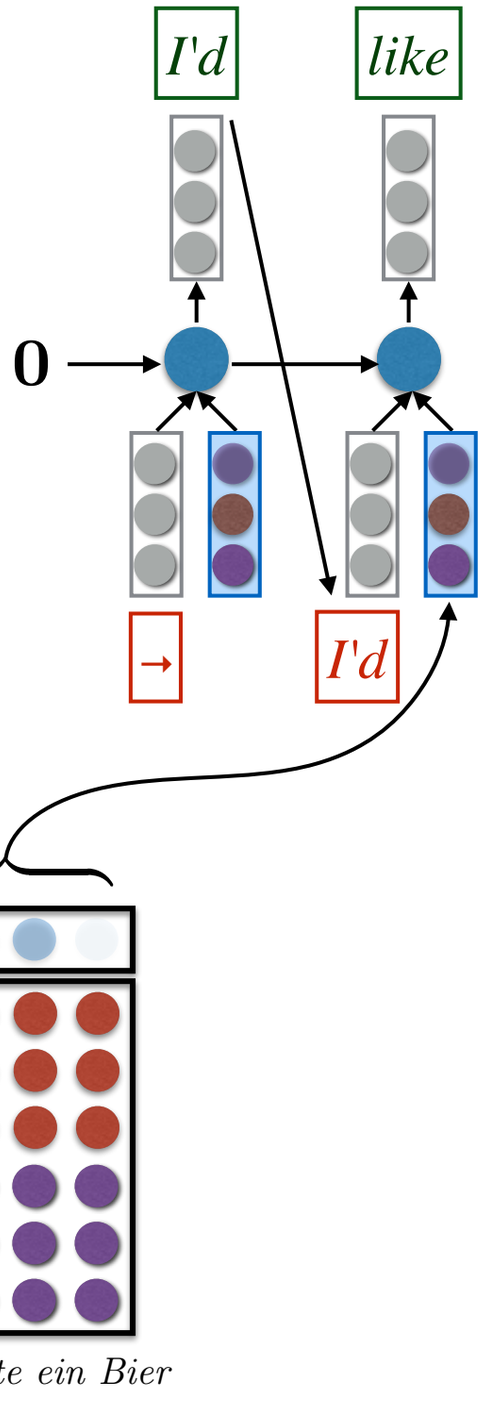
$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

$$c_i = \max_j \alpha_{ij} h_j$$

Hard attention

Difficult to train
(non-differentiable)

Attention v2



$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

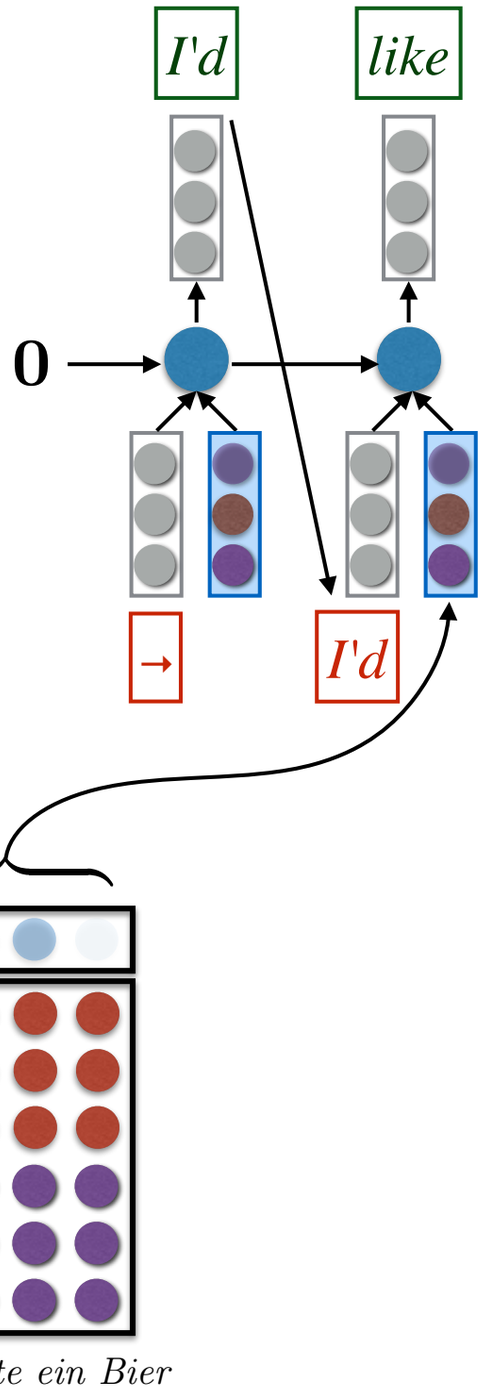
These models are all still:

$$p(\mathbf{f}|\mathbf{e}) = \prod_{i=1}^{|\mathbf{f}|} p(f_i|f_{i-1},...,f_1,\mathbf{e})$$

Summary

- Attention is closely related to “pooling” operations in convnets (and other architectures)
- Bahdanau’s attention model seems to only cares about “content”
    - No obvious bias in favor of diagonals, short jumps, fertility, etc.
    - Some work has begun to add other “structural” biases (Luong et al., 2015; Cohn et al., 2016), but there are lots more opportunities
- Attention is similar to **alignment**, but there are important differences
    - alignment makes stochastic but hard decisions. Even if the alignment probability distribution is “flat”, the model picks one word or phrase at a time
    - attention is “soft” (you add together all the words). Big difference between “flat” and “peaked” attention weights

Attention v2



$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i,p_i)$$

$$p_i = softmax(a_{ij})$$

Now alignment p is a latent variable
(but everything is differentiable;
no dependence between latent vars)

Attention v2

System	Ppl	BLEU
Winning WMT'14 system – *phrase-based + large LM* (Buck et al., 2014)		20.7
Existing NMT systems		
RNNsearch (Jean et al., 2015)		16.5
RNNsearch + unk replace (Jean et al., 2015)		19.0
RNNsearch + unk replace + large vocab + *ensemble* 8 models (Jean et al., 2015)		**21.6**
Our NMT systems		
Base	10.6	11.3
Base + reverse	9.9	12.6 (+*1.3*)
Base + reverse + dropout	8.1	14.0 (+*1.4*)
Base + reverse + dropout + global attention (*location*)	7.3	16.8 (+*2.8*)
Base + reverse + dropout + global attention (*location*) + feed input	6.4	18.1 (+*1.3*)
Base + reverse + dropout + local-p attention (*general*) + feed input	5.9	19.0 (+*0.9*)
Base + reverse + dropout + local-p attention (*general*) + feed input + unk replace		20.9 (+*1.9*)
Ensemble 8 models + unk replace		**23.0** (+*2.1*)

Attention v2

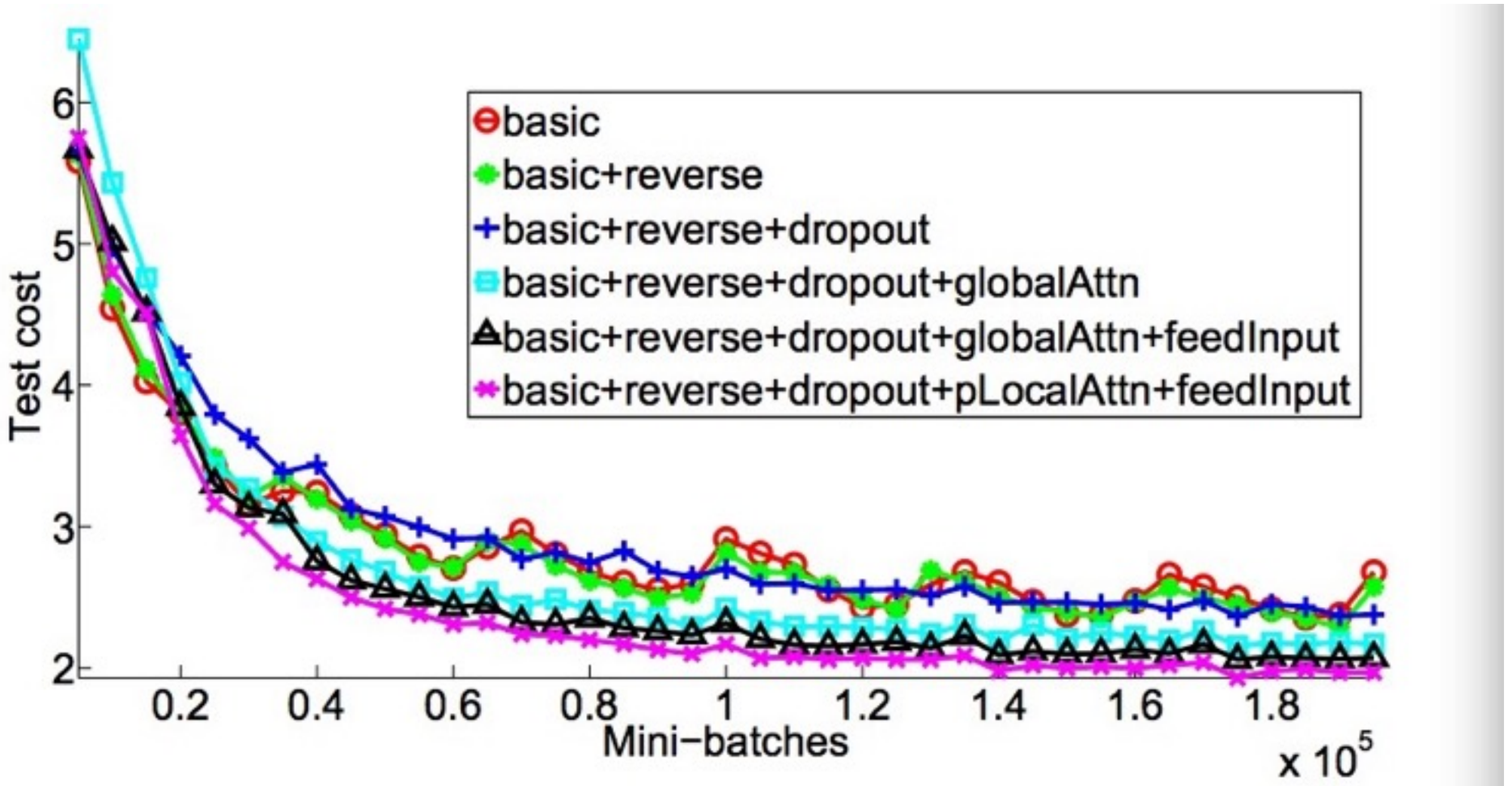


Figure 5: **Learning curves** – test cost (ln perplexity) on newstest2014 for English-German NMTs as training progresses.

Attention v2

Method	AER
global (location)	0.39
local-m (general)	0.34
local-p (general)	0.36
ensemble	0.34
Berkeley Aligner	0.32

Table 6: **AER scores** – results of various models on the RWTH English-German alignment data.

Attention v2

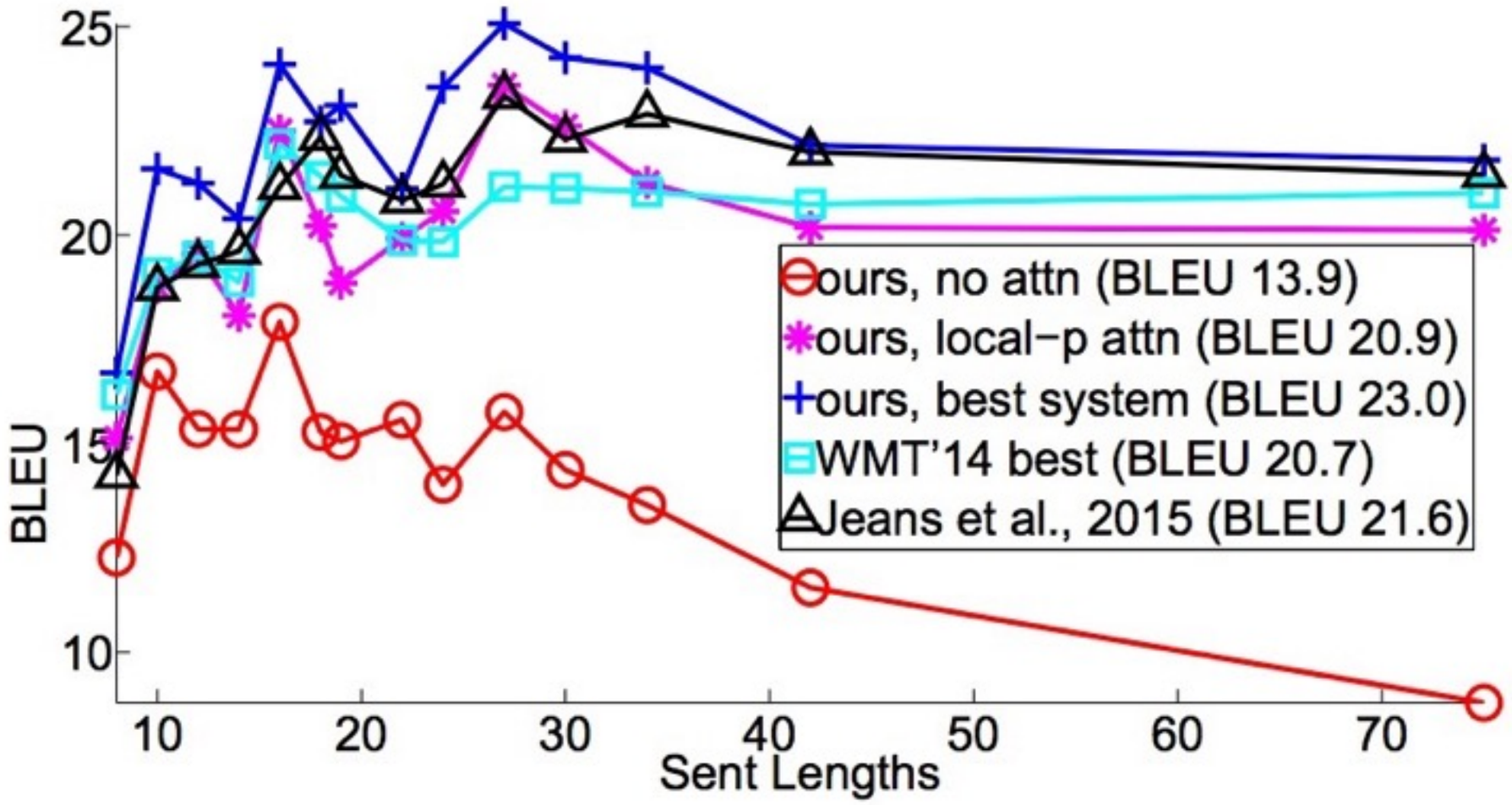


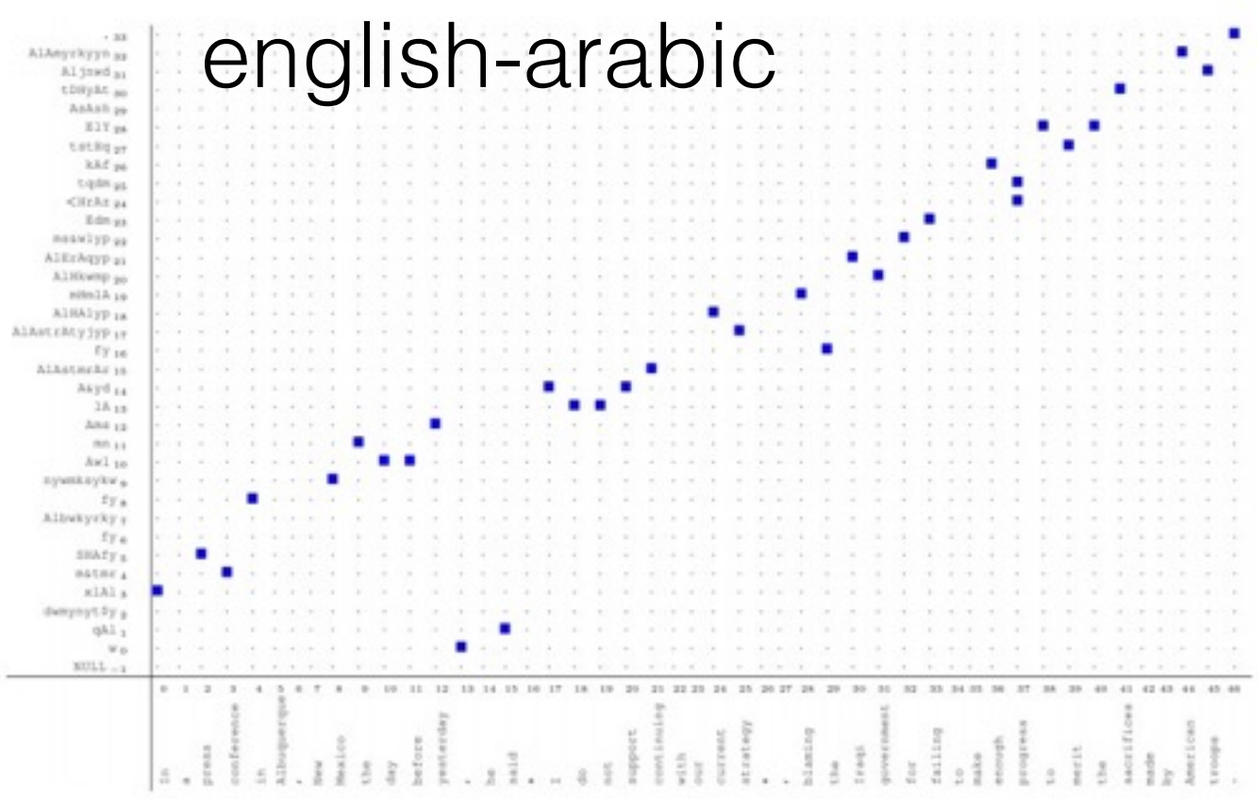
Figure 6: **Length Analysis** – translation qualities of different systems as sentences become longer.

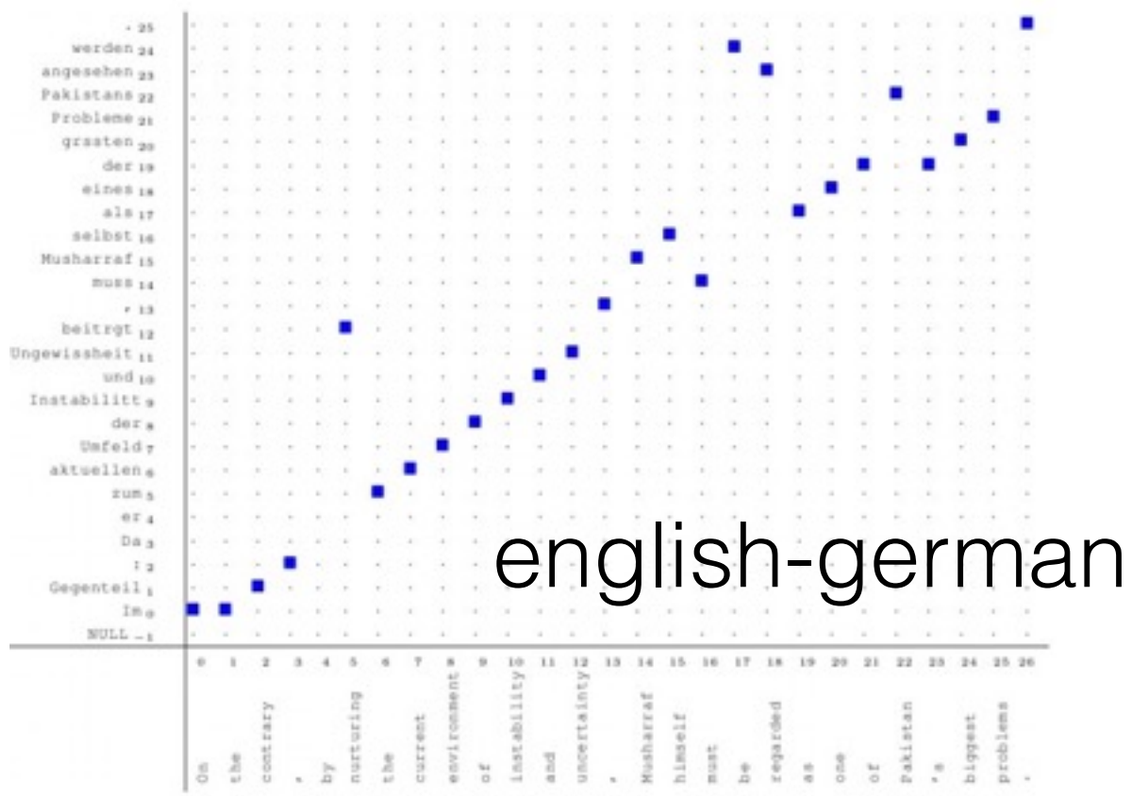
Recall IBM models

- Words have “fertilities”
- Bias towards translation along the diagonal.

Recall IBM models

- Words have “fertilities”
- Bias towards translation along the diagonal.

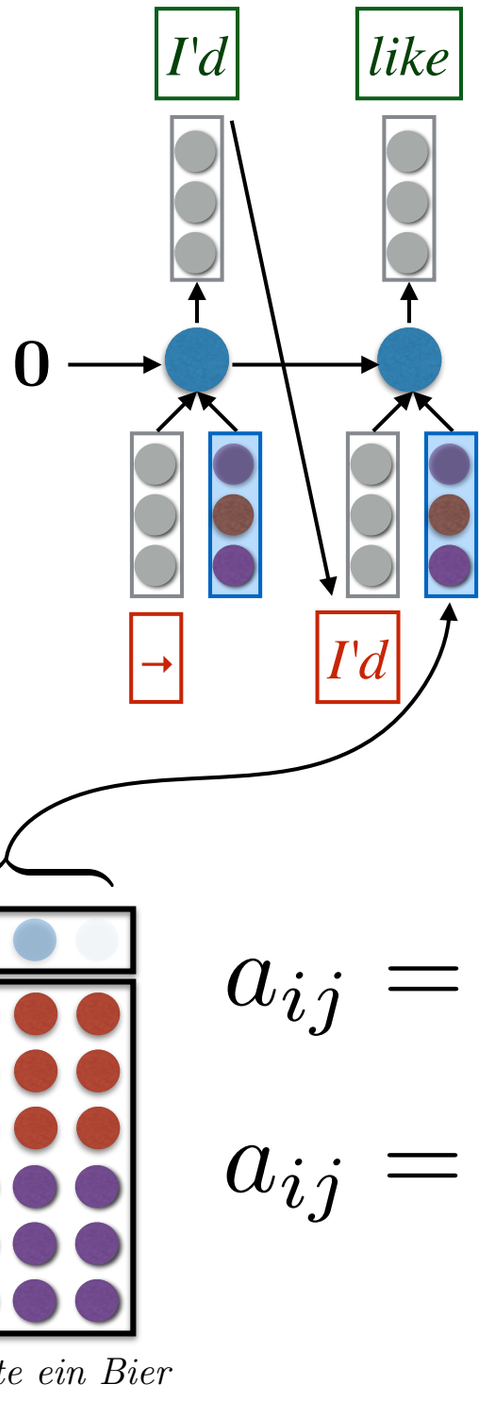




Recall IBM models

- Words have “fertilities”
- Bias towards translation along the diagonal.
- Are the same biases useful in NMT?

Attention v3



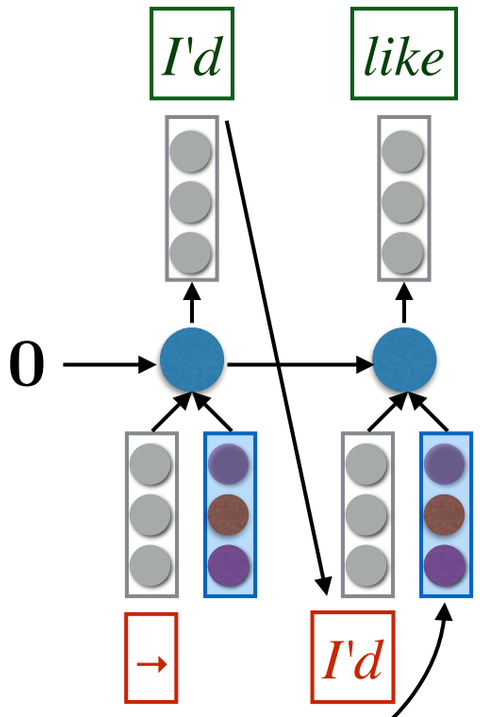
$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

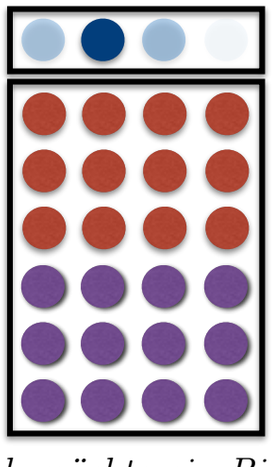
$$a_{ij} = a(s_{i-1},h_j)$$

Cohn et al. variant:

$$a_{ij} = w^\top \tanh(W_1 s_{i-1} + W_2 h_j + W_3 \psi(j,i,I))$$

Attention v3





Ich möchte ein Bier

$$p(f_i|f_{i-1},...,f_1,\mathbf{e}) = g(f_{i-1},s_i,c_i)$$

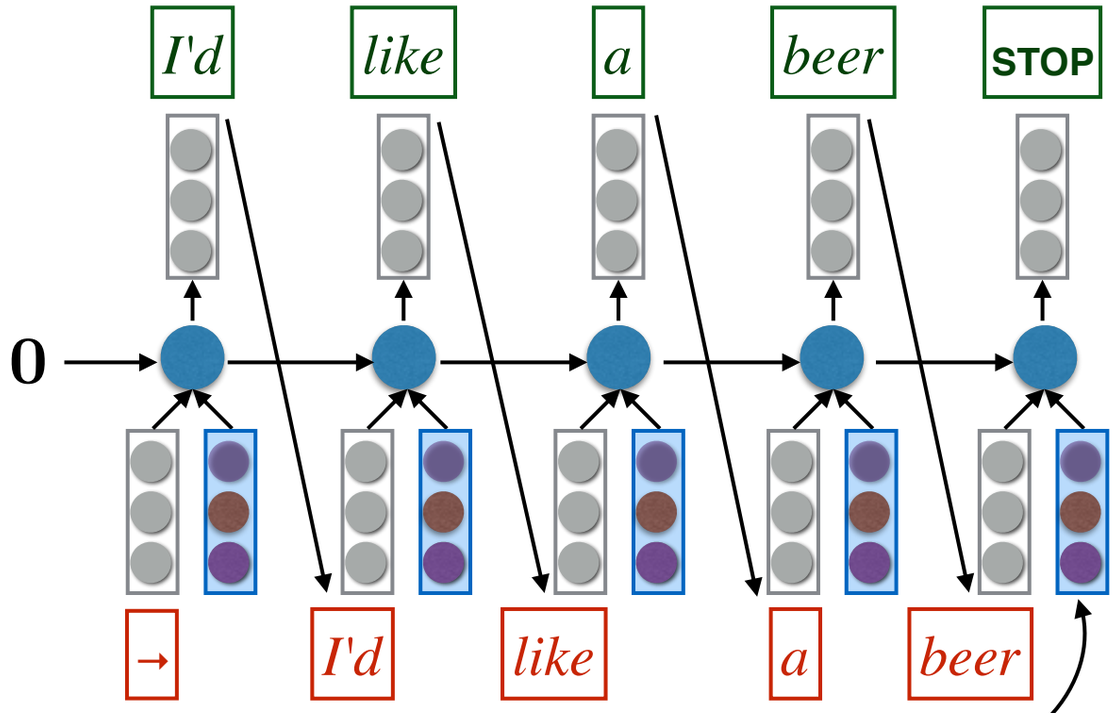
$$a_{ij} = a(s_{i-1},h_j)$$

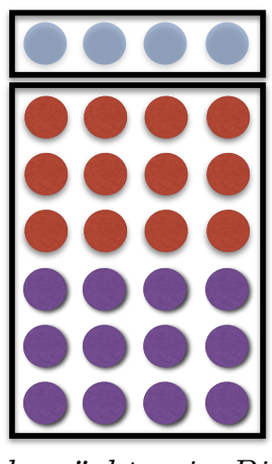
Cohn et al. variant:

$$a_{ij} = w^\top \tanh(W_1 s_{i-1} + W_2 h_j + W_3 \psi(j,i,I))$$

$$\psi(j,i,I) = [\log(1+j); \log(1+i); log(1+I)]$$

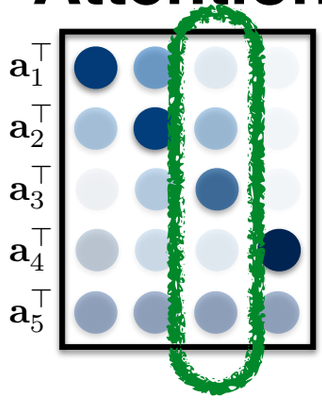
Attention v3





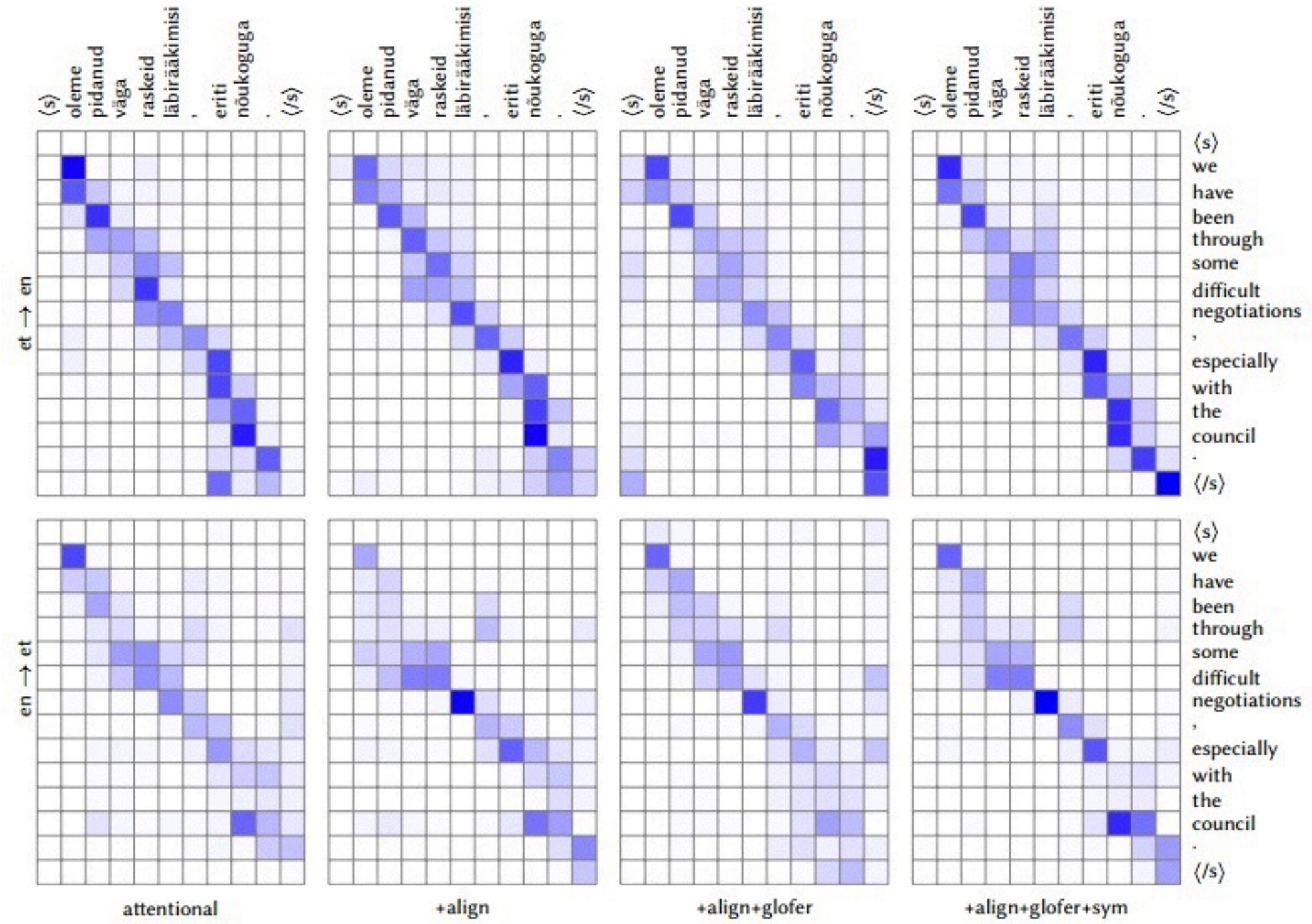
Ich möchte ein Bier

Attention history:



Cohn et al. fertility approximation: bias column sum towards a (Gaussian) mean in training (similar idea in Tu et al.)

Attention v3

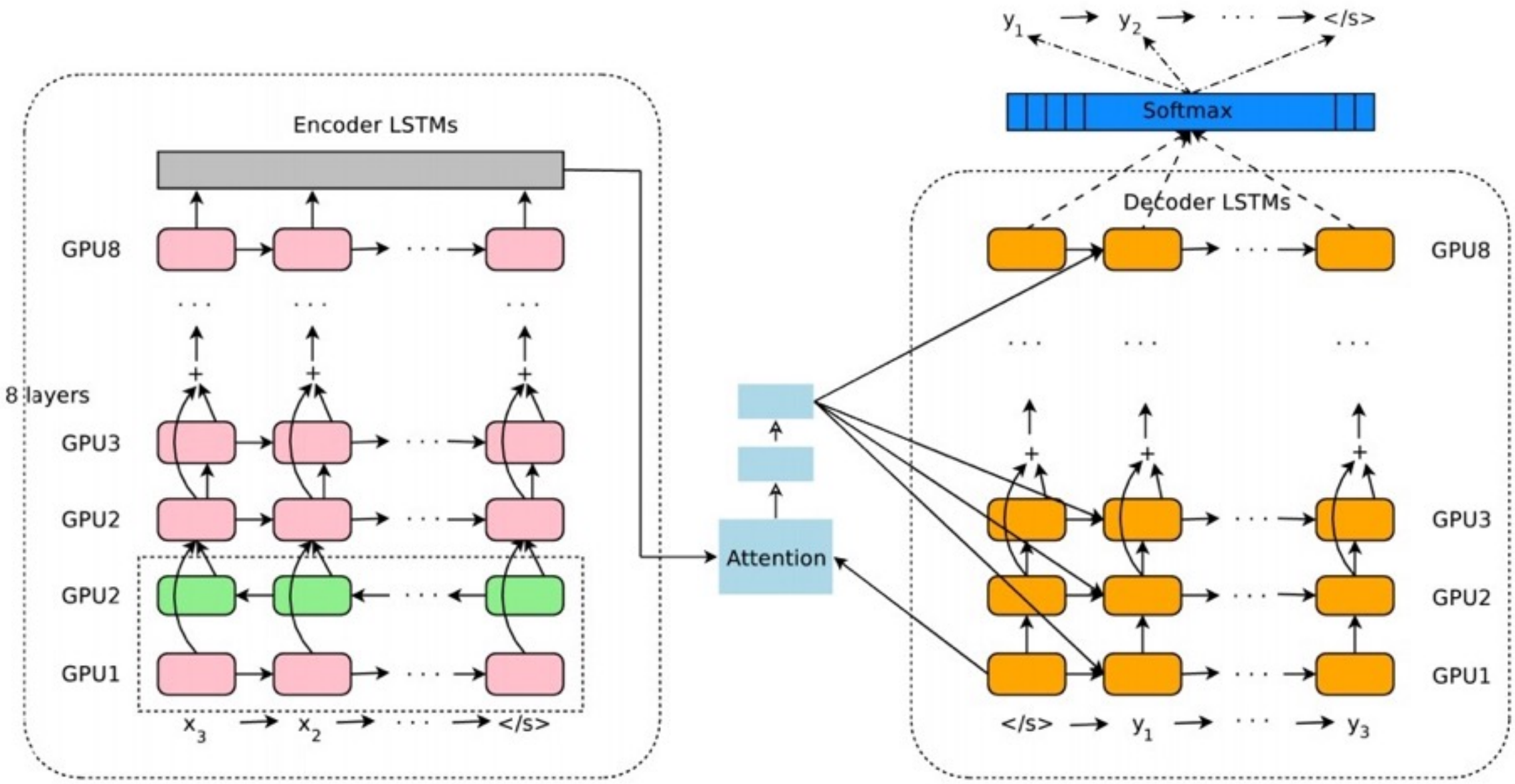


Attention v3

Lang. Pair	Zh-En	Ru-En	Et-En	Ro-En
Phrase-based	40.63	18.70	31.99	45.21
Enc-Dec	40.41	18.83	32.20	45.36
Attentional	41.16♣	19.79	32.78	46.83
Our Work	44.14♣♠	19.73	33.26♠	46.88

Table 4: BLEU scores on the test sets for re-ranking. **bold:** Best performance, ♠: Significantly better than Attentional, ♣: Using ensemble of models.

What happens here?



Things we don't know

- Are the information bottlenecks set up correctly?
- Are we incorporating prior knowledge in a useful way?
- What does the attention model learn?

Things we do know

- Attention is really, really useful
- Essentially: use an external memory the size of the input.
- Many variants on this idea …

speech

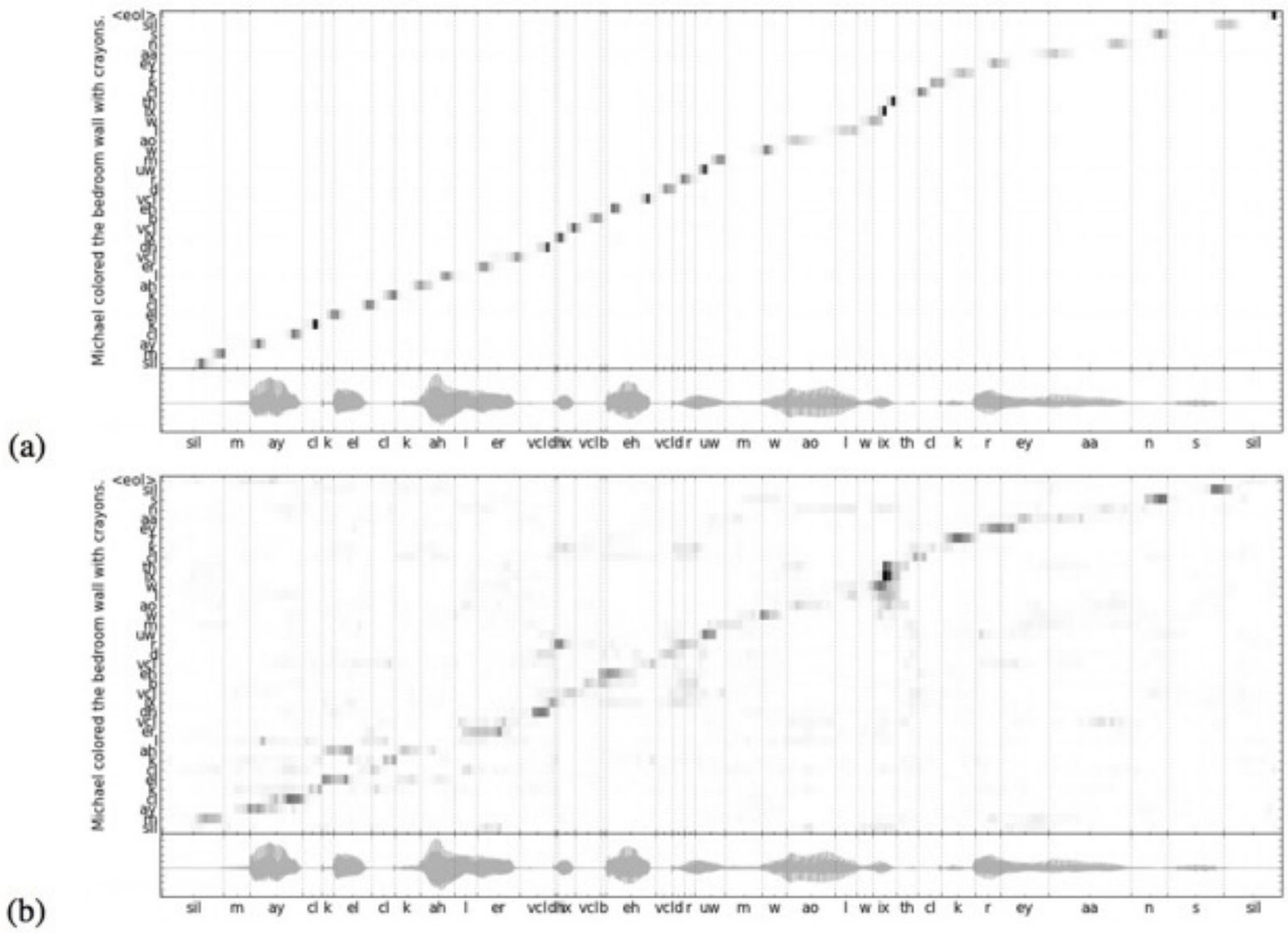


Figure 2: Alignments produced by the model: (a) when the alignment was successfully encouraged to be monotonic, and (b) when the model is free to select any frame in the input sequence. Each row in the plot contains the scores computed by the attention mechanism between the previous hidden state and all the input annotations. In (b), we observe how the absence of the learned preference for monotonicity makes the model confused by the repeated occurrence of the phonemes "cl k", and to a lesser degree, by the repetition of "w".

summarization

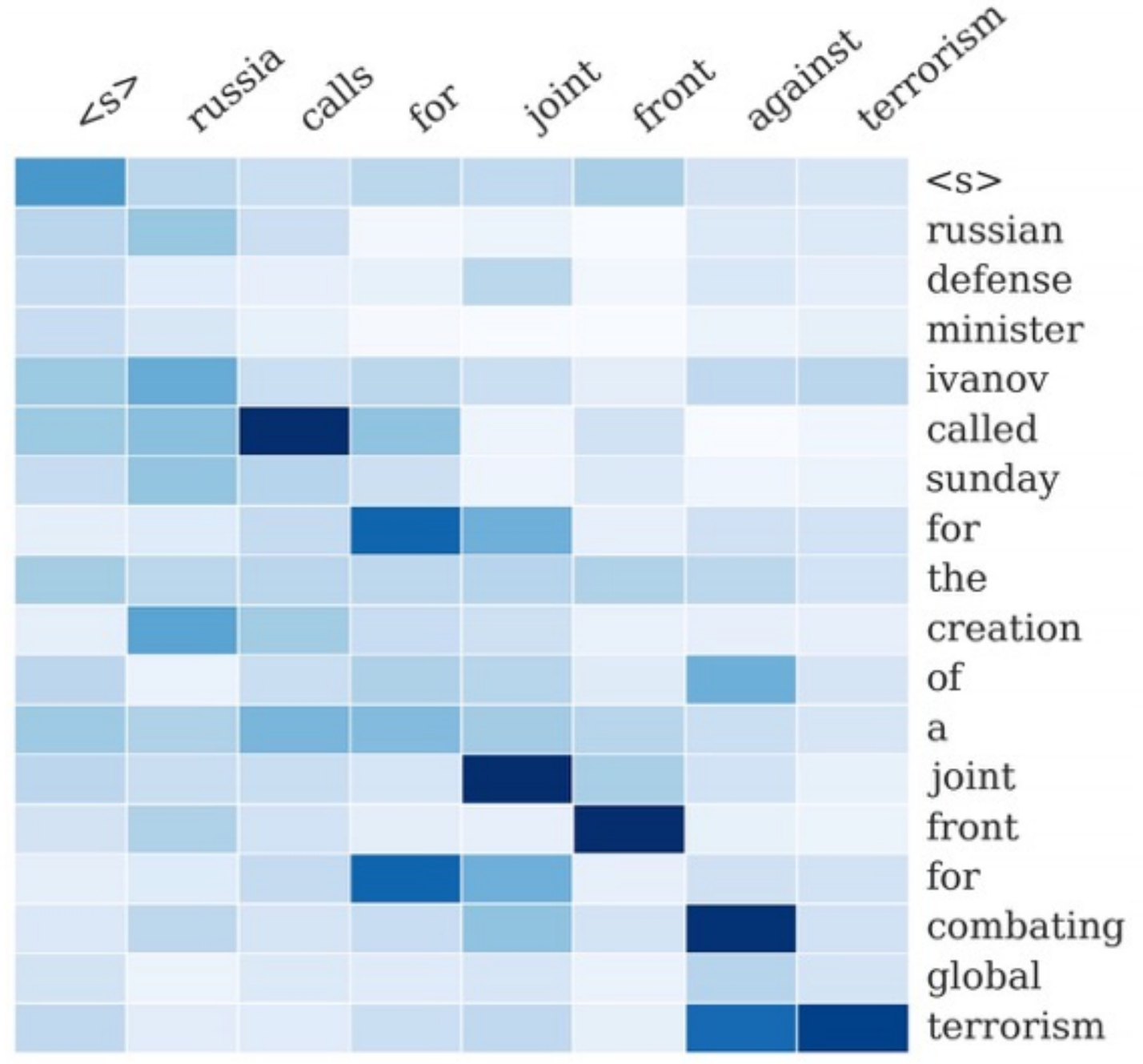


image captioning

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

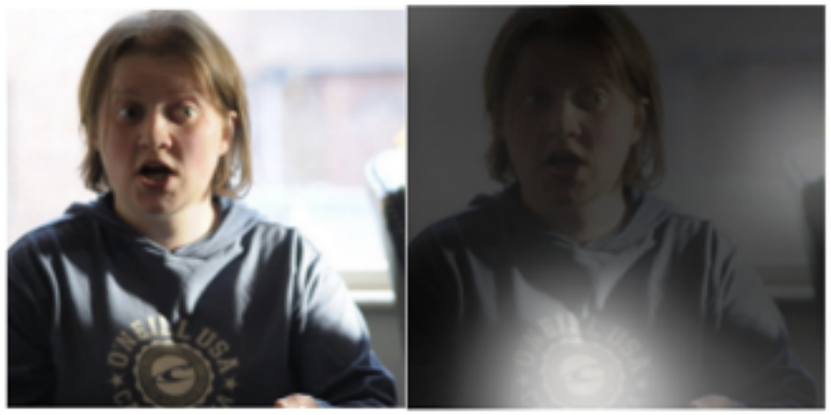
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

image captioning

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.

A large white bird standing in a forest.



A woman holding a clock in her hand.

A man wearing a hat and
a hat on a skateboard.

A person is standing on a beach
with a surfboard.

A woman is sitting at a table
with a large pizza.

A man is talking on his cell phone
while another man watches.