NLU: Semantic parsing

Adam Lopez
slide credits: Chris Dyer, Nathan Schneider
March 30, 2018

School of Informatics
University of Edinburgh
alopez@inf.ed.ac.uk

Recall: meaning representations

Sam likes Casey

likes(Sam, Casey)

Anna's dog Mr. PeanutButter misses her

misses(MrPB, Anna) $\wedge$ dog(MrPB)

Kim likes everyone

$\forall x$.likes(x, Kim)

Recall: meaning representations

- Meaning representations are verifiable, unambiguous, canonical.
- Predicate-argument structure is a good match for FOL, as well as structures with argument-like elements (e.g. NPs)
- Determiners, quantifiers (e.g. “everyone”, “anyone”), and negation can be expressed in first order logic.

Representing the meaning of arbitrary NL is **hard**

- Much of natural language is unverifiable, ambiguous, non-canonical.
- What is the finite set of predicates?
  - When do two words map to the same predicate?
  - When are homonyms mapped to different predicates?
  - Can we decompose lexical meanings into a finite set of predicates (possibly composed)?
- What is the finite set of constants?

Easier: representing a closed domain

Example: GEOQUERY dataset

What states border Texas?

$\lambda x.\ \mathrm{state}(x) \wedge \mathrm{borders}(x,\mathrm{texas})$

What is the largest state?

$\mathrm{argmax}(\lambda x.\ \mathrm{state}(x) \wedge \lambda x.\mathrm{size}(x))$

Semantic parsing is the problem of returning a logical form for an input natural language sentence.

Pairs of NL sentences with structured MR can be collected…

Example: IFTTT dataset (Quirk et al. 2015)

	INPUT	Park in garage when snow tomorrow		
(a)	IFTTT	Weather : Tomorrow's_forecast_calls_for	⟹	SMS : Send_me_an_SMS
	OUTPUT	Weather : Tomorrow's_forecast_calls_for	⟹	SMS : Send_me_an_SMS
	INPUT	Suas fotos do instagr.am salvas no dropbox		
(b)	IFTTT	Instagram : Any_new_photo_by_you	⟹	Dropbox : Add_file_from_URL
	OUTPUT	Instagram : Any_new_photo_by_you	⟹	Dropbox : Add_file_from_URL
	INPUT	Foursquare check-in archive		
(c)	IFTTT	Foursquare : Any_new_check-in	⟹	Evernote : Create_a_note
	OUTPUT	Foursquare : Any_new_check-in	⟹	Google_Drive : Add_row_to_spreadsheet
	INPUT	if i post something on blogger it will post it to wordpress		
(d)	IFTTT	Blogger : Any_new_post	⟹	WordPress : Create_a_post
	OUTPUT	Feed : New_feed_item	⟹	Blogger : Create_a_post
	INPUT	Endless loop!		
(e)	IFTTT	Gmail : New_email_in_inbox_from	⟹	Gmail : Send_an_email
	OUTPUT	SMS : Send_IFTTT_any_SMS	⟹	Philips_hue : Turn_on_color_loop

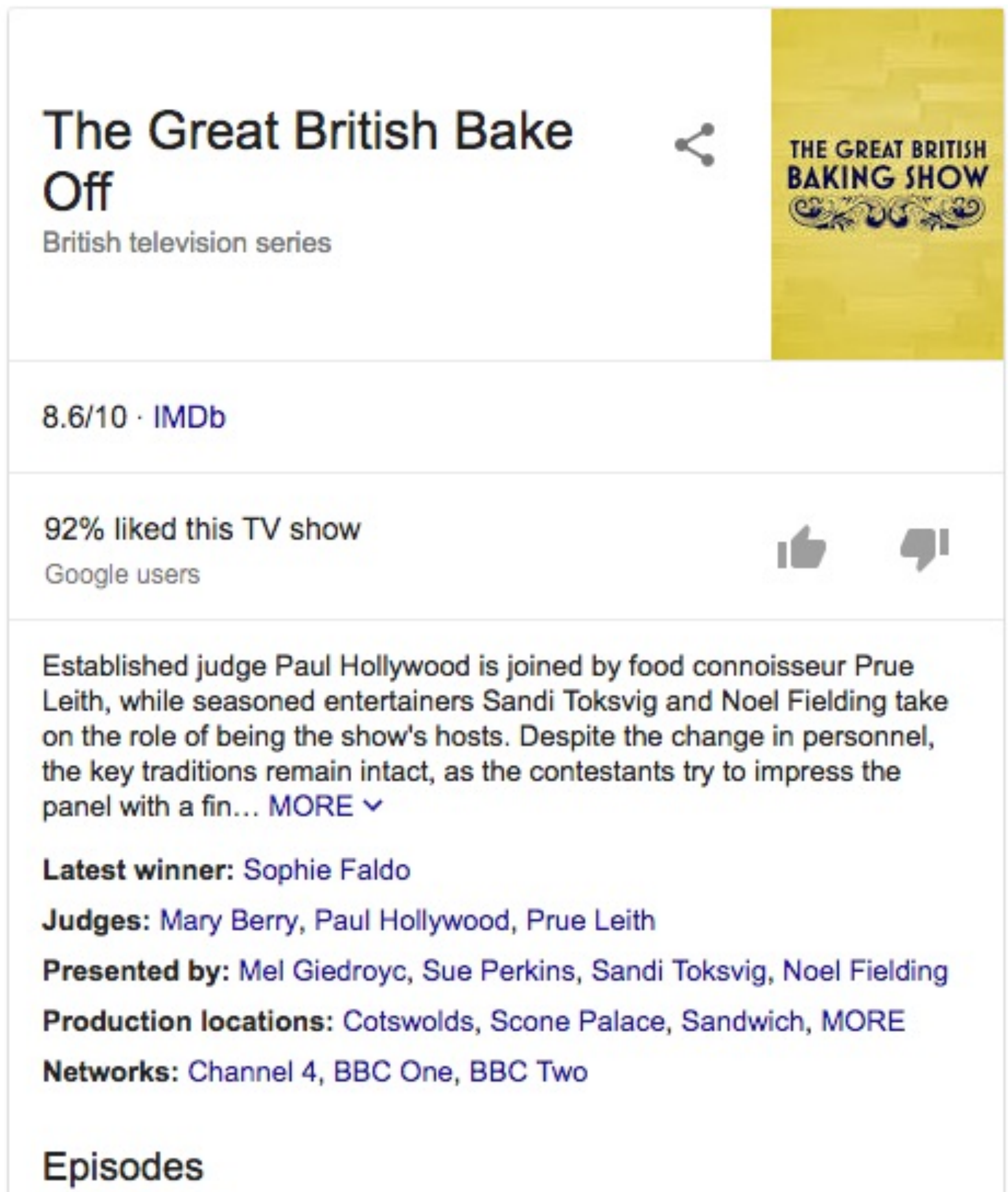
WikiTableQuestions

Year	City	Country	Nations
1896	Athens	Greece	14
1900	Paris	France	24
1904	St. Louis	USA	12
...	...	...	...
2004	Athens	Greece	201
2008	Beijing	China	204
2012	London	UK	204

x = Greece held its last Summer Olympics in which year?

y = 2004

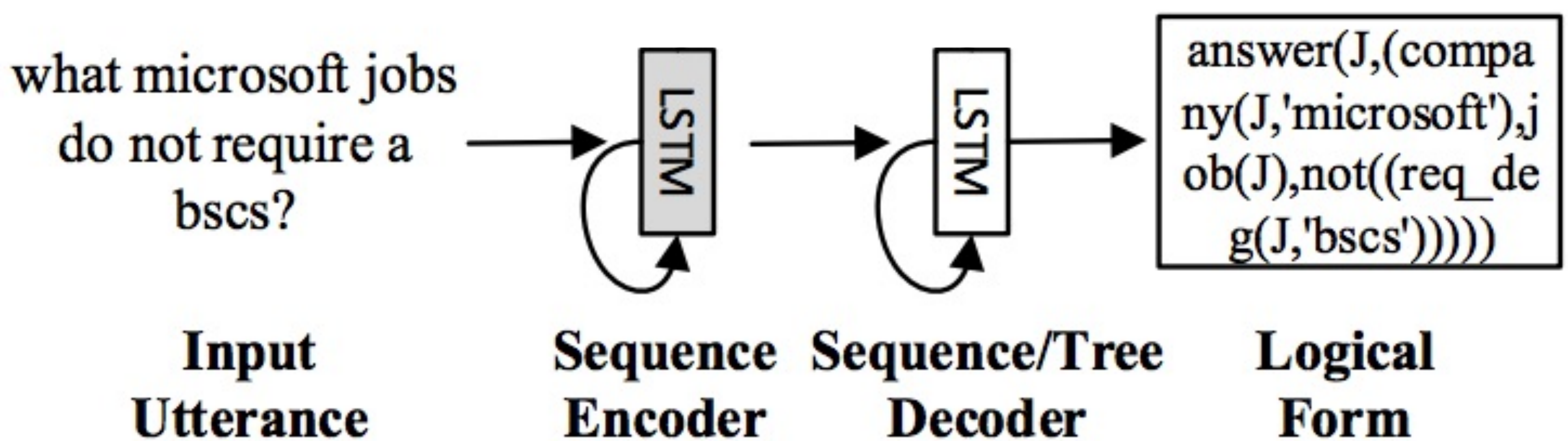
…similar information powers Google's knowledge graph



Viewing MR as a string, semantic parsing is just **conditional language modeling**

$$p(y_1, ..., y_{|y|} \mid x_1, ..., x_{|x|})$$

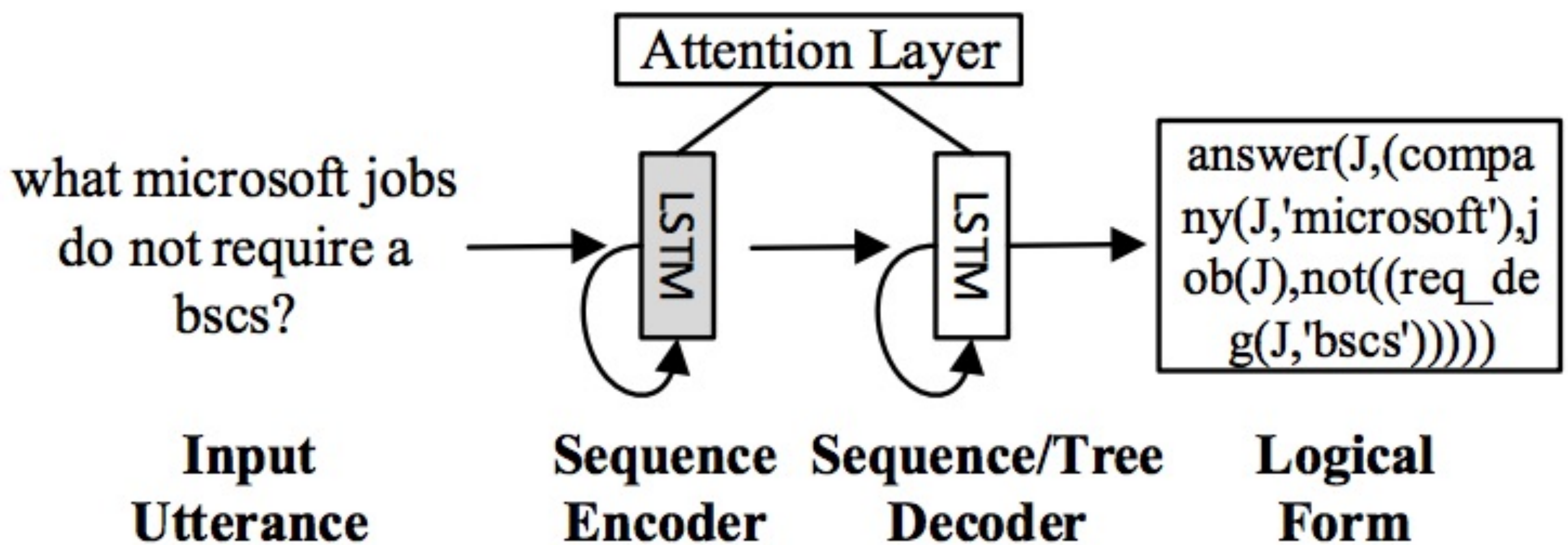
Model using standard sequence models…



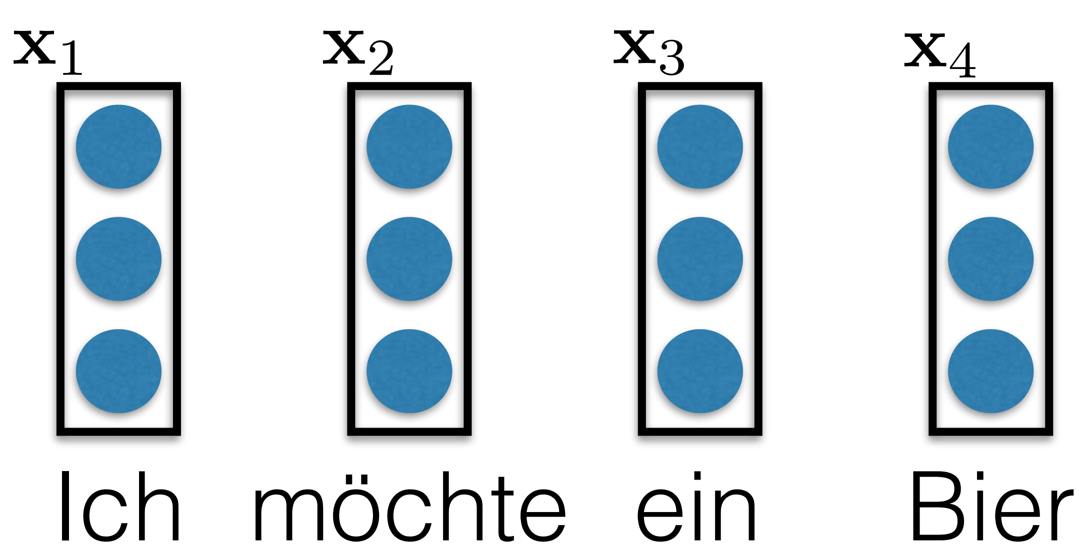
Viewing MR as a string, semantic parsing is just **conditional language modeling**

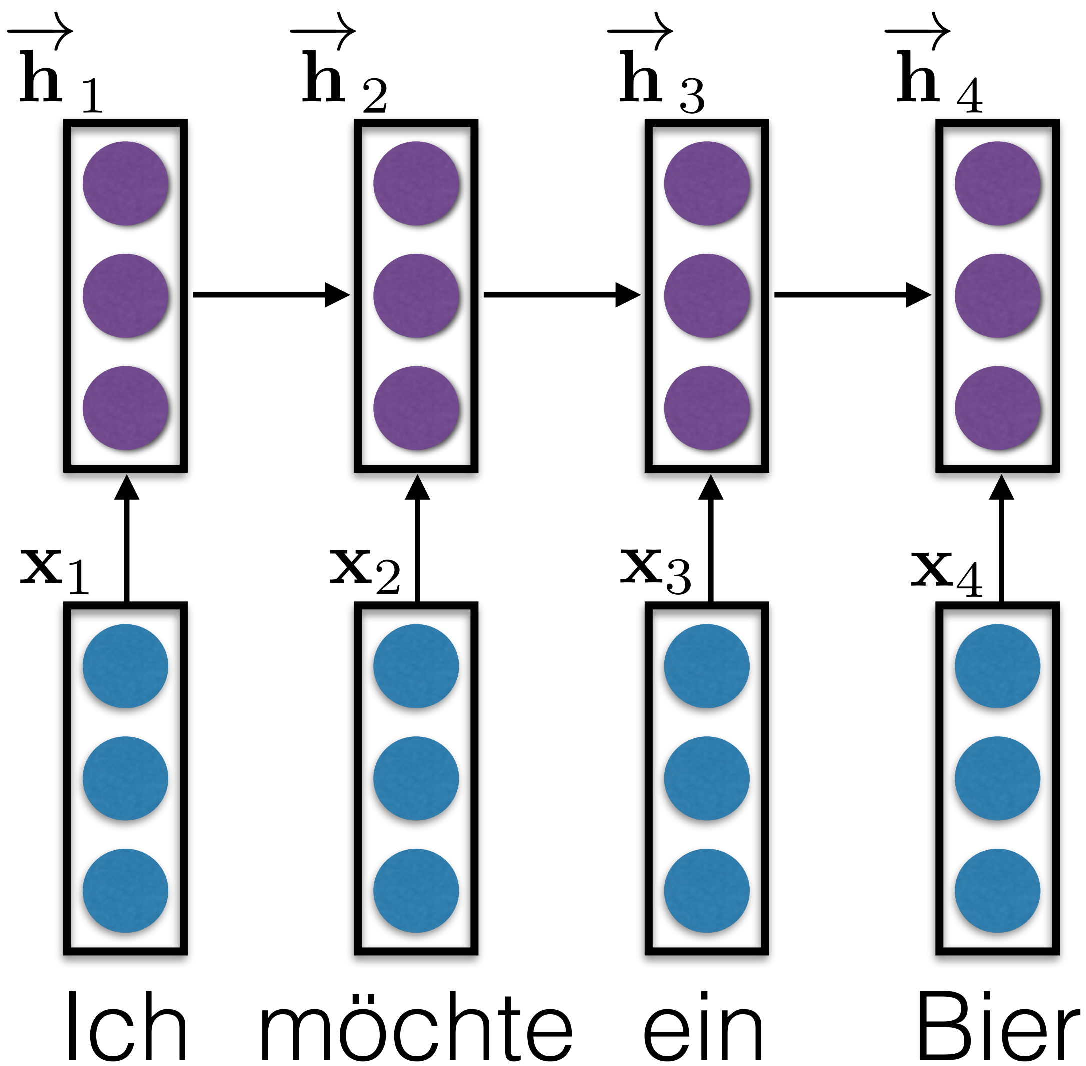
$$p(y_1, ..., y_{|y|} \mid x_1, ..., x_{|x|})$$

Model using standard sequence models…

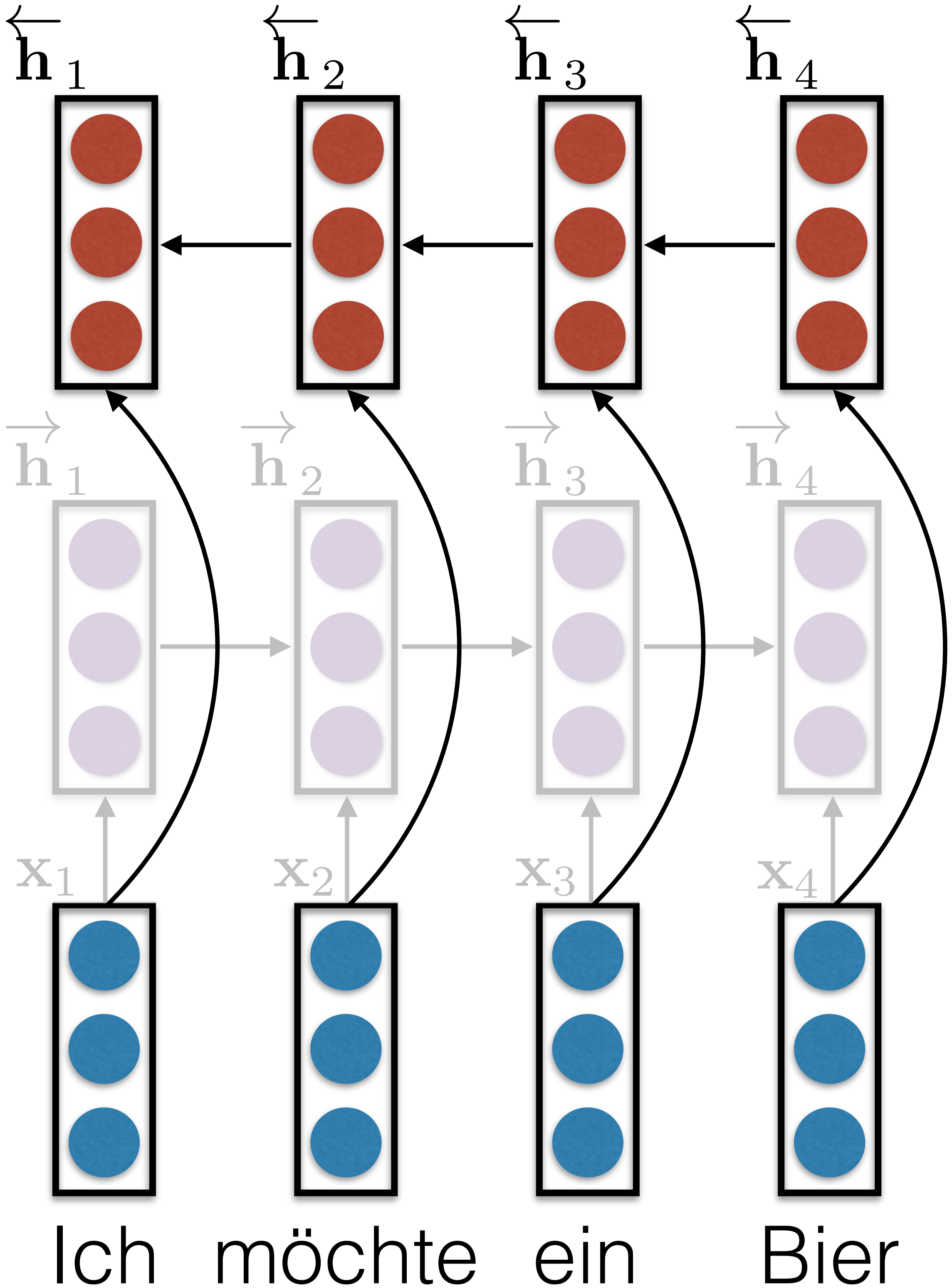


…with one additional element


$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_4$
Ich
möchte
ein
Bier

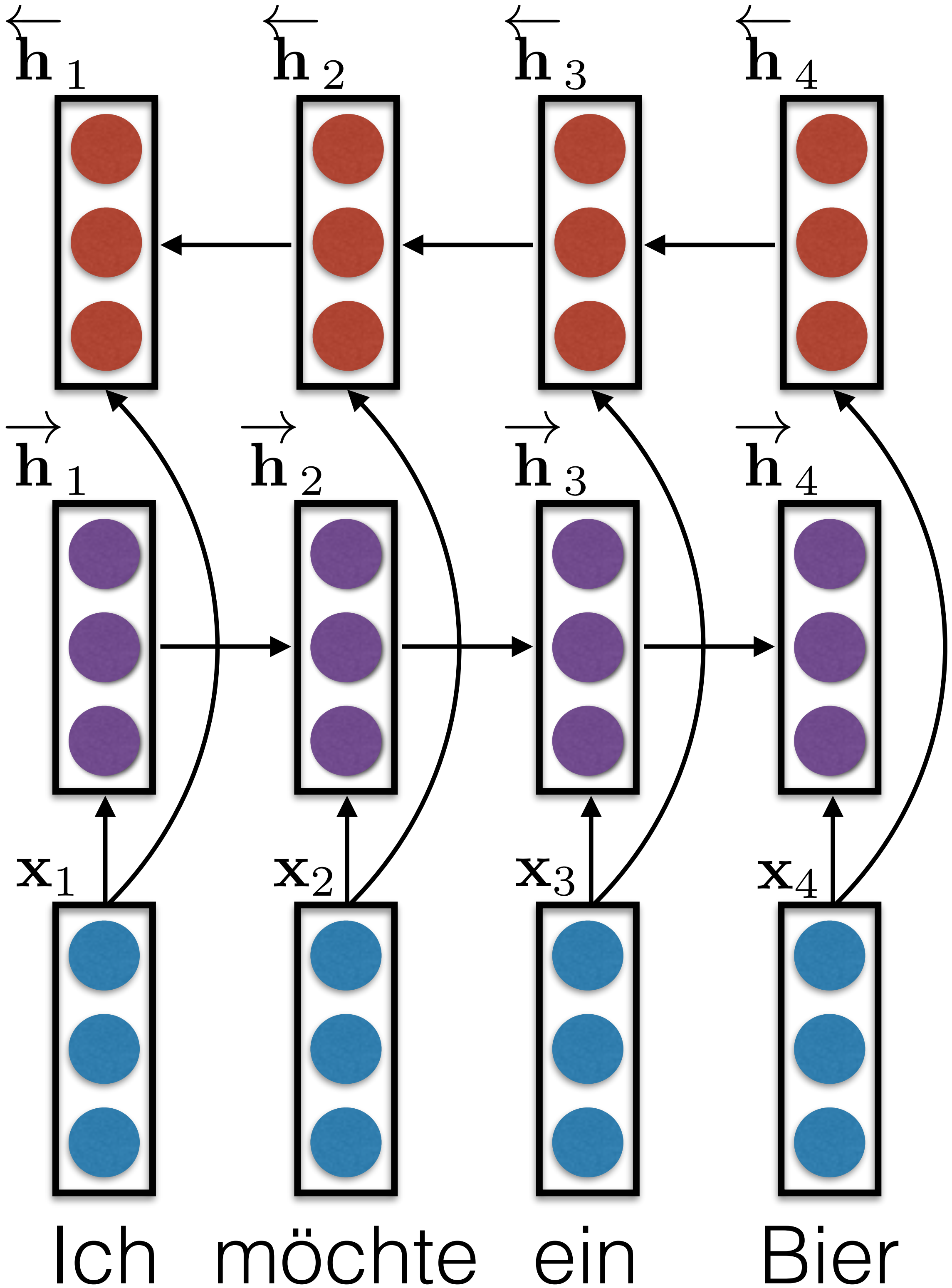


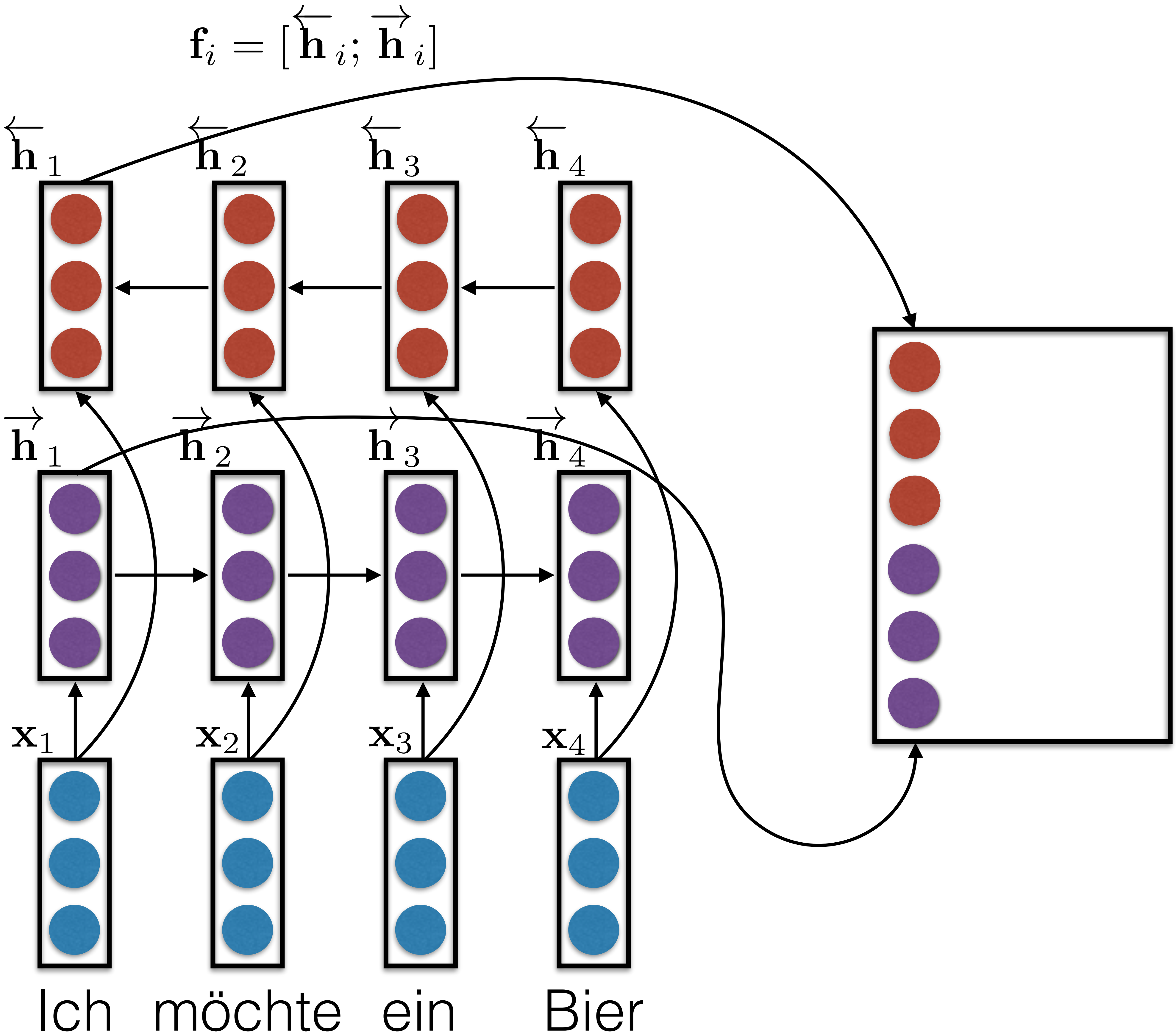
$\overrightarrow{\mathbf{h}}_1$
$\overrightarrow{\mathbf{h}}_2$
$\overrightarrow{\mathbf{h}}_3$
$\overrightarrow{\mathbf{h}}_4$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_4$
Ich
möchte
ein
Bier



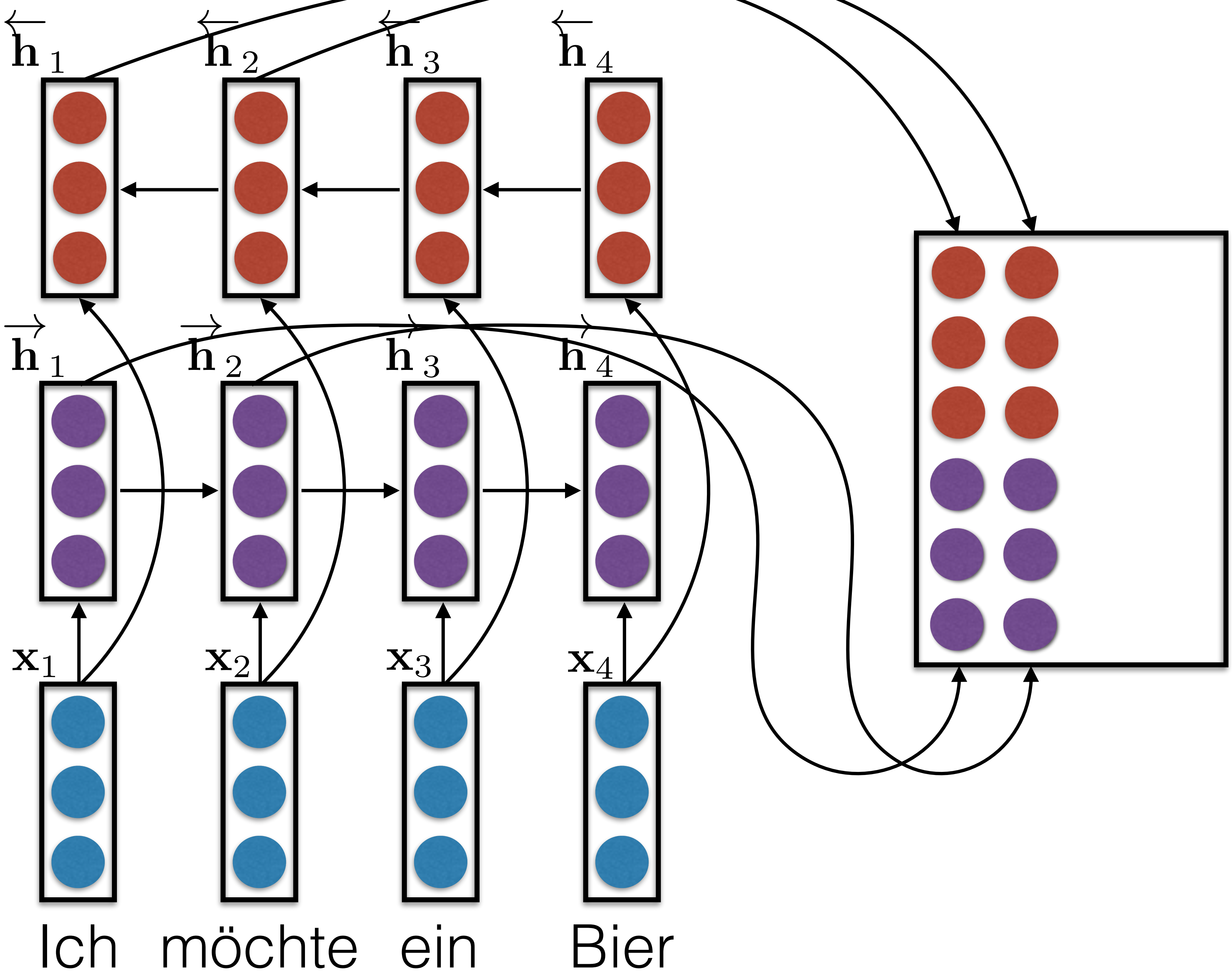
$\overleftarrow{\mathbf{h}}_1$
$\overleftarrow{\mathbf{h}}_2$
$\overleftarrow{\mathbf{h}}_3$
$\overleftarrow{\mathbf{h}}_4$
$\overrightarrow{\mathbf{h}}_1$
$\overrightarrow{\mathbf{h}}_2$
$\overrightarrow{\mathbf{h}}_3$
$\overrightarrow{\mathbf{h}}_4$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_4$
Ich
möchte
ein
Bier

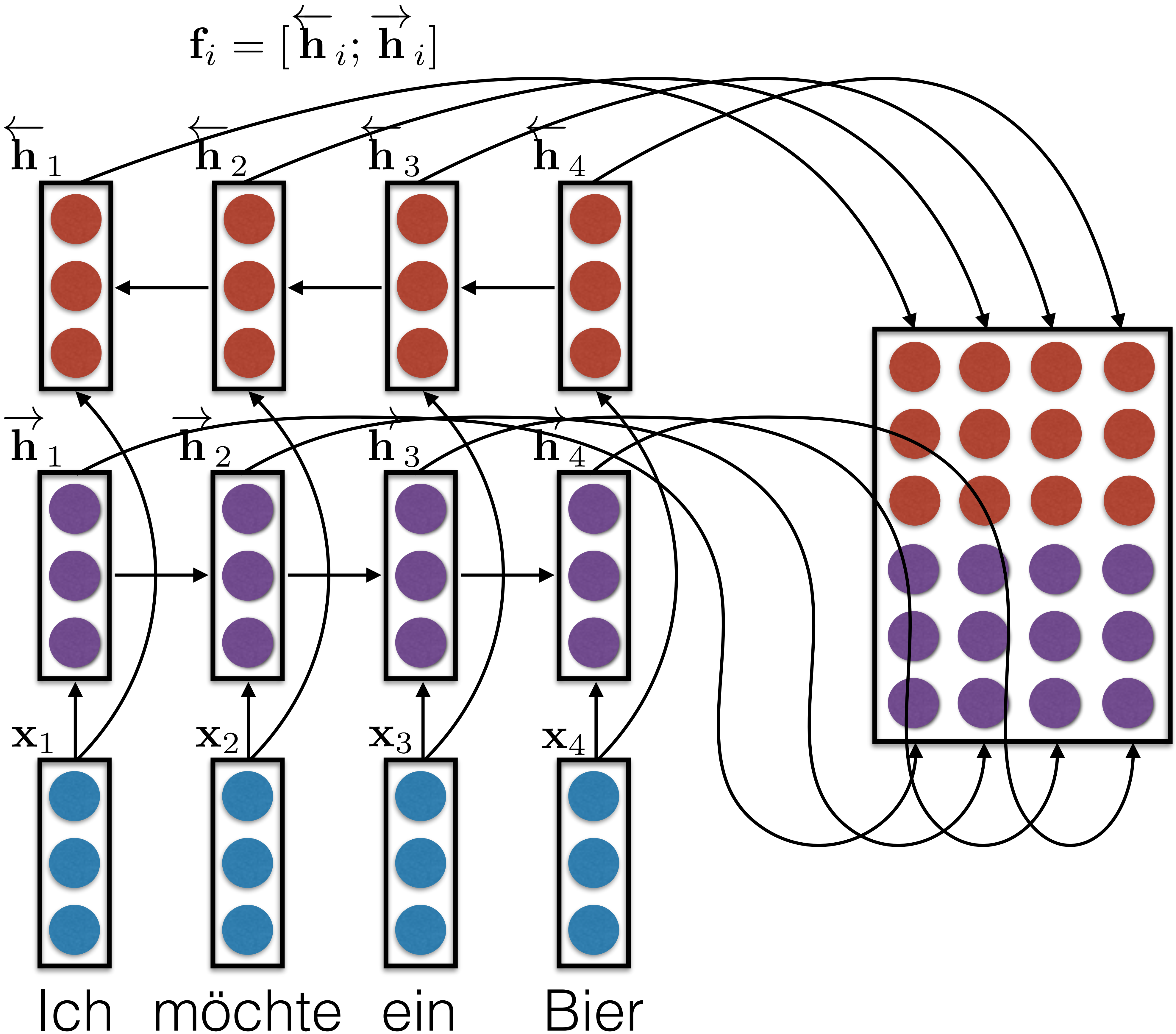
$$\mathbf{f}_i = [\overleftarrow{\mathbf{h}}_i; \overrightarrow{\mathbf{h}}_i]$$



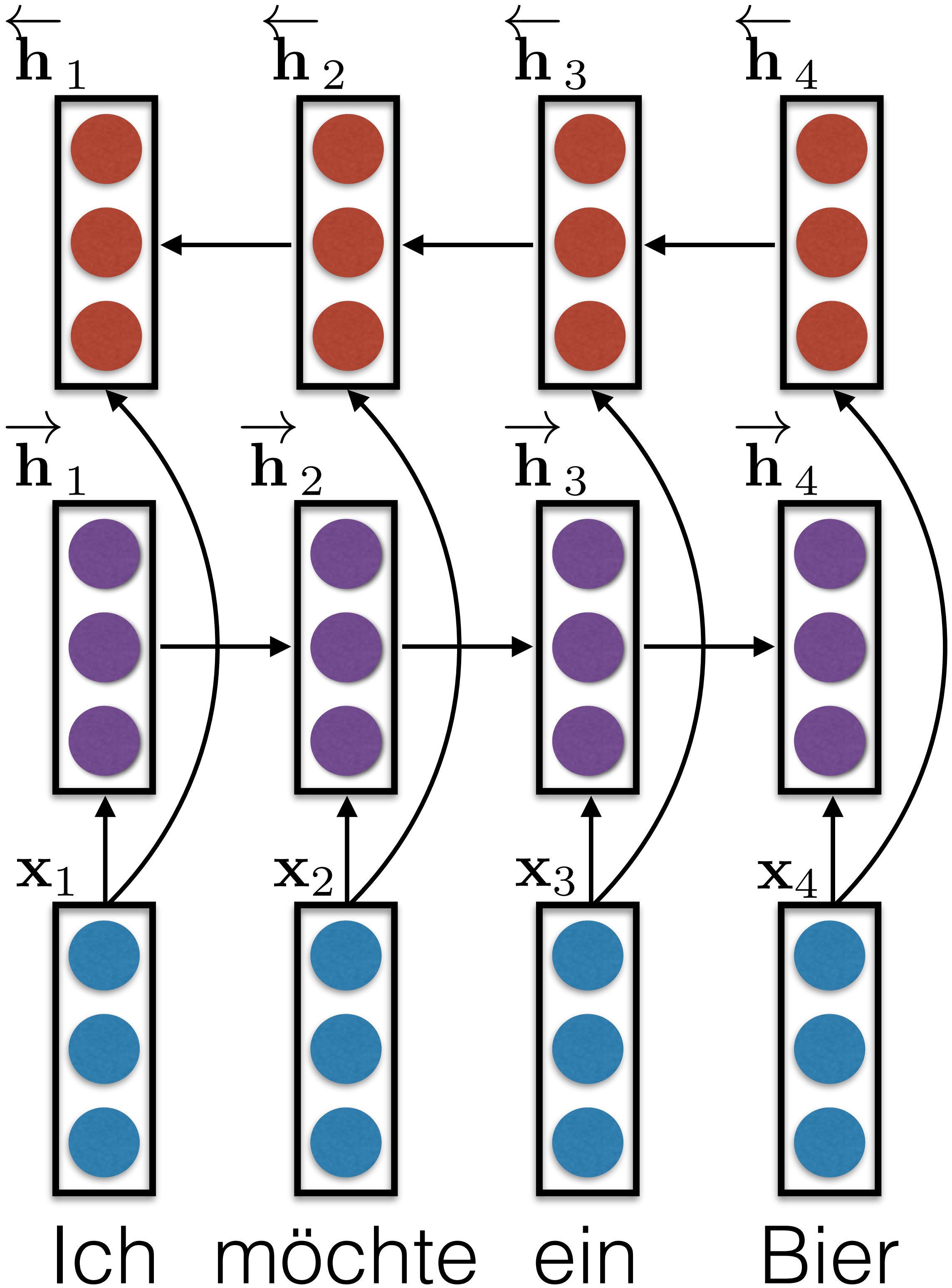

$\mathbf{f}_i = [\overleftarrow{\mathbf{h}}_i; \overrightarrow{\mathbf{h}}_i]$
$\overleftarrow{\mathbf{h}}_1$
$\overleftarrow{\mathbf{h}}_2$
$\overleftarrow{\mathbf{h}}_3$
$\overleftarrow{\mathbf{h}}_4$
$\overrightarrow{\mathbf{h}}_1$
$\overrightarrow{\mathbf{h}}_2$
$\overrightarrow{\mathbf{h}}_3$
$\overrightarrow{\mathbf{h}}_4$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

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

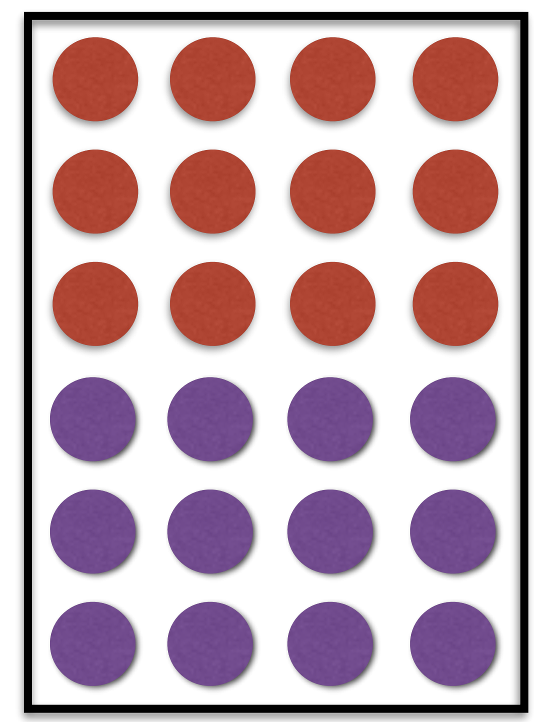



$\mathbf{f}_i = [\overleftarrow{\mathbf{h}}_i; \overrightarrow{\mathbf{h}}_i]$
$\overleftarrow{\mathbf{h}}_1$
$\overleftarrow{\mathbf{h}}_2$
$\overleftarrow{\mathbf{h}}_3$
$\overleftarrow{\mathbf{h}}_4$
$\overrightarrow{\mathbf{h}}_1$
$\overrightarrow{\mathbf{h}}_2$
$\overrightarrow{\mathbf{h}}_3$
$\overrightarrow{\mathbf{h}}_4$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

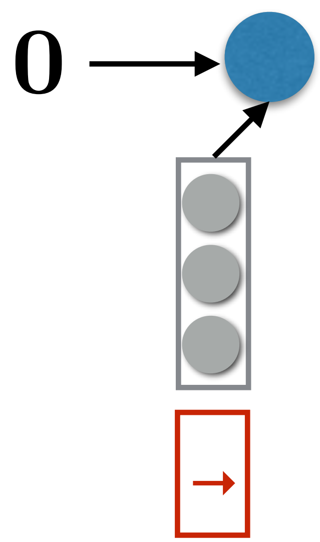
$$\mathbf{f}_i = [\overleftarrow{\mathbf{h}}_i; \overrightarrow{\mathbf{h}}_i]$$

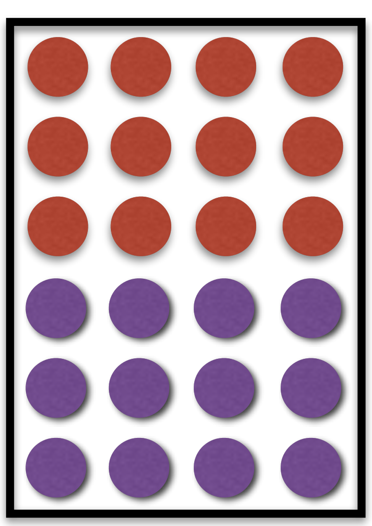


$\mathbf{F} \in \mathbb{R}^{2n \times |\boldsymbol{f}|}$

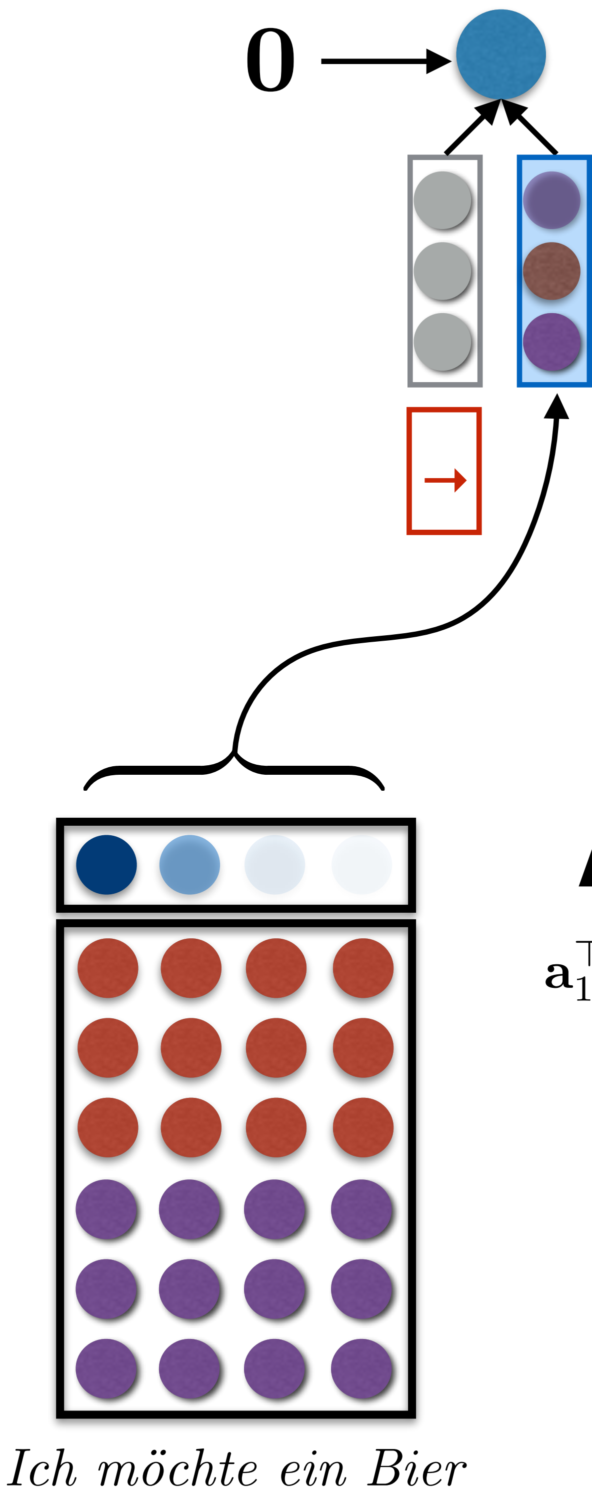


Ich möchte ein Bier

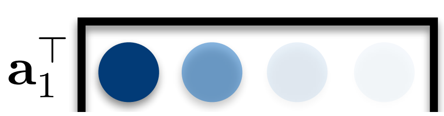


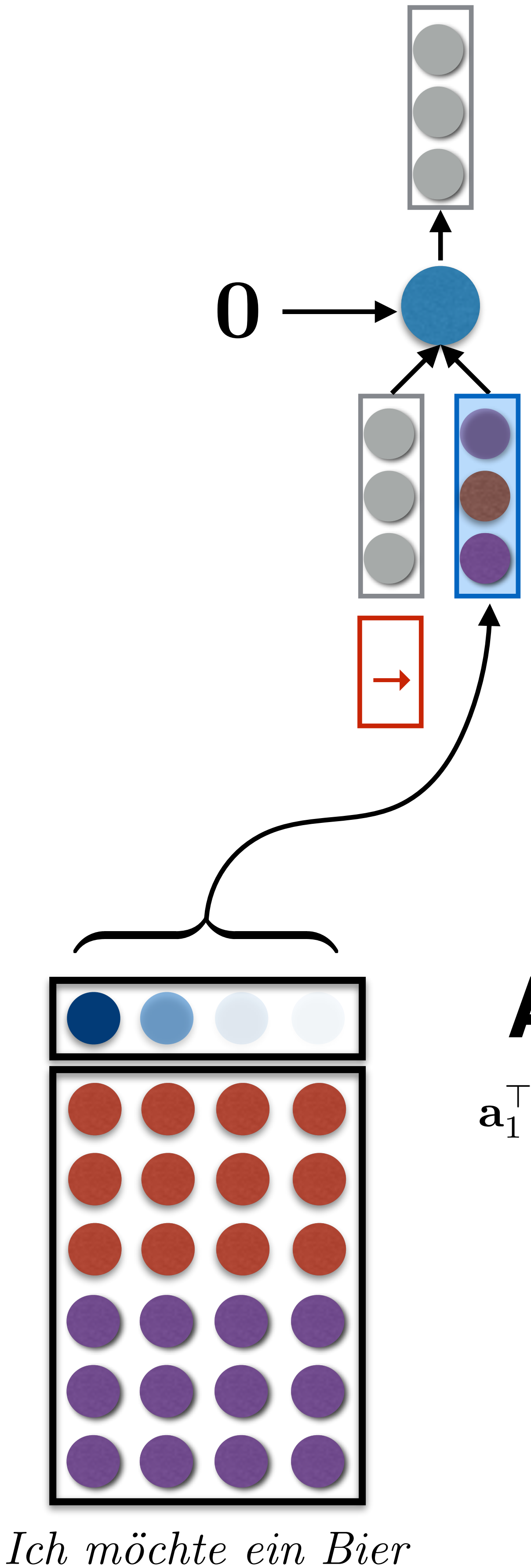


Ich möchte ein Bier

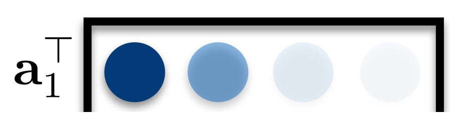


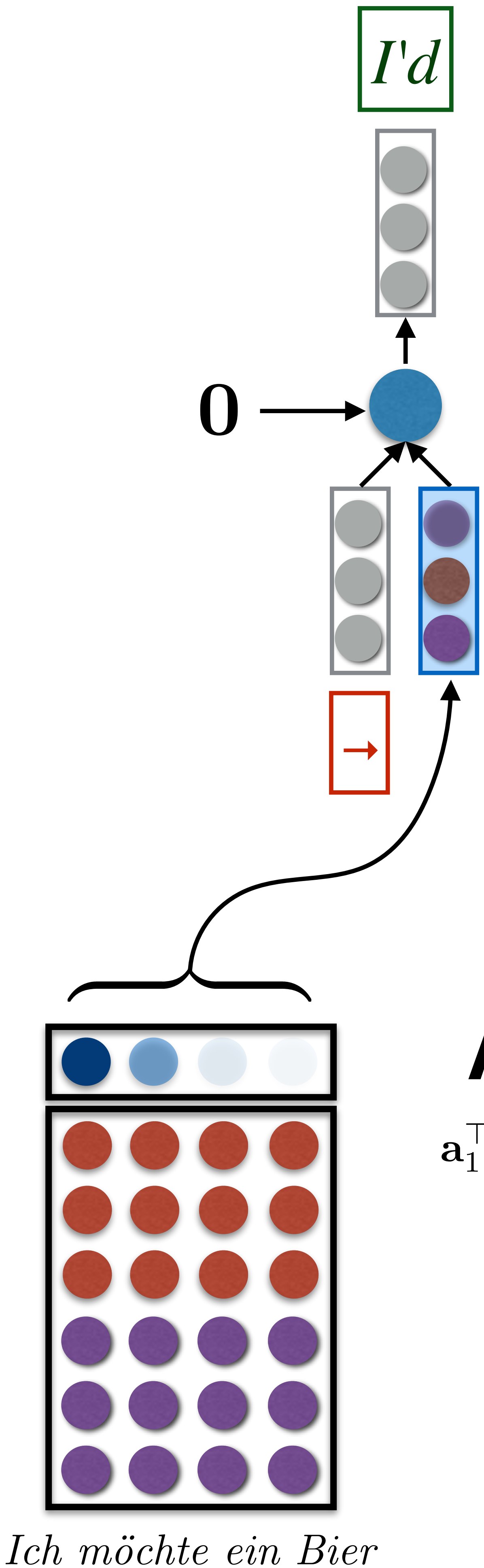
Attention history:



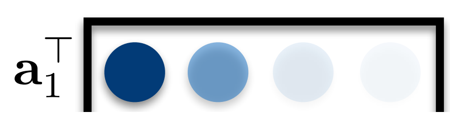


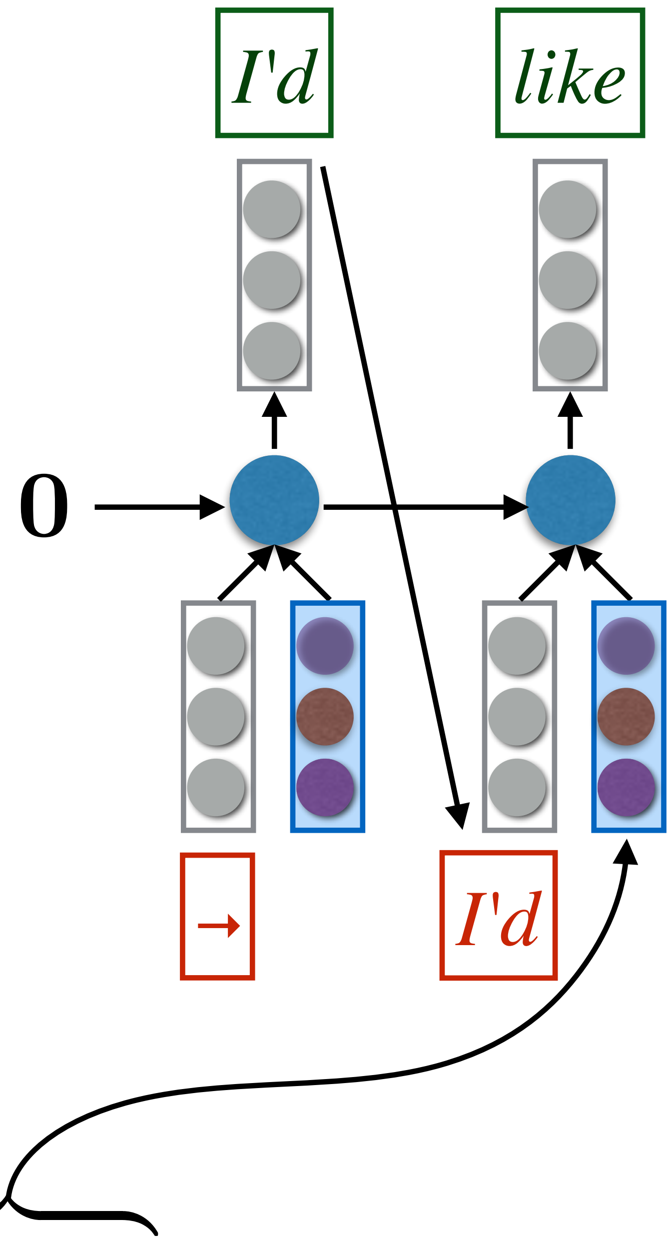
Attention history:

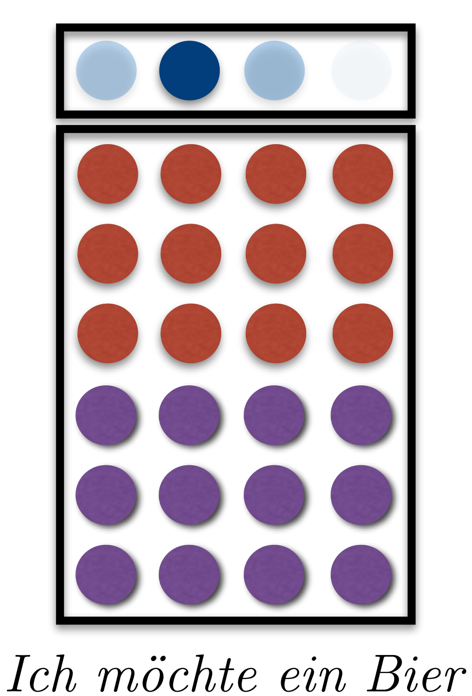




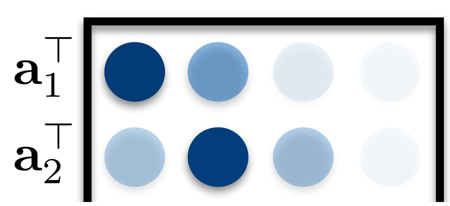
Attention history:

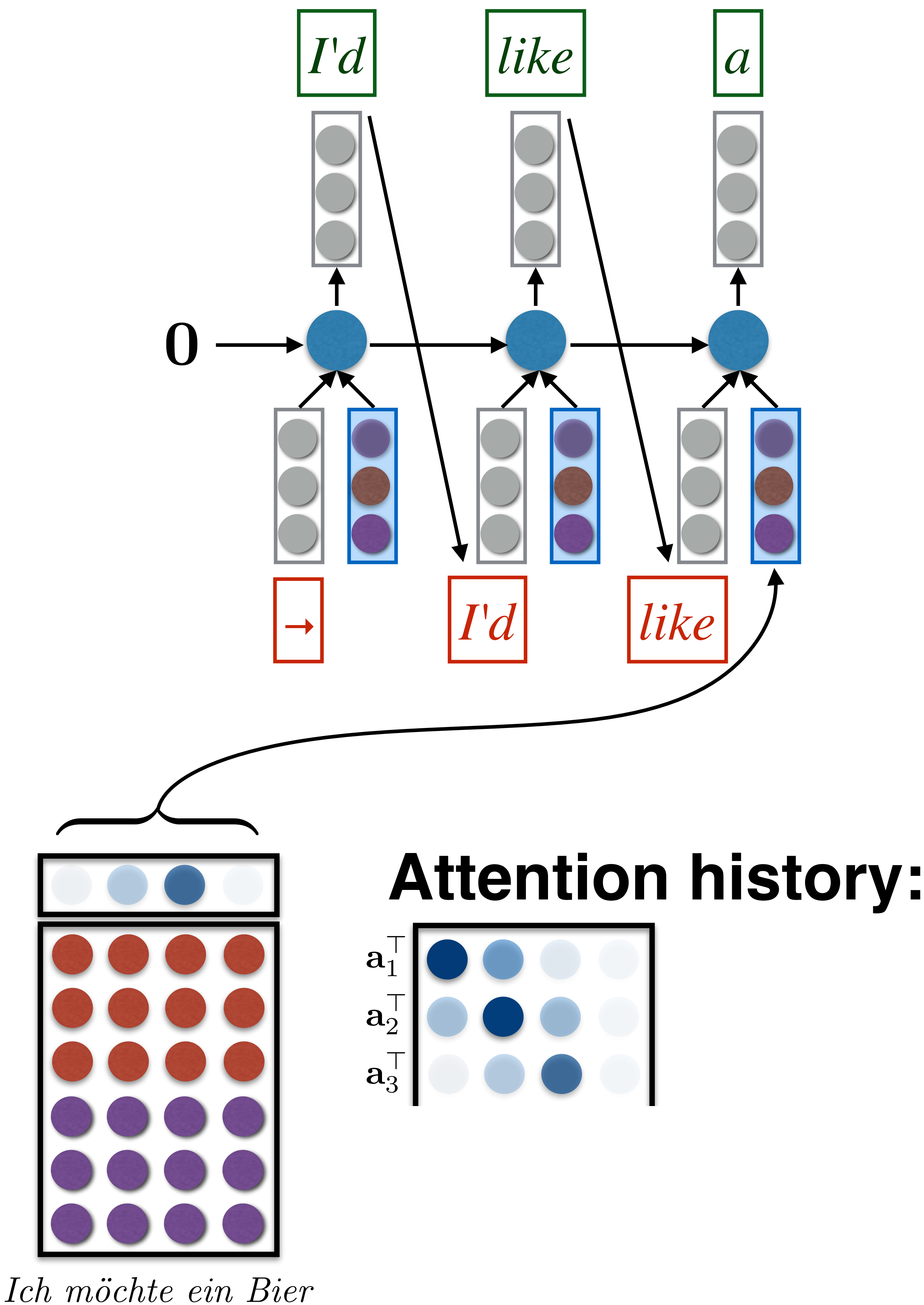


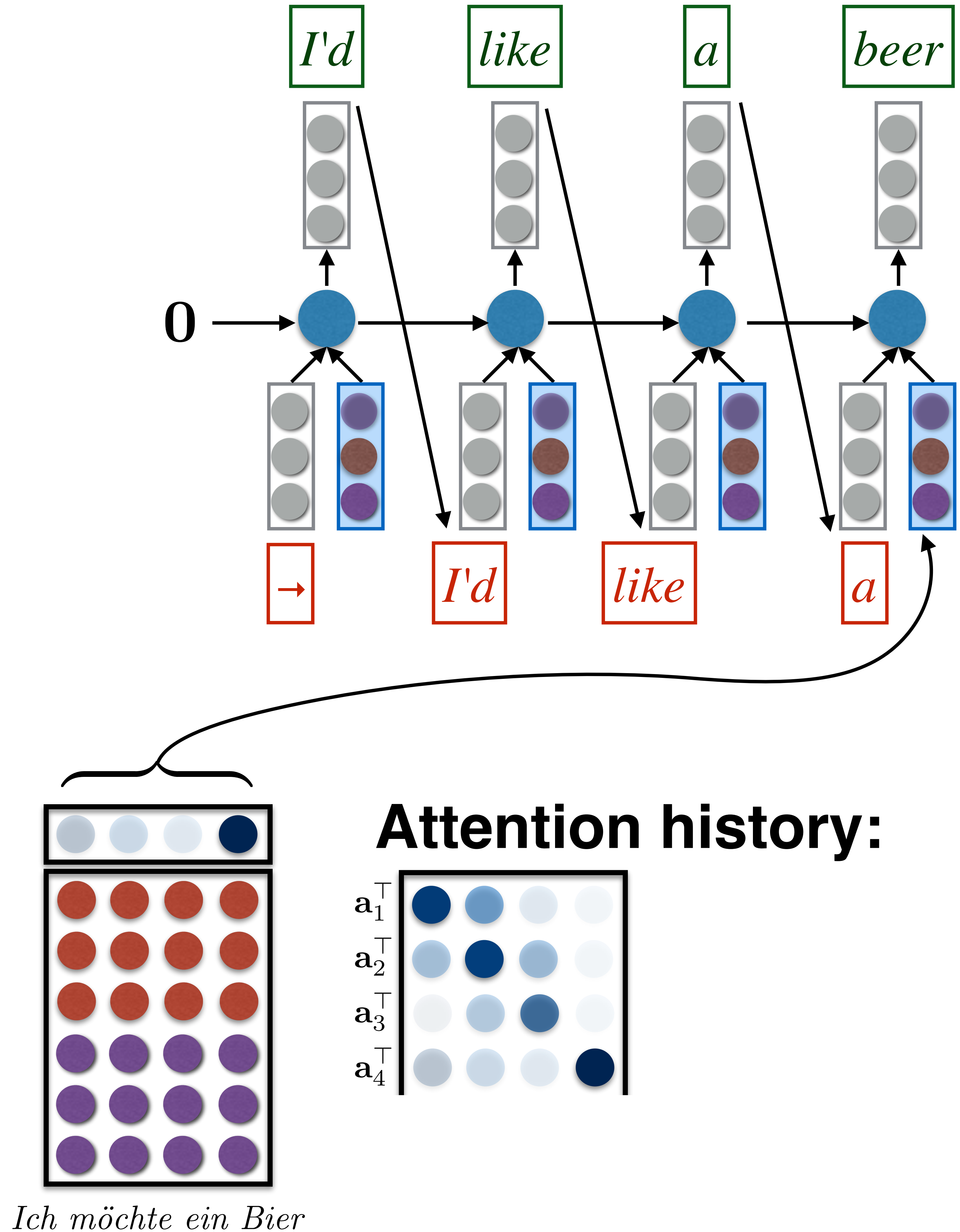


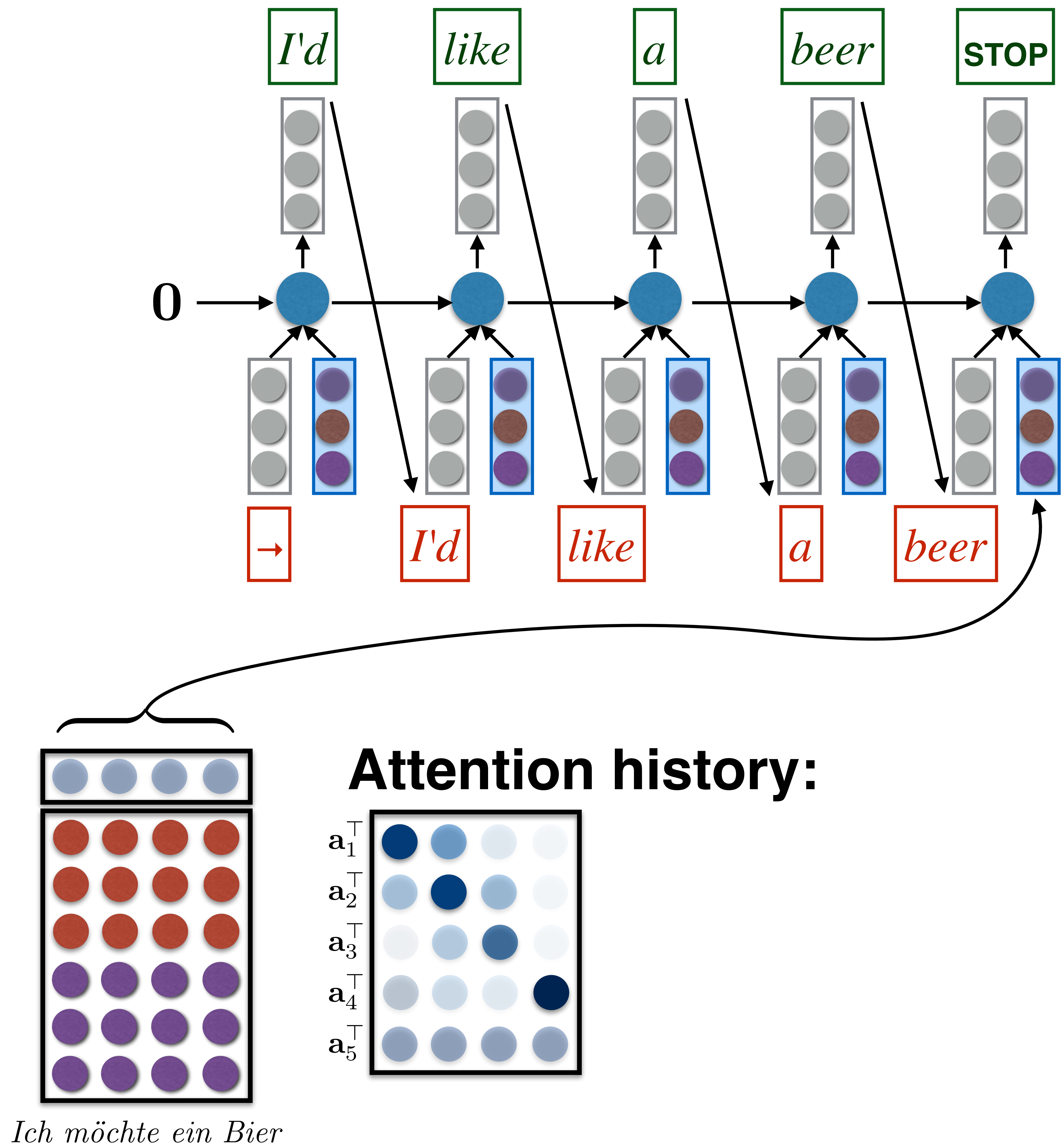


Attention history:

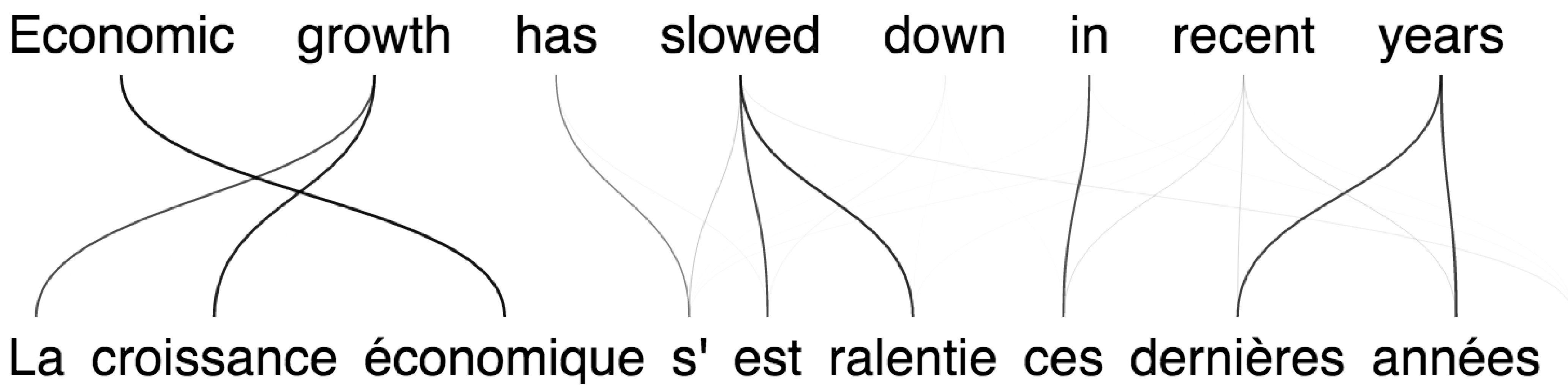



I'd
like
a
0
→
I'd
like
Ich möchte ein Bier
Attention history:
$\mathbf{a}_1^\top$
$\mathbf{a}_2^\top$
$\mathbf{a}_3^\top$

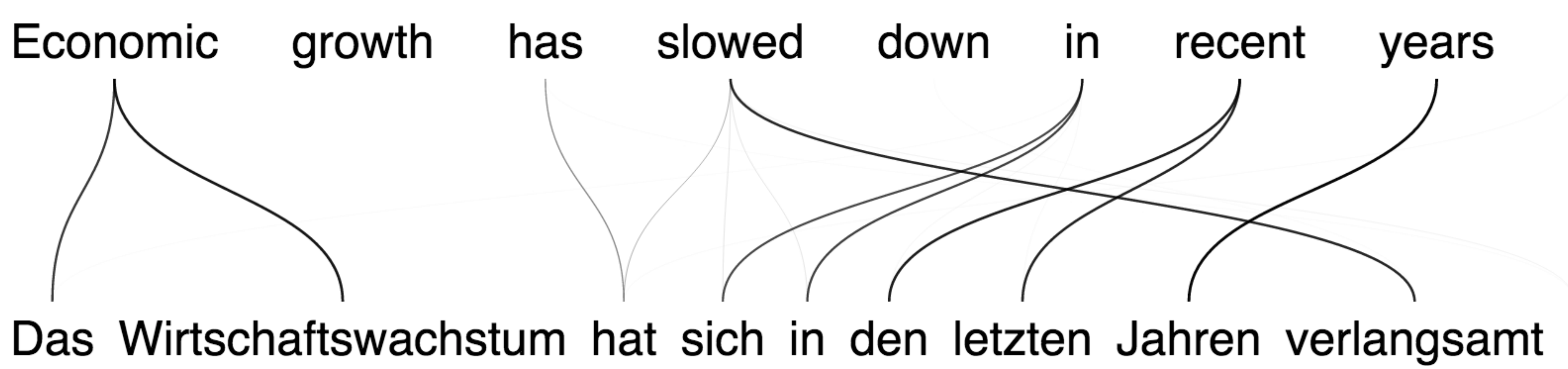

I'd
like
a
beer
0
→
I'd
like
a
Ich möchte ein Bier
Attention history:
$\mathbf{a}_1^\top$
$\mathbf{a}_2^\top$
$\mathbf{a}_3^\top$
$\mathbf{a}_4^\top$


I'd
like
a
beer
STOP
0
→
I'd
like
a
beer
Attention history:
$\mathbf{a}_1^\top$
$\mathbf{a}_2^\top$
$\mathbf{a}_3^\top$
$\mathbf{a}_4^\top$
$\mathbf{a}_5^\top$
Ich möchte ein Bier

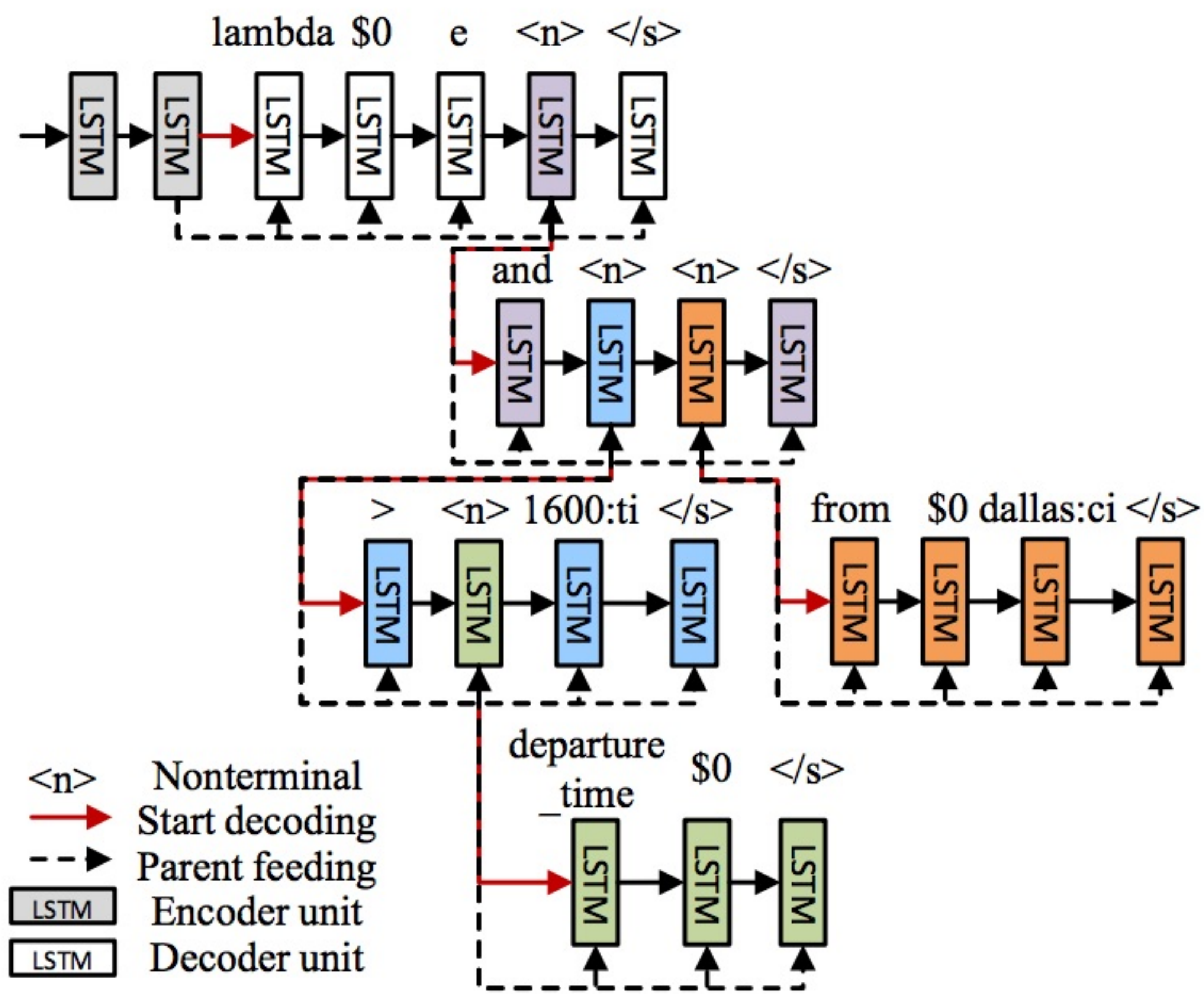
English-French



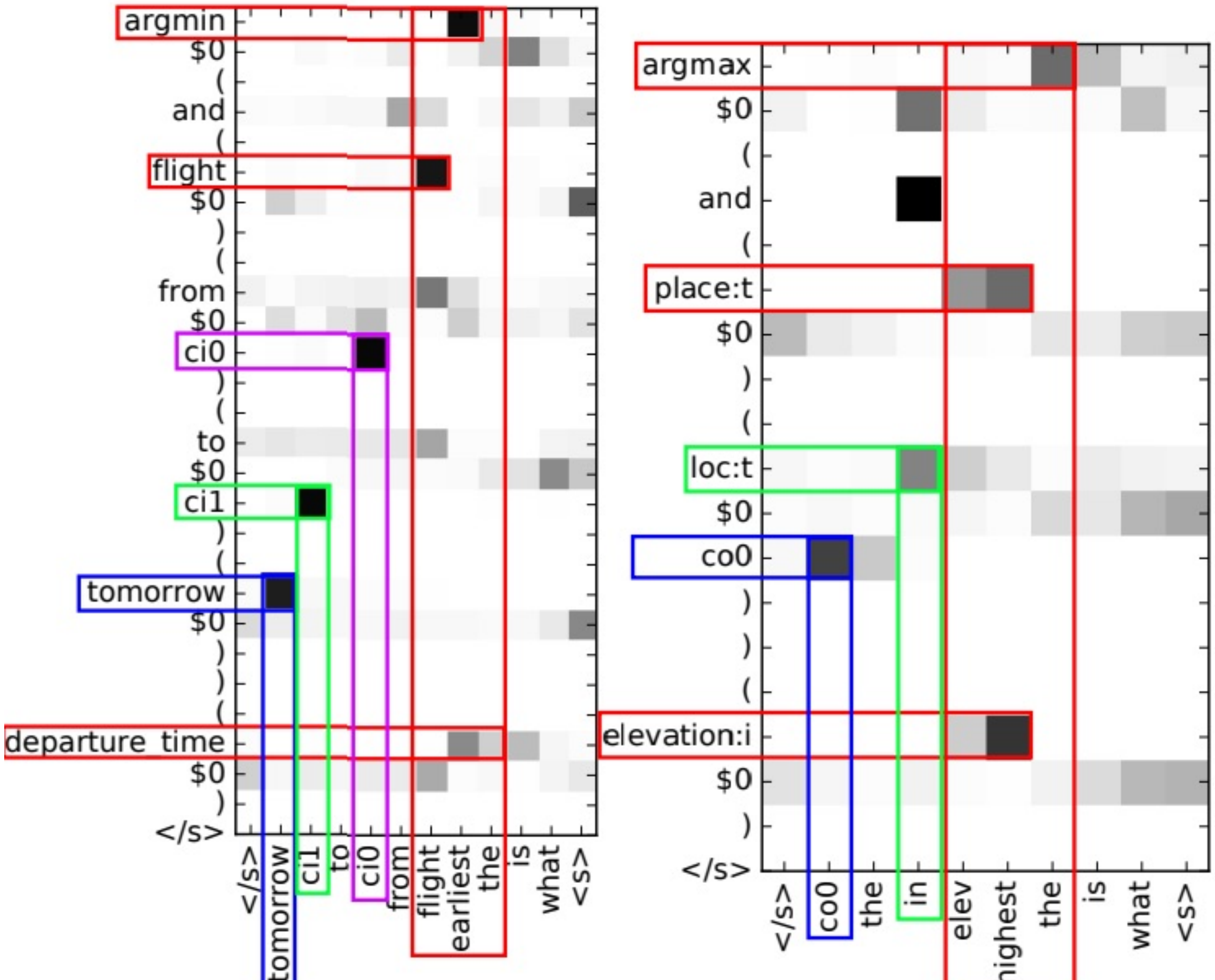
English-German



Since logical forms are tree-like, can use treeLSTM decoder

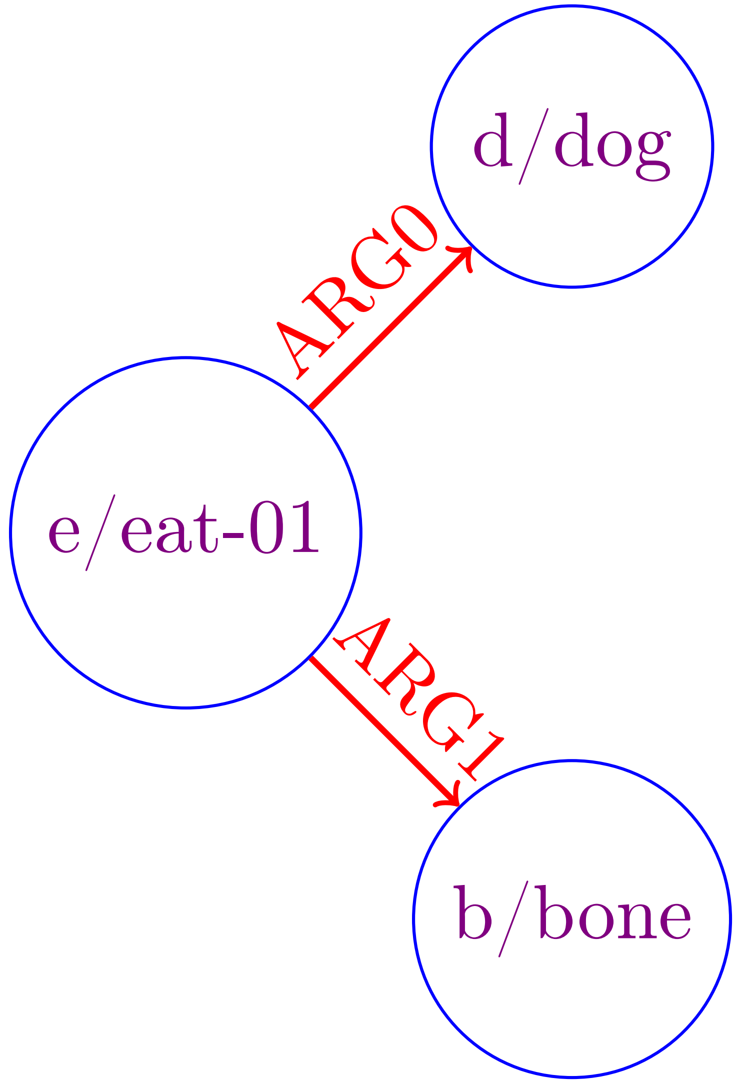


Model learns to “translate” words into predicates they invoke



Abstract meaning representation (AMR)

- The edges (ARG0 and ARG1) are **relations**
- Each node in the graph has a **variable**
- They are labeled with **concepts**
- **d** / **dog** means "**d** is an instance of **dog**"

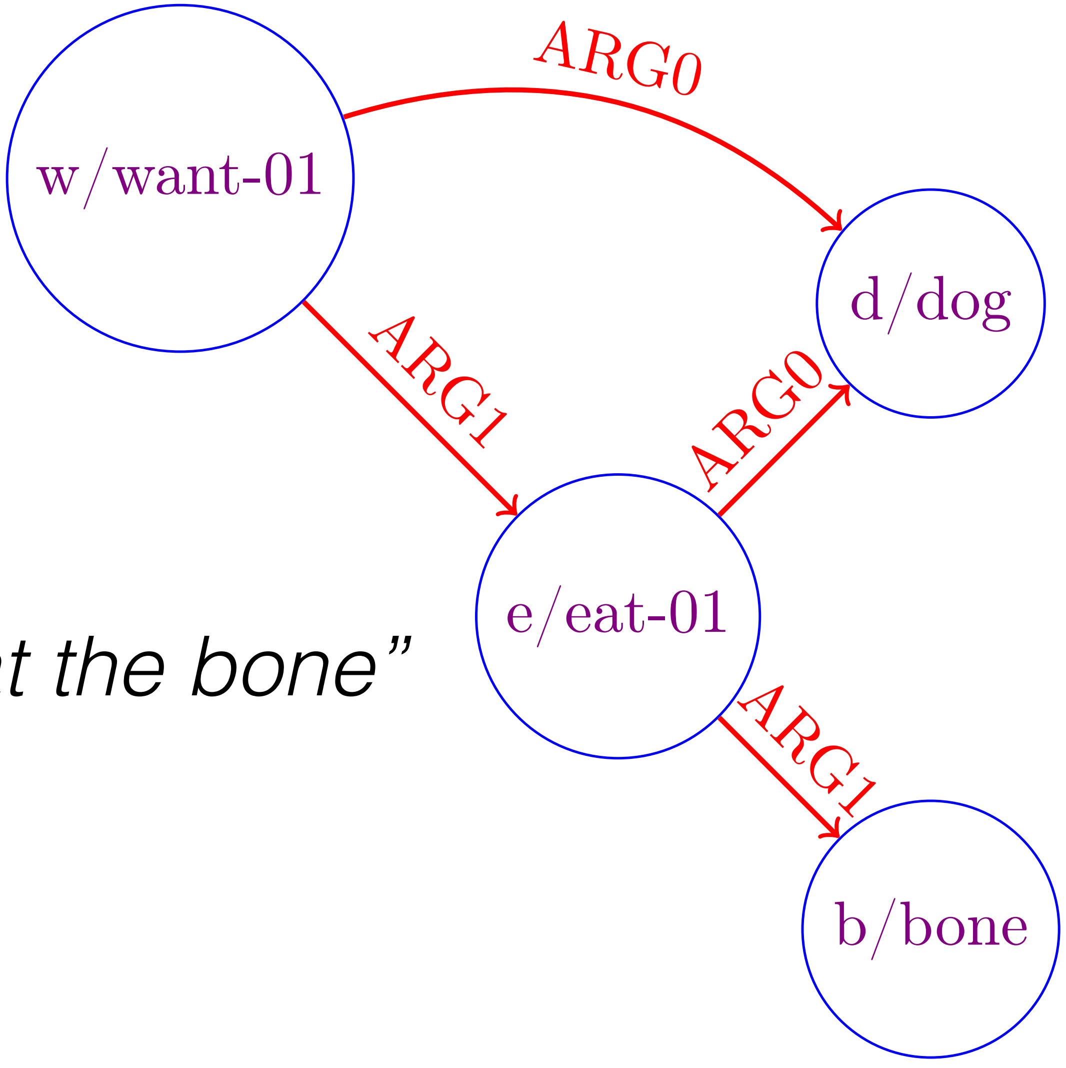


"The dog is eating a bone"

(**e** / **eat-01**
    :**ARG0** (**d** / **dog**)
    :**ARG1** (**b** / **bone**))

Abstract meaning representation (AMR)

- What if something is referenced multiple times?
- Notice how **dog** has two incoming roles now.
- To do this in PENMAN format, repeat the variable. We call this a **reentrancy**.



"The dog ***wants to*** *eat the bone"*

```
(want-01
    :ARG0 (d / dog)
    :ARG1 (e / eat-01
        :ARG0 d
        :ARG1 (b / bone)))
```

Coreference

Bob wants Anna to give **him** a job.

Q: who does **him** refer to?

Coreference

Charles just graduated, and now
Bob wants Anna to give **him** a job.

Q: who does **him** refer to?

Metonymy

Westminster decided to distribute funds throughout England, Wales, Northern Island, and Scotland

Metonymy

Westminster decided to distribute funds throughout England, Wales, Northern Island, and Scotland

decided(Westminster, …)

Metonymy

Westminster decided to distribute funds throughout England, Wales, Northern Island, and Scotland

decided(Westminster, …) ✗

decided(Parliament, …) ✓

Implicature

Implicature

What Rogelio was really thinking:
I would like a piece of that cake.

Even more phenomena…

- Abbreviations (e.g. National Health Service=NHS)
- Nicknames (JLaw=Jennifer Lawrence)
- Metaphor (crime is a virus infecting the city)
- Time expressions and change of state
- Many others

Summary

- In many cases, meaning representation can be captured in first-order logic.
- But wide-coverage meaning representation is hard; closed domains are easier, and can sometimes be harvested automatically.
- This leads to a proliferation of domain-specific MRs.
- Trainable alternative to compositional approaches: encoder-decoder neural models.
- The encoder and decoder can be mixed and matched: RNN, top-down tree RNN, etc.
- Works well on small, closed domains *if we have training data*, but there are many unsolved phenomena/ problems in semantics.