Large vocabulary language modeling

Essential epistemology

	Exact sciences
Deals with	Axioms & theorems
Truth is	Forever
Examples	Mathematics C.S. theory **F.L. theory**

Essential epistemology

	Exact sciences	Empirical sciences
Deals with	Axioms & theorems	Facts & theories
Truth is	Forever	Temporary
Examples	Mathematics C.S. theory **F.L. theory**	Physics Biology **Linguistics**

Essential epistemology

	Exact sciences	Empirical sciences	Engineering
Deals with	Axioms & theorems	Facts & theories	Artifacts
Truth is	Forever	Temporary	It works
Examples	Mathematics C.S. theory **F.L. theory**	Physics Biology **Linguistics**	Many, including applied C.S. e.g. **NLP**

Essential epistemology

Exact sciences	Empirical sciences	Engineering
	morphological properties of words (facts)	

Essential epistemology

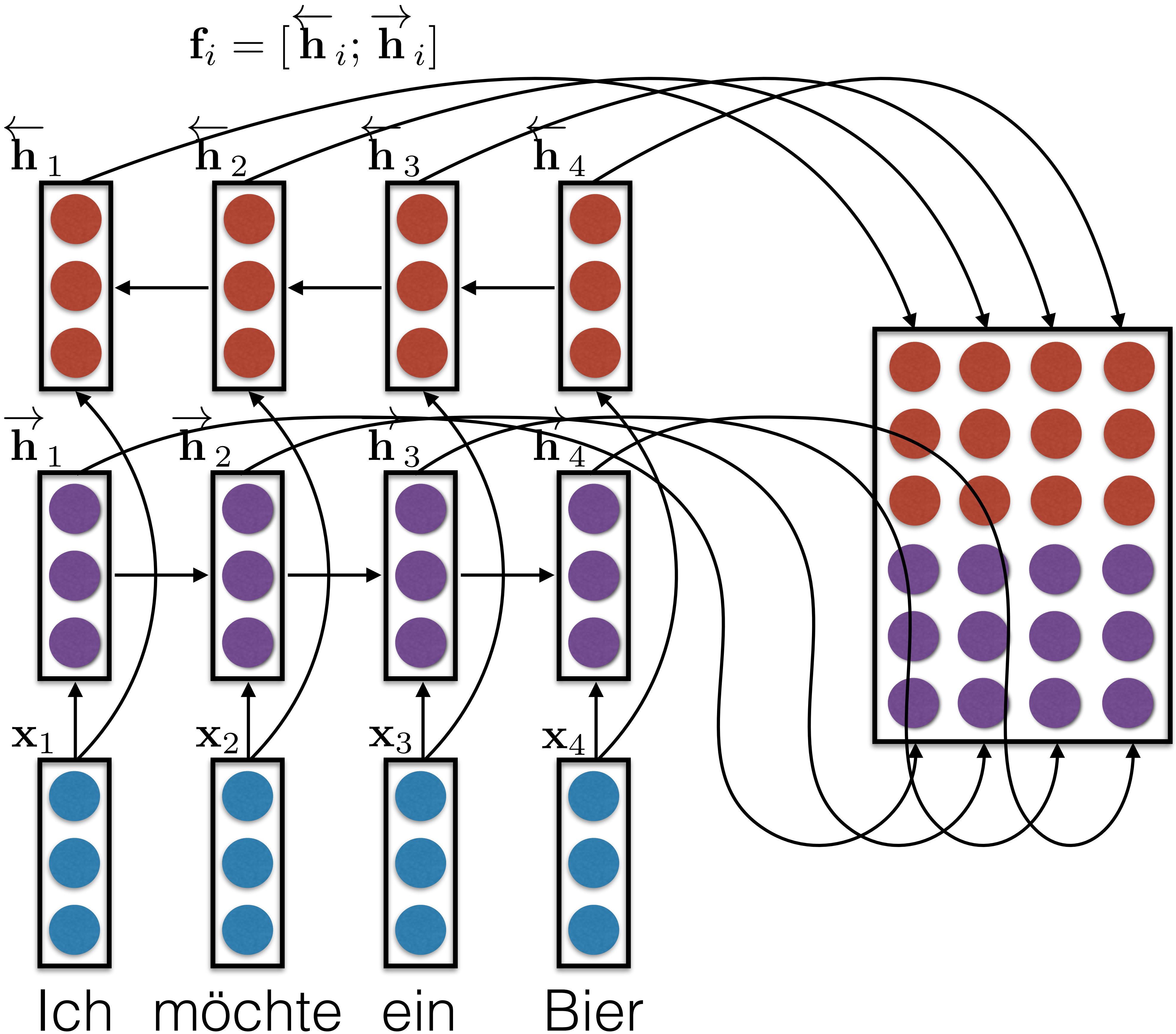
Exact sciences	Empirical sciences	Engineering
	morphological properties of words (facts) optimality theory	

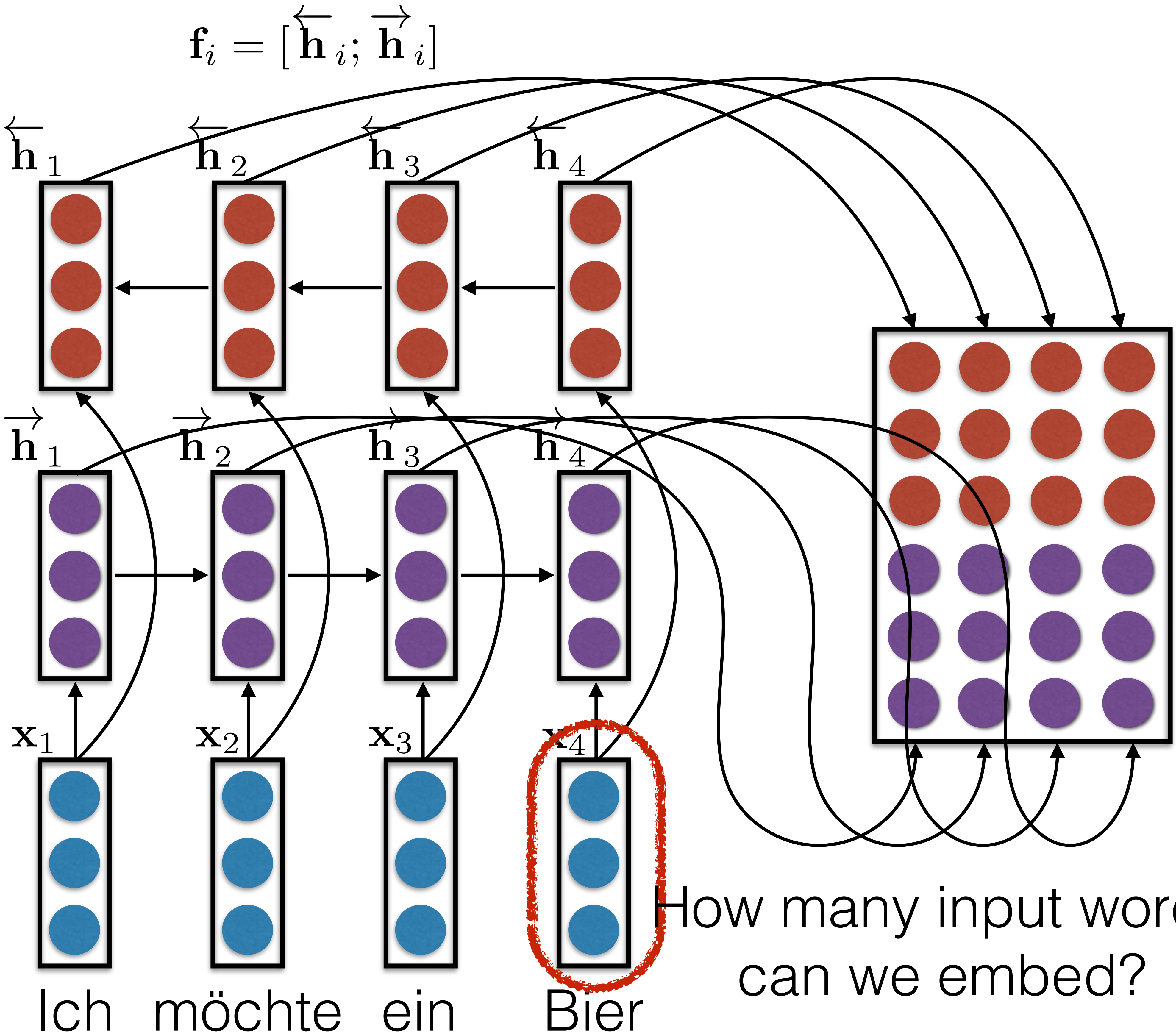
Essential epistemology

Exact sciences	Empirical sciences	Engineering
optimality theory is finite-state	morphological properties of words (facts)	
	optimality theory	

Essential epistemology

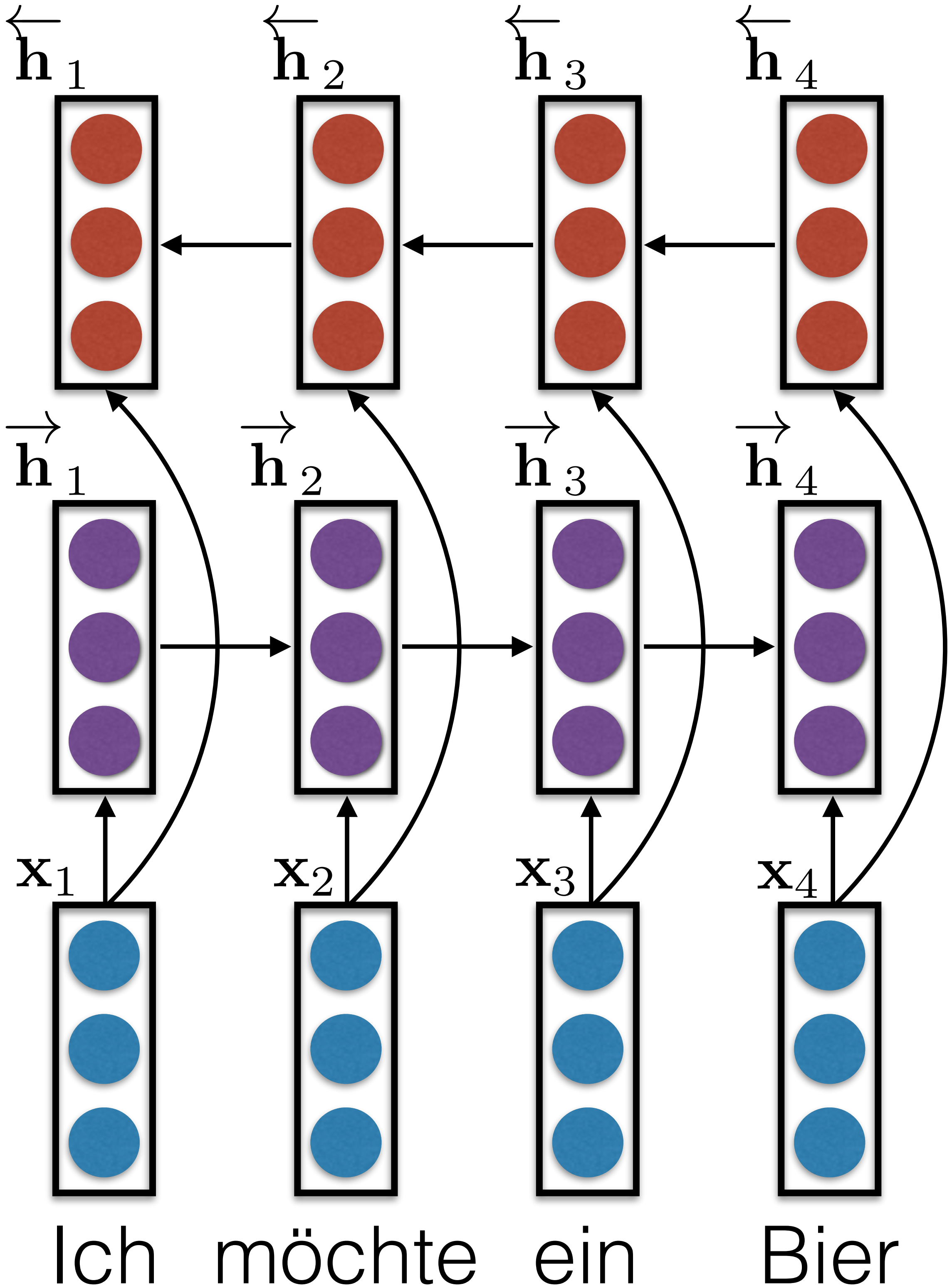
Exact sciences	Empirical sciences	Engineering
optimality theory is finite-state	morphological properties of words (facts)	we can represent words using finite-state machines (or *continuous*-state machines, i.e. NNs)
	optimality theory	


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

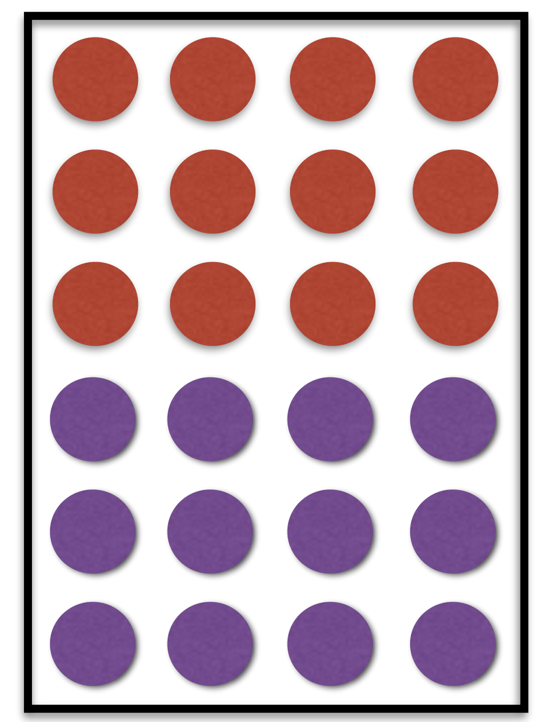


How many input words can we embed?

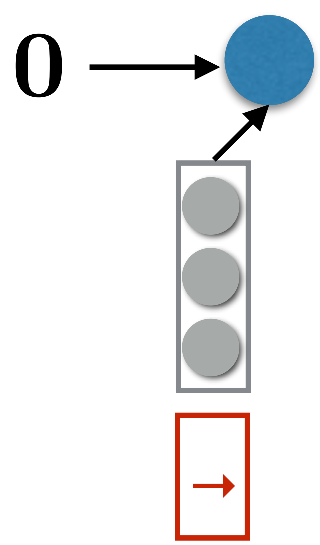
$$\mathbf{f}_i = [\overleftarrow{\mathbf{h}}_i; \overrightarrow{\mathbf{h}}_i]$$

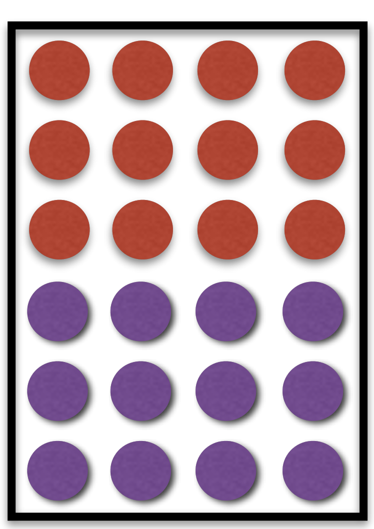


$\mathbf{F} \in \mathbb{R}^{2n \times |\boldsymbol{f}|}$

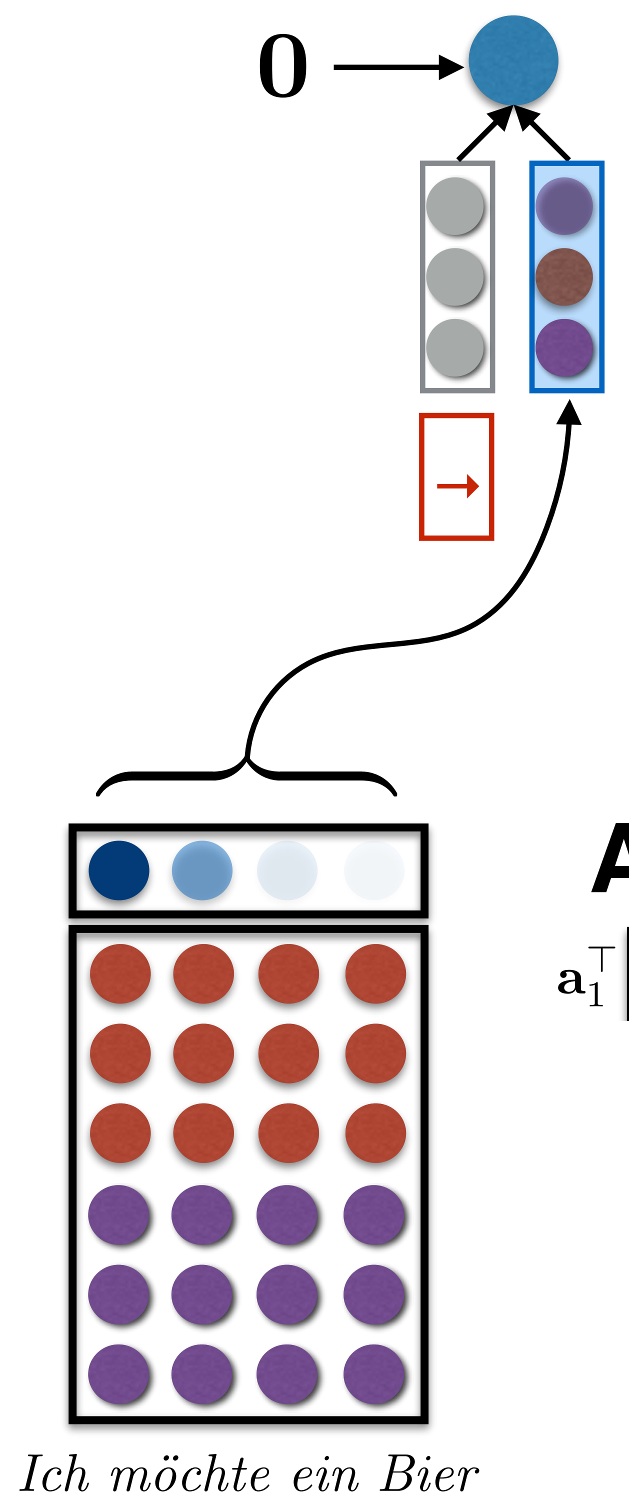


Ich möchte ein Bier

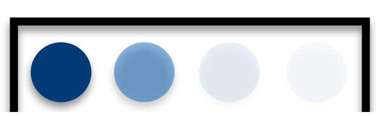


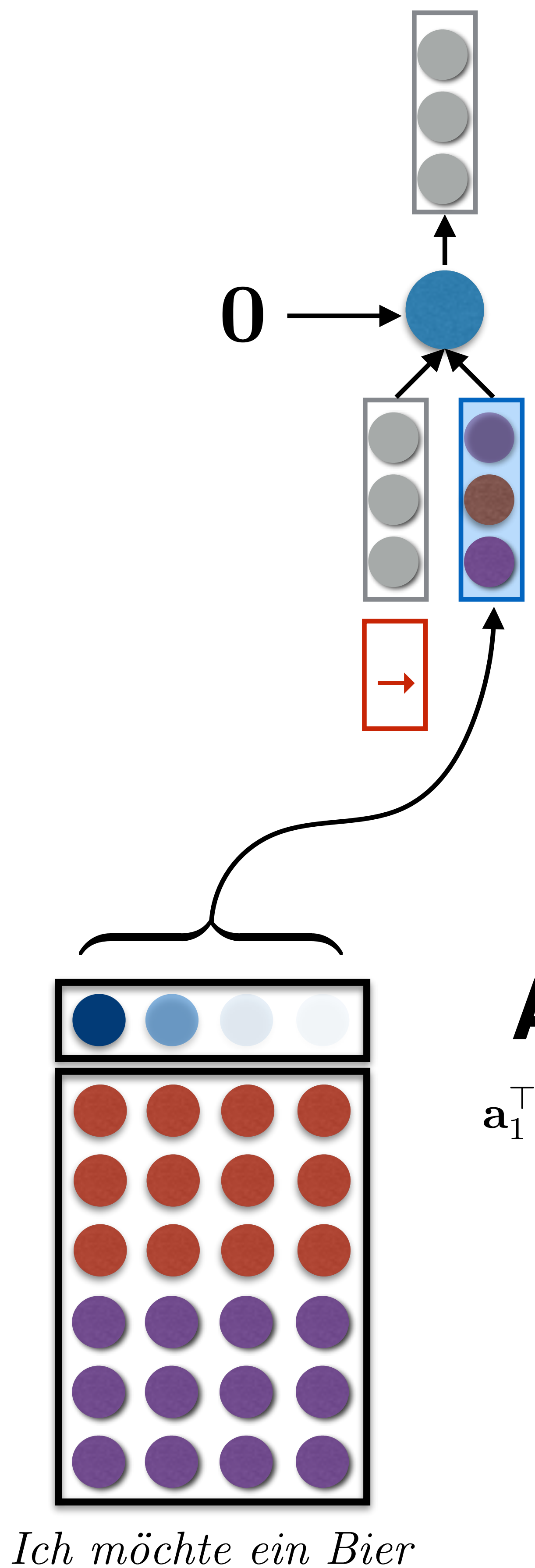


Ich möchte ein Bier

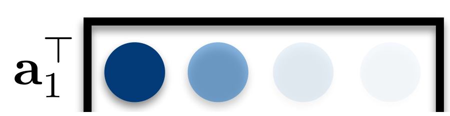


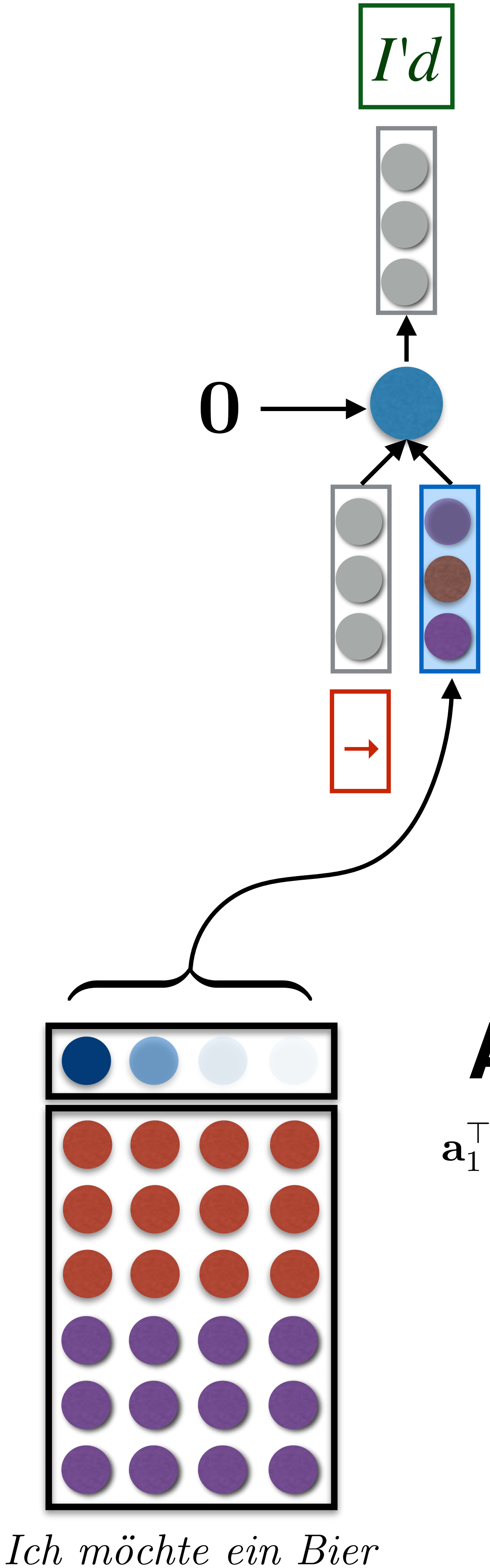
Attention history:

$\mathbf{a}_1^\top$ 

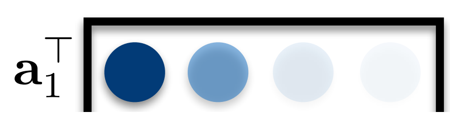


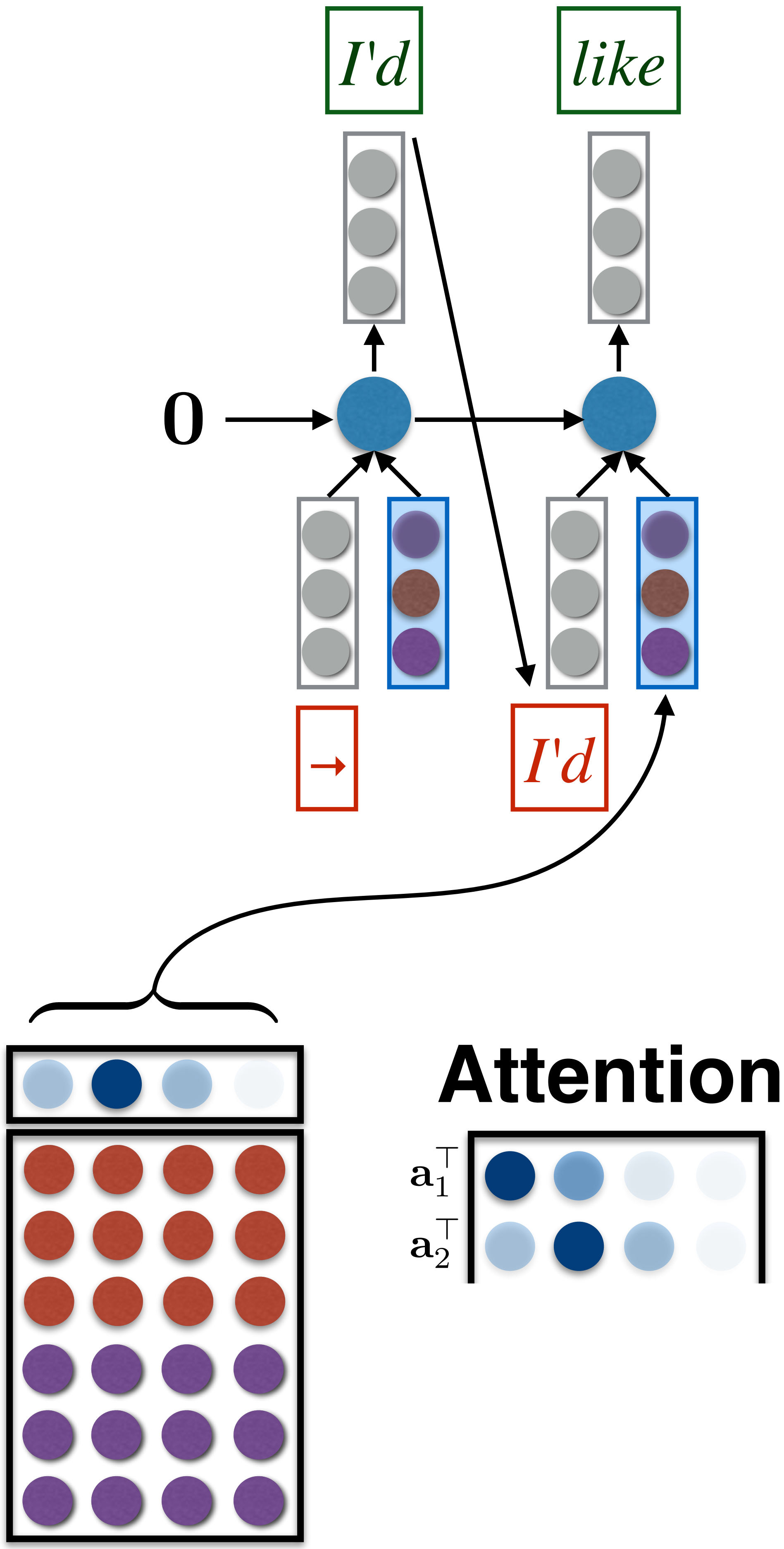
Attention history:





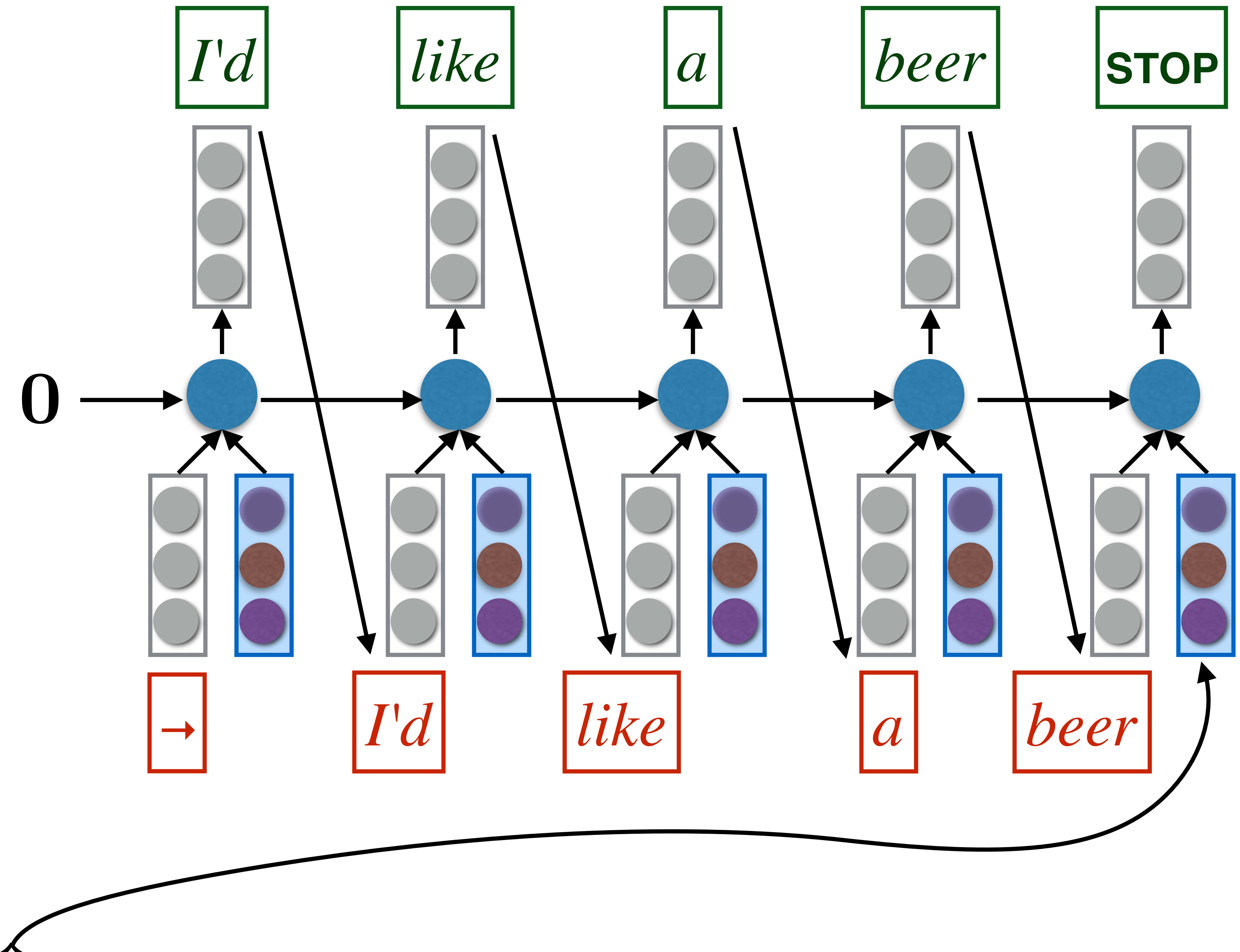
Attention history:

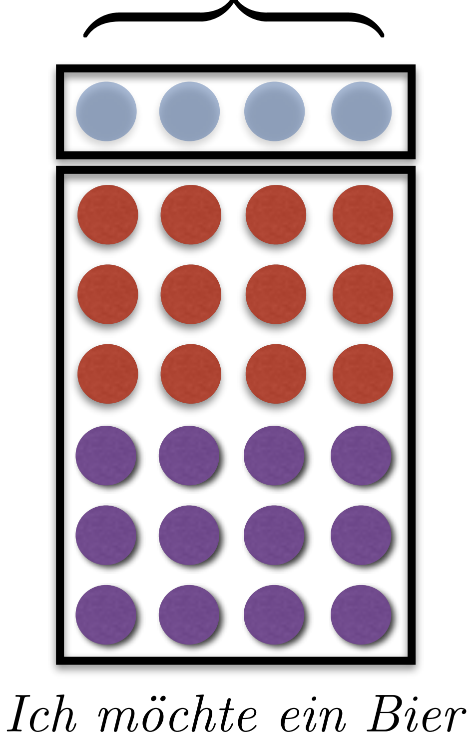




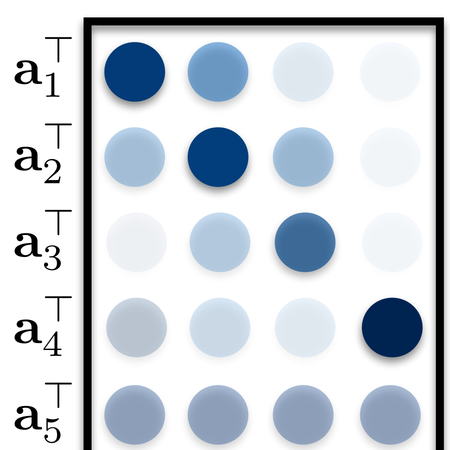
Attention history:

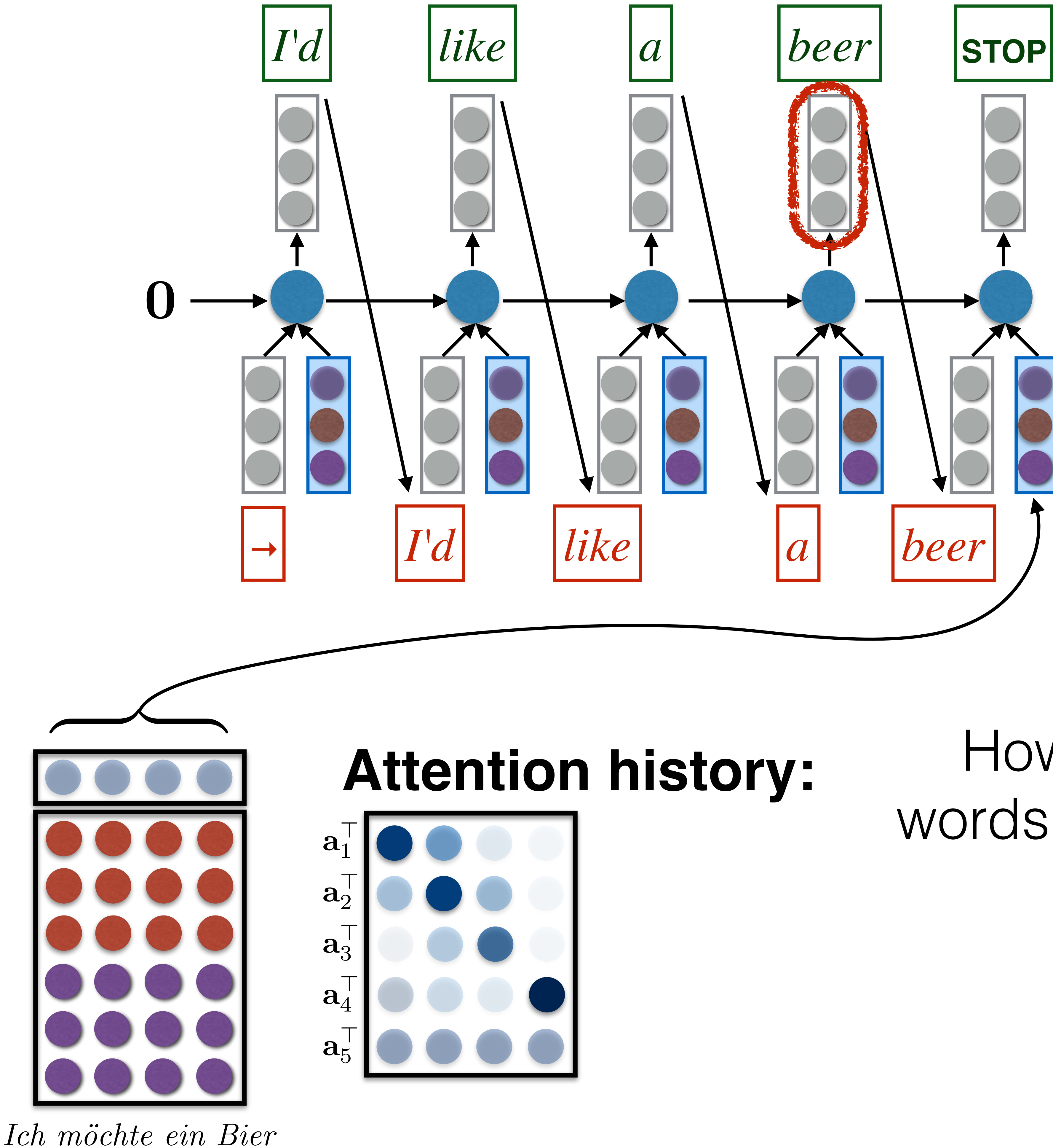
Ich möchte ein Bier





Attention history:



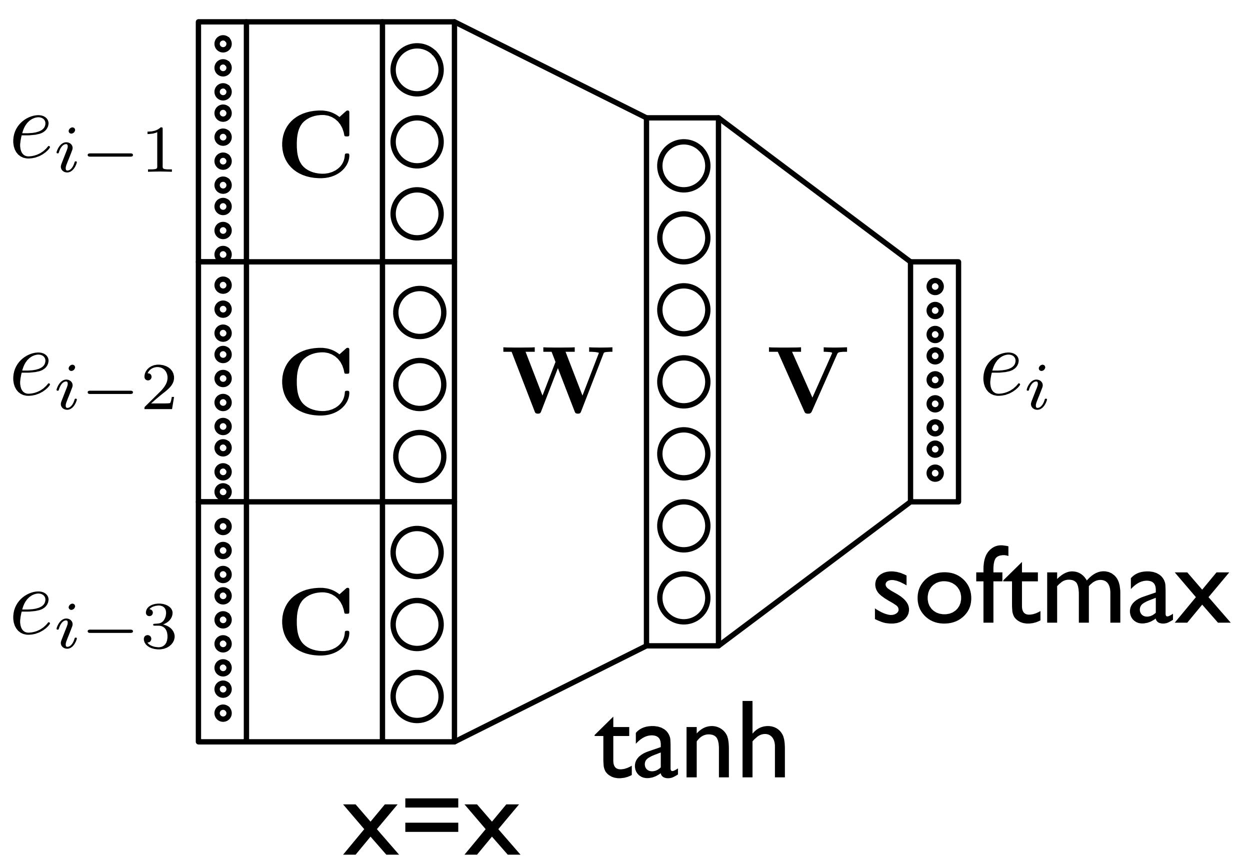


How many output words can we predict?

Bengio et al. (2003)

$$p(\mathbf{e}) = \prod_{i=1}^{|\mathbf{e}|} p(e_i \mid e_{i-n+1}, \ldots, e_{i-1})$$

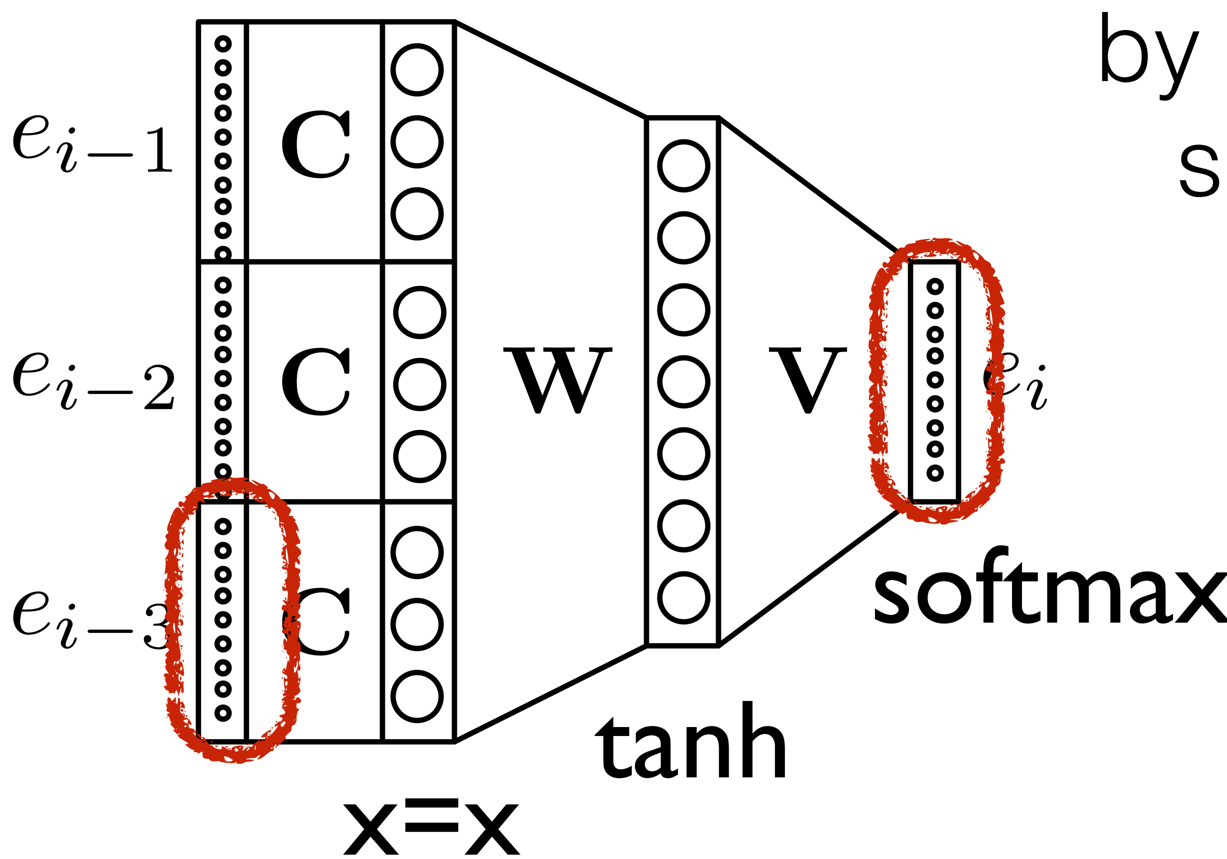
$$p(e_i \mid e_{i-n+1}, \ldots, e_{i-1}) =$$



Bengio et al. (2003)

$$p(\mathbf{e}) = \prod_{i=1}^{|\mathbf{e}|} p(e_i \mid e_{i-n+1}, \ldots, e_{i-1})$$

$$p(e_i \mid e_{i-n+1}, \ldots, e_{i-1}) =$$



Every model inspired by this one has the same problem!

Actually, this problem is even older…

Language modeling

For language modeling, we seek

$$p(\mathbf{e}) > 0 \quad \forall \mathbf{e} \in \Sigma^*$$

We will assume that Σ is known and finite.

Actually, this problem is even older…

Language modeling

For language modeling, we seek

$$p(\mathbf{e}) > 0 \quad \forall \mathbf{e} \in \Sigma^*$$

We will assume that Σ is known and finite.

When does this assumption make sense for language modeling?

Known and finite

- Practical problem: softmax computation is linear in vocabulary size.

Problems with this?

- Bengio et al.: “Rare words with frequency ≤ 3 were merged into a single symbol, reducing the vocabulary size to $|V| = 16{,}383$.”

- Bahdanau et al.: “we use a shortlist of 30,000 most frequent words in each language to train our models. Any word not included in the shortlist is mapped to a special token ([UNK]).”

Problems with this?

- Bengio et al.: “Rare words with frequency ≤ 3 were merged into a single symbol, reducing the vocabulary size to |V| = 16,383.”

- Bahdanau et al.: “we use a shortlist of 30,000 most frequent words in each language to train our models. Any word not included in the shortlist is mapped to a special token ([UNK]).”

```
-----------------------------------------------
Src | 日本 の 主要 作物 は 米 で あ る 。
Ref | the main crop of japan is rice .
Hyp | the _UNK is popular of _UNK . _EOS
-----------------------------------------------
```

Approaches

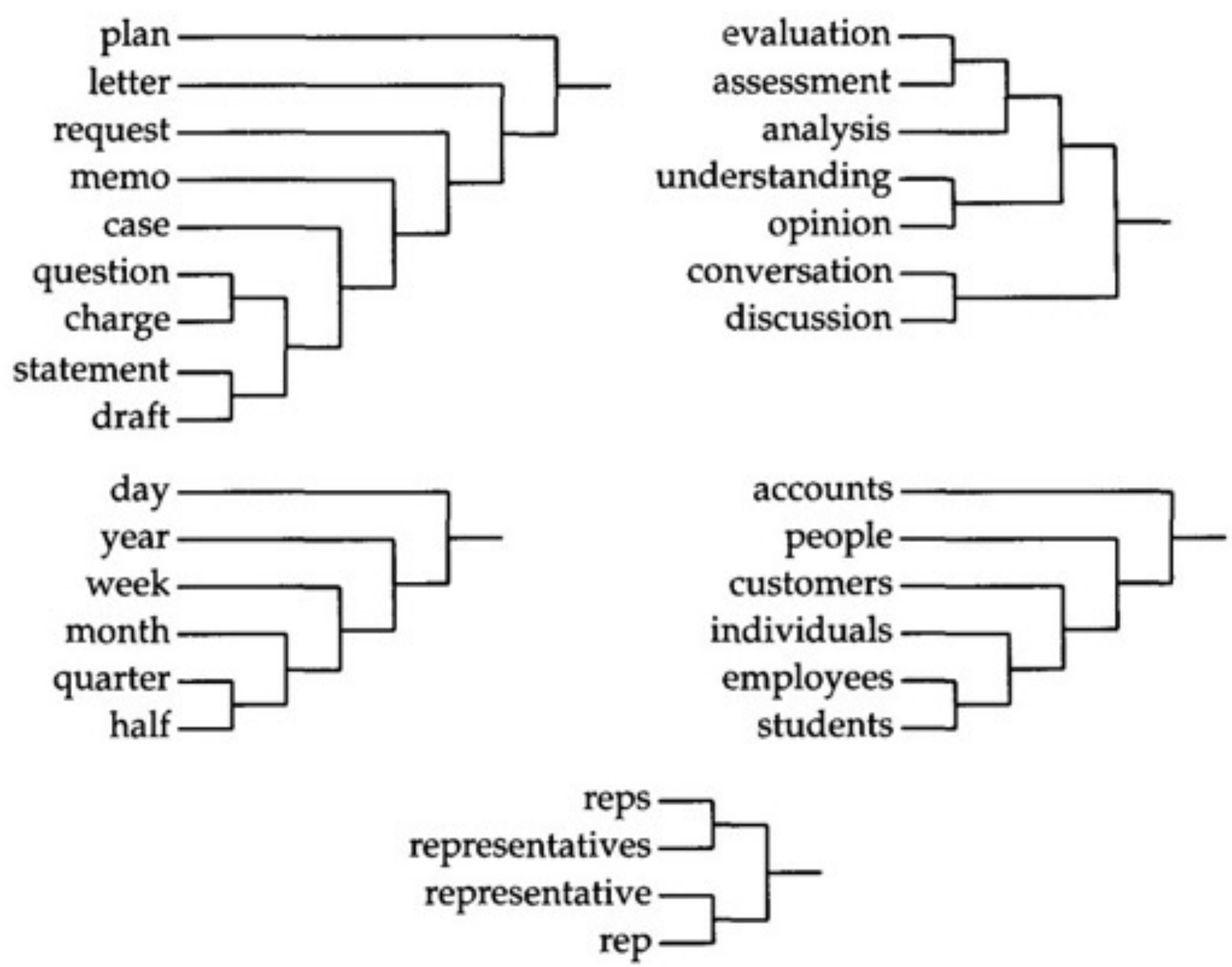
- Partition the vocabulary into smaller pieces.

$$p(w_i|h_i) = p(c_i|h_i)p(w_i|c_i, h_i)$$

Class-based LM

Approaches

- Partition the vocabulary into smaller pieces hierarchically (*hierarchical* softmax).



Brown clustering: hard clustering based on mutual information

Approaches

- Differentiated softmax: assign more parameters to more frequent words, fewer to less frequent words.

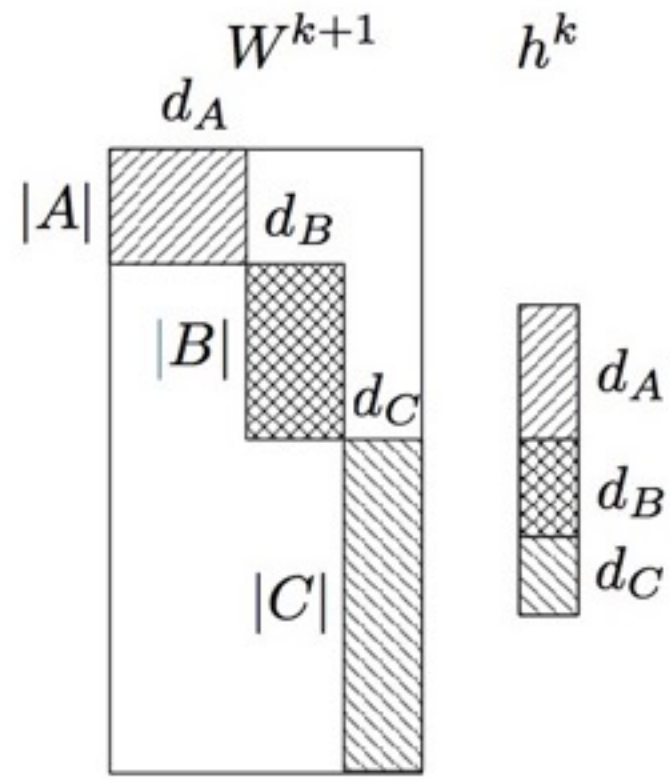
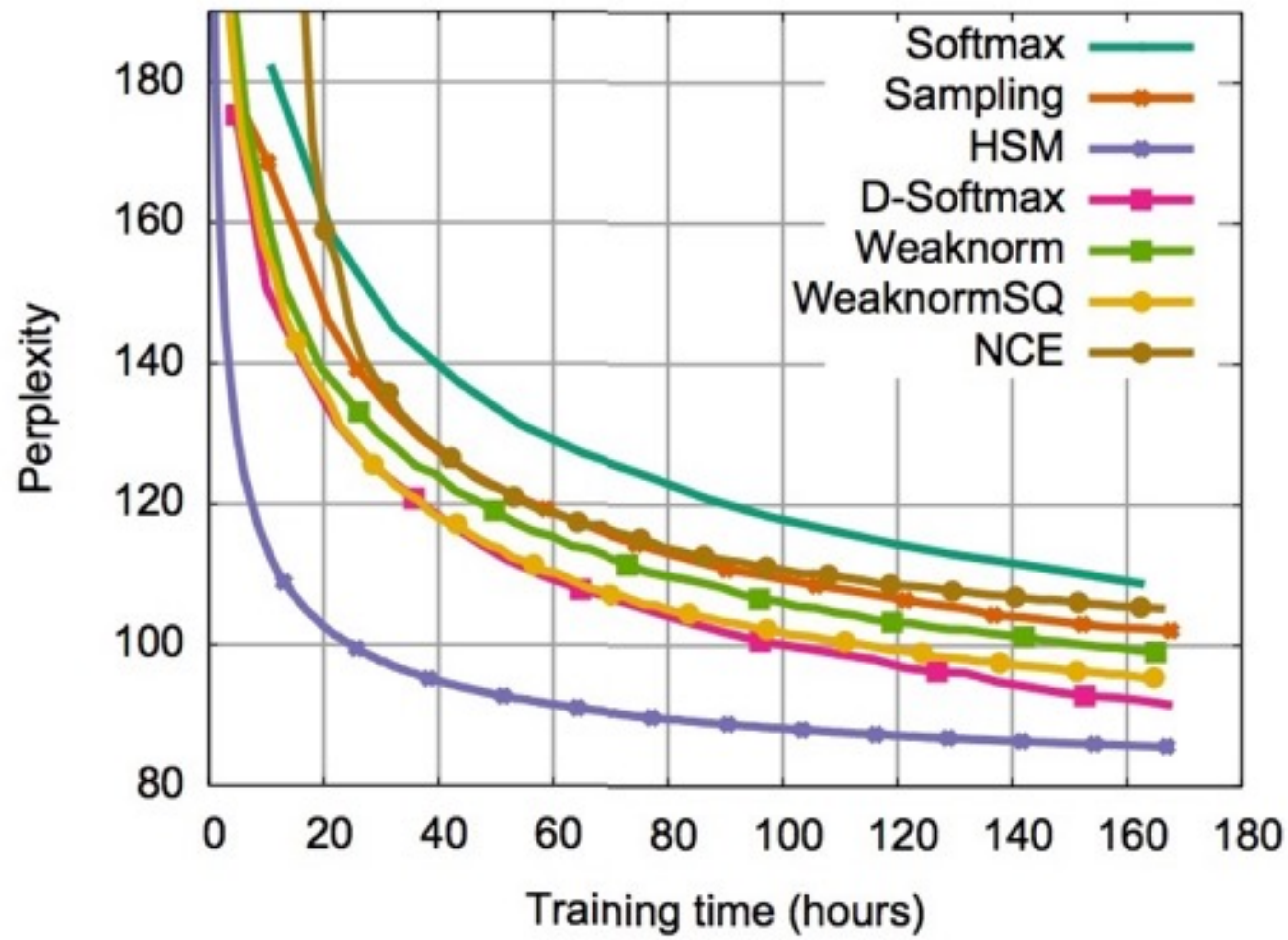


Figure 1: Final weight matrix W^{k+1} and hidden layer h^k for differentiated softmax for partitions A, B, C of the output vocabulary with embedding dimensions d_A, d_B, d_C; non-shaded areas are zero.

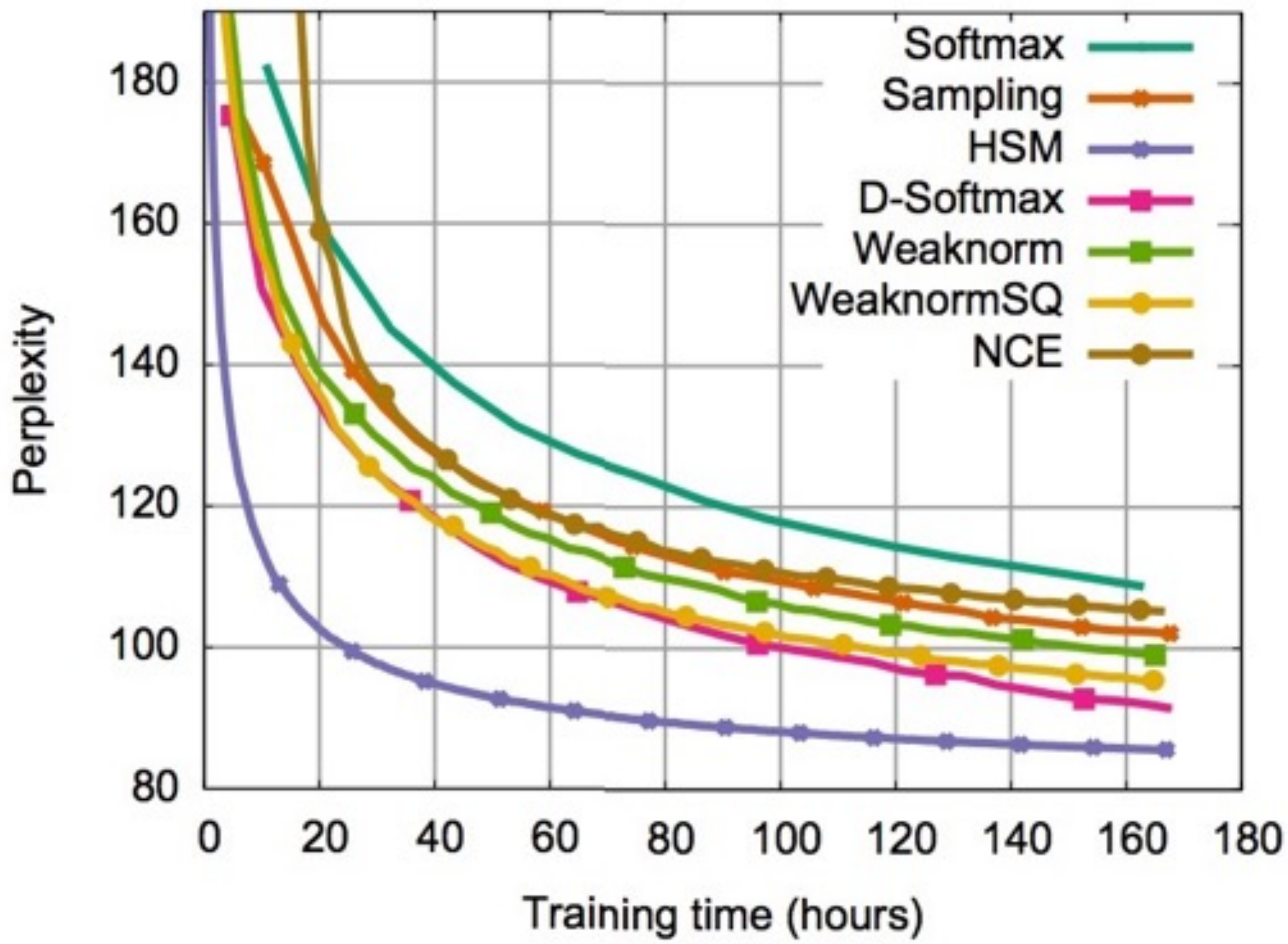
Comparisons



Dataset	Train	Test	Vocab	OOV
PTB	1M	0.08M	10k	5.8%
gigaword	4,631M	279M	100k	5.6%
billionW	799M	8.1M	793k	0.3%

Table 1: Dataset statistics. Number of tokens for train and test set, vocabulary size and ratio of out-of-vocabulary words in the test set.

Comparisons



Dataset	Train	Test	Vocab	OOV
PTB	1M	0.08M	10k	5.8%
gigaword	4,631M	279M	100k	5.6%
billionW	799M	8.1M	793k	0.3%

Table 1: Dataset statistics. Number of tokens for train and test set, vocabulary size and ratio of out-of-vocabulary words in the test set.

Comparisons

	PTB	gigaword	billionW
KN	141.2	57.1	70.2
Softmax	123.8	56.5	108.3
D-Softmax	121.1	52.0	91.2
Sampling	124.2	57.6	101.0
HSM	138.2	57.1	85.2
NCE	143.1	78.4	104.7
Weaknorm	124.4	56.9	98.7
WeaknormSQ	122.1	56.1	94.9
KN+Softmax	108.5	43.6	59.4
KN+D-Softmax	107.0	42.0	56.3
KN+Sampling	109.4	43.8	58.1
KN+HSM	115.0	43.9	55.6
KN+NCE	114.6	49.0	58.8
KN+Weaknorm	109.2	43.8	58.1
KN+WeaknormSQ	108.8	43.8	57.7

Table 2: Test perplexity of individual models and interpolation with Kneser-Ney.

Comparisons

Noise contrastive estimation

	PTB	gigaword	billionW
KN	141.2	57.1	70.2
Softmax	123.8	56.5	108.3
D-Softmax	121.1	52.0	91.2
Sampling	124.2	57.6	101.0
HSM	138.2	57.1	85.2
NCE	143.1	78.4	104.7
Weaknorm	124.4	56.9	98.7
WeaknormSQ	122.1	56.1	94.9
KN+Softmax	108.5	43.6	59.4
KN+D-Softmax	107.0	42.0	56.3
KN+Sampling	109.4	43.8	58.1
KN+HSM	115.0	43.9	55.6
KN+NCE	114.6	49.0	58.8
KN+Weaknorm	109.2	43.8	58.1
KN+WeaknormSQ	108.8	43.8	57.7

Table 2: Test perplexity of individual models and interpolation with Kneser-Ney.

Comparisons

Skip normalization step altogether

	PTB	gigaword	billionW
KN	141.2	57.1	70.2
Softmax	123.8	56.5	108.3
D-Softmax	121.1	52.0	91.2
Sampling	124.2	57.6	101.0
HSM	138.2	57.1	85.2
NCE	143.1	78.4	104.7
Weaknorm	124.4	56.9	98.7
WeaknormSQ	122.1	56.1	94.9
KN+Softmax	108.5	43.6	59.4
KN+D-Softmax	107.0	42.0	56.3
KN+Sampling	109.4	43.8	58.1
KN+HSM	115.0	43.9	55.6
KN+NCE	114.6	49.0	58.8
KN+Weaknorm	109.2	43.8	58.1
KN+WeaknormSQ	108.8	43.8	57.7

Table 2: Test perplexity of individual models and interpolation with Kneser-Ney.

Comparisons

	PTB	gigaword	billionW
KN	141.2	57.1	70.2
Softmax	123.8	56.5	108.3
D-Softmax	121.1	52.0	91.2
Sampling	124.2	57.6	101.0
HSM	138.2	57.1	85.2
NCE	143.1	78.4	104.7
Weaknorm	124.4	56.9	98.7
WeaknormSQ	122.1	56.1	94.9
KN+Softmax	108.5	43.6	59.4
KN+D-Softmax	107.0	42.0	56.3
KN+Sampling	109.4	43.8	58.1
KN+HSM	115.0	43.9	55.6
KN+NCE	114.6	49.0	58.8
KN+Weaknorm	109.2	43.8	58.1
KN+WeaknormSQ	108.8	43.8	57.7

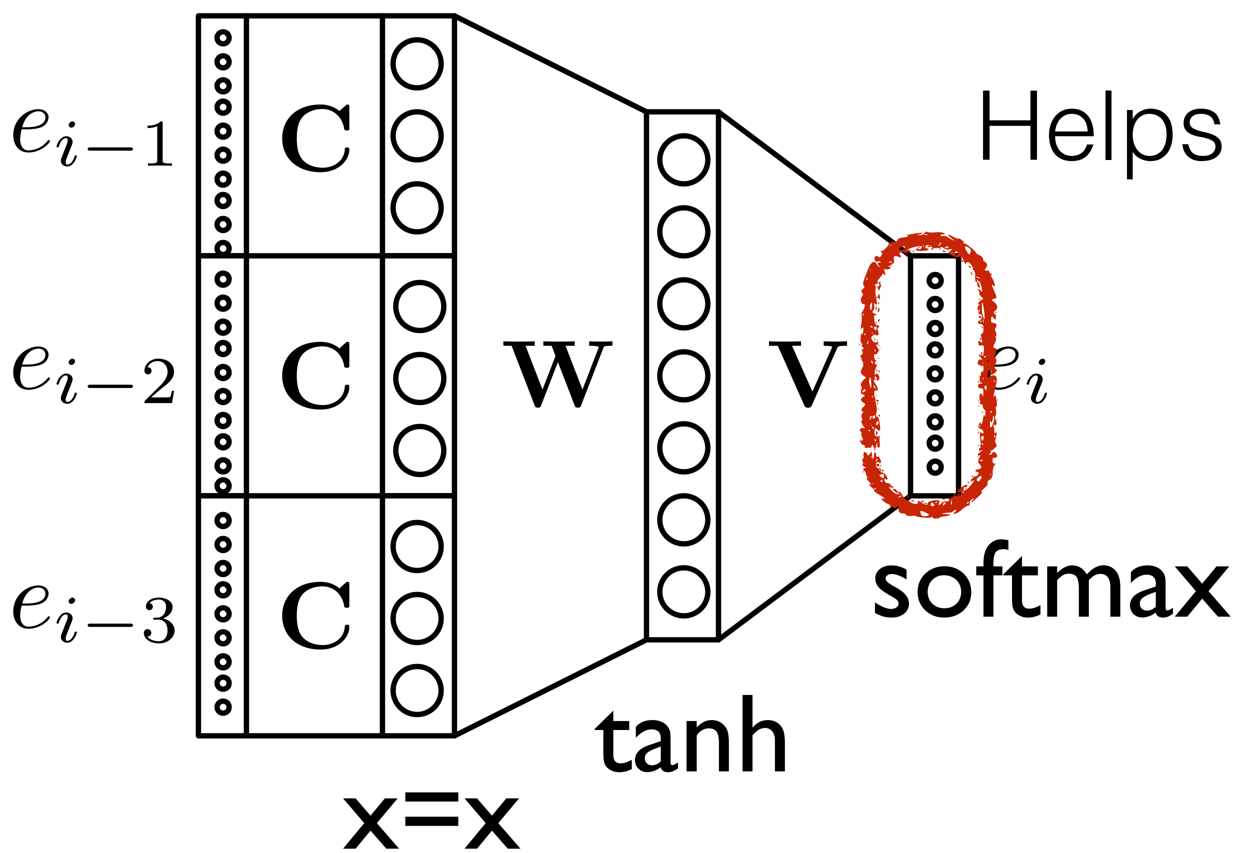
Room for improvement

Table 2: Test perplexity of individual models and interpolation with Kneser-Ney.

Bengio et al. (2003)

$$p(\mathbf{e}) = \prod_{i=1}^{|\mathbf{e}|} p(e_i \mid e_{i-n+1}, \ldots, e_{i-1})$$

$$p(e_i \mid e_{i-n+1}, \ldots, e_{i-1}) =$$



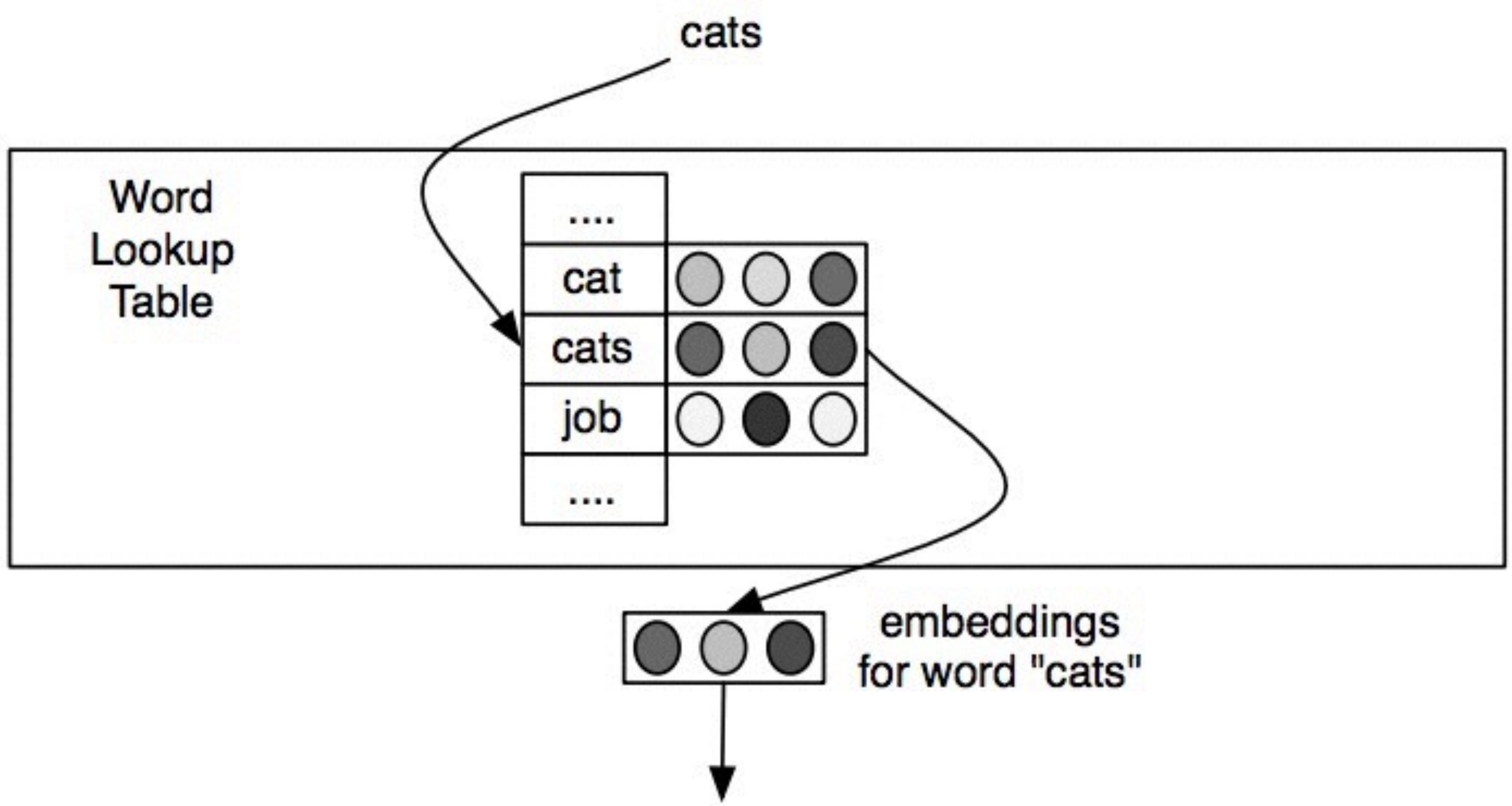
Helps with this problem

Known and finite

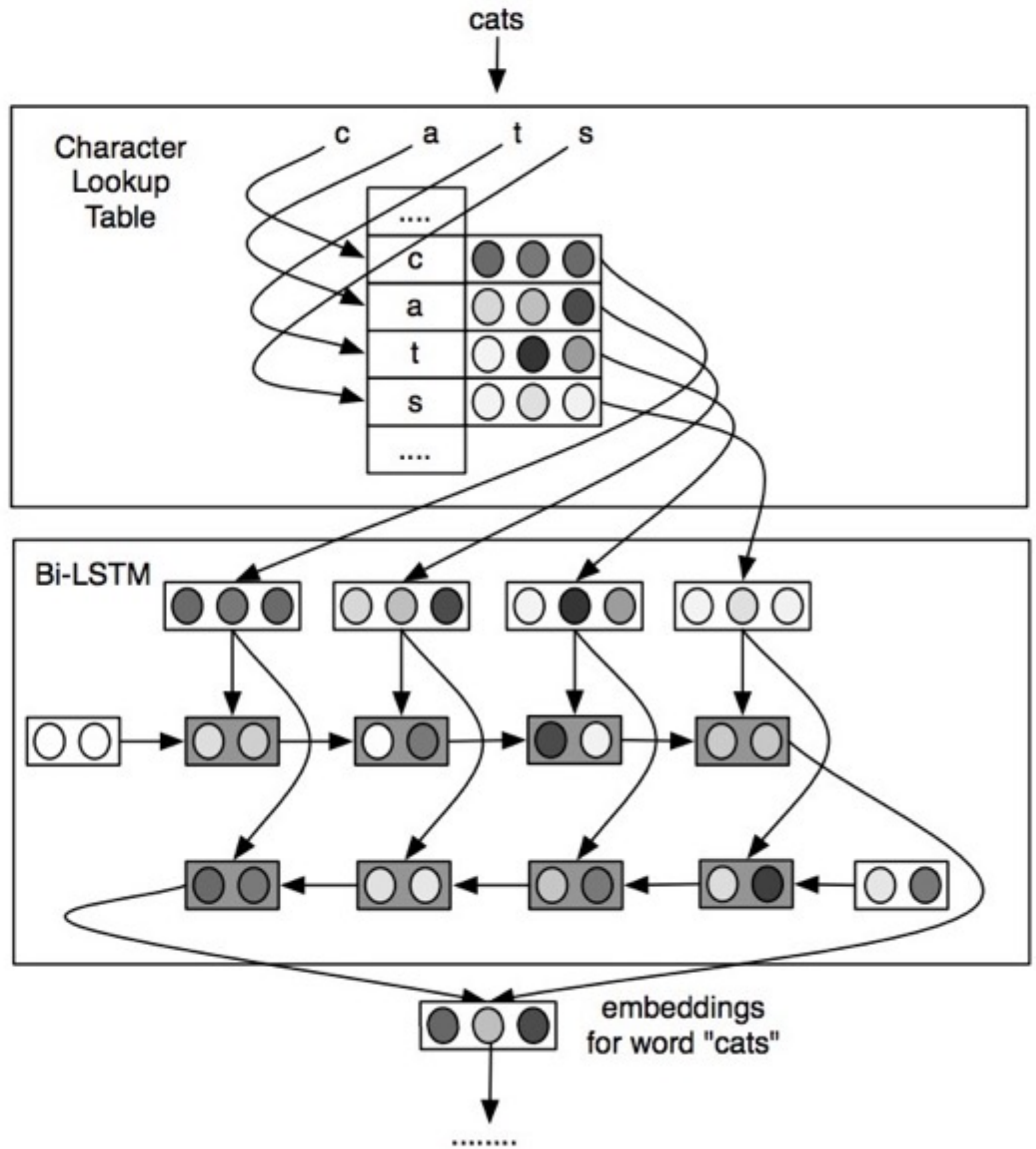
- Practical problem: softmax computation is linear in vocabulary size.
- **Theorem.** The vocabulary of word types is infinite.
  Proof 1. productive morphology and loanwords.
  Proof 2. 1, 2, 3, 4, …

What set is finite?

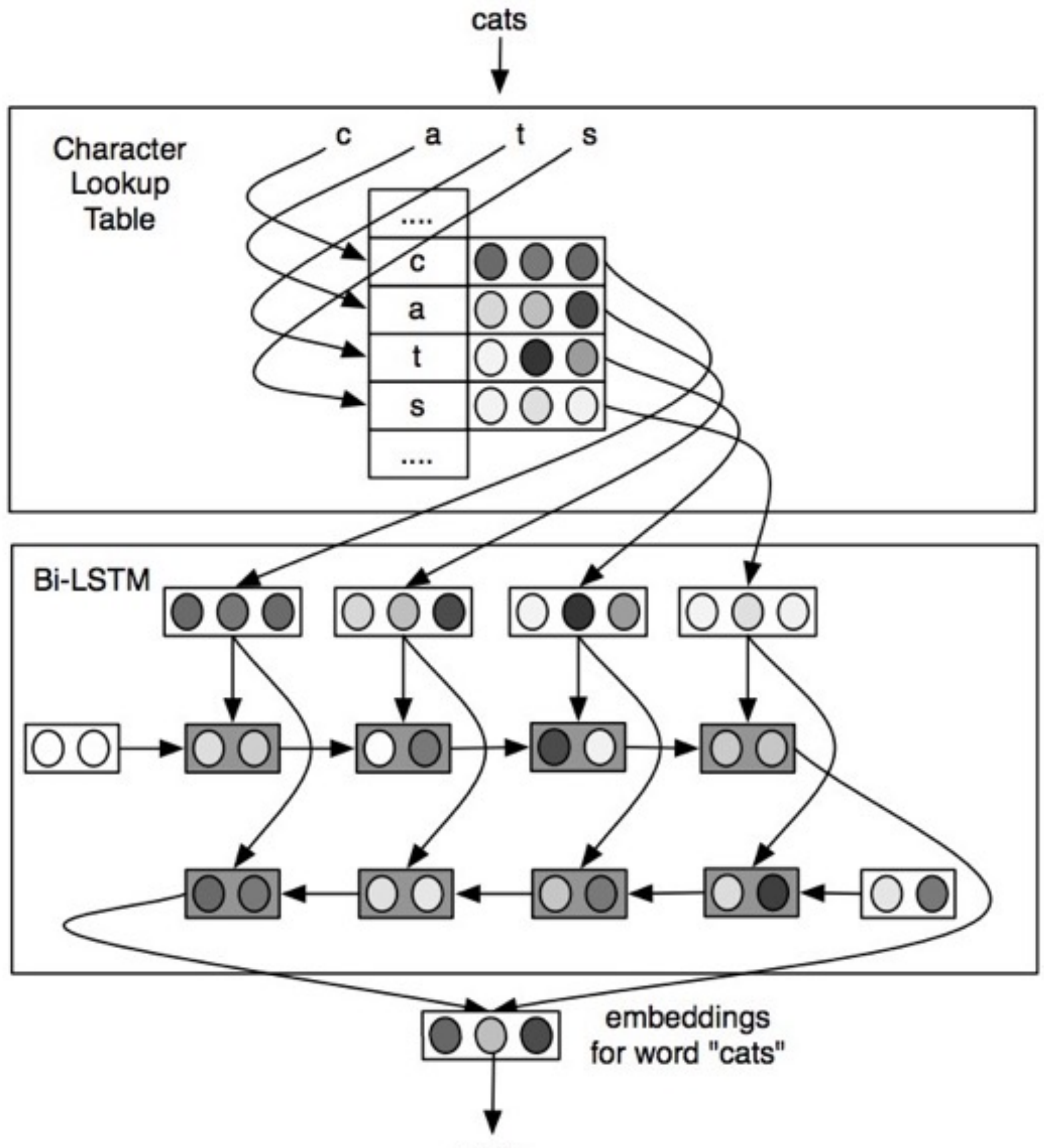
What set is finite?



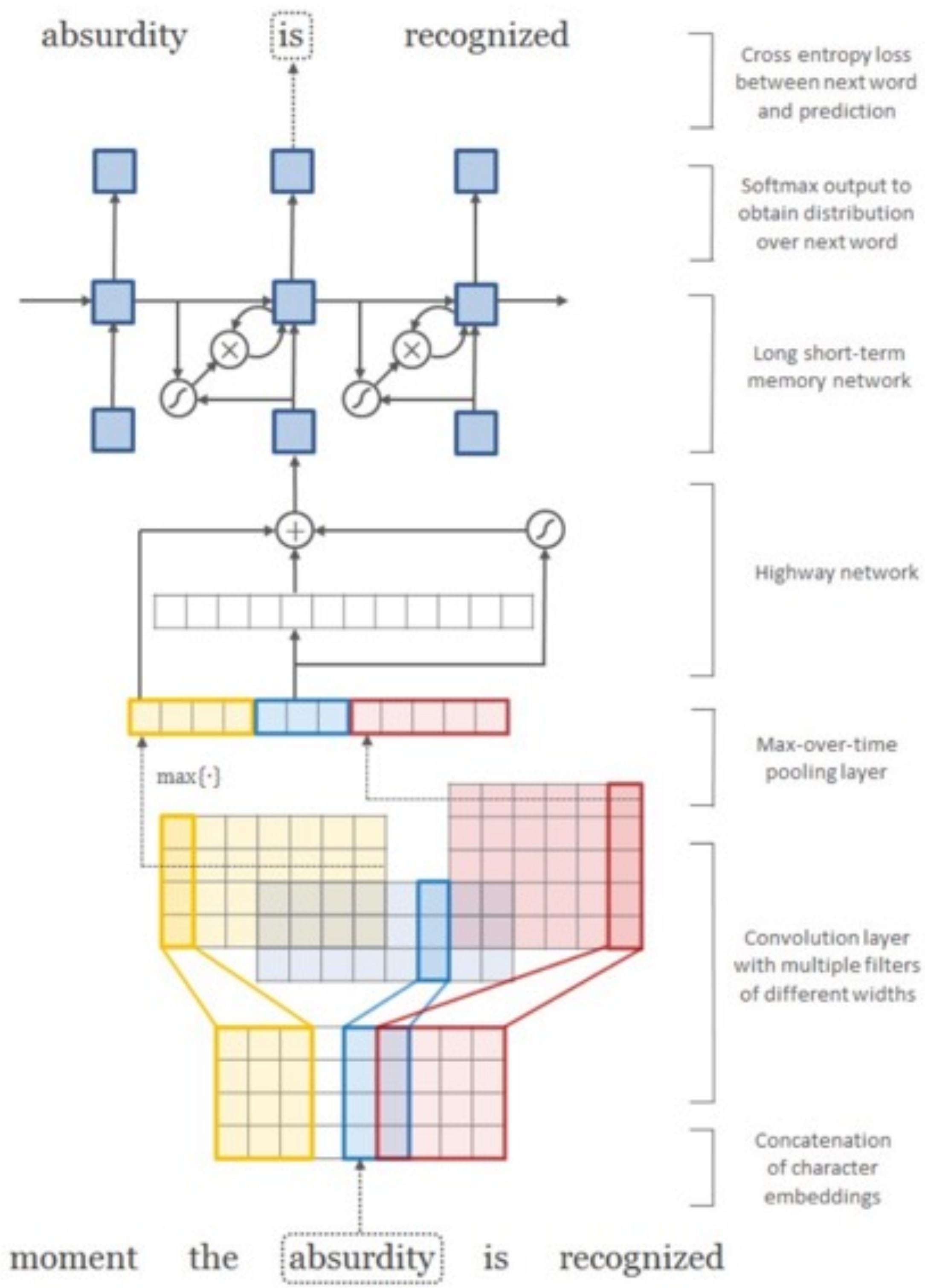
What set is finite?



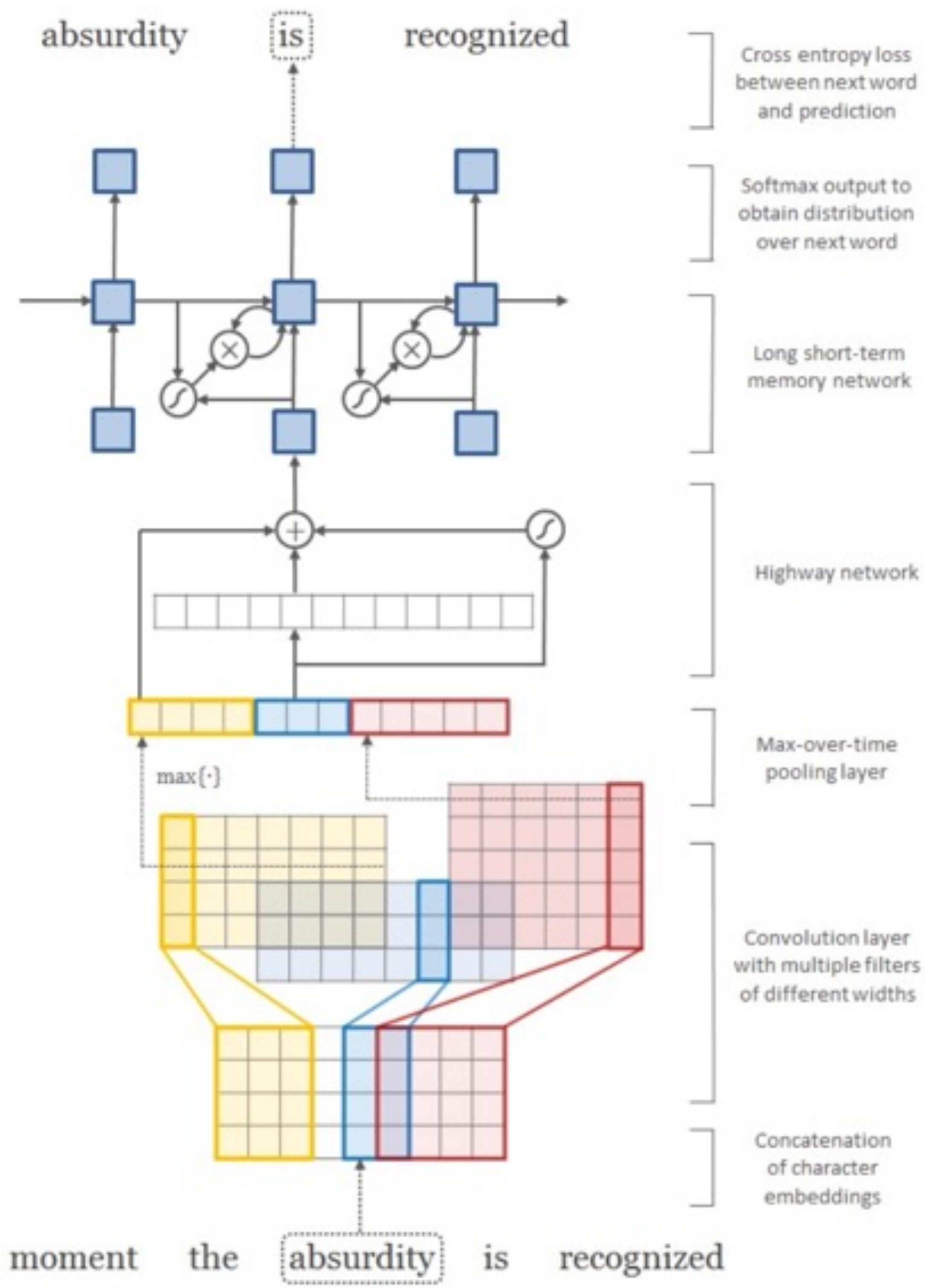
What set is finite?



What set is finite?



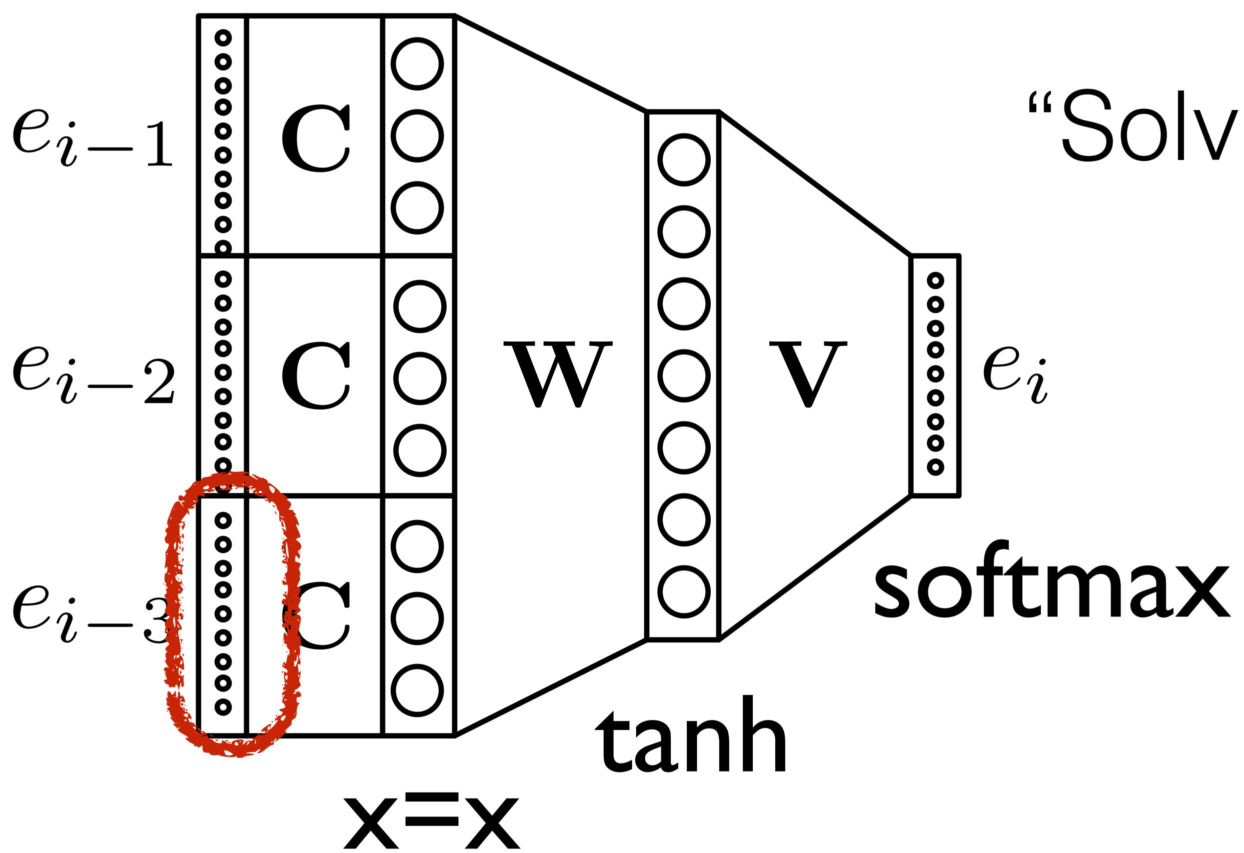
What set is finite?



Bengio et al. (2003)

$$p(\mathbf{e}) = \prod_{i=1}^{|\mathbf{e}|} p(e_i \mid e_{i-n+1}, \ldots, e_{i-1})$$

$$p(e_i \mid e_{i-n+1}, \ldots, e_{i-1}) =$$

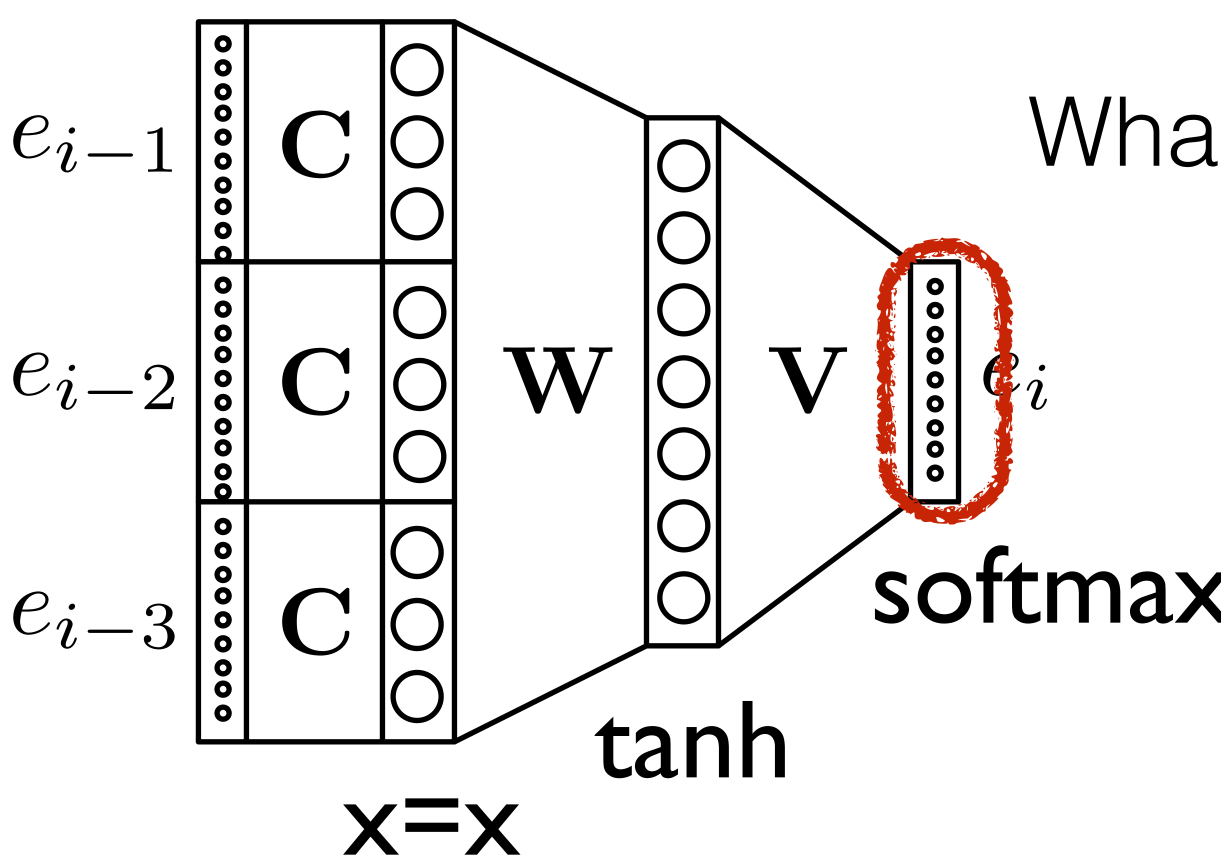


"Solves" this problem

Bengio et al. (2003)

$$p(\mathbf{e}) = \prod_{i=1}^{|\mathbf{e}|} p(e_i \mid e_{i-n+1}, \ldots, e_{i-1})$$

$$p(e_i \mid e_{i-n+1}, \ldots, e_{i-1}) =$$



What about this one?