Machine Translation

Philipp Koehn

18 June 2018

language models

N-Gram Backoff Language Model

- We approximate

$$p(W) = p(w_1, w_2, ..., w_n)$$

- ... by applying the chain rule

$$p(W) = \sum_i p(w_i|w_1, ..., w_{i-1})$$

- ... and limiting the history (Markov order)

$$p(w_i|w_1, ..., w_{i-1}) \simeq p(w_i|w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1})$$

- Each $p(w_i|w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1})$ may not have enough statistics to estimate

$\rightarrow$ we back off to $p(w_i|w_{i-3}, w_{i-2}, w_{i-1})$, $p(w_i|w_{i-2}, w_{i-1})$, etc., all the way to $p(w_i)$

– exact details of backing off get complicated — "interpolated Kneser-Ney"

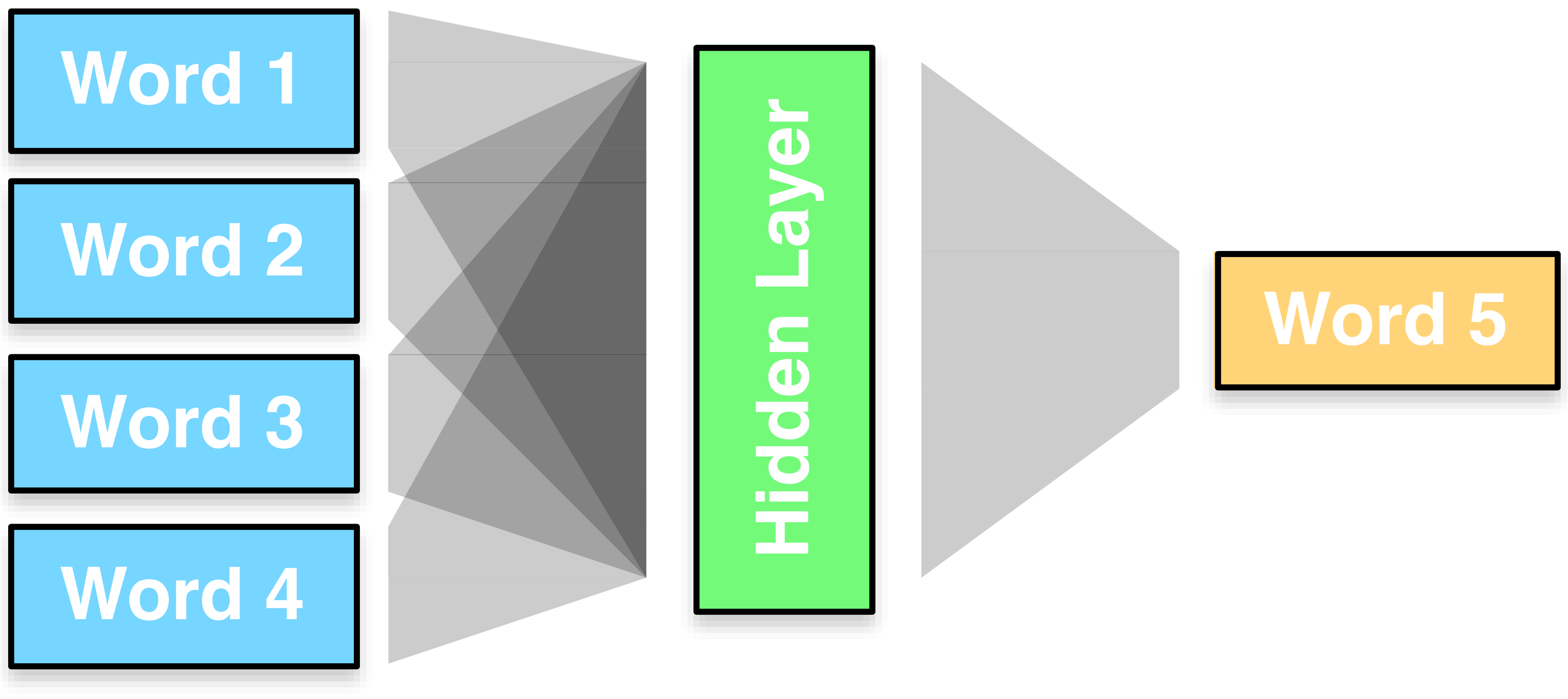
Refinements

- A whole family of back-off schemes
- Skip-n gram models that may back off to $p(w_i|w_{i-2})$
- Class-based models $p(C(w_i)|C(w_{i-4}), C(w_{i-3}), C(w_{i-2}), C(w_{i-1}))$

⇒ We are wrestling here with

- using as much relevant evidence as possible
- pooling evidence between words

First Sketch



Representing Words

- Words are represented with a one-hot vector, e.g.,
  - dog = (0,0,0,0,1,0,0,0,0,....)
  - cat = (0,0,0,0,0,0,0,1,0,....)
  - eat = (0,1,0,0,0,0,0,0,0,....)

- That's a large vector!

- Remedies
  - limit to, say, 20,000 most frequent words, rest are OTHER
  - place words in $\sqrt{n}$ classes, so each word is represented by
    - 1 class label
    - 1 word in class label

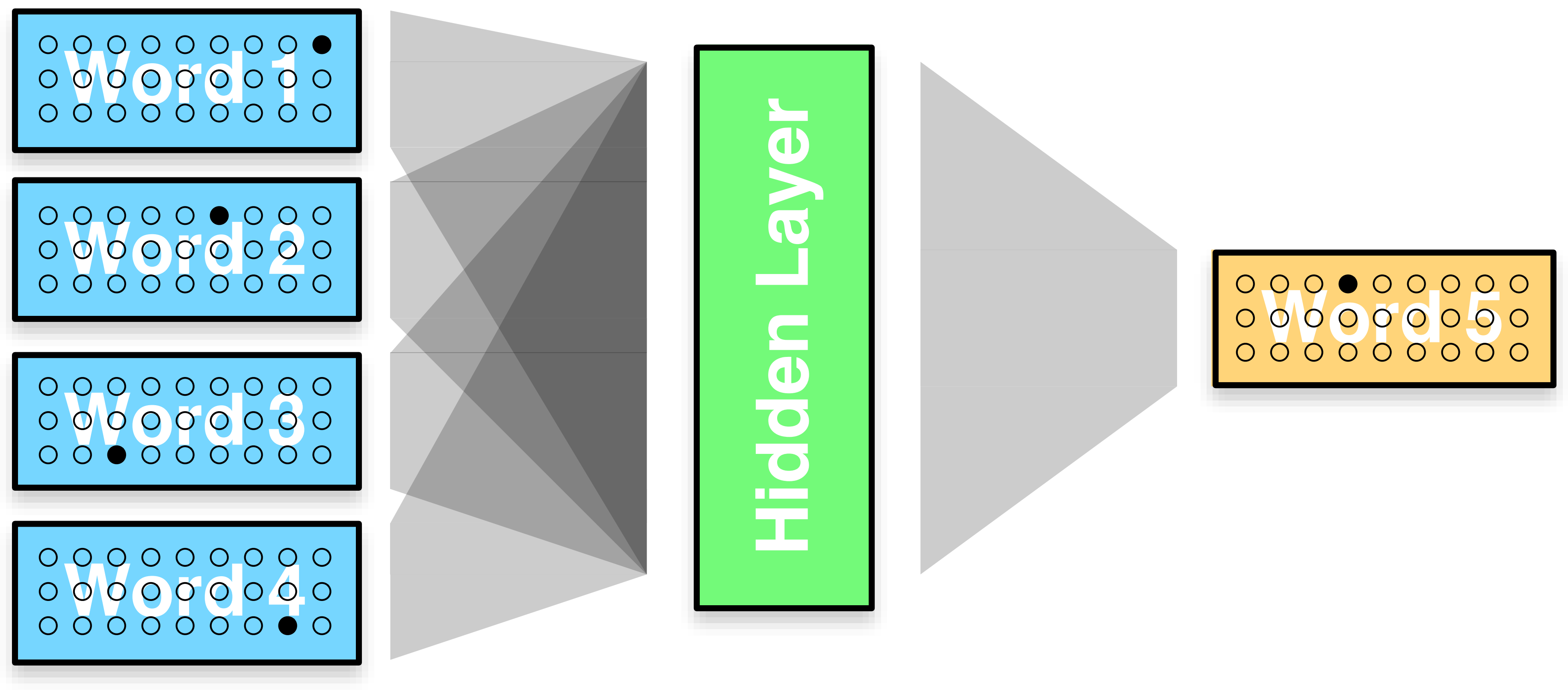
Word Classes for Two-Hot Representations

- WordNet classes
- Brown clusters
- Frequency binning
  - sort words by frequency
  - place them in order into classes
  - each class has same token count

  → very frequent words have their own class

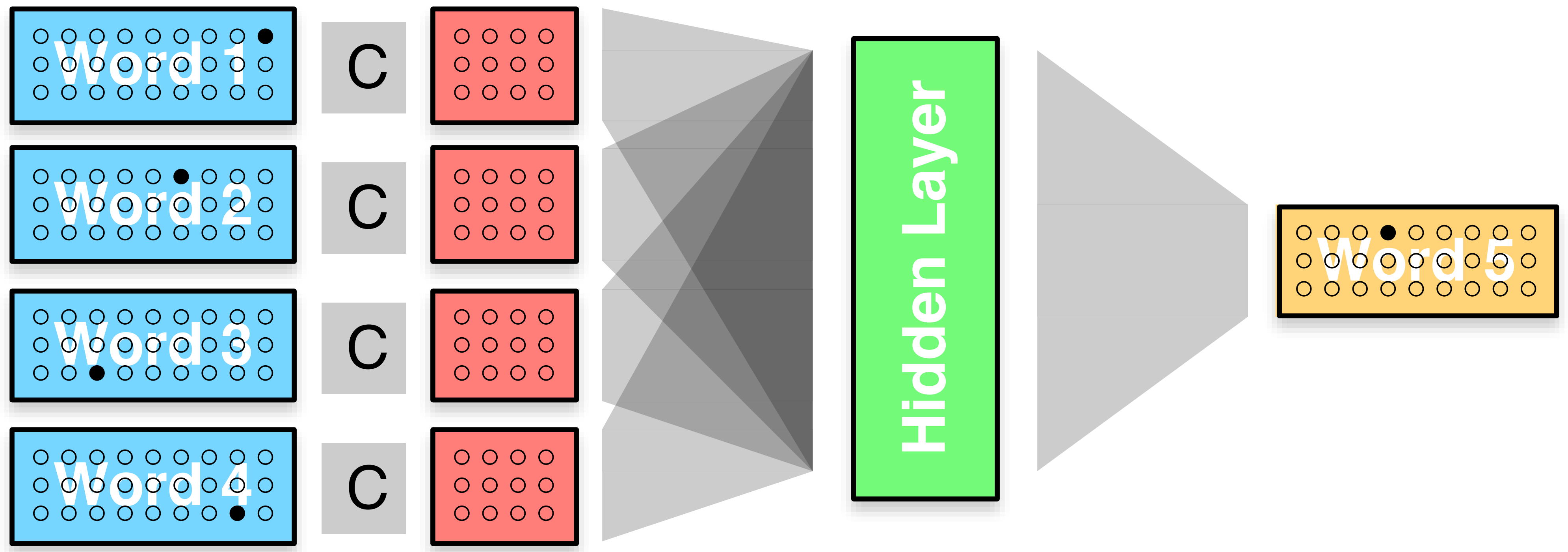
  → rare words share class with many other words
- Anything goes: assign words randomly to classes

Second Sketch



word embeddings

Add a Hidden Layer

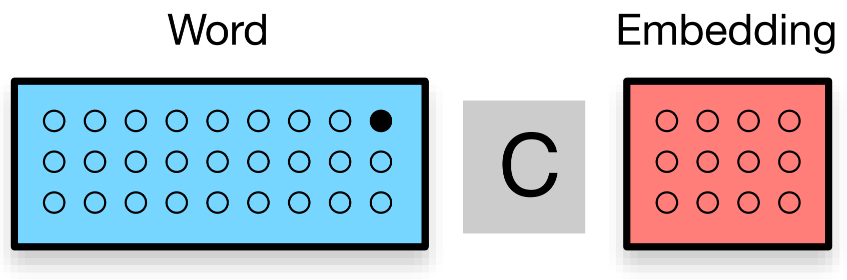


- Map each word first into a lower-dimensional real-valued space
- Shared weight matrix C

Details (Bengio et al., 2003)

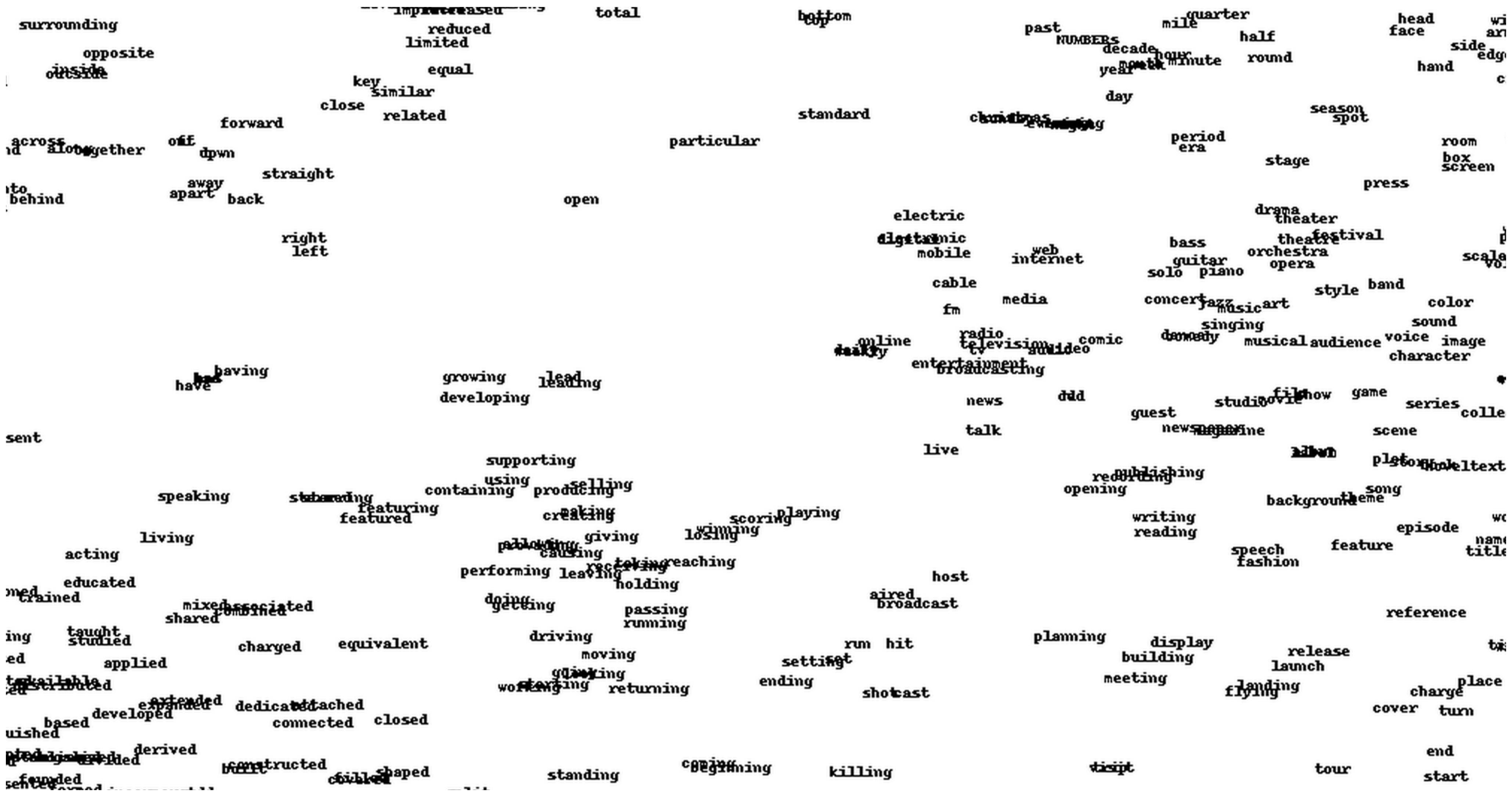
- Add direct connections from embedding layer to output layer
- Activation functions
  - input→embedding: none
  - embedding→hidden: tanh
  - hidden→output: softmax
- Training
  - loop through the entire corpus
  - update between predicted probabilities and 1-hot vector for output word

Word Embeddings

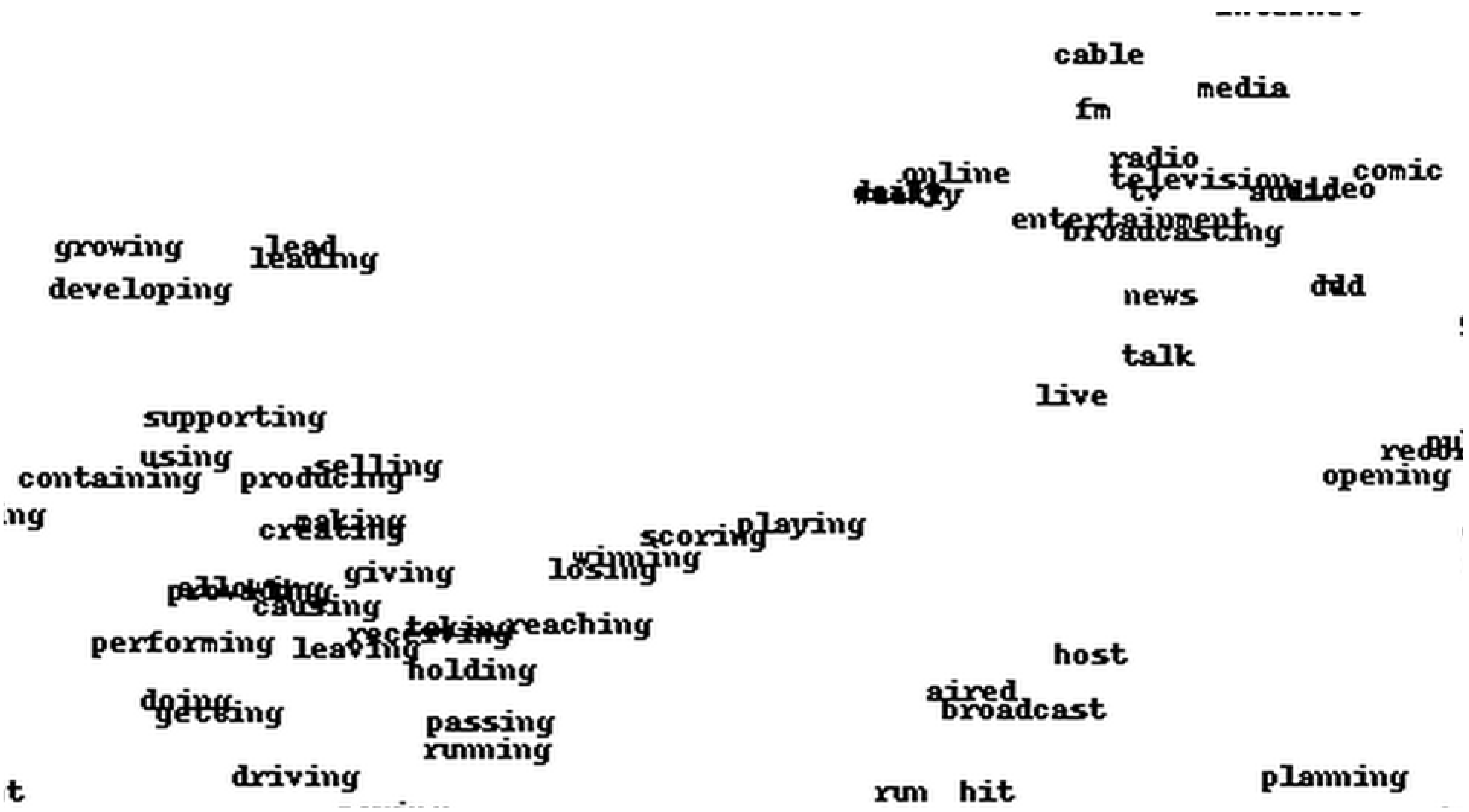


- By-product: embedding of word into continuous space
- Similar contexts $\rightarrow$ similar embedding
- Recall: distributional semantics

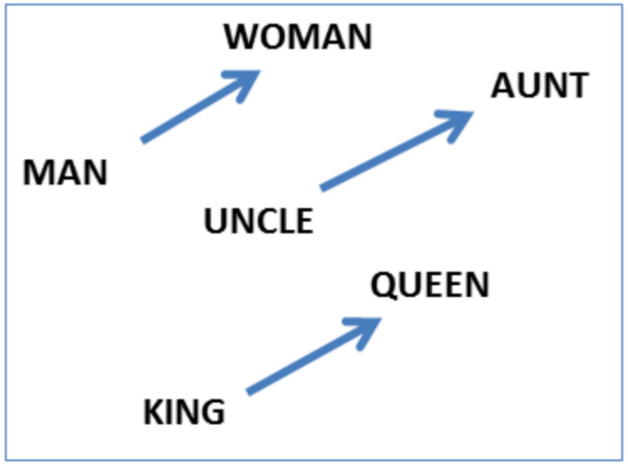
Word Embeddings

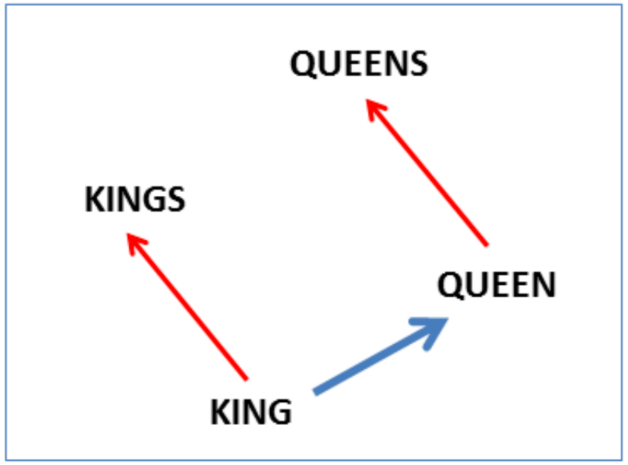


Word Embeddings



Are Word Embeddings Magic?



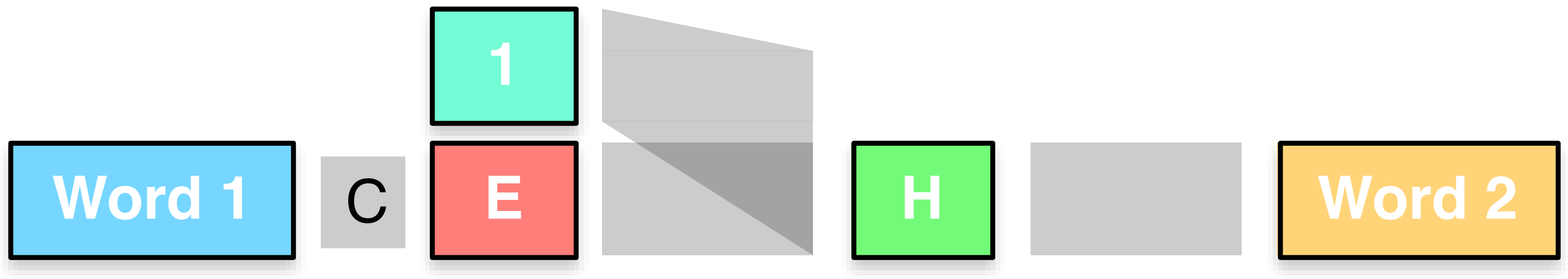


- Morphosyntactic regularities (Mikolov et al., 2013)
  - adjectives base form vs. comparative, e.g., good, better
  - nouns singular vs. plural, e.g., year, years
  - verbs present tense vs. past tense, e.g., see, saw

- Semantic regularities
  - clothing is to shirt as dish is to bowl
  - evaluated on human judgment data of semantic similarities

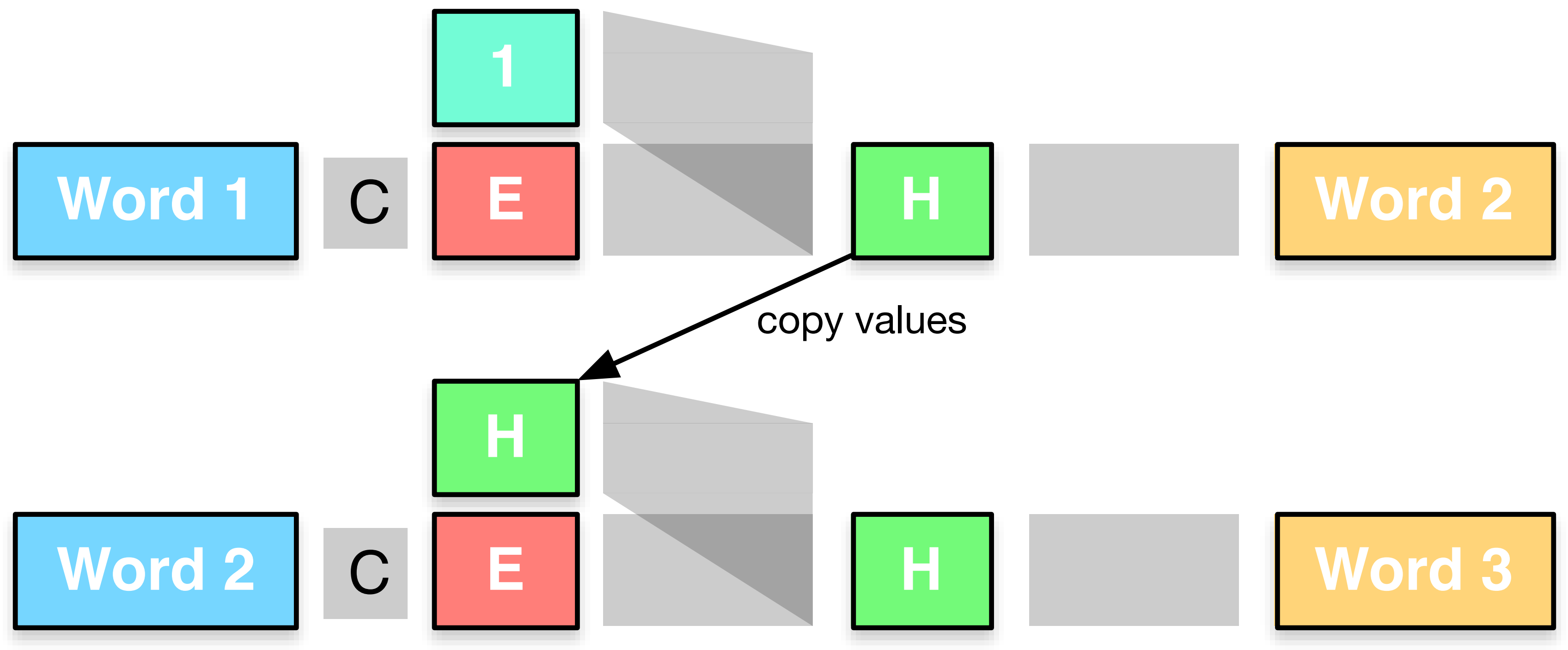
recurrent neural networks

Recurrent Neural Networks

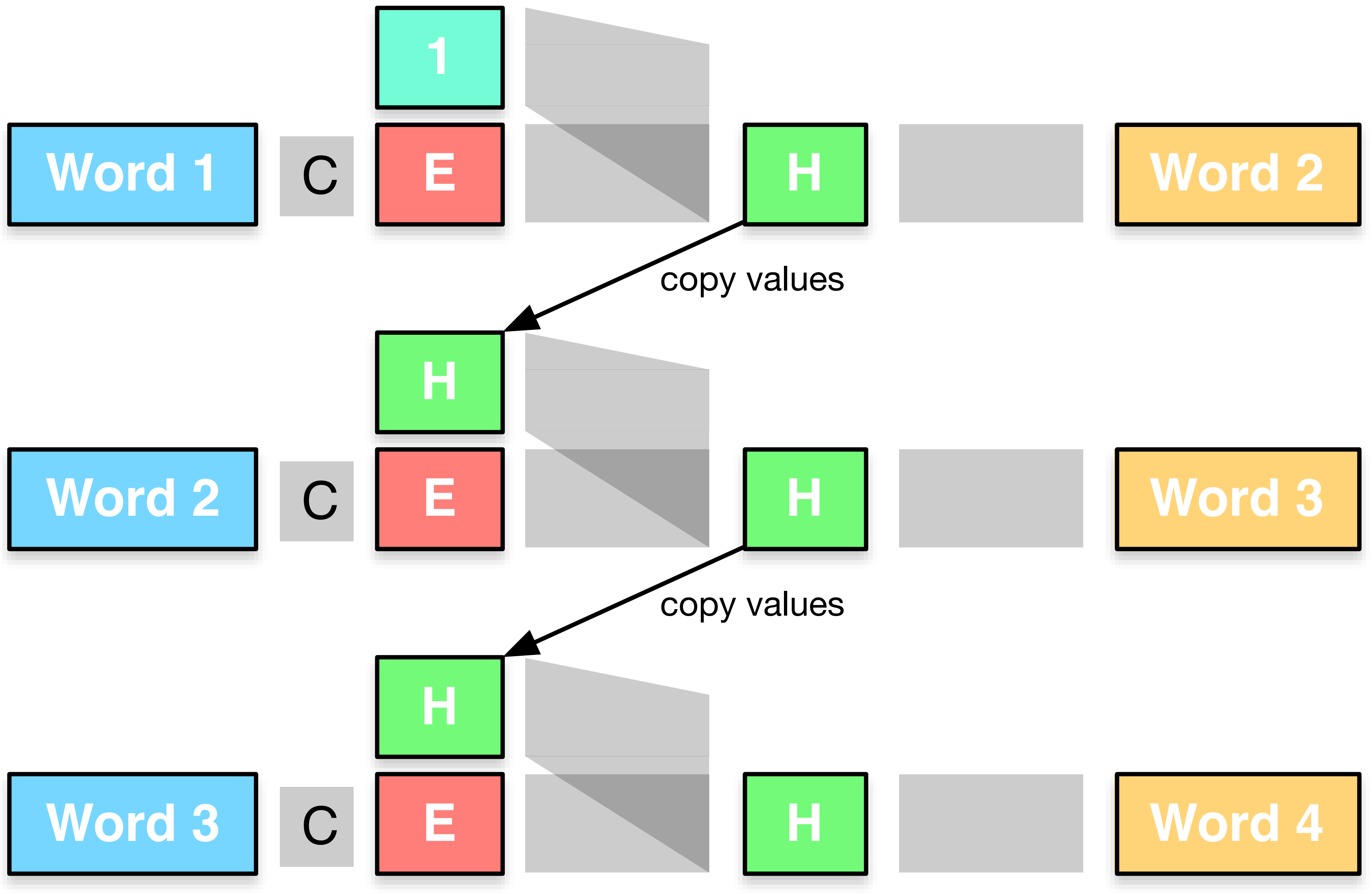


- Start: predict second word from first
- Mystery layer with nodes all with value 1

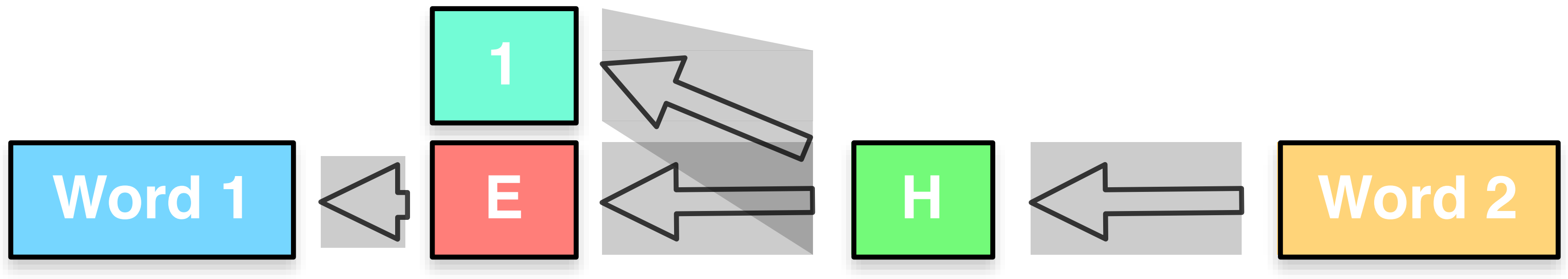
Recurrent Neural Networks



Recurrent Neural Networks

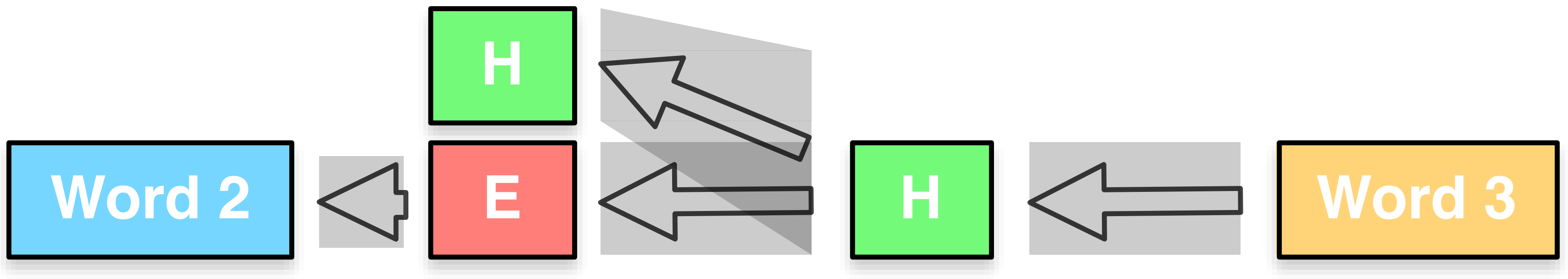


Training



- Process first training example
- Update weights with back-propagation

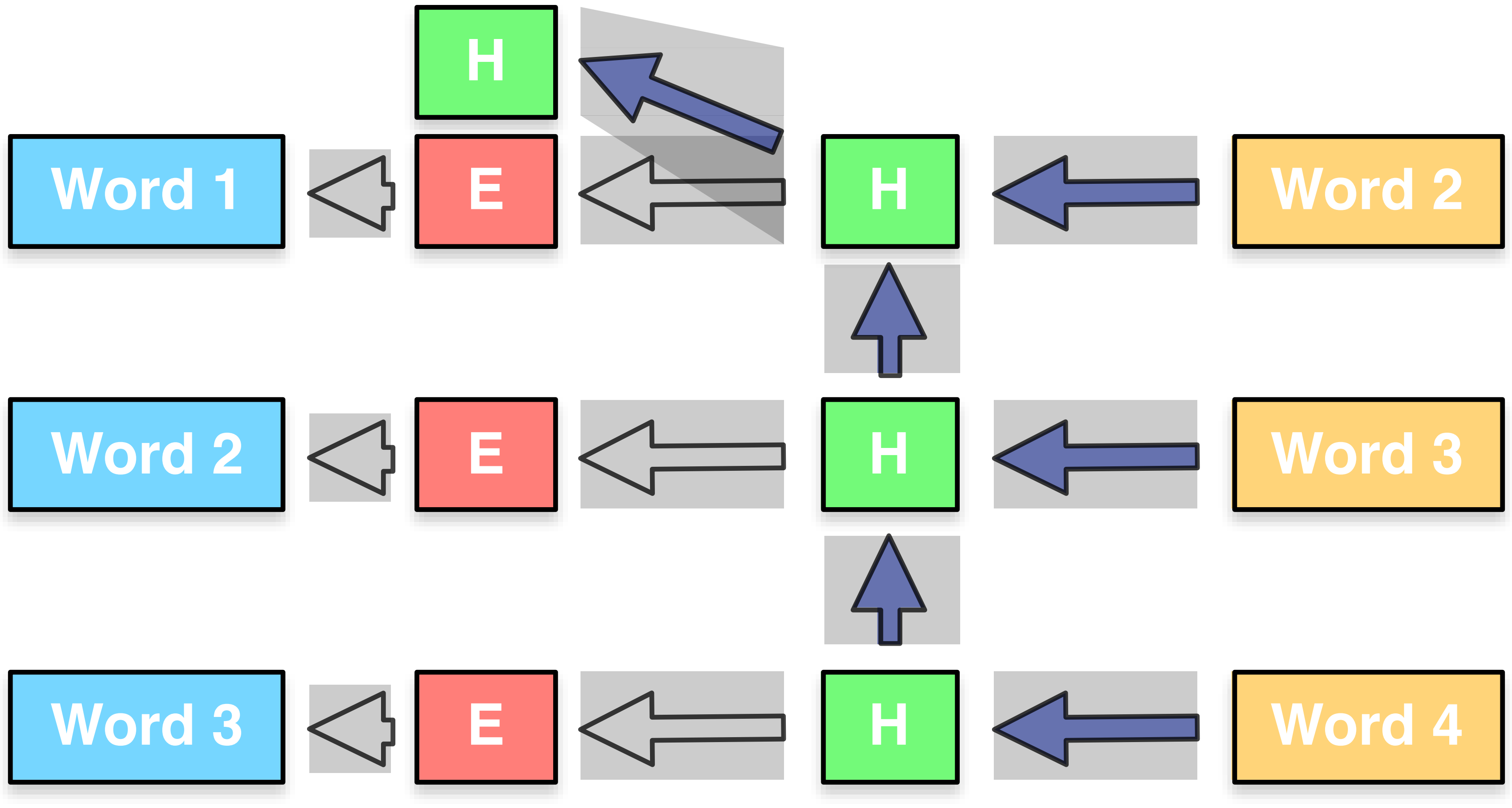
Training



- Process second training example
- Update weights with back-propagation
- And so on...

- But: no feedback to previous history

Back-Propagation Through Time



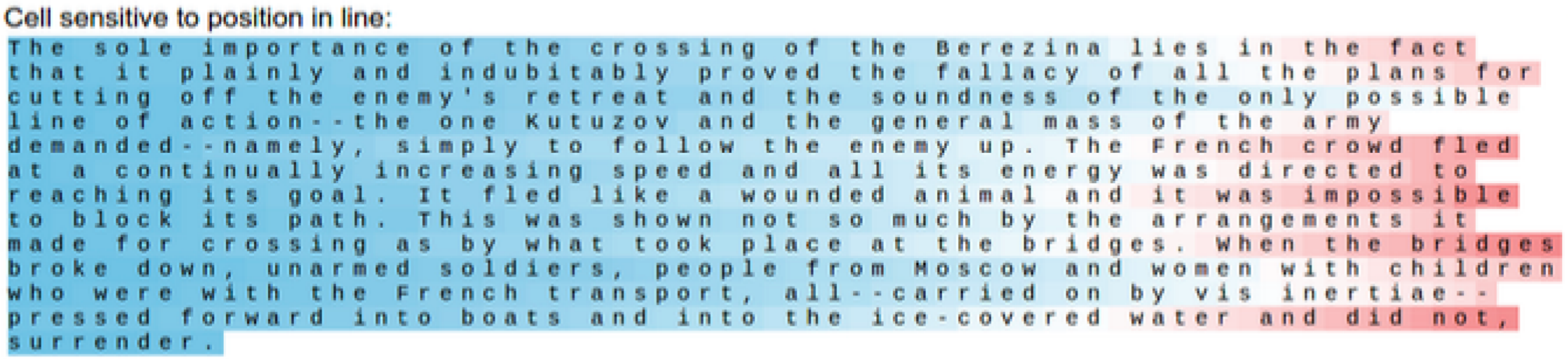
- After processing a few training examples,
  update through the unfolded recurrent neural network

Back-Propagation Through Time

- Carry out back-propagation though time (BPTT) after each training example
  - 5 time steps seems to be sufficient
  - network learns to store information for more than 5 time steps

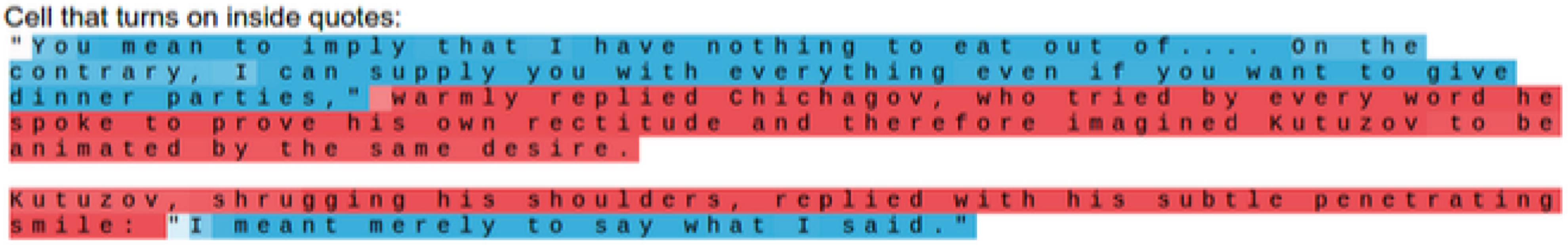
- Or: update in mini-batches
  - process 10-20 training examples
  - update backwards through all examples
  - removes need for multiple steps for each training example

Visualizing Individual Cells



Cell sensitive to position in line:

The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action--the one Kutuzov and the general mass of the army demanded--namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all--carried on by vis inertiae--pressed forward into boats and into the ice-covered water and did not, surrender.



Cell that turns on inside quotes:

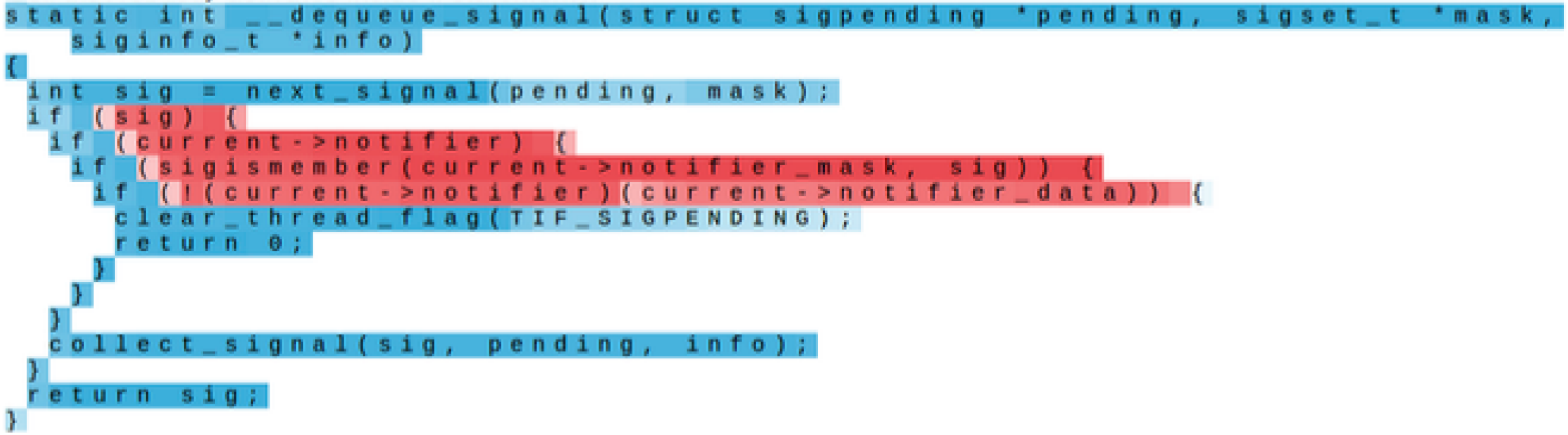
"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

Karpathy et al. (2015): "Visualizing and Understanding Recurrent Networks"

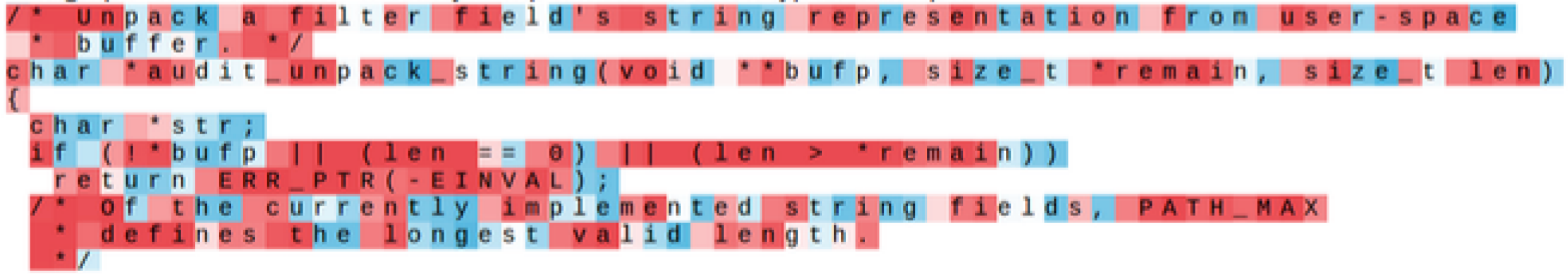
Visualizing Individual Cells

Cell that robustly activates inside if statements:



```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
   siginfo_t *info)
{
 int sig = next_signal(pending, mask);
 if (sig) {
  if (current->notifier) {
   if (sigismember(current->notifier_mask, sig)) {
    if (!(current->notifier)(current->notifier_data)) {
     clear_thread_flag(TIF_SIGPENDING);
     return 0;
    }
   }
  }
  collect_signal(sig, pending, info);
 }
 return sig;
}
```

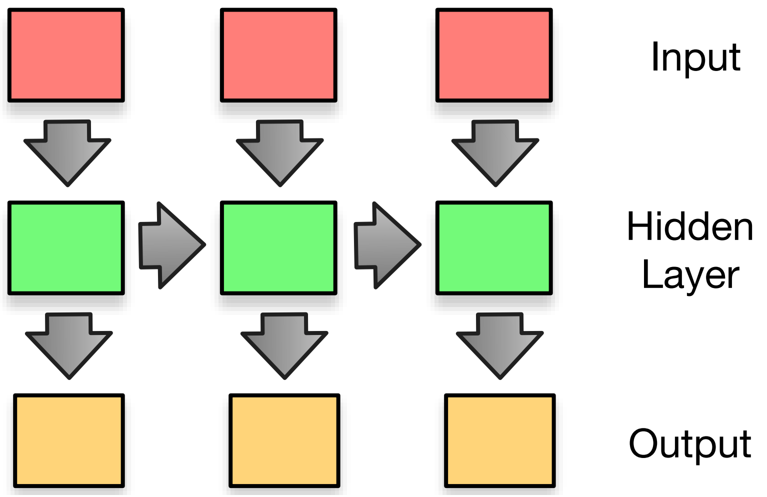
A large portion of cells are not easily interpretable. Here is a typical example:



```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
 char *str;
 if (!*bufp || (len == 0) || (len > *remain))
  return ERR_PTR(-EINVAL);
 /* Of the currently implemented string fields, PATH_MAX
  * defines the longest valid length.
  */
```

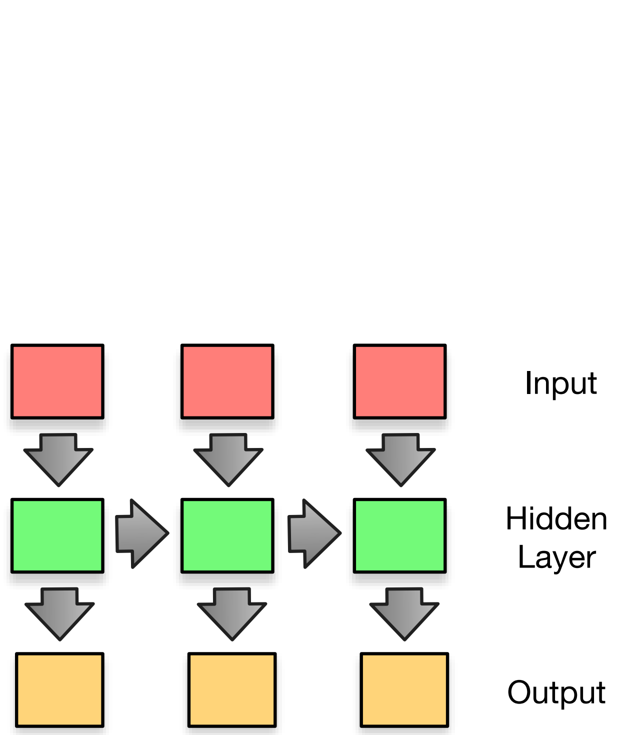
deeper models

Deep Learning?

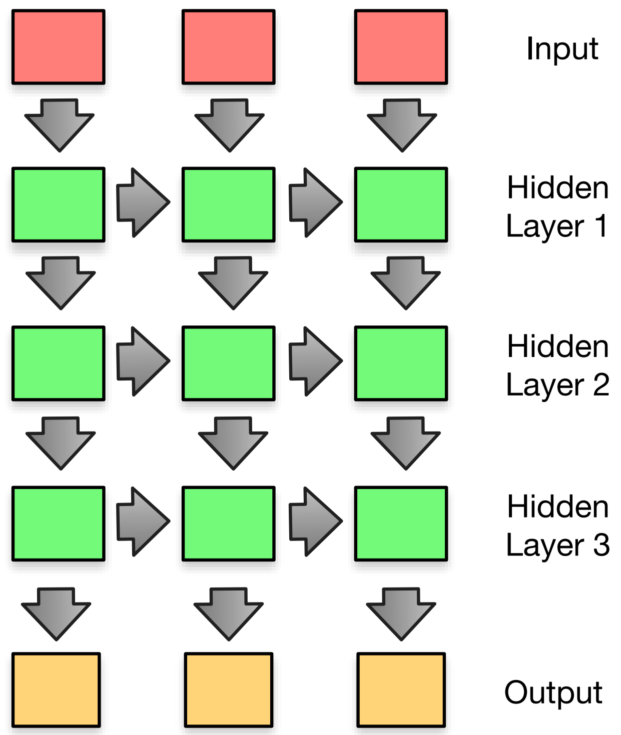


- Not much **deep** learning so far
- Between prediction from input to output: only 1 hidden layer
- How about more hidden layers?

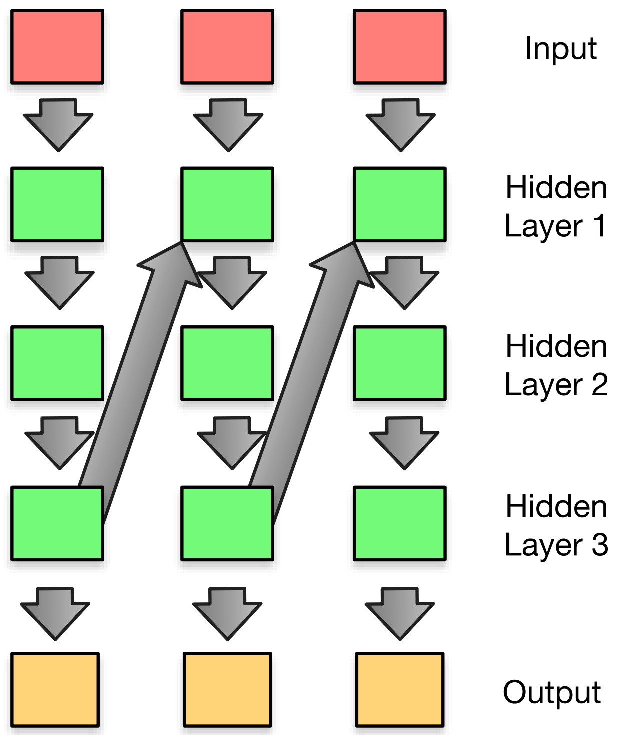
Deep Models



Shallow



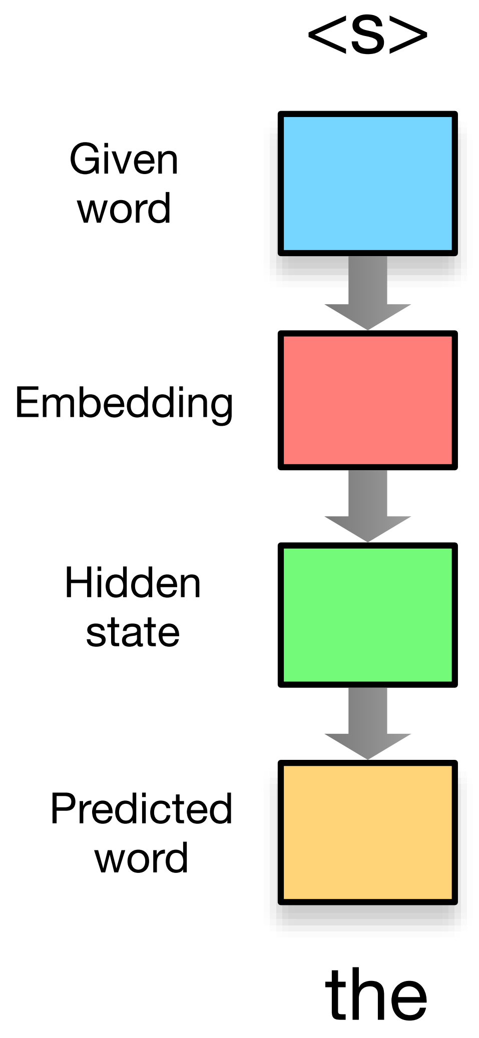
Deep Stacked



Deep Transition

towards translation models

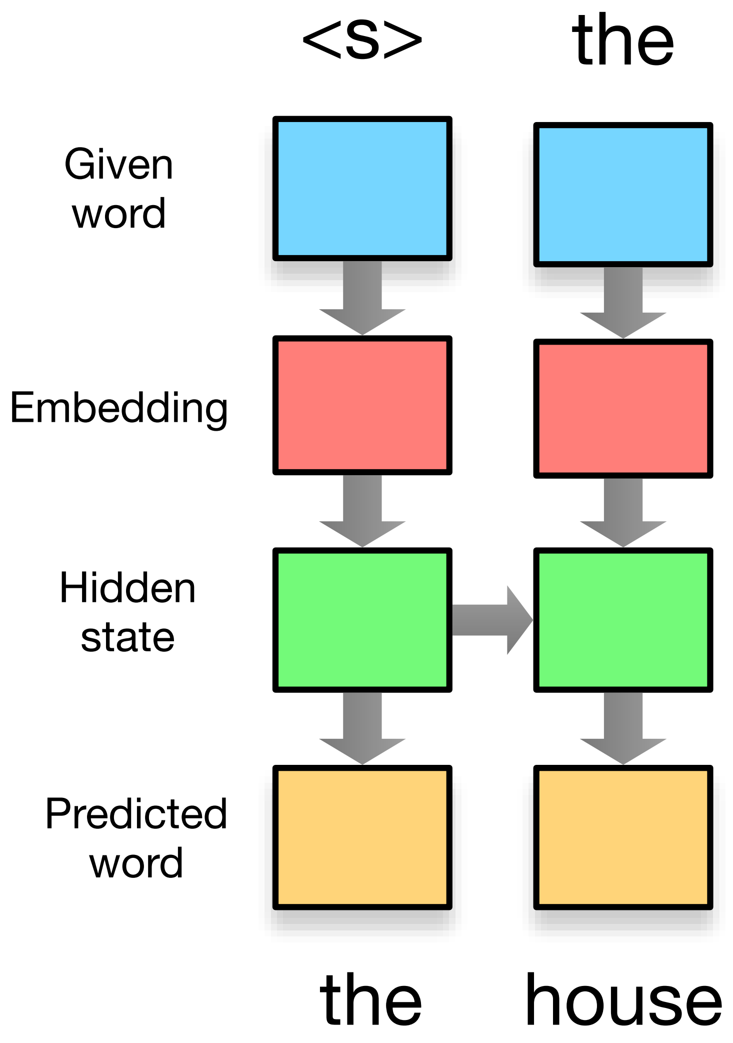
Recurrent Neural Language Model



Predict
the first word
of a sentence

Same as before,
just drawn top-down

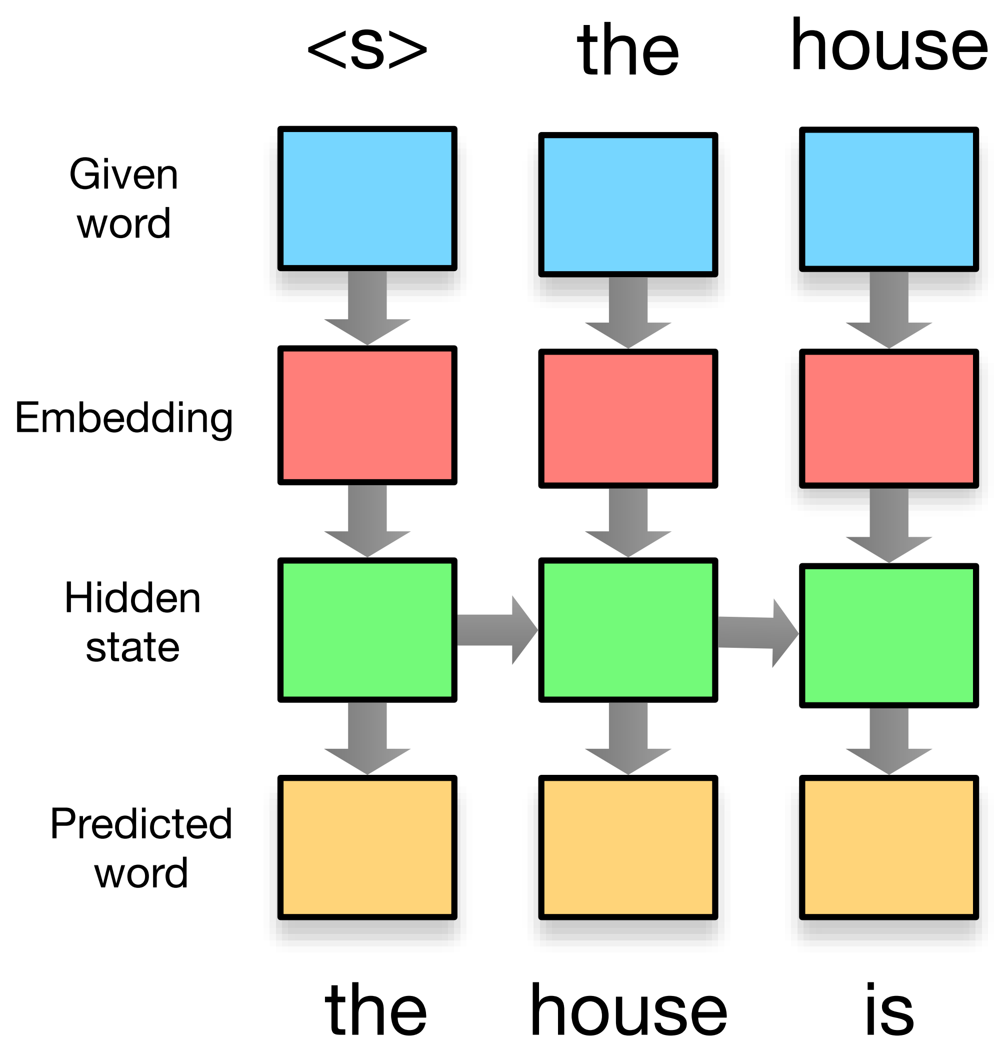
Recurrent Neural Language Model



Predict
the second word
of a sentence

Re-use hidden state
from
first word prediction

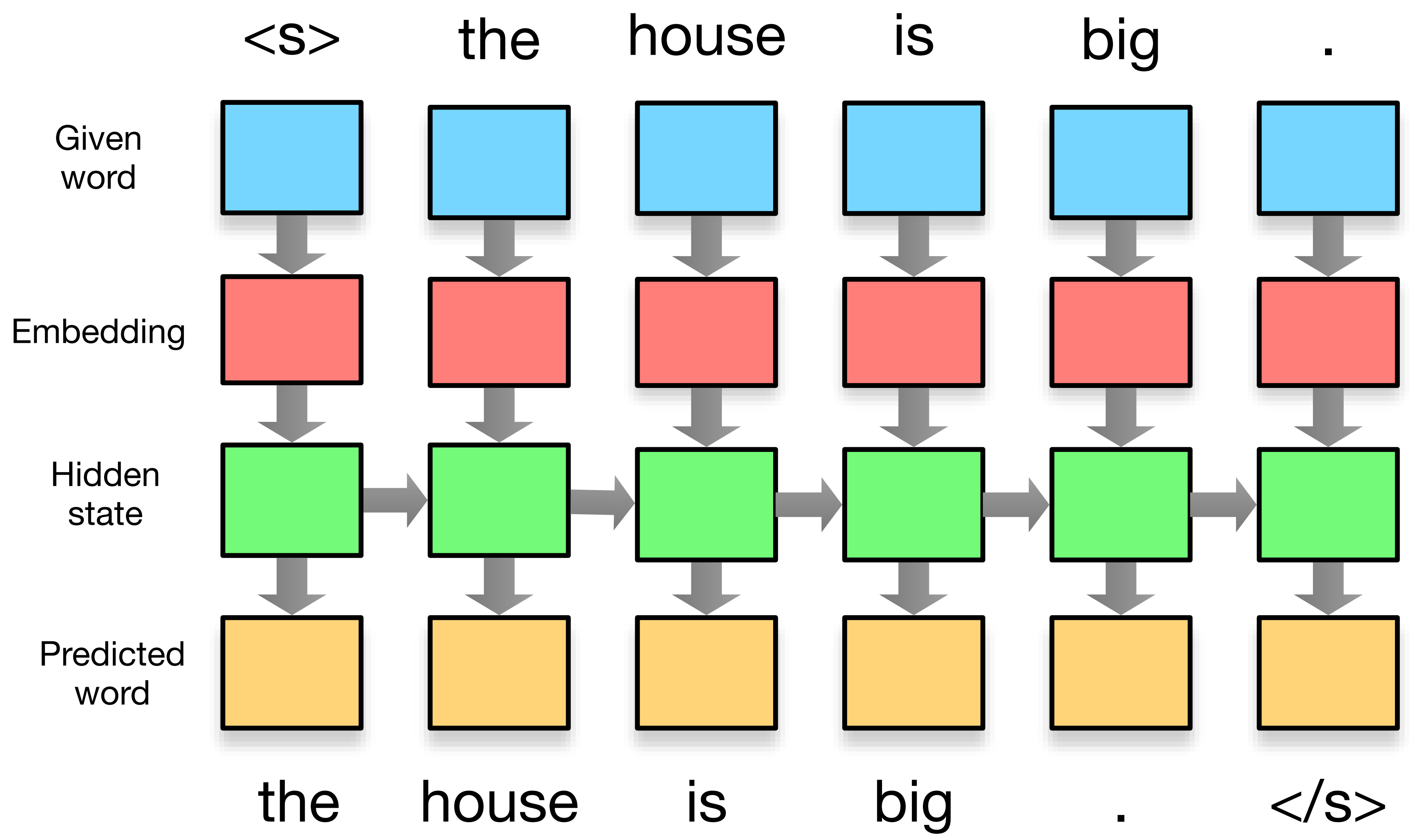
Recurrent Neural Language Model



Predict
the third word
of a sentence

... and so on

Recurrent Neural Language Model

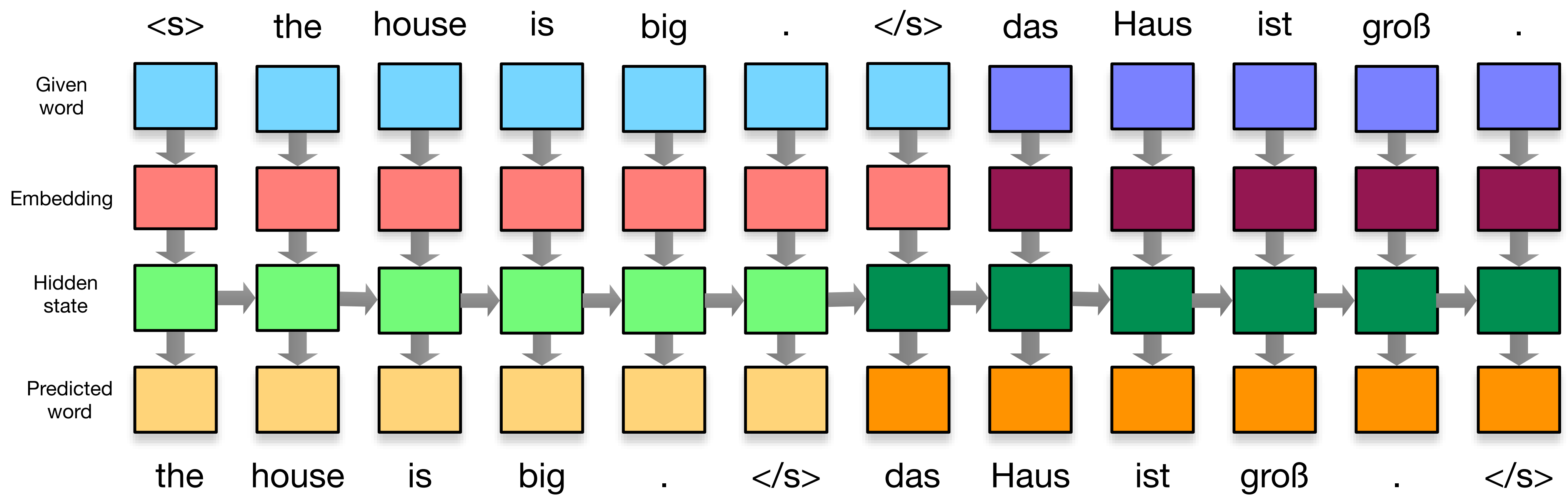


Recurrent Neural Translation Model

- We predicted the words of a sentence

- Why not also predict their translations?

Encoder-Decoder Model



- Obviously madness
- Proposed by Google (Sutskever et al. 2014)

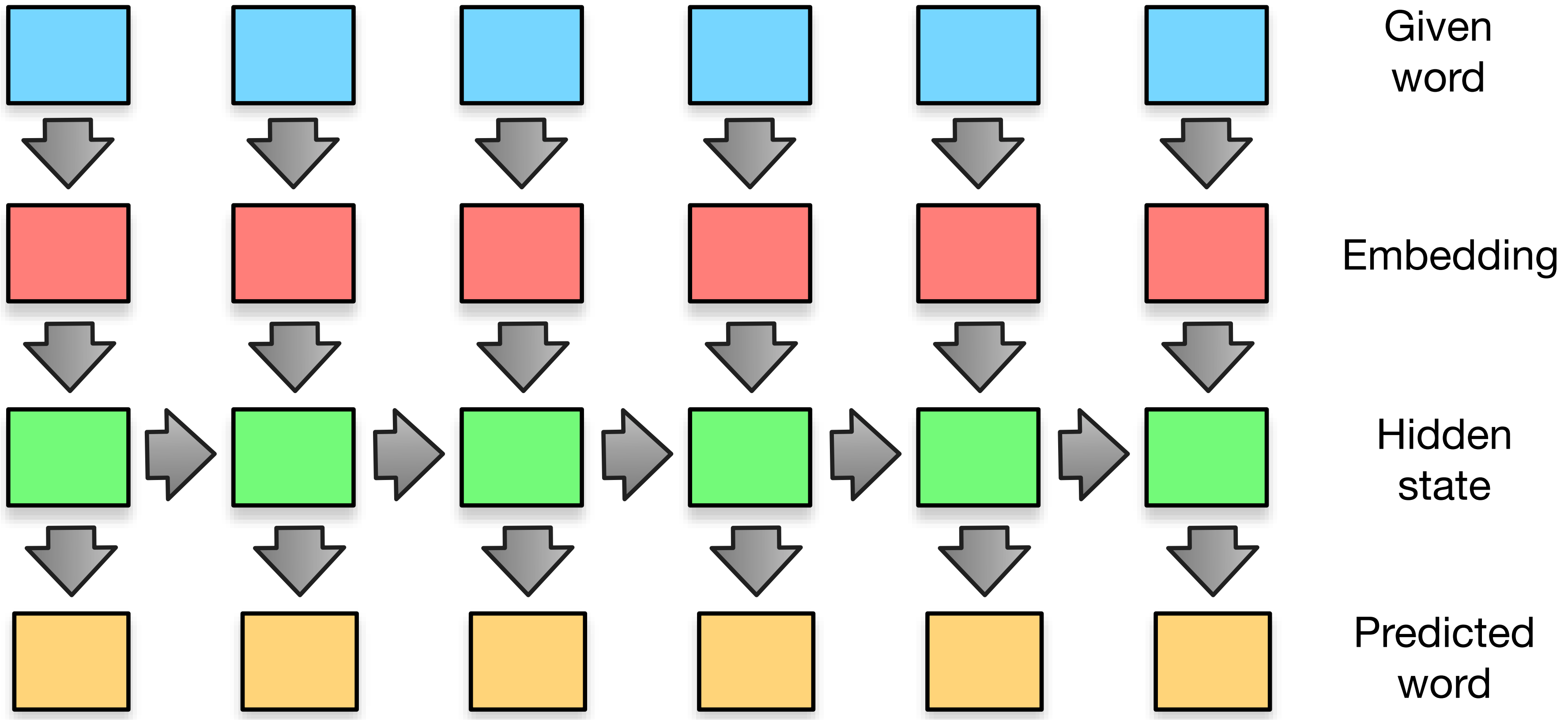
What is missing?

- Alignment of input words to output words

⇒ Solution: attention mechanism

neural translation model with attention

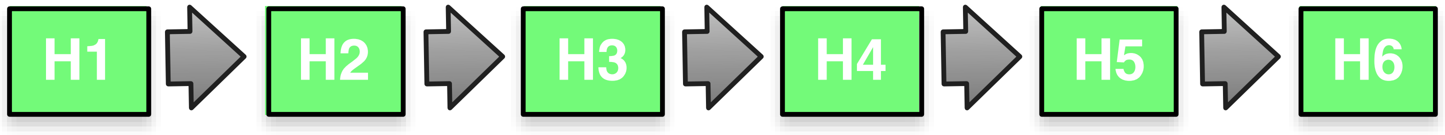
Input Encoding



- Inspiration: recurrent neural network language model on the input side

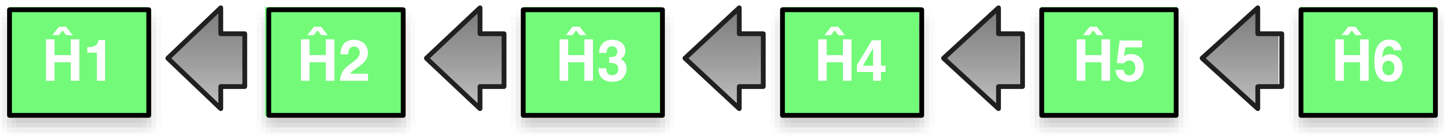
Hidden Language Model States

- This gives us the hidden states

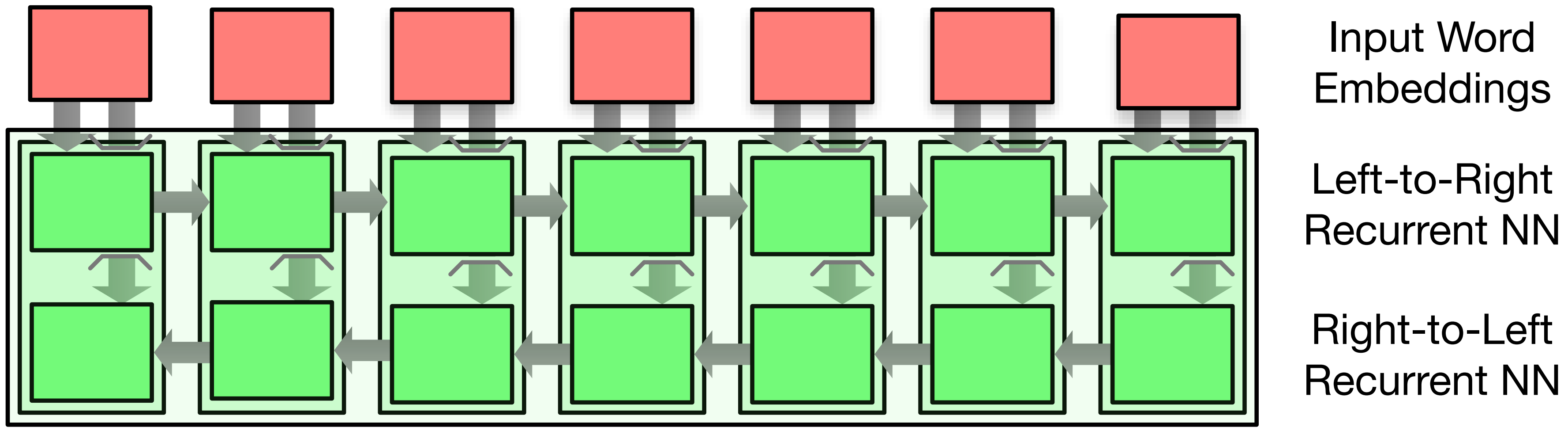


- These encode left context for each word

- Same process in reverse: right context for each word

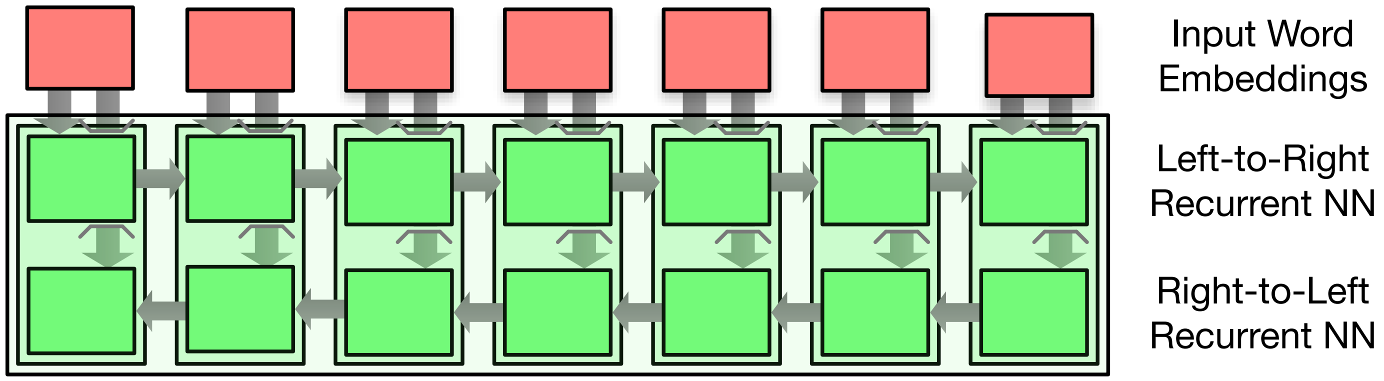


Input Encoder



- Input encoder: concatenate bidirectional RNN states
- Each word representation includes full left and right sentence context

Encoder: Math



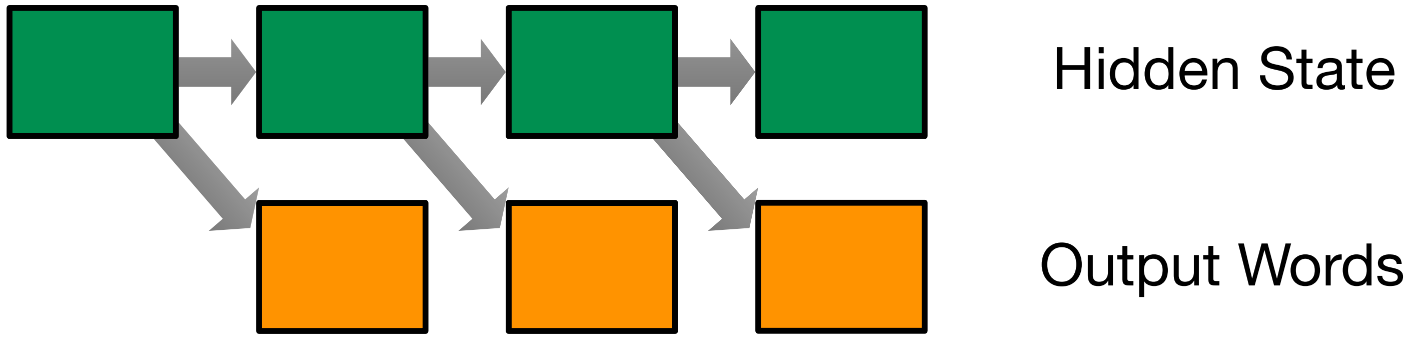
- Input is sequence of words x_j, mapped into embedding space $\bar{E}\ x_j$
- Bidirectional recurrent neural networks

$$\overleftarrow{h_j} = f(\overleftarrow{h_{j+1}}, \bar{E}\ x_j)$$
$$\overrightarrow{h_j} = f(\overrightarrow{h_{j-1}}, \bar{E}\ x_j)$$

- Various choices for the function $f()$: feed-forward layer, GRU, LSTM, ...

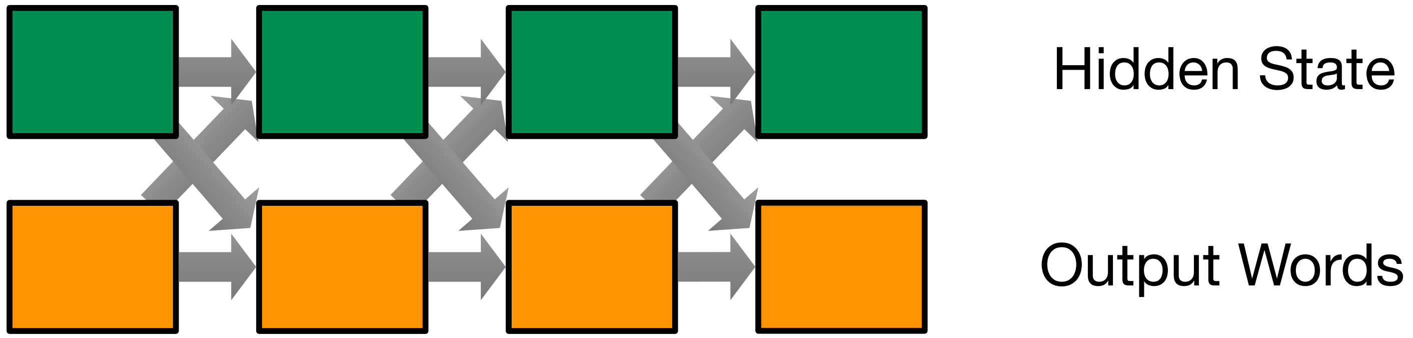
Decoder

- We want to have a recurrent neural network predicting output words



Decoder

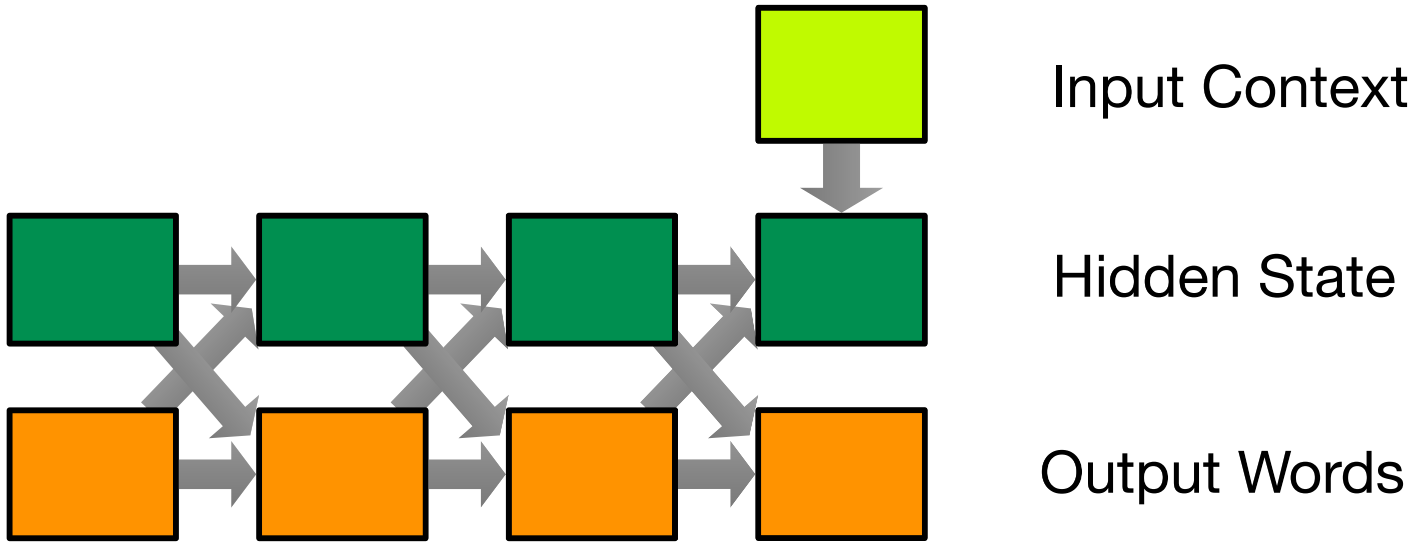
- We want to have a recurrent neural network predicting output words



- We feed decisions on output words back into the decoder state

Decoder

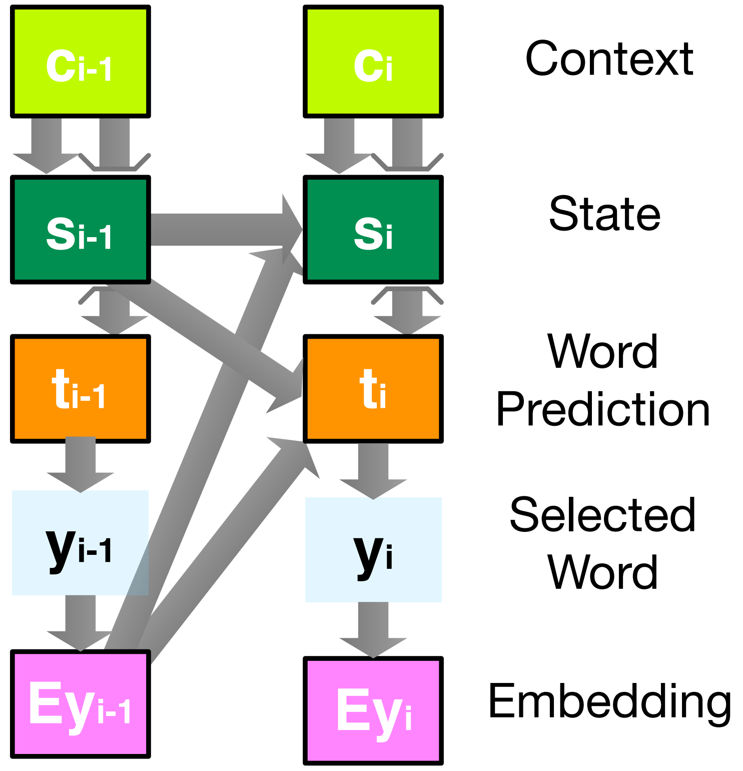
- We want to have a recurrent neural network predicting output words



- We feed decisions on output words back into the decoder state

- Decoder state is also informed by the input context

More Detail



- Decoder is also recurrent neural network over sequence of hidden states s_i

$$s_i = f(s_{i-1},\ Ey_{-1}, c_i)$$

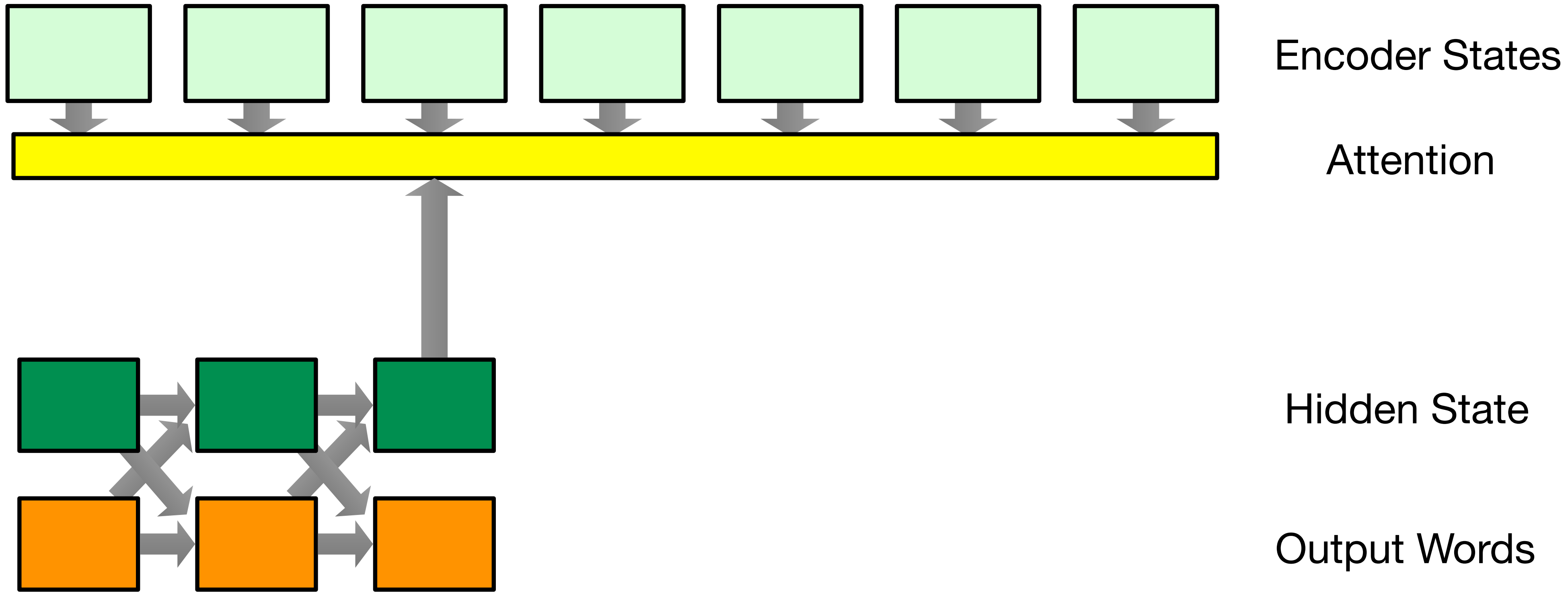
- Again, various choices for the function $f()$: feed-forward layer, GRU, LSTM, ...
- Output word y_i is selected by computing a vector t_i (same size as vocabulary)

$$t_i = W(Us_{i-1} + VEy_{i-1} + Cc_i)$$

then finding the highest value in vector t_i

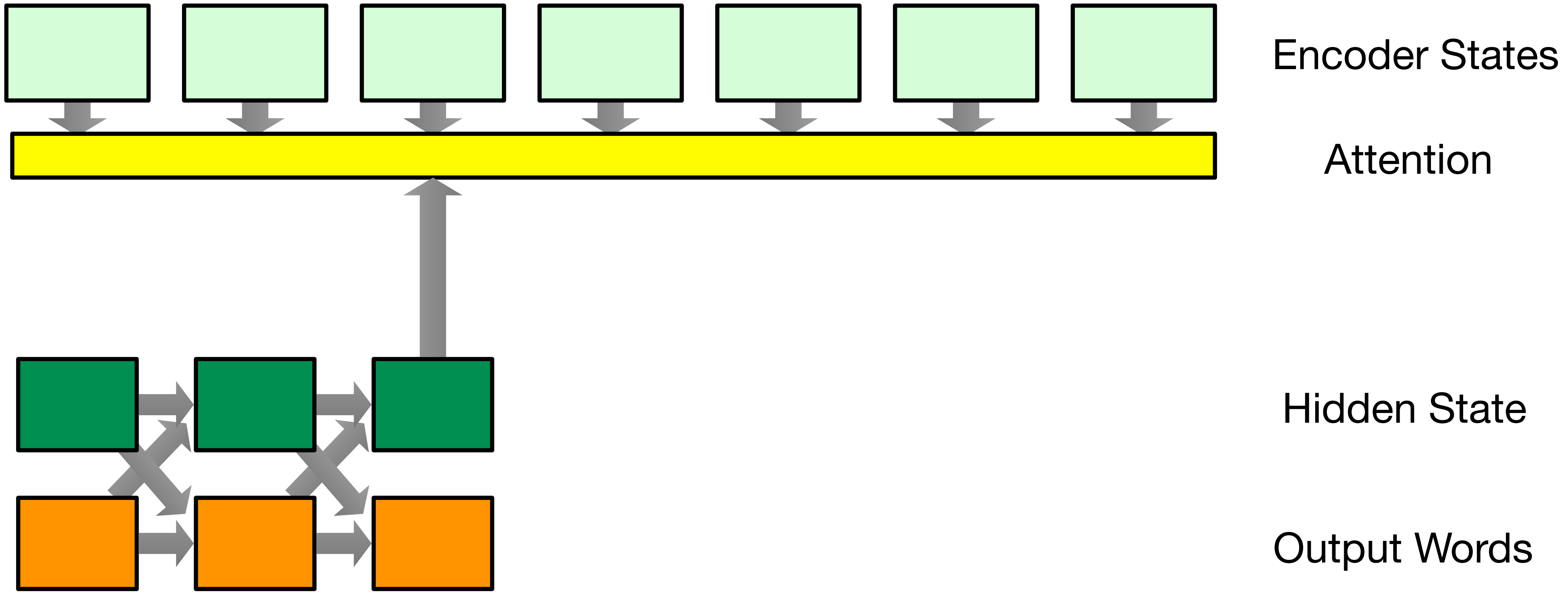
- If we normalize t_i, we can view it as a probability distribution over words
- Ey_i is the embedding of the output word y_i

Attention



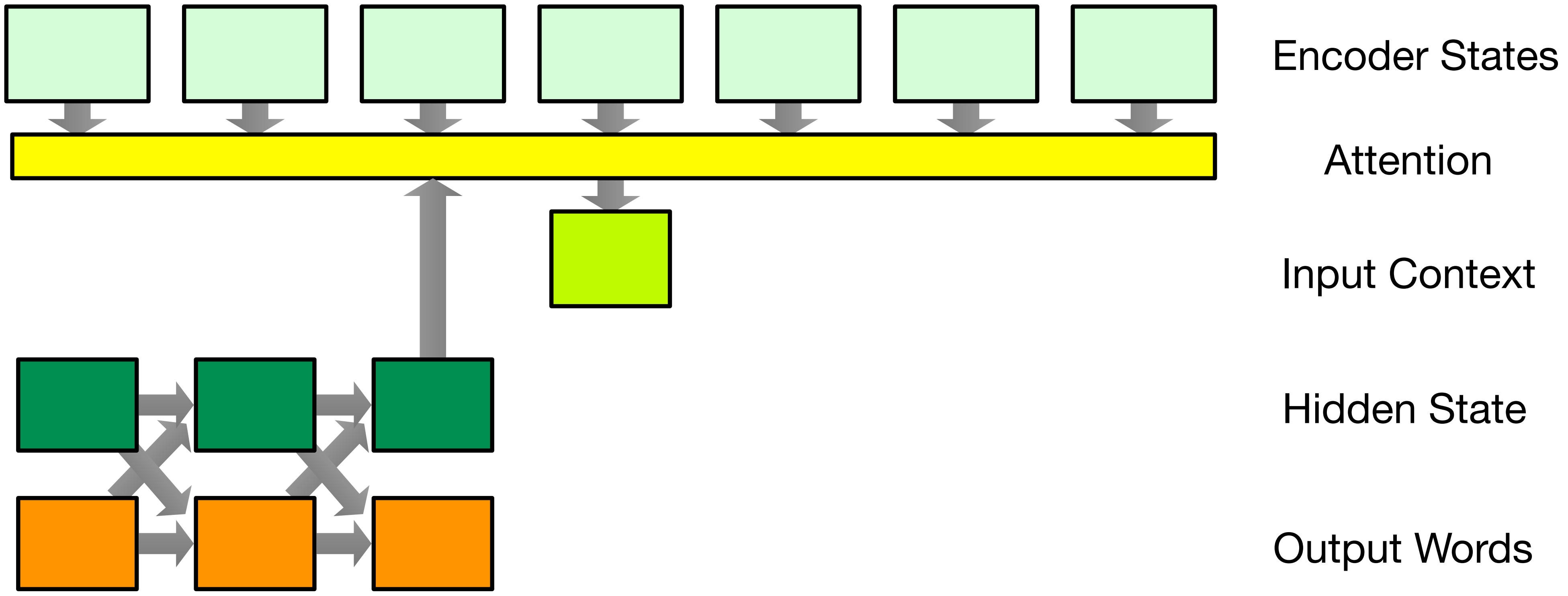
- Given what we have generated so far (decoder hidden state)
- ... which words in the input should we pay attention to (encoder states)?

Attention



- Given: – the previous hidden state of the decoder s_{i-1}
  – the representation of input words $h_j = (\overleftarrow{h_j}, \overrightarrow{h_j})$
- Predict an alignment probability $a(s_{i-1}, h_j)$ to each input word j (modeled with with a feed-forward neural network layer)

Attention

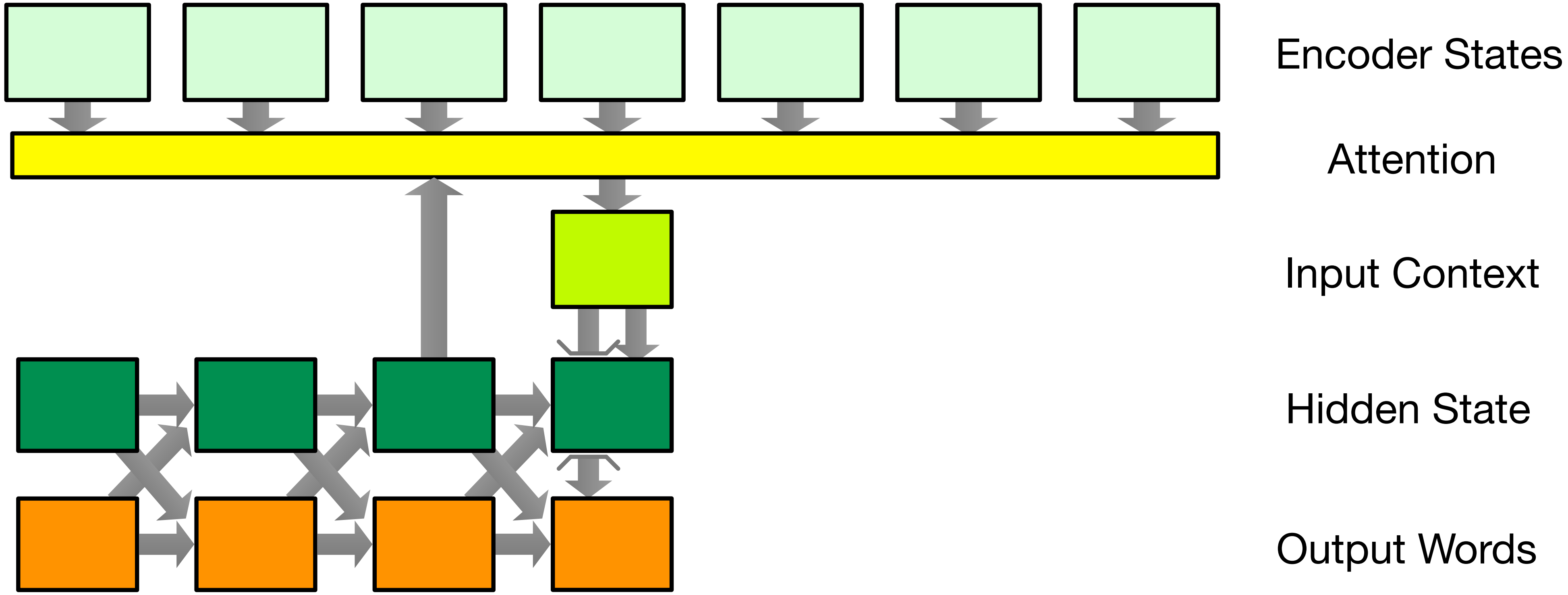


- Normalize attention (softmax)

$$\alpha_{ij} = \frac{\exp(a(s_{i-1}, h_j))}{\sum_k \exp(a(s_{i-1}, h_k))}$$

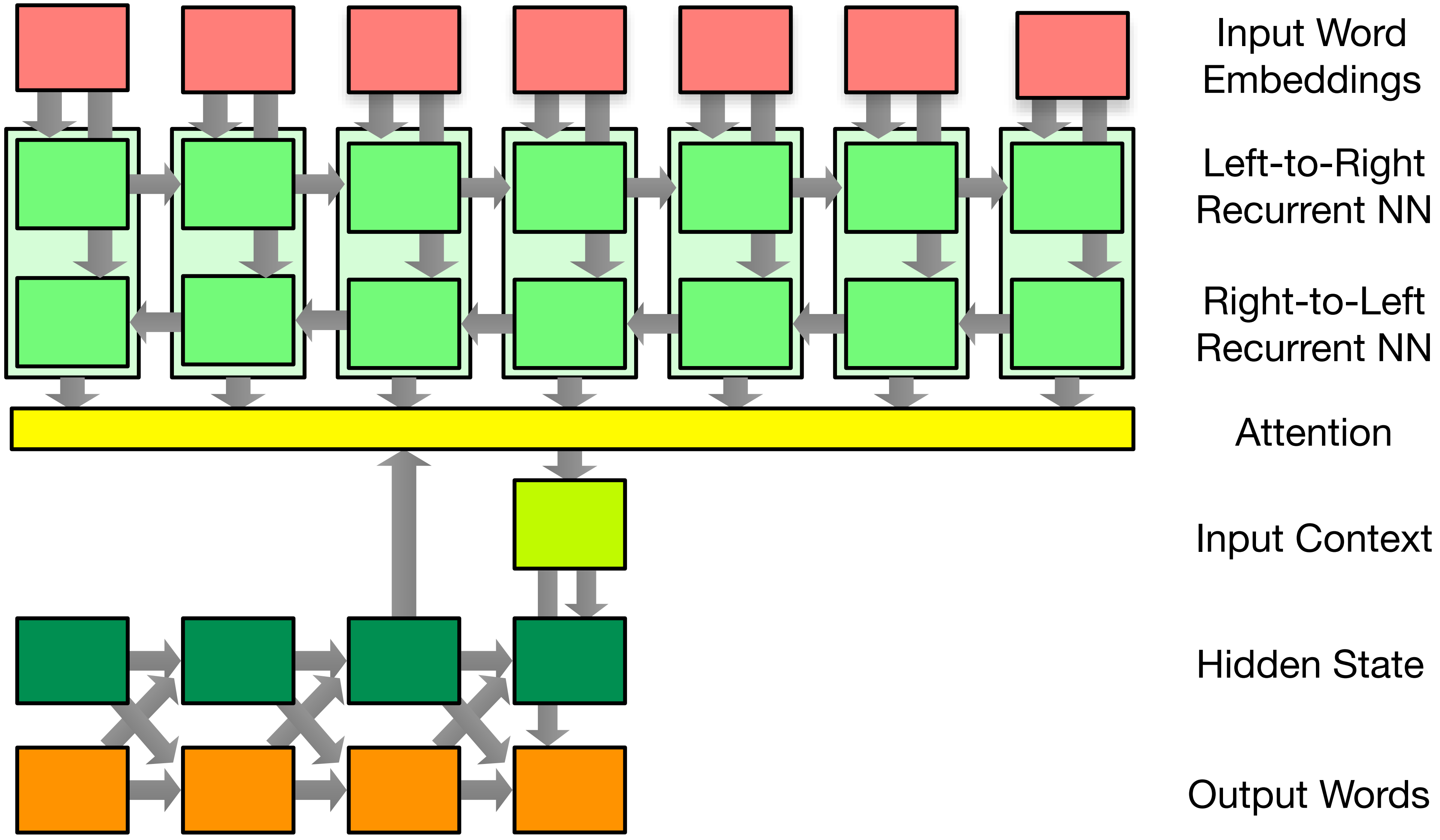
- Relevant input context: weigh input words according to attention: $c_i = \sum_j \alpha_{ij} h_j$

Attention



- Use context to predict next hidden state and output word

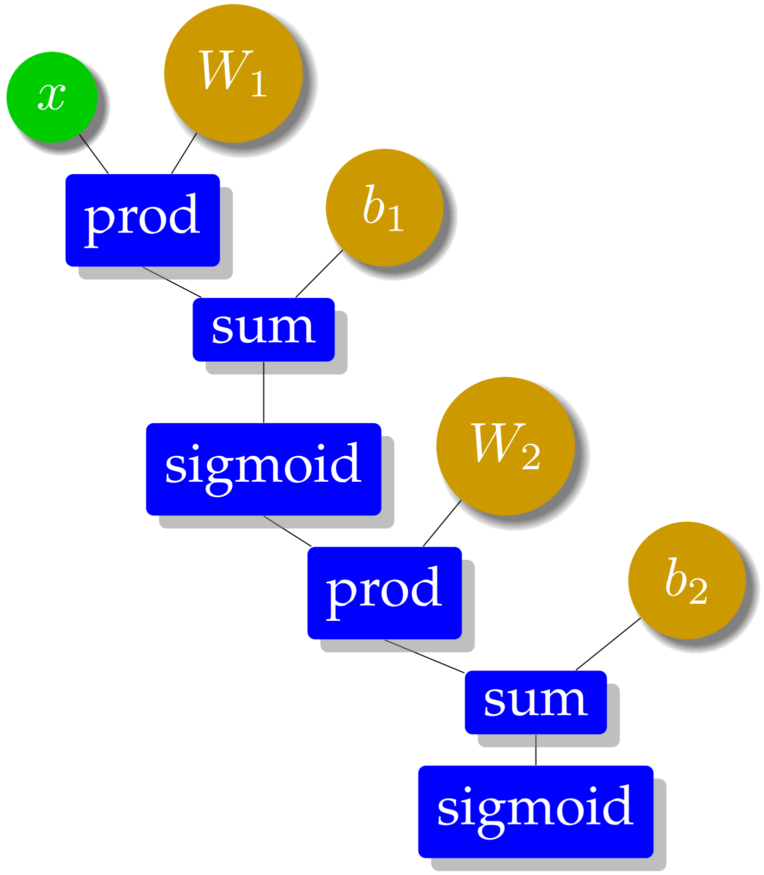
Encoder-Decoder with Attention



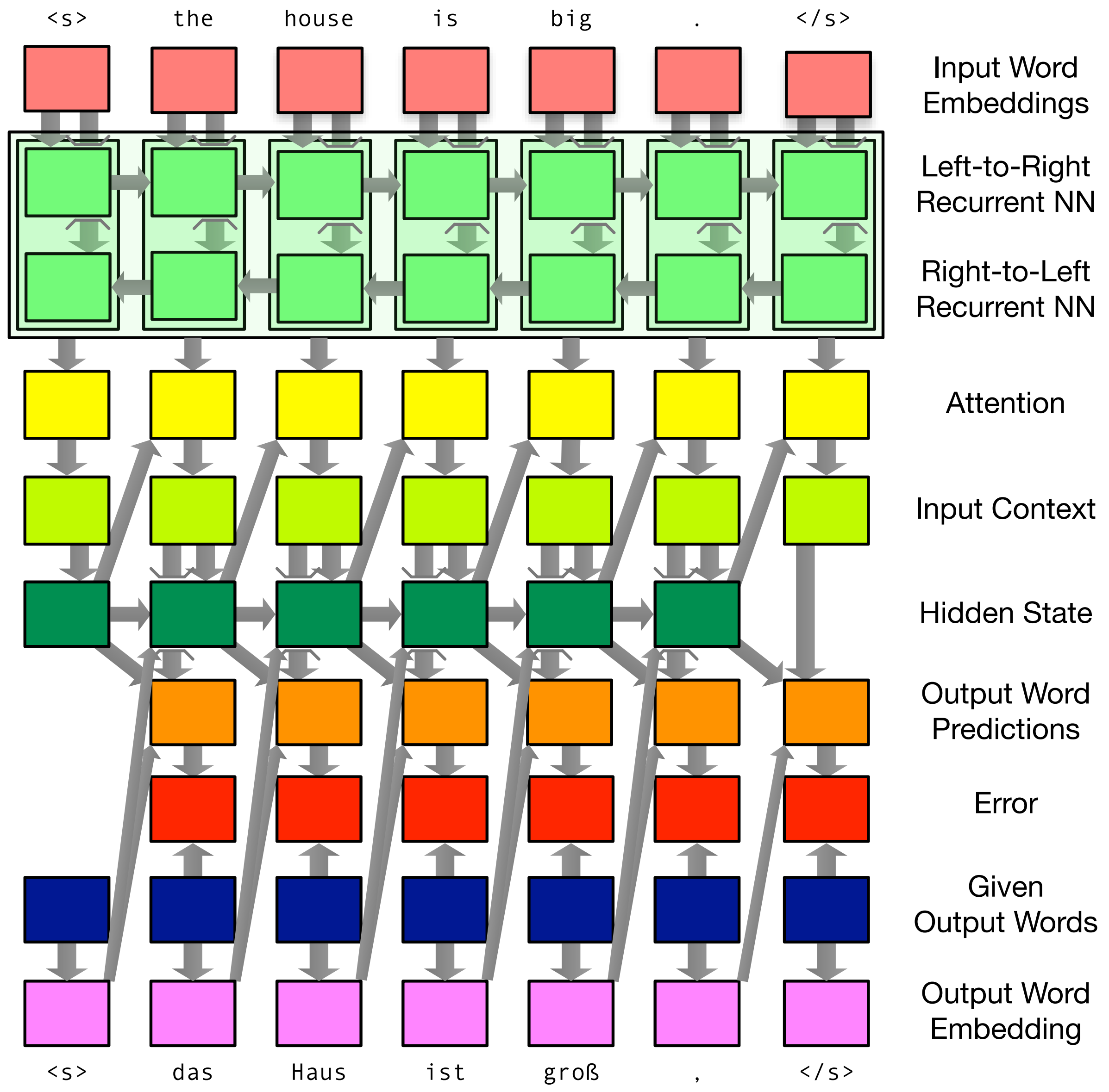
training

Computation Graph

- Math behind neural machine translation defines a computation graph
- Forward and backward computation to compute gradients for model training



Unrolled Computation Graph

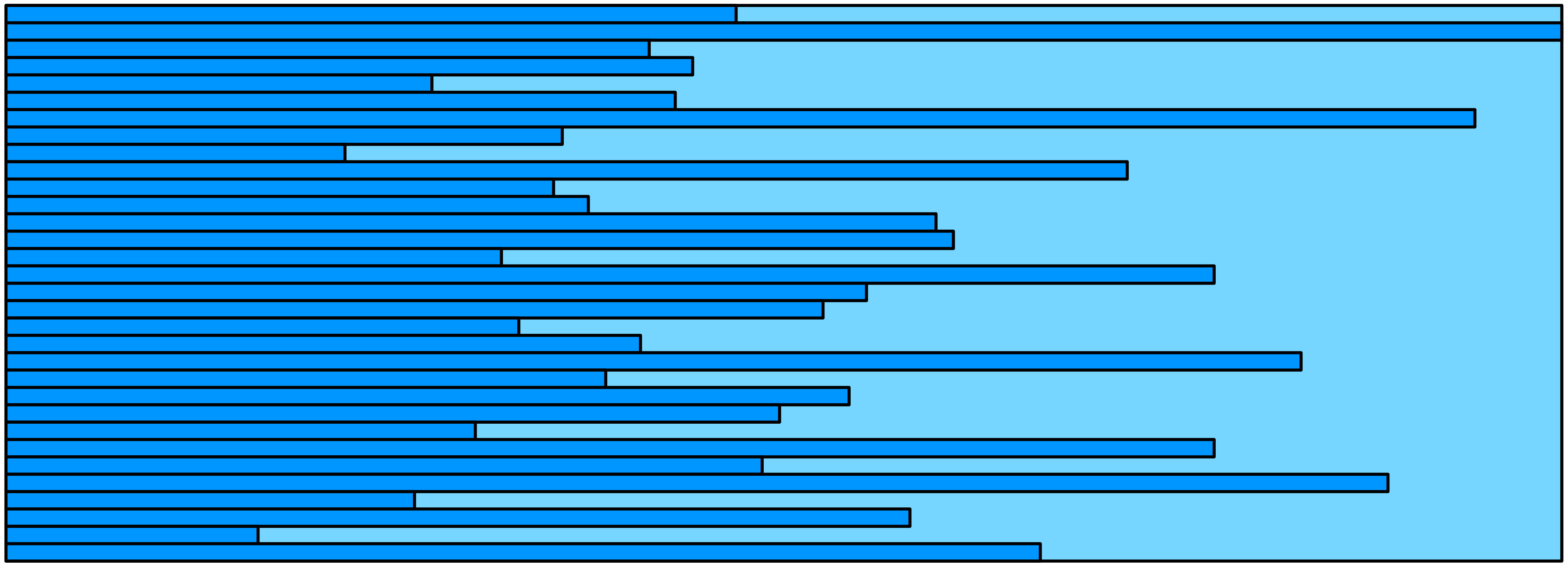


Batching

- Already large degree of parallelism
  - most computations on vectors, matrices
  - efficient implementations for CPU and GPU
- Further parallelism by batching
  - processing several sentence pairs at once
  - scalar operation $\rightarrow$ vector operation
  - vector operation $\rightarrow$ matrix operation
  - matrix operation $\rightarrow$ 3d tensor operation
- Typical batch sizes 50–100 sentence pairs

Batches

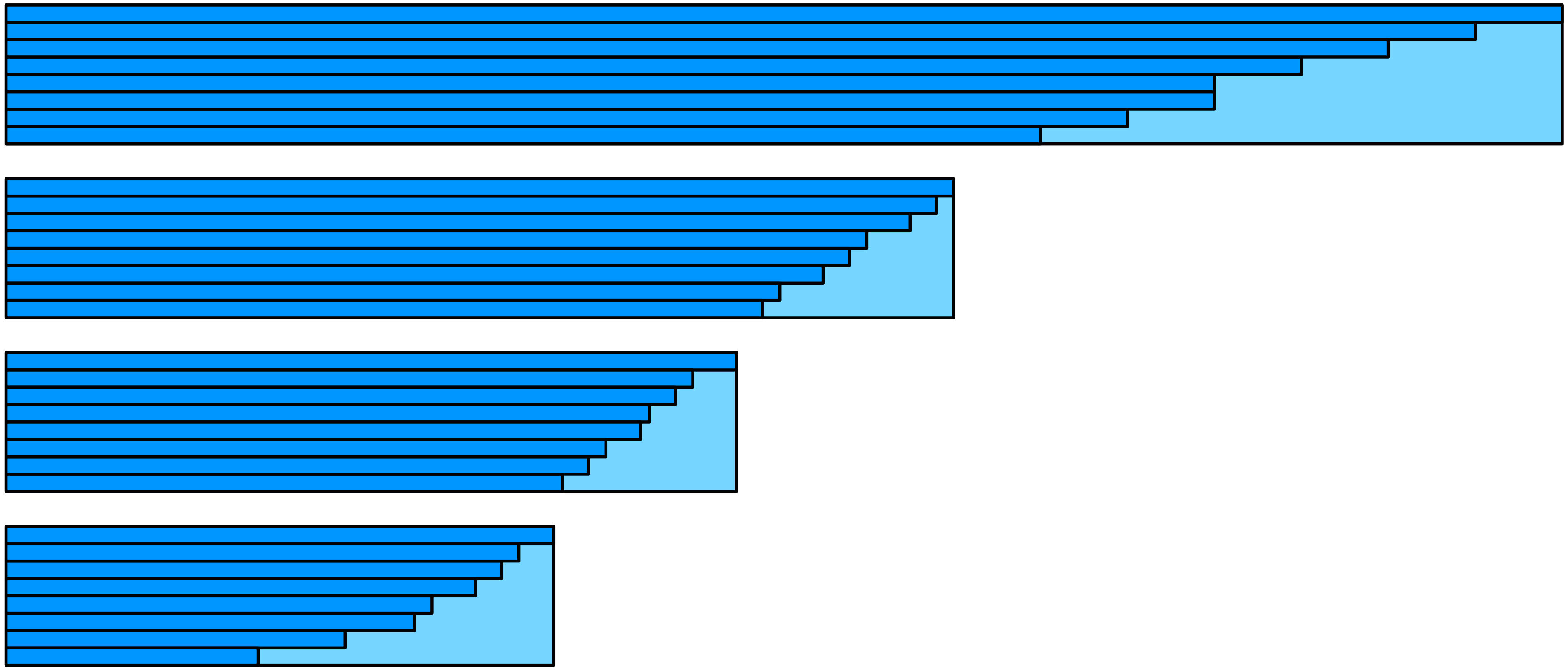
- Sentences have different length
- When batching, fill up unneeded cells in tensors



⇒ A lot of wasted computations

Mini-Batches

- Sort sentences by length, break up into mini-batches



- Example: Maxi-batch 1600 sentence pairs, mini-batch 80 sentence pairs

Overall Organization of Training

- Shuffle corpus
- Break into maxi-batches
- Break up each maxi-batch into mini-batches
- Process mini-batch, update parameters
- Once done, repeat
- Typically 5-15 epochs needed (passes through entire training corpus)

inference

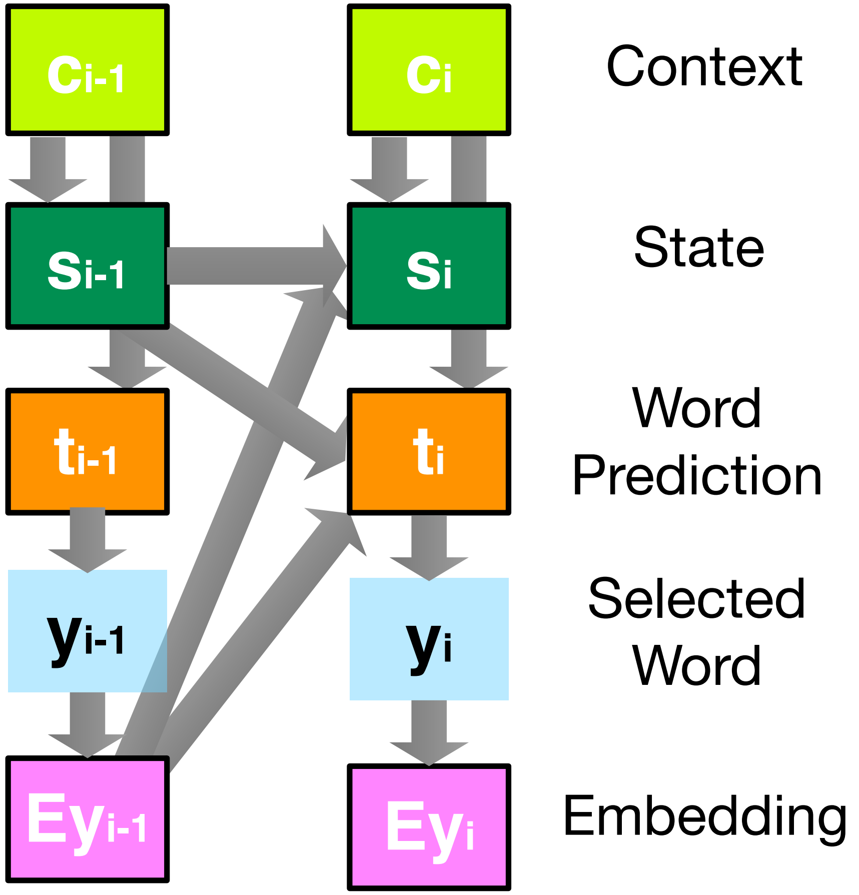
Inference

- Given a trained model

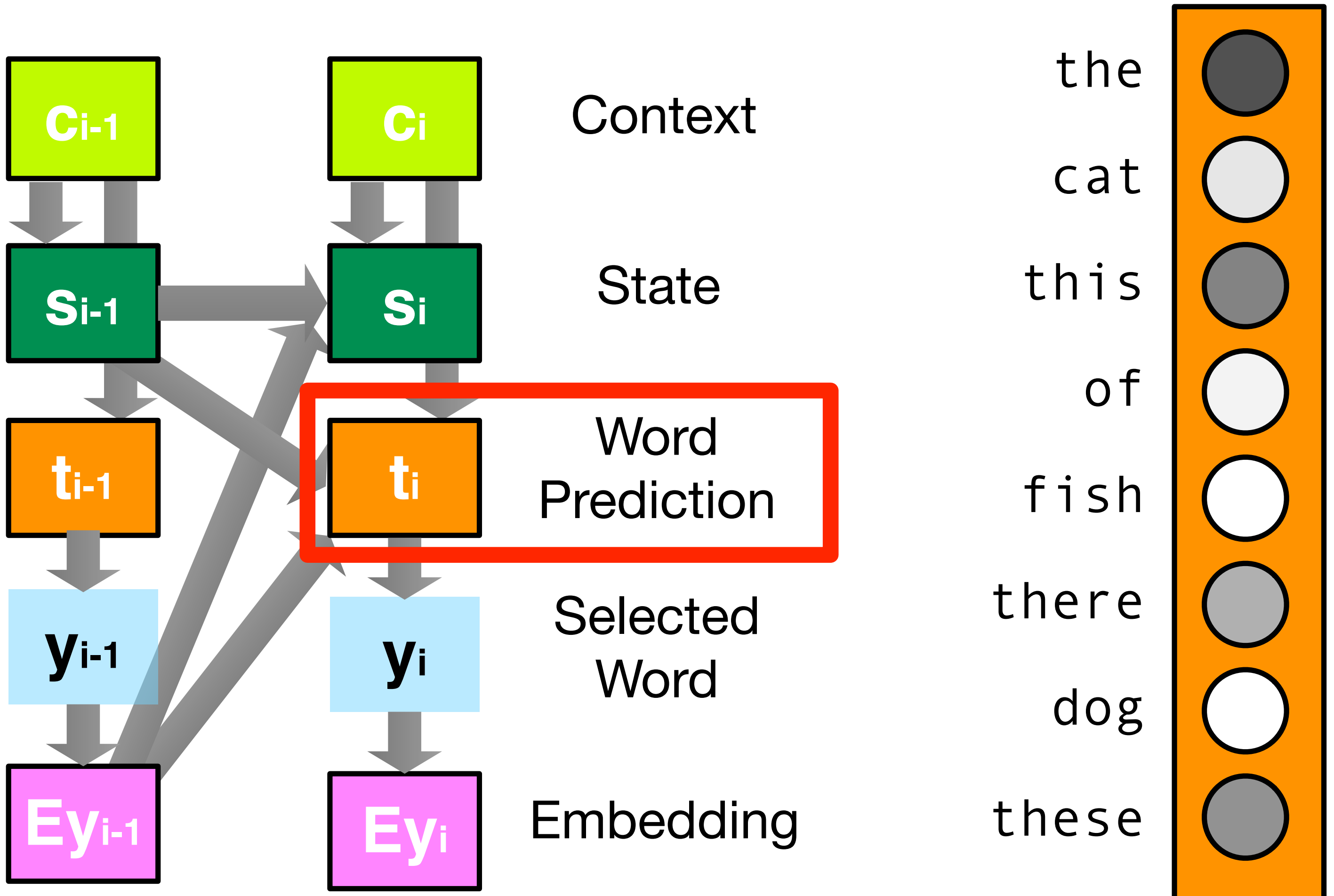
  ... we now want to translate test sentences

- We only need execute the ”forward” step in the computation graph

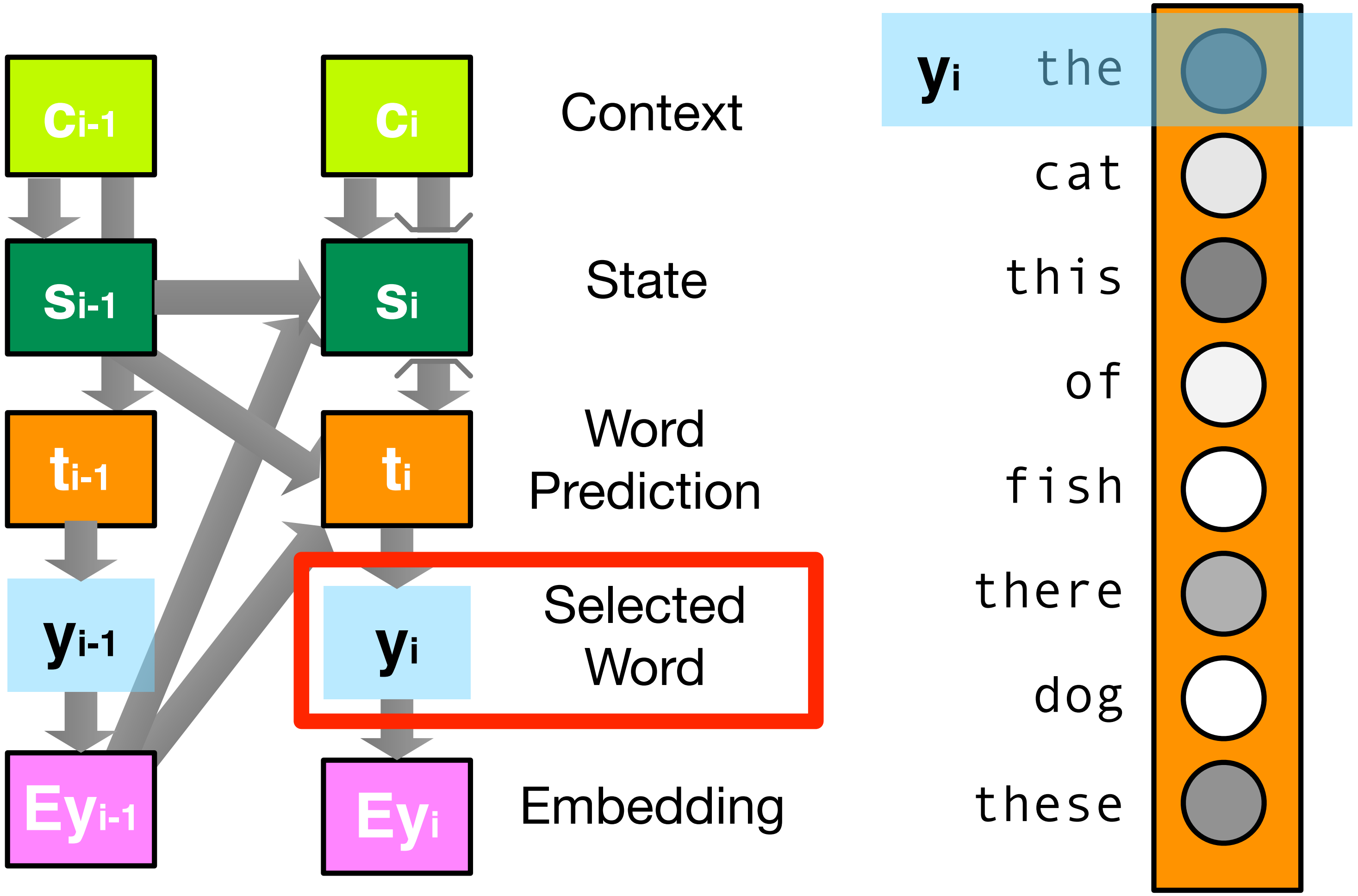
Word Prediction



Selected Word

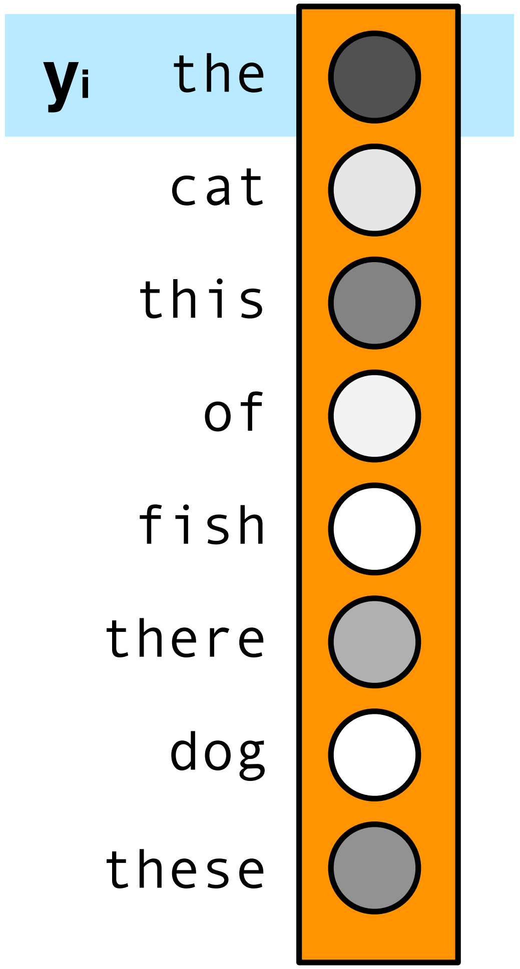


Embedding

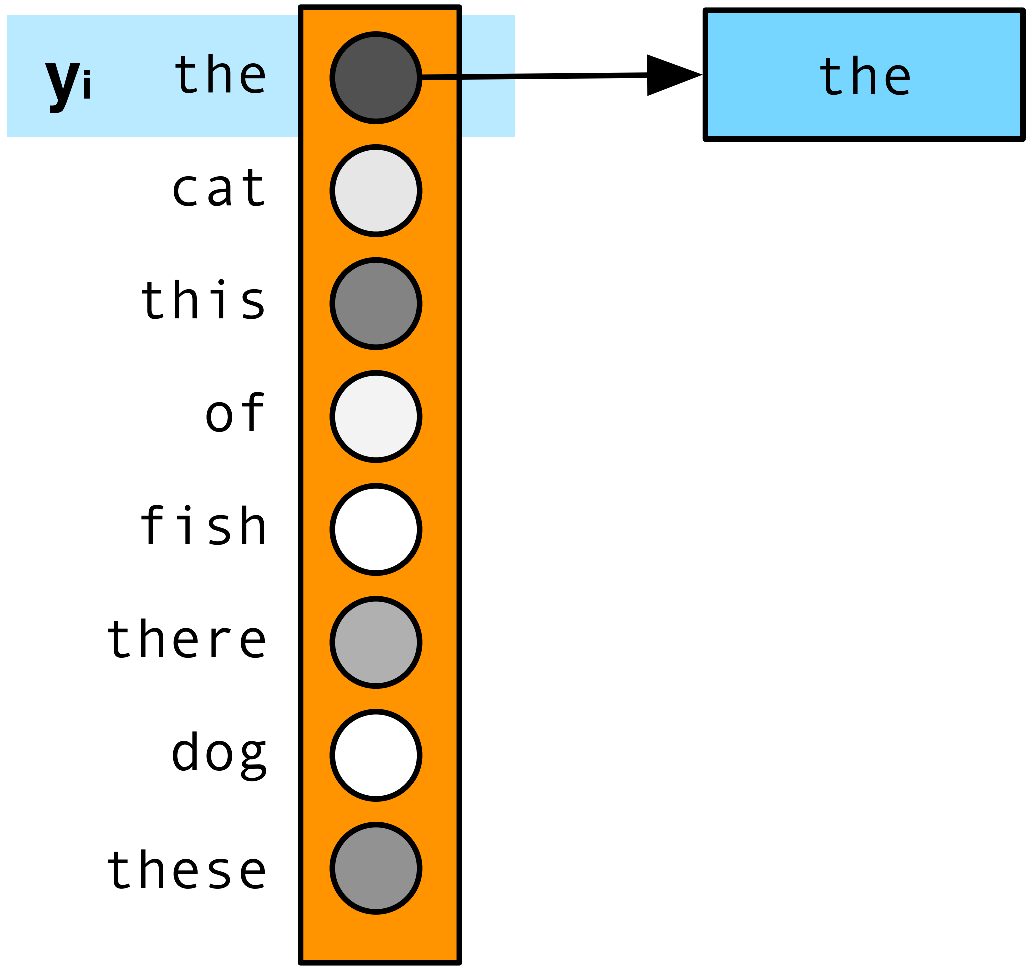


c_{i-1}
c_i
Context
s_{i-1}
s_i
State
t_{i-1}
t_i
Word Prediction
y_{i-1}
y_i
Selected Word
Ey_{i-1}
Ey_i
Embedding
y_i
the
cat
this
of
fish
there
dog
these

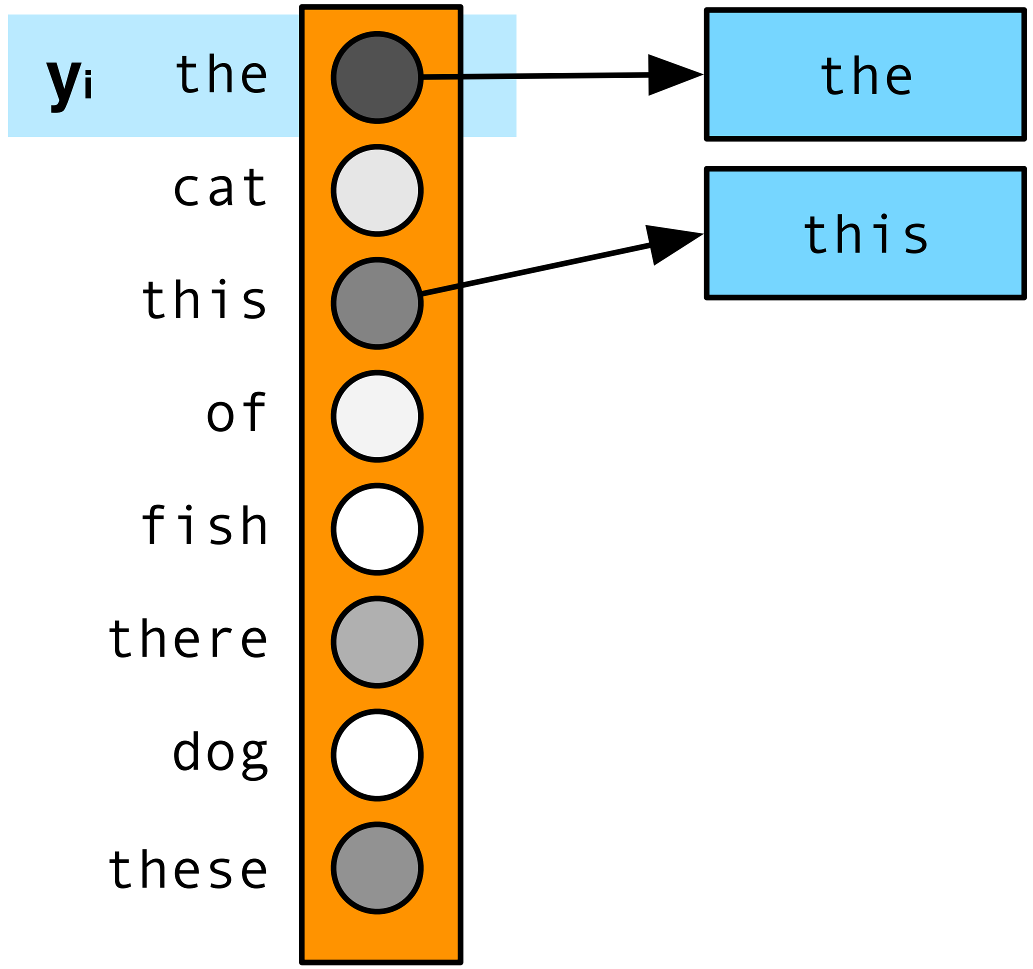
Distribution of Word Predictions



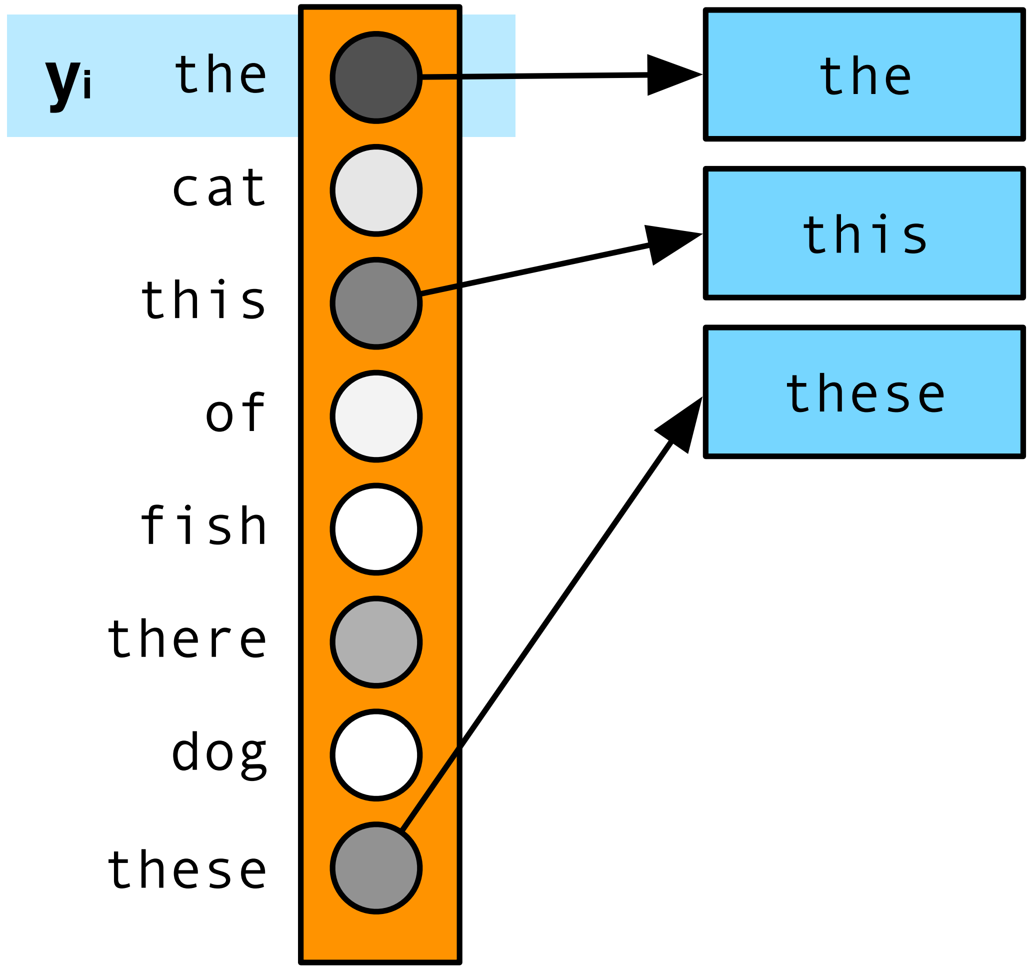
Select Best Word



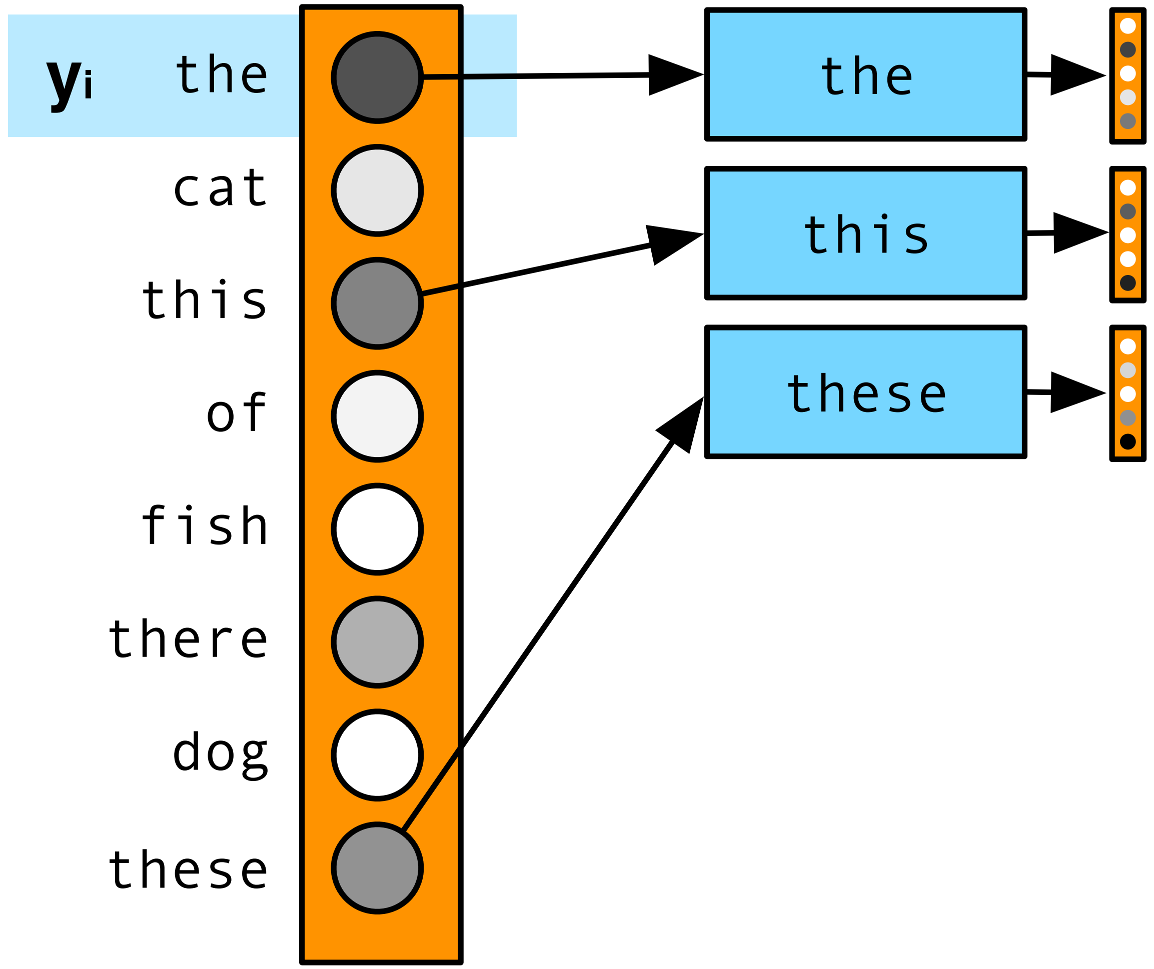
Select Second Best Word



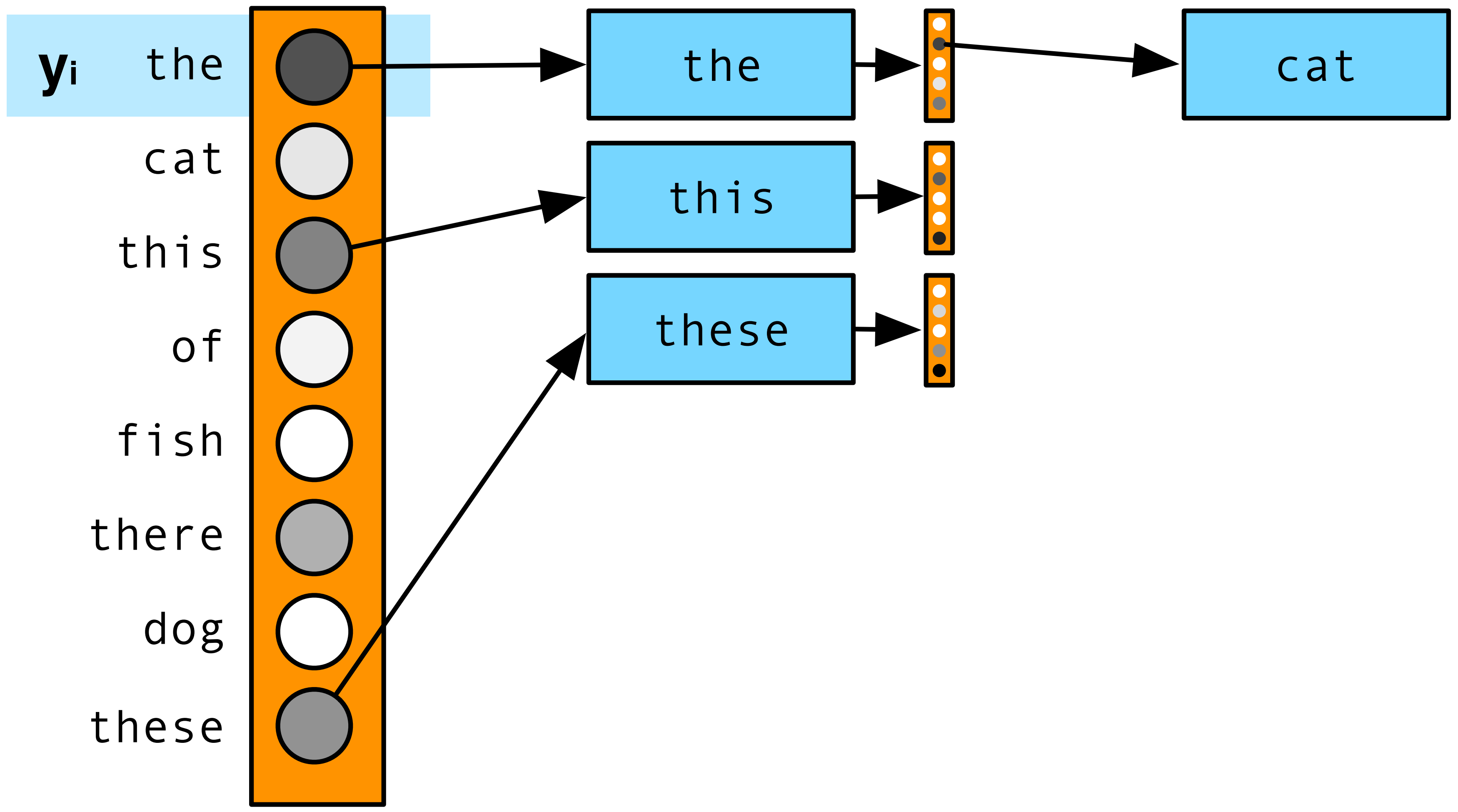
Select Third Best Word



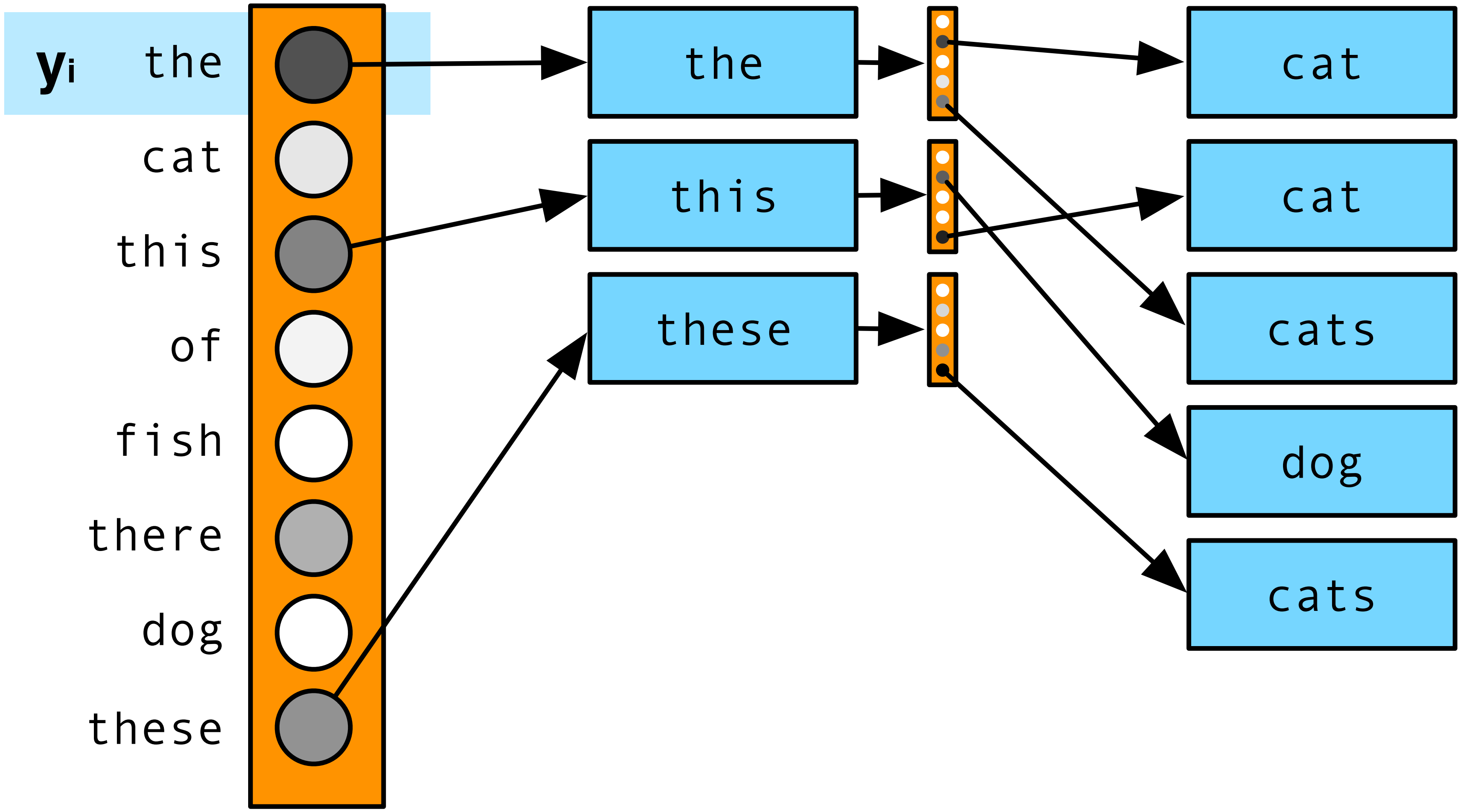
Use Selected Word for Next Predictions



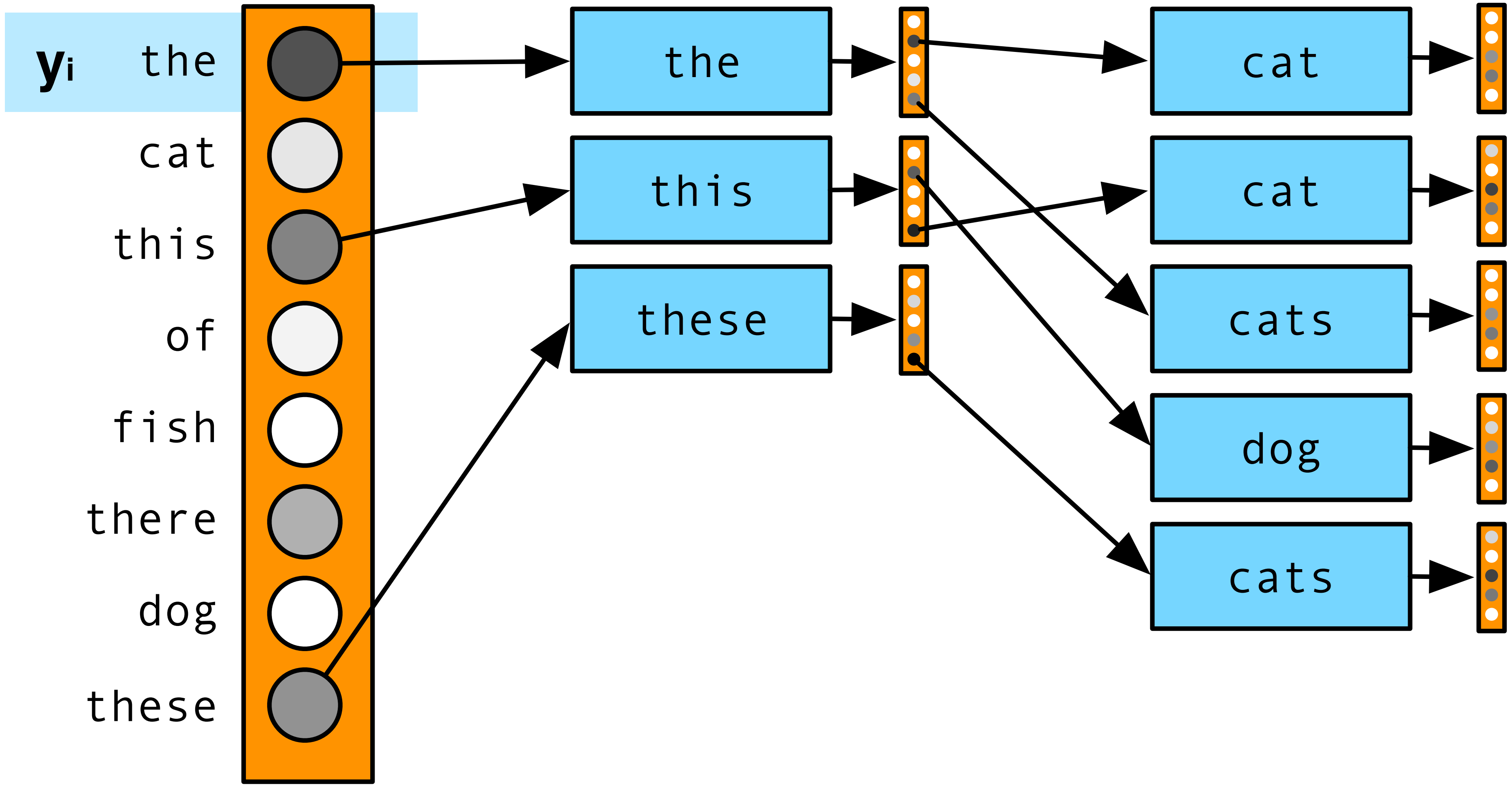
Select Best Continuation



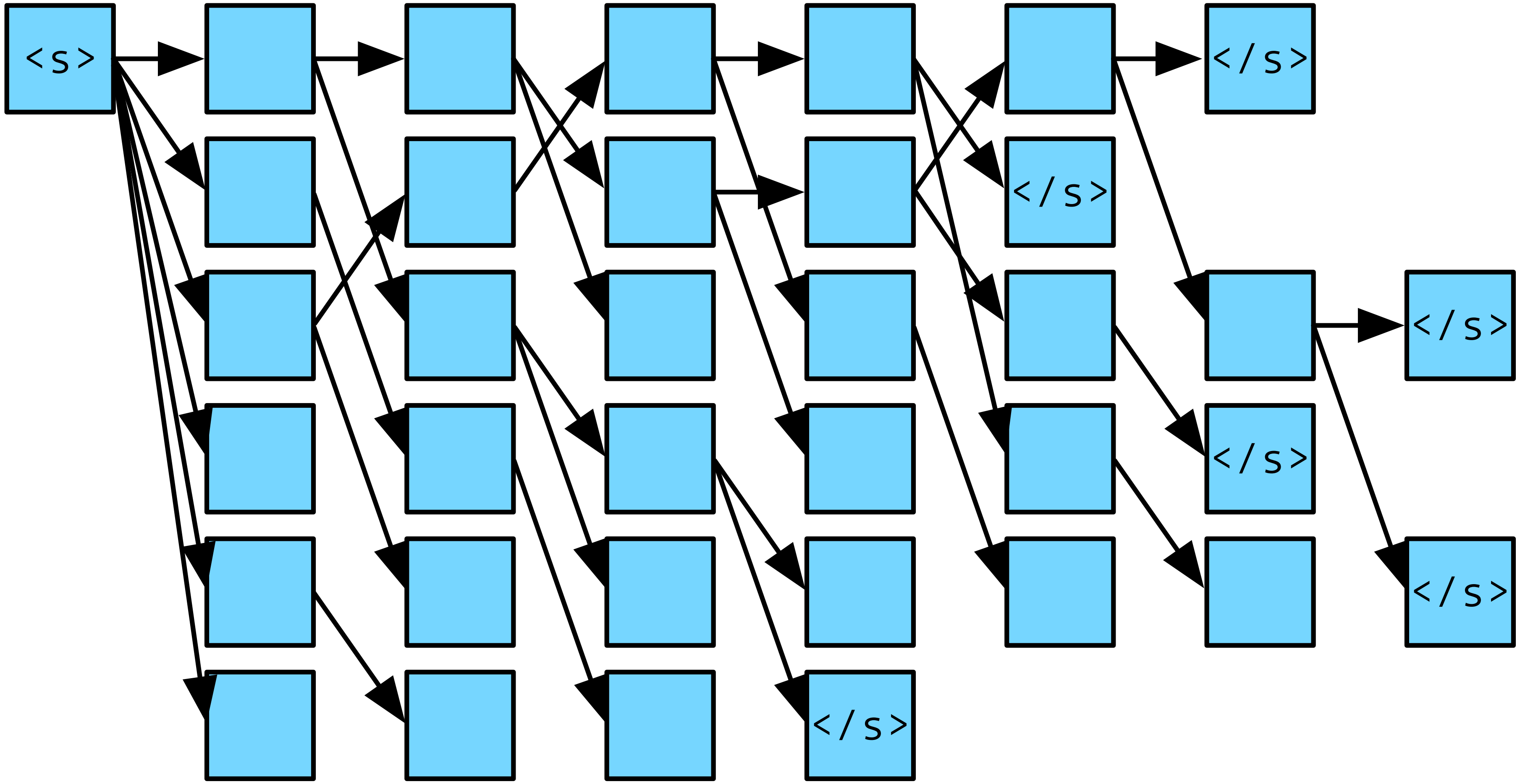
Select Next Best Continuations



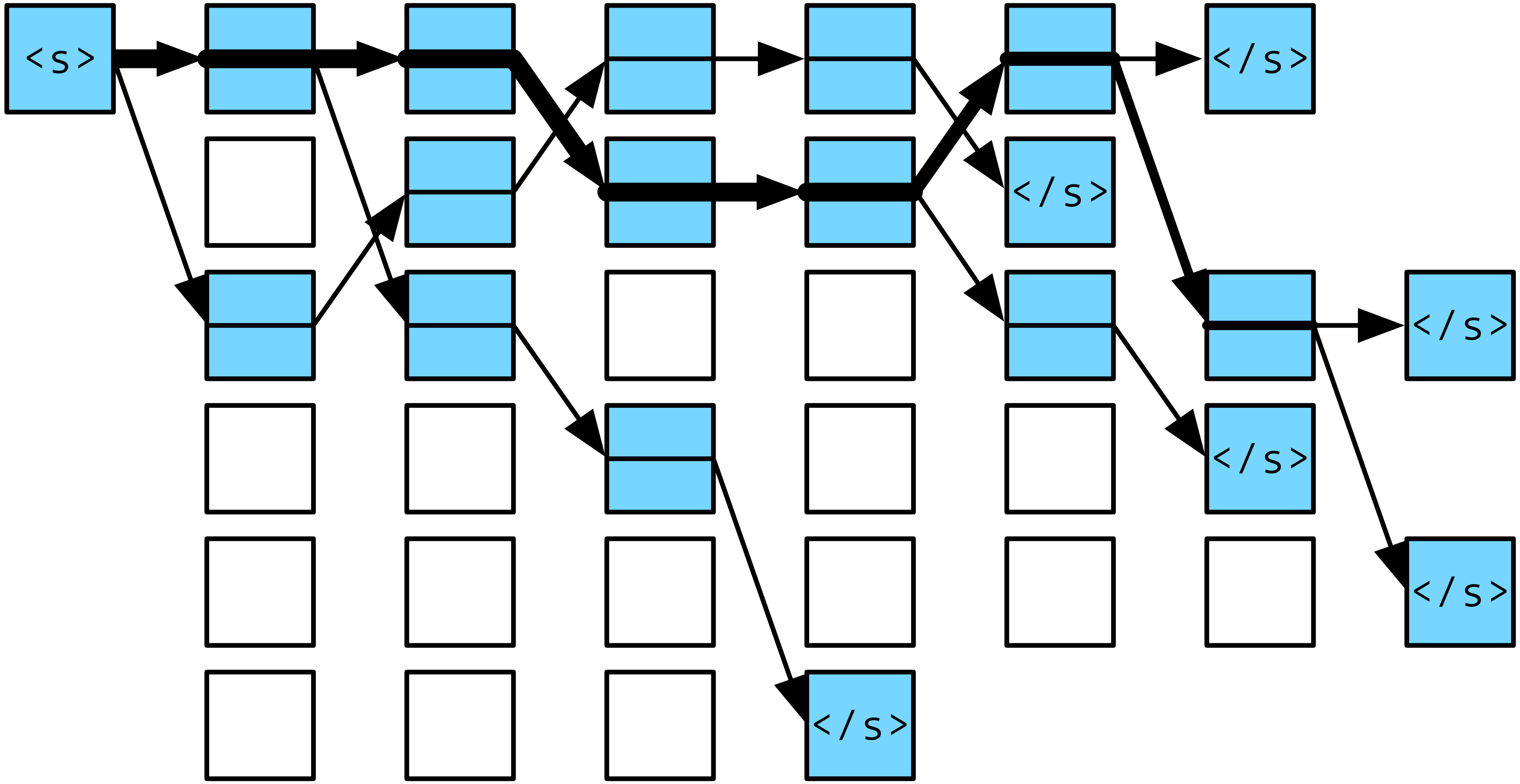
Continue...


y_i
the
cat
this
of
fish
there
dog
these
the
this
these
cat
cat
cats
dog
cats

Beam Search



Best Paths



Beam Search Details

- Normalize score by length

- No recombination (paths cannot be merged)

Output Word Predictions

Input Sentence: *ich glaube aber auch , er ist clever genug um seine Aussagen vage genug zu halten , so dass sie auf verschiedene Art und Weise interpretiert werden können .*

Best		**Alternatives**
but	(42.1%)	*however* (25.3%), *I* (20.4%), *yet* (1.9%), *and* (0.8%), *nor* (0.8%), ...
I	(80.4%)	*also* (6.0%), *,* (4.7%), *it* (1.2%), *in* (0.7%), *nor* (0.5%), *he* (0.4%), ...
also	(85.2%)	*think* (4.2%), *do* (3.1%), *believe* (2.9%), *,* (0.8%), *too* (0.5%), ...
believe	(68.4%)	*think* (28.6%), *feel* (1.6%), *do* (0.8%), ...
he	(90.4%)	*that* (6.7%), *it* (2.2%), *him* (0.2%), ...
is	(74.7%)	*'s* (24.4%), *has* (0.3%), *was* (0.1%), ...
clever	(99.1%)	*smart* (0.6%), ...
enough	(99.9%)	
to	(95.5%)	*about* (1.2%), *for* (1.1%), *in* (1.0%), *of* (0.3%), *around* (0.1%), ...
keep	(69.8%)	*maintain* (4.5%), *hold* (4.4%), *be* (4.2%), *have* (1.1%), *make* (1.0%), ...
his	(86.2%)	*its* (2.1%), *statements* (1.5%), *what* (1.0%), *out* (0.6%), *the* (0.6%), ...
statements	(91.9%)	*testimony* (1.5%), *messages* (0.7%), *comments* (0.6%), ...
vague	(96.2%)	*v@@* (1.2%), *in* (0.6%), *ambiguous* (0.3%), ...
enough	(98.9%)	*and* (0.2%), ...
so	(51.1%)	*,* (44.3%), *to* (1.2%), *in* (0.6%), *and* (0.5%), *just* (0.2%), *that* (0.2%), ...
they	(55.2%)	*that* (35.3%), *it* (2.5%), *can* (1.6%), *you* (0.8%), *we* (0.4%), *to* (0.3%), ...
can	(93.2%)	*may* (2.7%), *could* (1.6%), *are* (0.8%), *will* (0.6%), *might* (0.5%), ...
be	(98.4%)	*have* (0.3%), *interpret* (0.2%), *get* (0.2%), ...
interpreted	(99.1%)	*interpre@@* (0.1%), *constru@@* (0.1%), ...
in	(96.5%)	*on* (0.9%), *differently* (0.5%), *as* (0.3%), *to* (0.2%), *for* (0.2%), *by* (0.1%), ...
different	(41.5%)	*a* (25.2%), *various* (22.7%), *several* (3.6%), *ways* (2.4%), *some* (1.7%), ...
ways	(99.3%)	*way* (0.2%), *manner* (0.2%), ...
.	(99.2%)	*</S>* (0.2%), *,* (0.1%), ...
</s>	(100.0%)	

refinements

Refinements

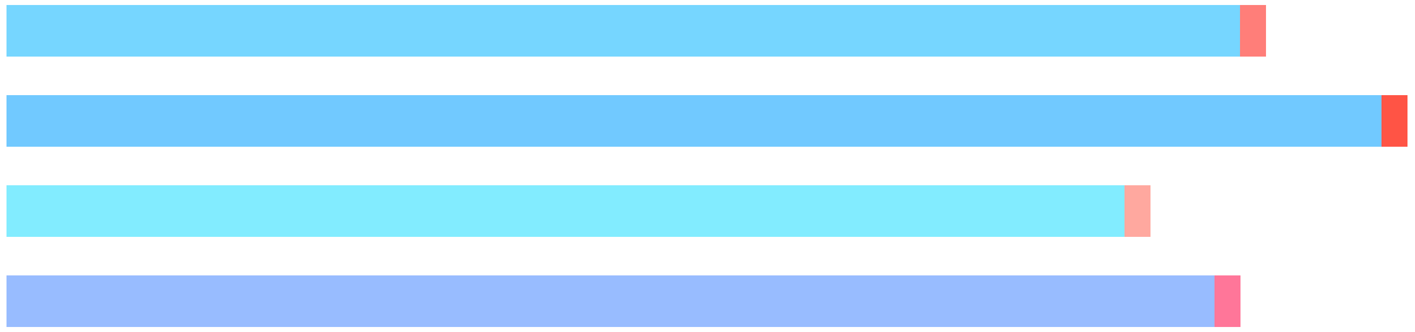
- Last lecture: architecture of attentional sequence-to-sequence neural model
- Today: practical considerations and refinements
  - ensembling
  - handling large vocabularies
  - using monolingual data
  - deep models
  - alignment and coverage
  - use of linguistic annotation
  - multiple language pairs

ensembling

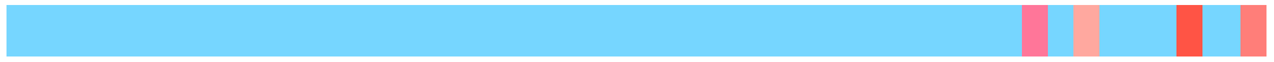
Ensembling

- Train multiple models

- Say, by different random initializations

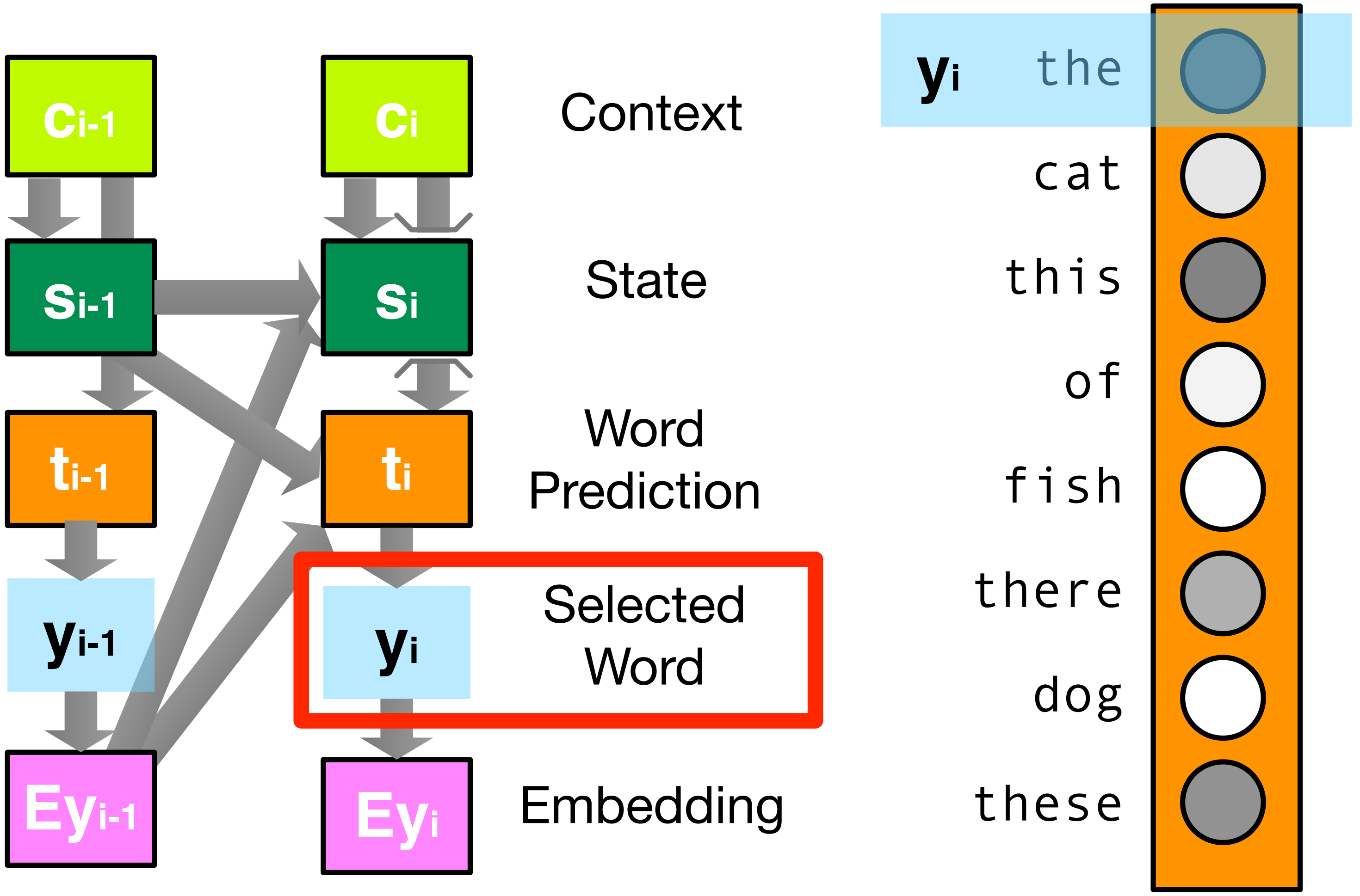


- Or, by using model dumps from earlier iterations

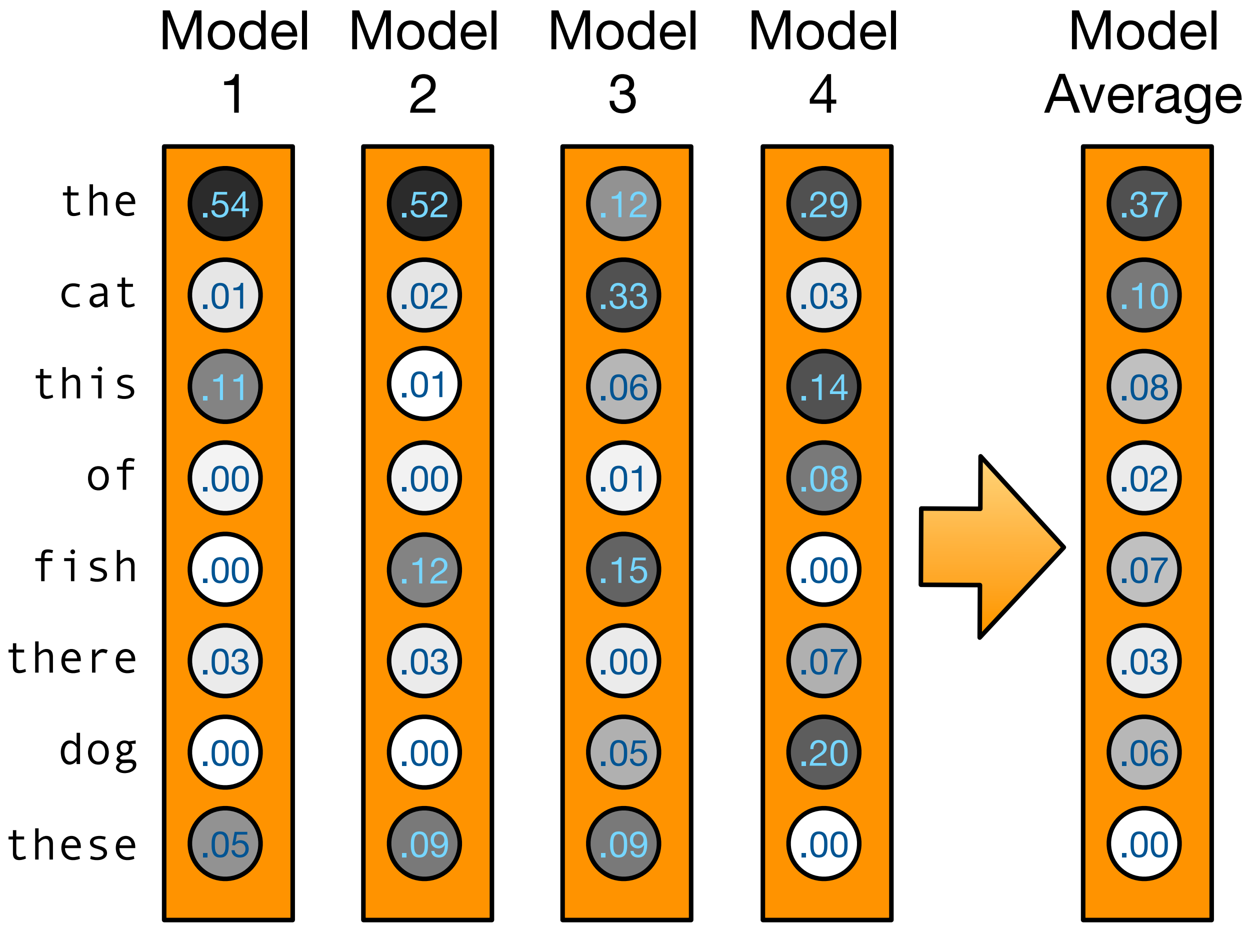


(most recent, or interim models with highest validation score)

Decoding with Single Model



Combine Predictions



Ensembling

- Surprisingly reliable method in machine learning

- Long history, many variants:
  bagging, ensemble, model averaging, system combination, ...

- Works because errors are random, but correct decisions unique

Right-to-Left Inference

- Neural machine translation generates words right to left (L2R)

$$\text{the} \rightarrow \text{cat} \rightarrow \text{is} \rightarrow \text{in} \rightarrow \text{the} \rightarrow \text{bag} \rightarrow .$$

- But it could also generate them right to left (R2L)

$$\text{the} \leftarrow \text{cat} \leftarrow \text{is} \leftarrow \text{in} \leftarrow \text{the} \leftarrow \text{bag} \leftarrow .$$

Obligatory notice: Some languages (Arabic, Hebrew, ...) have writing systems that are right-to-left, so the use of ”right-to-left” is not precise here.

Right-to-Left Reranking

- Train both L2R and R2L model

- Score sentences with both

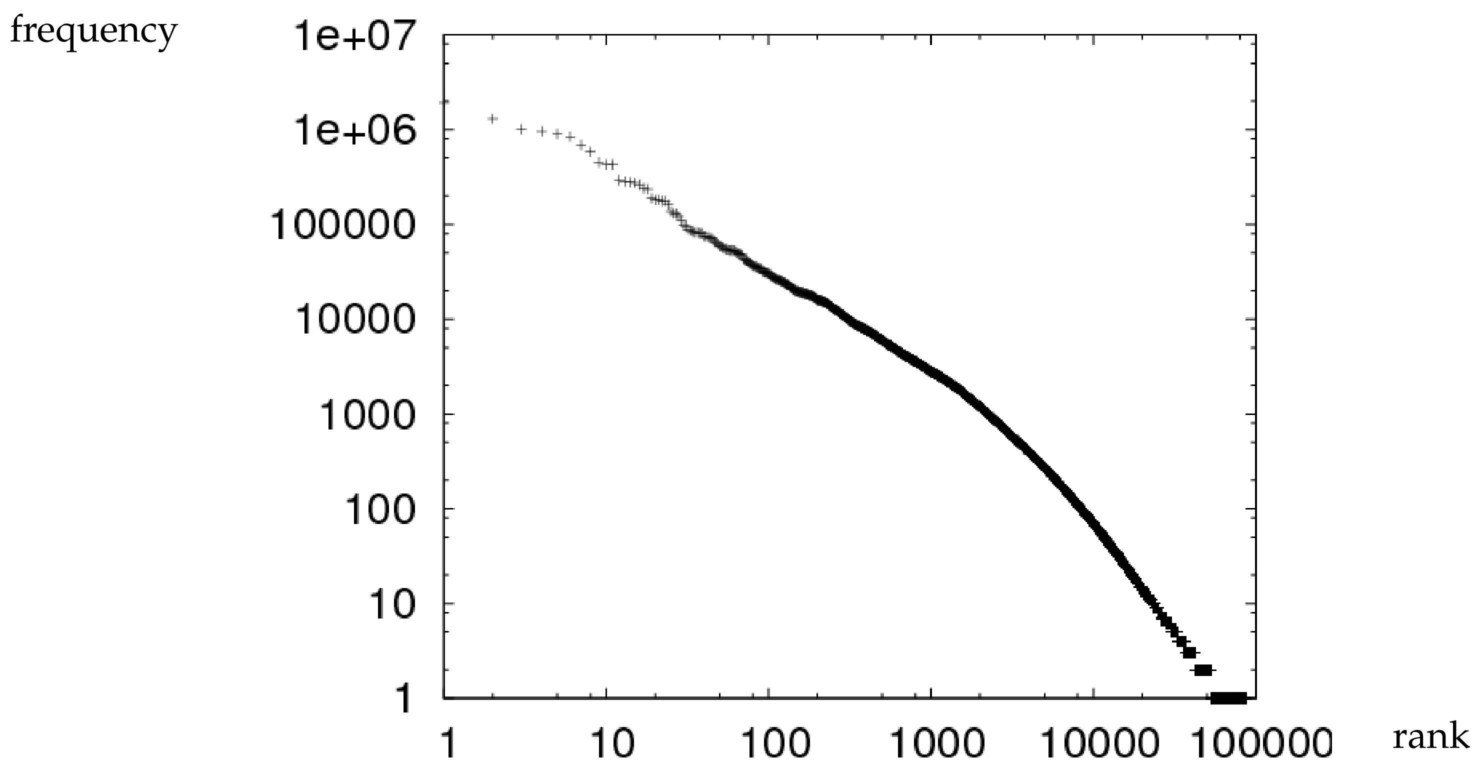
  ⇒ use both left and right context during translation

- Only possible once full sentence produced → re-ranking

  1. generate n-best list with L2R model
  2. score candidates in n-best list with R2L model
  3. chose translation with best average score

large vocabularies

Zipf's Law: Many Rare Words



frequency × rank = constant

Many Problems

- Sparse data
  - words that occur once or twice have unreliable statistics

- Computation cost
  - input word embedding matrix: $|V| \times 1000$
  - outout word prediction matrix: $1000 \times |V|$

Some Causes for Large Vocabularies

- Morphology

  tweet, tweets, tweeted, tweeting, retweet, ...

  → morphological analysis?

- Compounding

  homework, website, ...

  → compound splitting?

- Names

  Netanyahu, Jones, Macron, Hoboken, ...

  → transliteration?

⇒ Breaking up words into **subwords** may be a good idea

Byte Pair Encoding

- Start by breaking up words into characters

```
t h e ␣ f a t ␣ c a t ␣ i s ␣ i n ␣ t h e ␣ t h i n ␣ b a g
```

- Merge frequent pairs

```
t h→th     th e ␣ f a t ␣ c a t ␣ i s ␣ i n ␣ th e ␣ th i n ␣ b a g
a t→at     th e ␣ f at ␣ c at ␣ i s ␣ i n ␣ th e ␣ th i n ␣ b a g
i n→in     th e ␣ f at ␣ c at ␣ i s ␣ in ␣ th e ␣ th in ␣ b a g
th e→the   the ␣ f at ␣ c at ␣ i s ␣ in ␣ the ␣ th in ␣ b a g
```

- Each merge operation increases the vocabulary size
  - starting with the size of the character set (maybe 100 for Latin script)
  - stopping at, say, 50,000

Example: 49,500 BPE Operations

```
Obama receives Net@@ any@@ ahu

the relationship between Obama and Net@@ any@@ ahu is not exactly
friendly .  the two wanted to talk about the implementation of the
international agreement and about Teheran ’s destabil@@ ising activities
in the Middle East .  the meeting was also planned to cover the conflict
with the Palestinians and the disputed two state solution .  relations
between Obama and Net@@ any@@ ahu have been stra@@ ined for years .
Washington critic@@ ises the continuous building of settlements in
Israel and acc@@ uses Net@@ any@@ ahu of a lack of initiative in the
peace process .  the relationship between the two has further
deteriorated because of the deal that Obama negotiated on Iran ’s
atomic programme .  in March , at the invitation of the Republic@@ ans
, Net@@ any@@ ahu made a controversial speech to the US Congress , which
was partly seen as an aff@@ ront to Obama .  the speech had not been
agreed with Obama , who had rejected a meeting with reference to the
election that was at that time im@@ pending in Israel .
```

using monolingual data

Traditional View

- Two core objectives for translation

Adequacy	**Fluency**
meaning of source and target match	target is well-formed
translation model	language model
parallel data	monolingual data

- Language model is key to good performance in statistical models
- But: current neural translation models only trained on parallel data

Integrating a Language Model

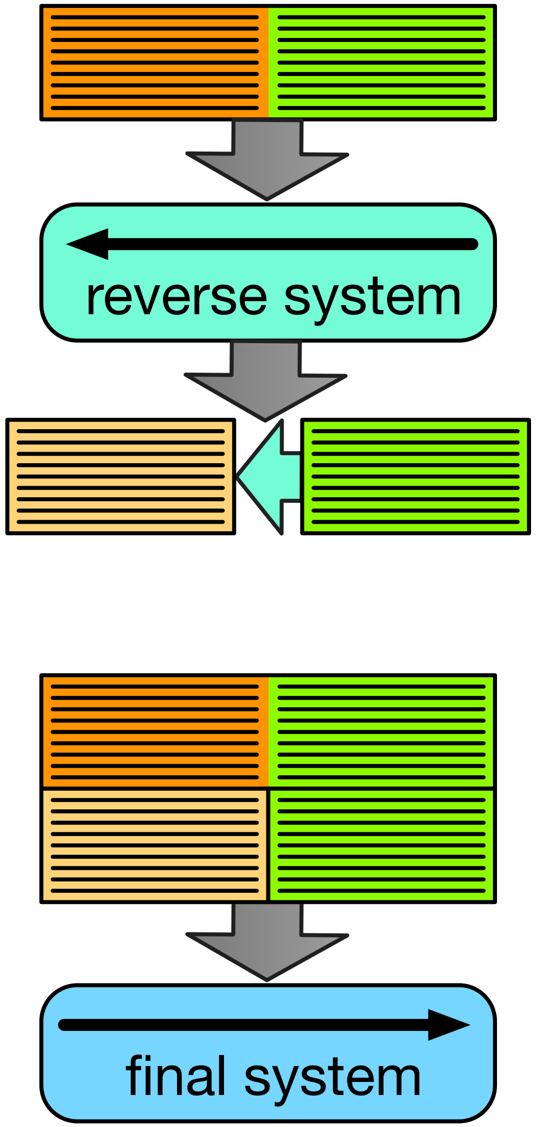
- Integrating a language model into neural architecture
  - word prediction informed by translation model and language model
  - gated unit that decides balance

- Use of language model in decoding
  - train language model in isolation
  - add language model score during inference (similar to ensembling)

- Proper balance between models (amount of training data, weights) unclear

Backtranslation

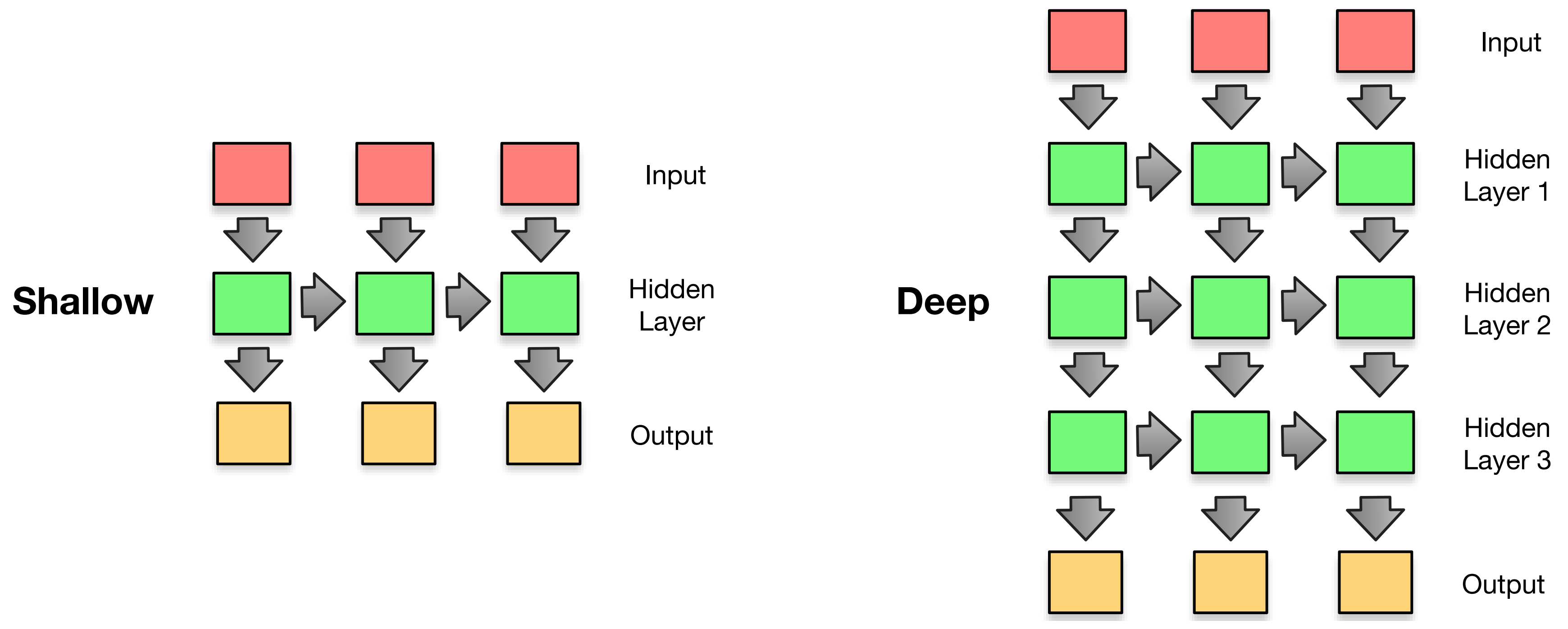
- No changes to model architecture
- Create synthetic parallel data
  - train a system in reverse direction
  - translate target-side monolingual data into source language
  - add as additional parallel data
- Simple, yet effective



deeper models

Deeper Models

- Encoder and decoder are recurrent neural networks
- We can add additional layers for each step
- Recall shallow and deep language models

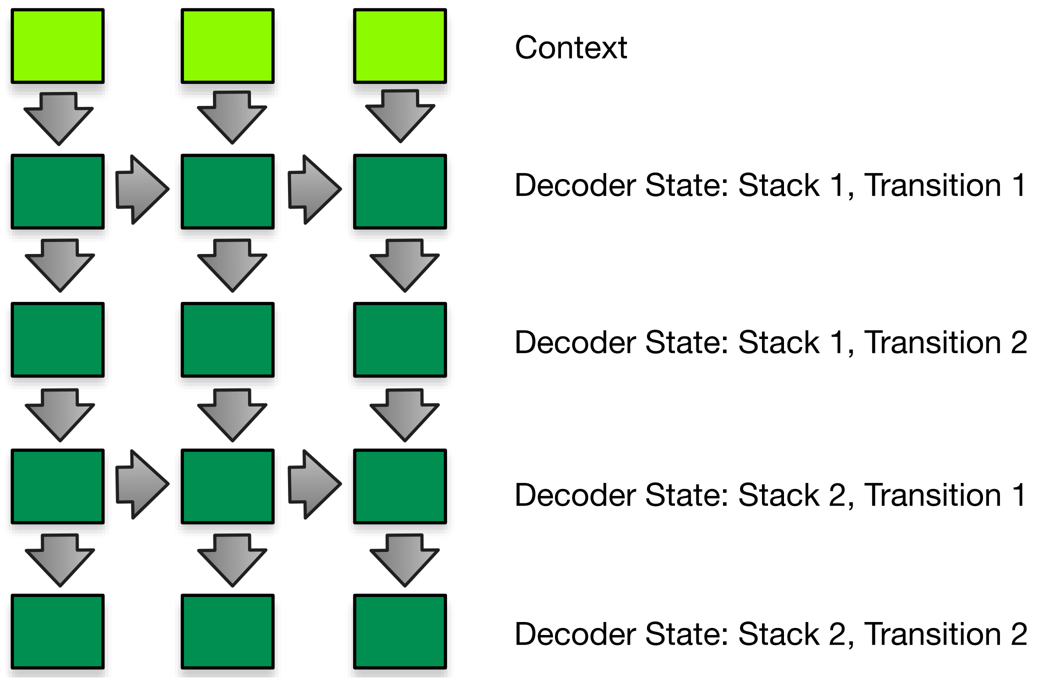


- Adding residual connections (short-cuts through deep layers) help

Deep Decoder

- Two ways of adding layers
  - deep transitions: several layers on path to output
  - deeply stacking recurrent neural networks

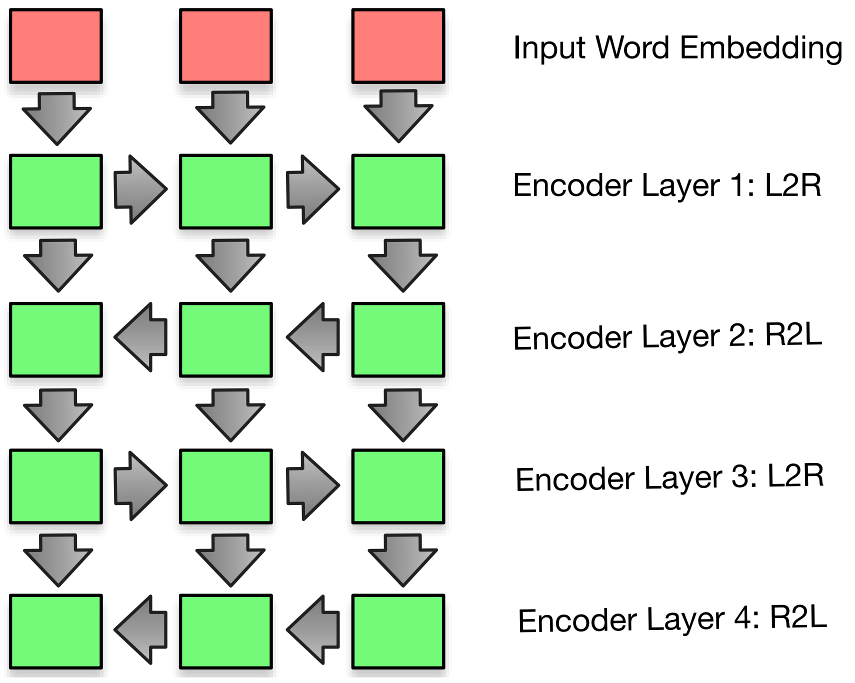
- Why not both?


Context
Decoder State: Stack 1, Transition 1
Decoder State: Stack 1, Transition 2
Decoder State: Stack 2, Transition 1
Decoder State: Stack 2, Transition 2

Deep Encoder

- Previously proposed encoder already has 2 layers
  - left-to-right recurrent network, to encode left context
  - right-to-left recurrent network, to encode right context

⇒ Third way of adding layers



Reality Check: Edinburgh WMT 2017

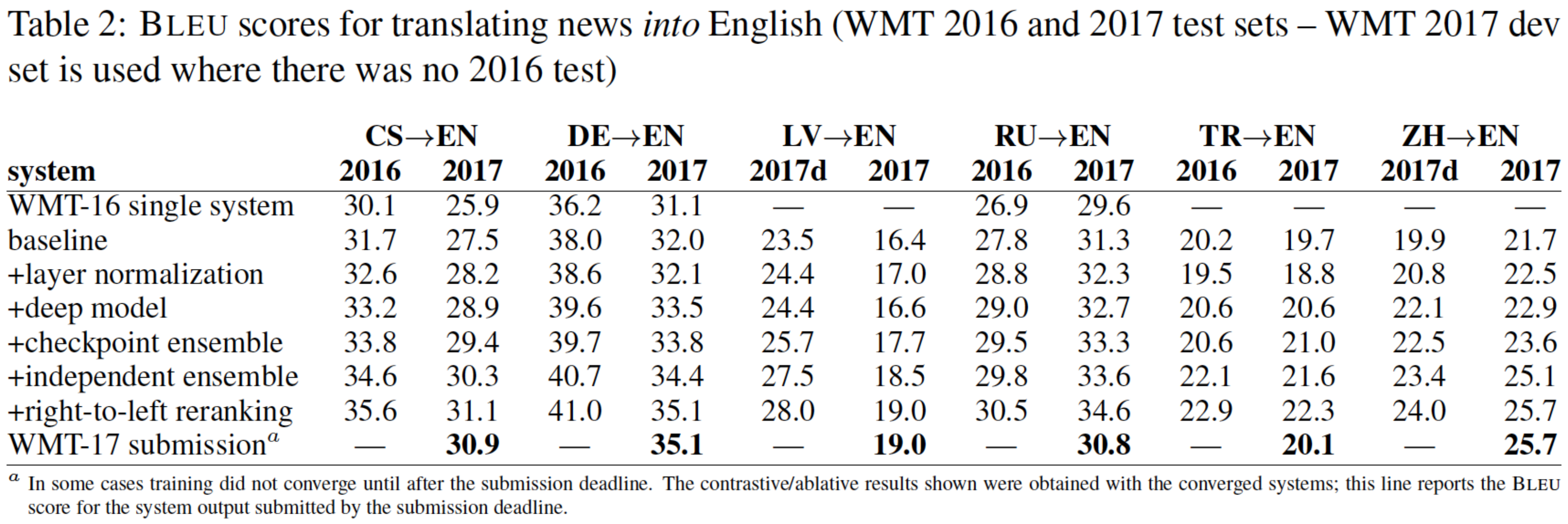


Table 2: BLEU scores for translating news *into* English (WMT 2016 and 2017 test sets – WMT 2017 dev set is used where there was no 2016 test)

	CS→EN		DE→EN		LV→EN		RU→EN		TR→EN		ZH→EN	
system	**2016**	**2017**	**2016**	**2017**	**2017d**	**2017**	**2016**	**2017**	**2016**	**2017**	**2017d**	**2017**
WMT-16 single system	30.1	25.9	36.2	31.1	—	—	26.9	29.6	—	—	—	—
baseline	31.7	27.5	38.0	32.0	23.5	16.4	27.8	31.3	20.2	19.7	19.9	21.7
+layer normalization	32.6	28.2	38.6	32.1	24.4	17.0	28.8	32.3	19.5	18.8	20.8	22.5
+deep model	33.2	28.9	39.6	33.5	24.4	16.6	29.0	32.7	20.6	20.6	22.1	22.9
+checkpoint ensemble	33.8	29.4	39.7	33.8	25.7	17.7	29.5	33.3	20.6	21.0	22.5	23.6
+independent ensemble	34.6	30.3	40.7	34.4	27.5	18.5	29.8	33.6	22.1	21.6	23.4	25.1
+right-to-left reranking	35.6	31.1	41.0	35.1	28.0	19.0	30.5	34.6	22.9	22.3	24.0	25.7
WMT-17 submission[a]	—	**30.9**	—	**35.1**	—	**19.0**	—	**30.8**	—	**20.1**	—	**25.7**

[a] In some cases training did not converge until after the submission deadline. The contrastive/ablative results shown were obtained with the converged systems; this line reports the BLEU score for the system output submitted by the submission deadline.

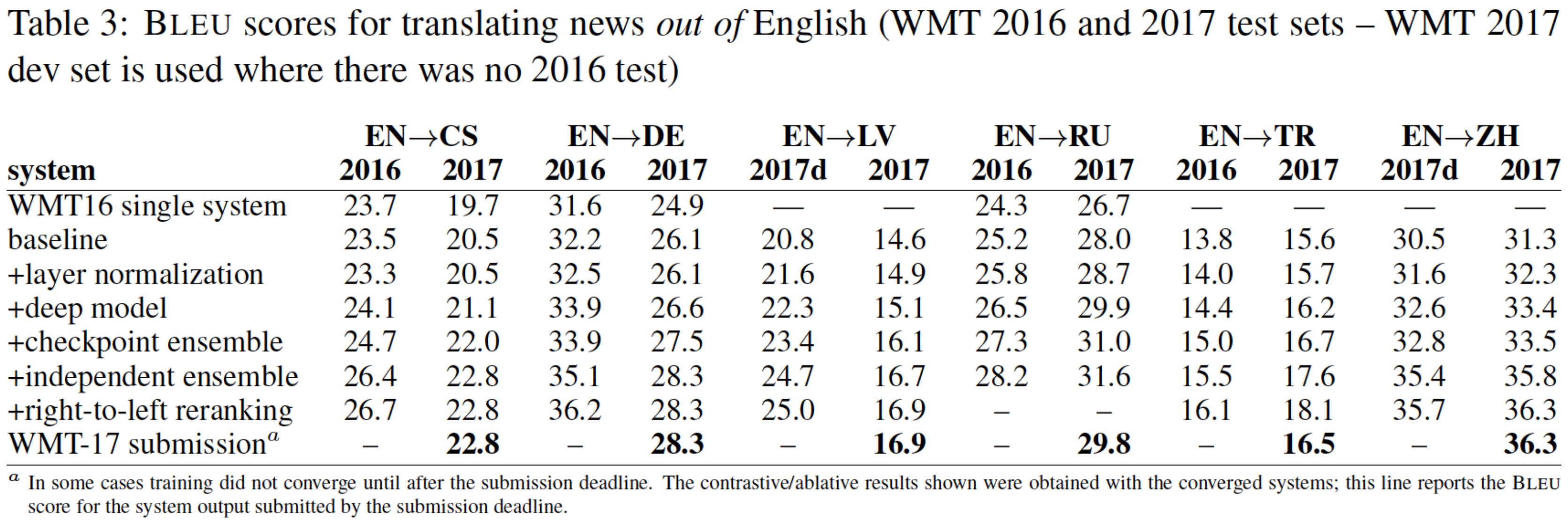


Table 3: BLEU scores for translating news *out of* English (WMT 2016 and 2017 test sets – WMT 2017 dev set is used where there was no 2016 test)

	EN→CS		EN→DE		EN→LV		EN→RU		EN→TR		EN→ZH	
system	**2016**	**2017**	**2016**	**2017**	**2017d**	**2017**	**2016**	**2017**	**2016**	**2017**	**2017d**	**2017**
WMT16 single system	23.7	19.7	31.6	24.9	—	—	24.3	26.7	—	—	—	—
baseline	23.5	20.5	32.2	26.1	20.8	14.6	25.2	28.0	13.8	15.6	30.5	31.3
+layer normalization	23.3	20.5	32.5	26.1	21.6	14.9	25.8	28.7	14.0	15.7	31.6	32.3
+deep model	24.1	21.1	33.9	26.6	22.3	15.1	26.5	29.9	14.4	16.2	32.6	33.4
+checkpoint ensemble	24.7	22.0	33.9	27.5	23.4	16.1	27.3	31.0	15.0	16.7	32.8	33.5
+independent ensemble	26.4	22.8	35.1	28.3	24.7	16.7	28.2	31.6	15.5	17.6	35.4	35.8
+right-to-left reranking	26.7	22.8	36.2	28.3	25.0	16.9	–	–	16.1	18.1	35.7	36.3
WMT-17 submission[a]	–	**22.8**	–	**28.3**	–	**16.9**	–	**29.8**	–	**16.5**	–	**36.3**

[a] In some cases training did not converge until after the submission deadline. The contrastive/ablative results shown were obtained with the converged systems; this line reports the BLEU score for the system output submitted by the submission deadline.

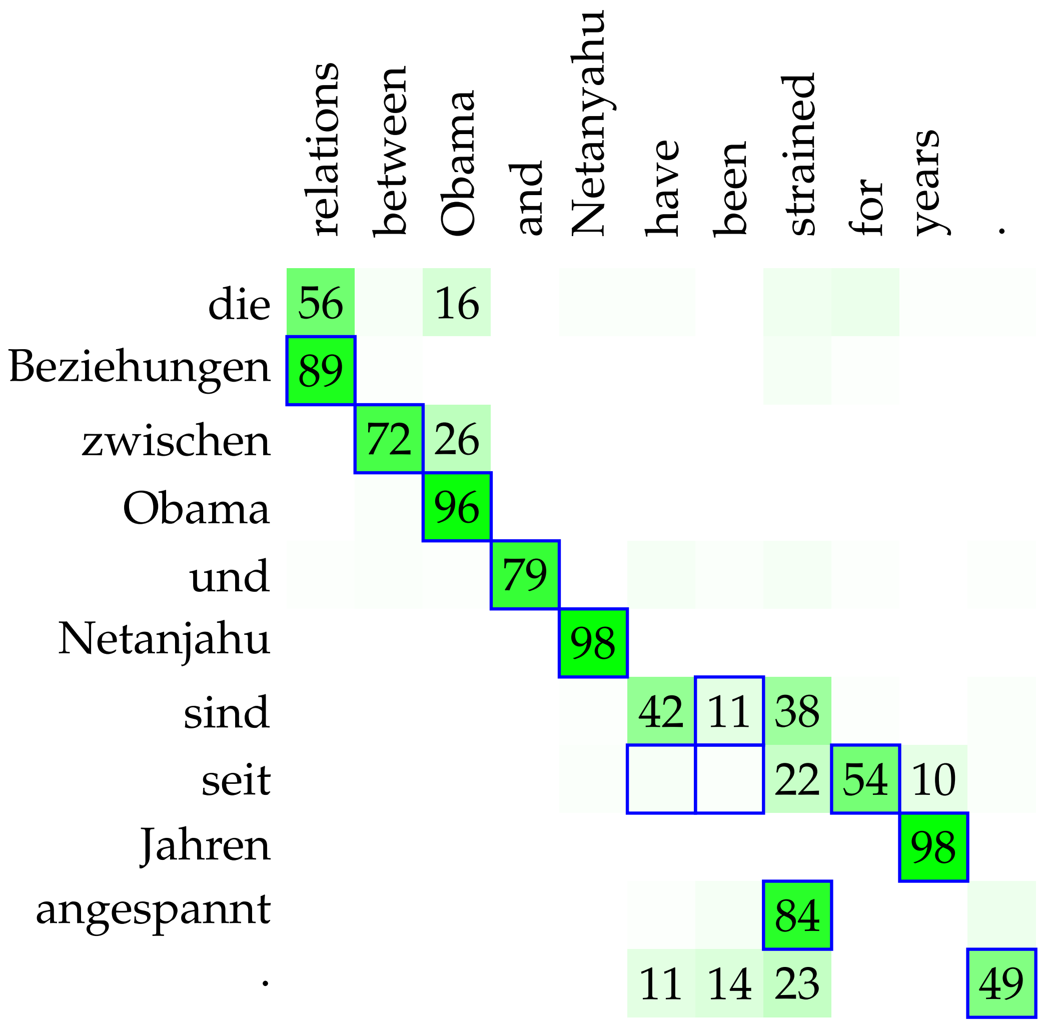
alignment and coverage

Alignment

- Attention model fulfills role of alignment

- Traditional methods for word alignment
  - based on co-occurence, word position, etc.
  - expectation maximization (EM) algorithm
  - popular: IBM models, fast-align

Attention vs. Alignment



	relations	between	Obama	and	Netanyahu	have	been	strained	for	years	.
die	56		16								
Beziehungen	89										
zwischen		72	26								
Obama			96								
und				79							
Netanjahu					98						
sind						42	11	38			
seit								22	54	10	
Jahren										98	
angespannt								84			
.						11	14	23			49

Guided Alignment

- Guided alignment training for neural networks
  - traditional objective function: match output words
  - now: also match given word alignments

- Add as cost to objective function
  - given alignment matrix A, with $\sum_j A_{ij} = 1$ (from IBM Models)
  - computed attention α_{ij} (also $\sum_j \alpha_{ij} = 1$ due to softmax)
  - added training objective (cross-entropy)

$$\text{cost}_{\text{CE}} = -\frac{1}{I} \sum_{i=1}^{I} \sum_{j=1}^{J} A_{ij} \log \alpha_{ij}$$

Coverage

	in	order	to	solve	the	problem	,	the	„	Social	Housing	"	alliance	suggests	a	fresh	start	.
um	37		33					13										
das					84													
Problem					10	80												
zu				63			12											
lösen				81														
,							40							30				
schlägt														80				
das								71	18									
Unternehmen									86			10						
der									84									
Gesellschaft									80									
für									45	41								
soziale									40	44	10							
Bildung											89							
vor									12			40	10					
.									10			37			11	13		
	43	7	46	161	108	89	62	112	392	121	110	130	26	132	22	19	6	6

Tracking Coverage

- Neural machine translation may drop or duplicate content

- Track coverage during decoding

$$\text{coverage}(j) = \sum_i \alpha_{i,j}$$

$$\text{over-generation} = \max\Big(0, \sum_j \text{coverage}(j) - 1\Big)$$

$$\text{under-generation} = \min\Big(1, \sum_j \text{coverage}(j)\Big)$$

- Add as cost to hypotheses

Coverage Models

- Use as information for state progression

$$a(s_{i-1}, h_j) = W^a s_{i-1} + U^a h_j + V^a \text{coverage}(j) + b^a$$

- Add to objective function

$$\log \sum_i P(y_i|x) + \lambda \sum_j (1 - \text{coverage}(j))^2$$

- May also model fertility
  - some words are typically dropped
  - some words produce multiple output words

linguistic annotation

Example

Words	*the*	*girl*	*watched*	*attentively*	*the*	*beautiful*	*fireflies*
Part of speech	DET	NN	VFIN	ADV	DET	JJ	NNS
Lemma	*the*	*girl*	*watch*	*attentive*	*the*	*beautiful*	*firefly*
Morphology	-	SING.	PAST	-	-	-	PLURAL
Noun phrase	BEGIN	CONT	OTHER	OTHER	BEGIN	CONT	CONT
Verb phrase	OTHER	OTHER	BEGIN	CONT	CONT	CONT	CONT
Synt. dependency	*girl*	*watched*	-	*watched*	*fireflies*	*fireflies*	*watched*
Depend. relation	DET	SUBJ	-	ADV	DET	ADJ	OBJ
Semantic role	-	ACTOR	-	MANNER	-	MOD	PATIENT
Semantic type	-	HUMAN	VIEW	-	-	-	ANIMATE

Input Annotation

- Input words are encoded in one-hot vectors
- Additional linguistic annotation
  - part-of-speech tag
  - morphological features
  - etc.
- Encode each annotation in its own one-hot vector space
- Concatenate one-hot vecors
- Essentially:
  - each annotation maps to embedding
  - embeddings are added

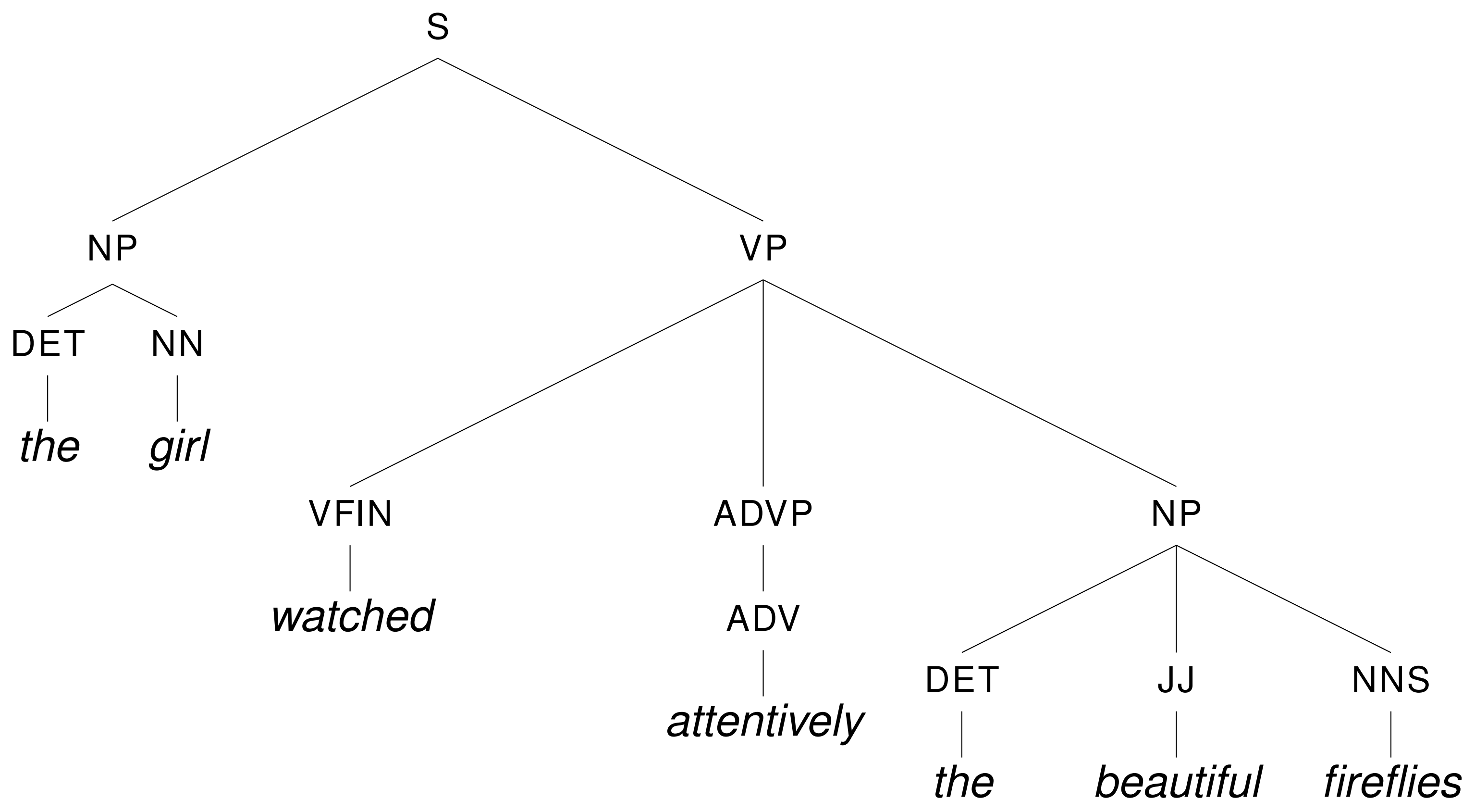
Output Annotation

- Same can be done for output

- Additional output annotation is latent feature
  - ultimately, we do not care if right part-of-speech tag is predicted
  - only right output words matter

- Optimizing for correct output annotation $\rightarrow$ better prediction of output words

Linearized Output Syntax

Sentence	*the girl watched attentively the beautiful fireflies*
Syntax tree	 S NP VP DET NN the girl VFIN ADVP NP watched ADV attentively DET JJ NNS the beautiful fireflies
Linearized	(S (NP (DET *the*) (NN *girl*)) (VP (VFIN *watched*) (ADVP (ADV *attentively*)) (NP (DET *the*) (JJ *beautiful*) (NNS *fireflies*))))

multiple language pairs

One Model, Multiple Language Pairs

- One language pair $\rightarrow$ train one model
- Multiple language pairs $\rightarrow$ train one model for each
- Multiple language pair $\rightarrow$ train one model for all

Multiple Input Languages

- Given
  - French–English corpus
  - German–English corpus

- Train one model on concatenated corpora

- Benefit: sharing monolingual target language data

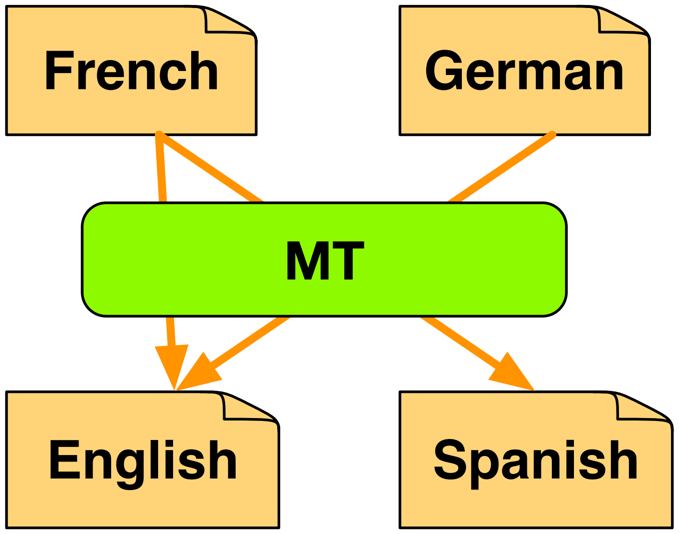
Multiple Output Languages

- Multiple output languages
  - French–English corpus
  - French–Spanish corpus
- Need to mark desired output language with special token

[ENGLISH] *N'y a-t-il pas ici deux poids, deux mesures?*
$\Rightarrow$ *Is this not a case of double standards?*

[SPANISH] *N'y a-t-il pas ici deux poids, deux mesures?*
$\Rightarrow$ *No puede verse con toda claridad que estamos utilizando un doble rasero?*

Zero Shot

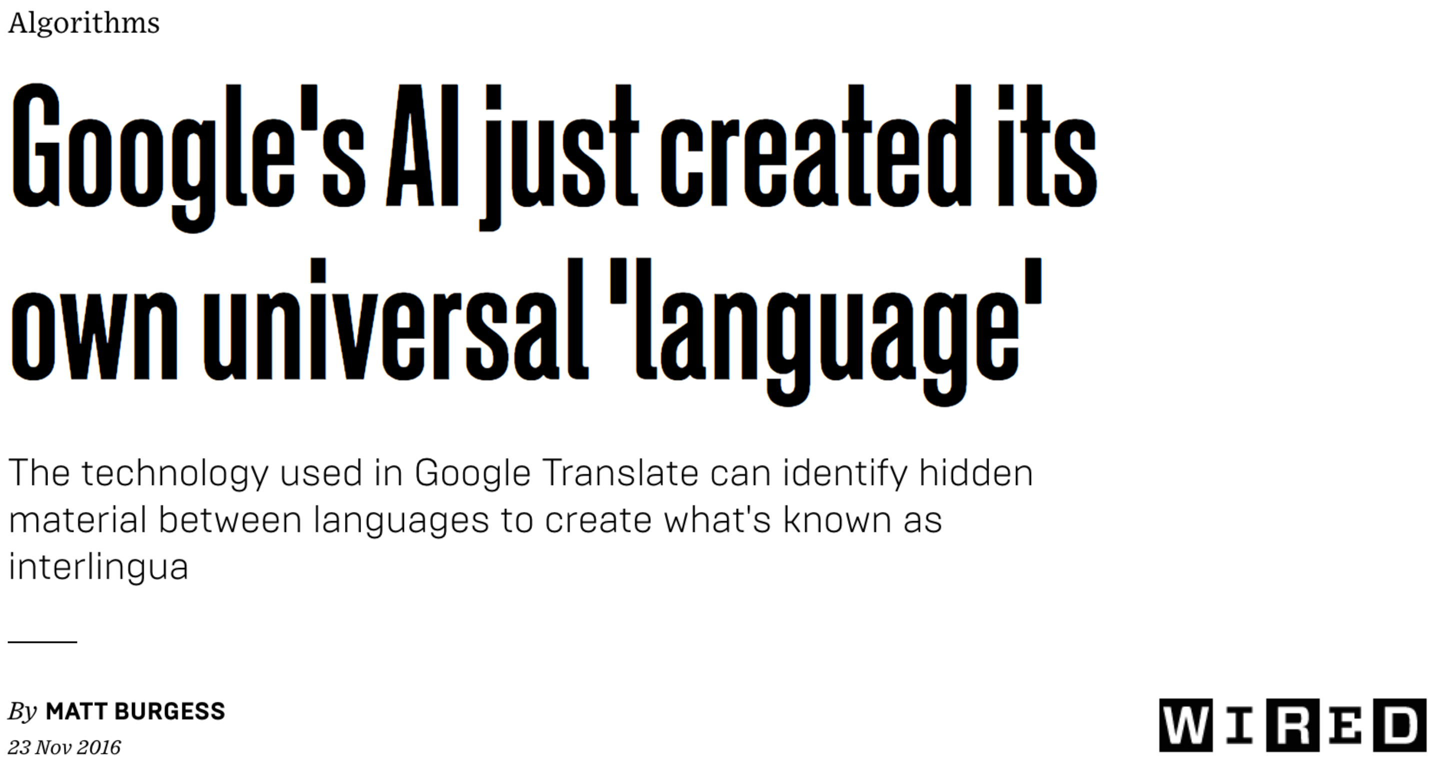


- Can the model translate German to Spanish?

[SPANISH] *Messen wir hier nicht mit zweierlei Maß?*
$\Rightarrow$ *No puede verse con toda claridad que estamos utilizando un doble rasero?*

Zero Shot: Vision

- Direct translation only requires bilingual mapping
- Zero shot requires interlingual representation



Algorithms

Google's AI just created its own universal 'language'

The technology used in Google Translate can identify hidden material between languages to create what's known as interlingua

By **MATT BURGESS**

23 Nov 2016

WIRED

Zero Shot: Reality

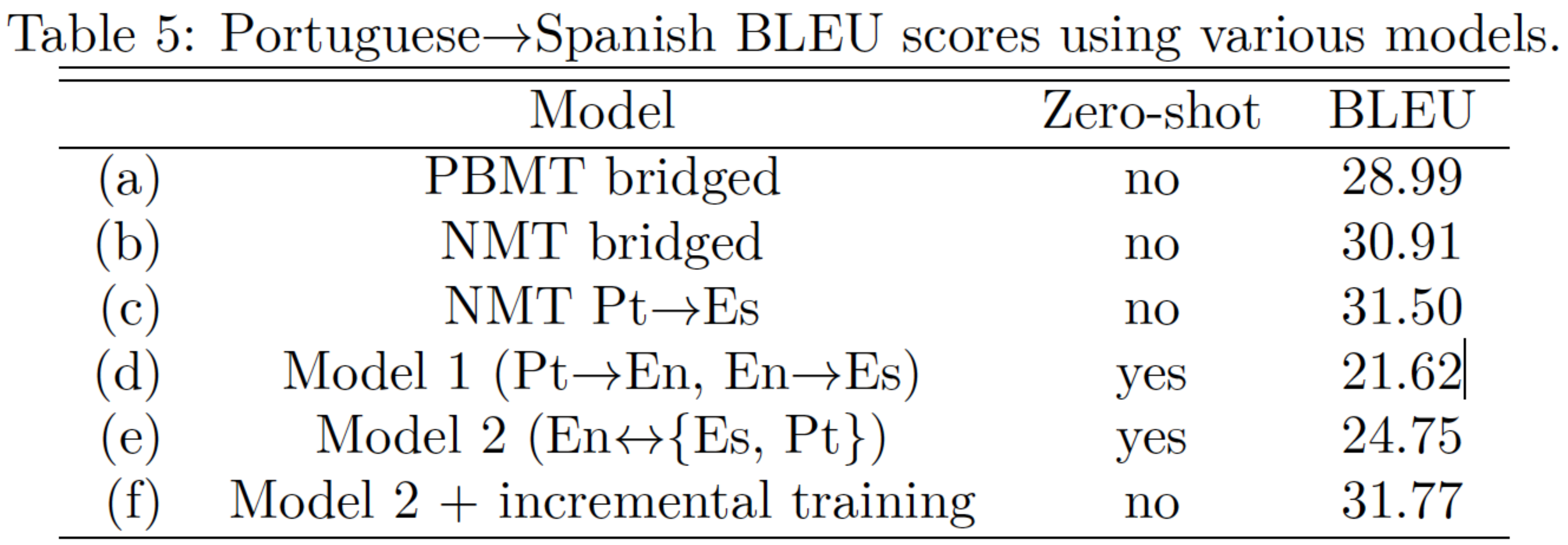


Table 5: Portuguese→Spanish BLEU scores using various models.

	Model	Zero-shot	BLEU
(a)	PBMT bridged	no	28.99
(b)	NMT bridged	no	30.91
(c)	NMT Pt→Es	no	31.50
(d)	Model 1 (Pt→En, En→Es)	yes	21.62
(e)	Model 2 (En↔{Es, Pt})	yes	24.75
(f)	Model 2 + incremental training	no	31.77

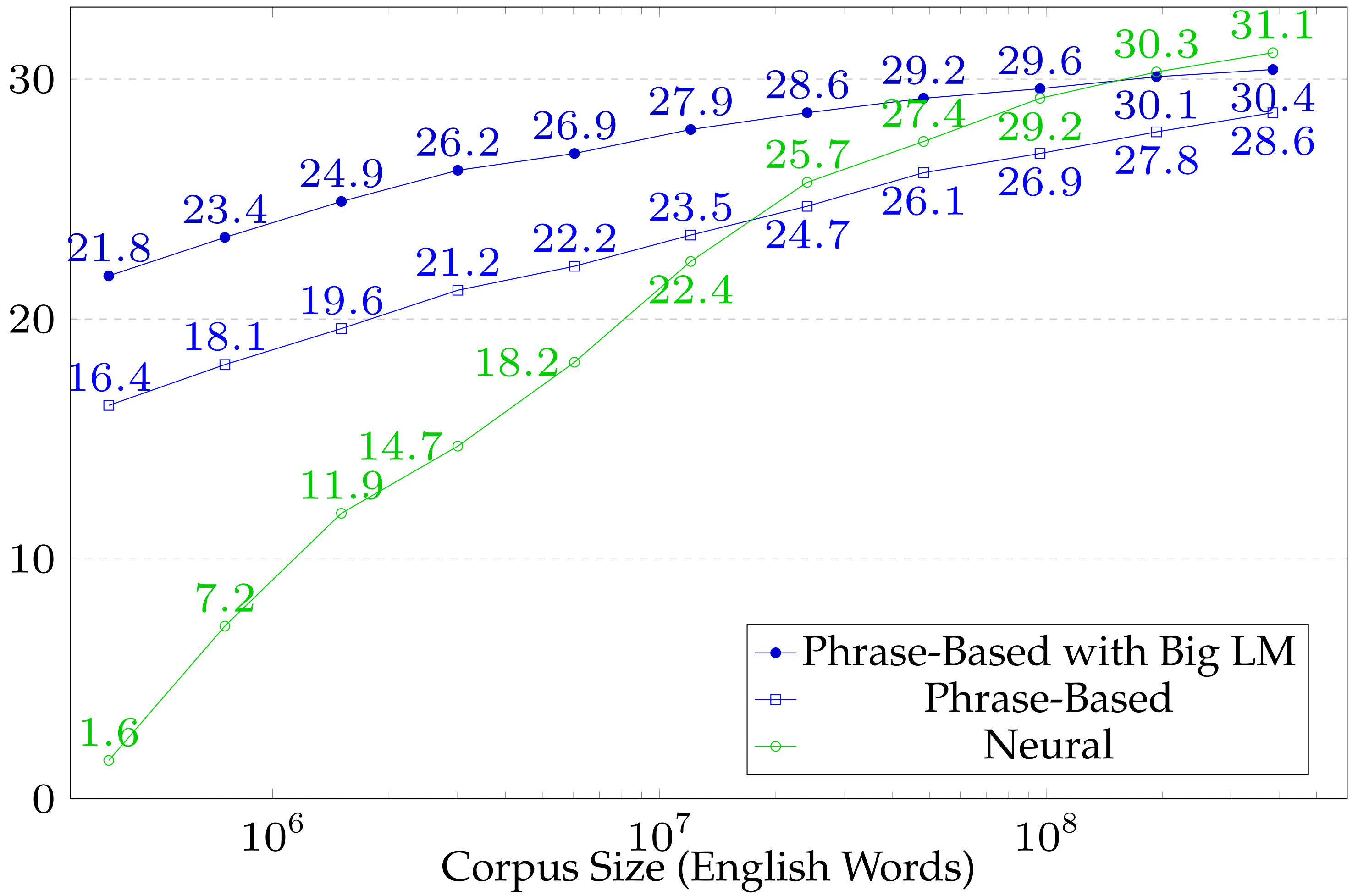
challenges

Challenges

- Challenges
  - lack of training data
  - domain mismatch
  - noisy data
  - sentence length
  - word alignment
  - beam search
- Alternative architectures
  - convolutional neural networks
  - self-attention

challenges

Amount of Training Data



English-Spanish systems trained on 0.4 million to 385.7 million words

Translation Examples

Source	A Republican strategy to counter the re-election of Obama
$\frac{1}{1024}$	Un órgano de coordinación para el anuncio de libre determinación
$\frac{1}{512}$	Lista de una estrategia para luchar contra la elección de hojas de Ohio
$\frac{1}{256}$	Explosión realiza una estrategia divisiva de luchar contra las elecciones de autor
$\frac{1}{128}$	Una estrategia republicana para la eliminación de la reelección de Obama
$\frac{1}{64}$	Estrategia siria para contrarrestar la reelección del Obama .
$\frac{1}{32}+$	Una estrategia republicana para contrarrestar la reelección de Obama

domain mismatch

Domain Mismatch

System ↓	Law	Medical	IT	Koran	Subtitles
All Data	30.5 32.8	45.1 42.2	35.3 44.7	17.9 17.9	26.4 20.8
Law	31.1 34.4	12.1 18.2	3.5 6.9	1.3 2.2	2.8 6.0
Medical	3.9 10.2	39.4 43.5	2.0 8.5	0.6 2.0	1.4 5.8
IT	1.9 3.7	6.5 5.3	42.1 39.8	1.8 1.6	3.9 4.7
Koran	0.4 1.8	0.0 2.1	0.0 2.3	15.9 18.8	1.0 5.5
Subtitles	7.0 9.9	9.3 17.8	9.2 13.6	9.0 8.4	25.9 22.1

Translation Examples

Source	Schaue um dich herum.
Ref.	Look around you.
All	NMT: Look around you. SMT: Look around you.
Law	NMT: Sughum gravecorn. SMT: In order to implement dich Schaue .
Medical	NMT: EMEA / MB / 049 / 01-EN-Final Work progamme for 2002 SMT: Schaue by dich around .
IT	NMT: Switches to paused. SMT: To Schaue by itself . \t \t
Koran	NMT: Take heed of your own souls. SMT: And you see.
Subtitles	NMT: Look around you. SMT: Look around you .

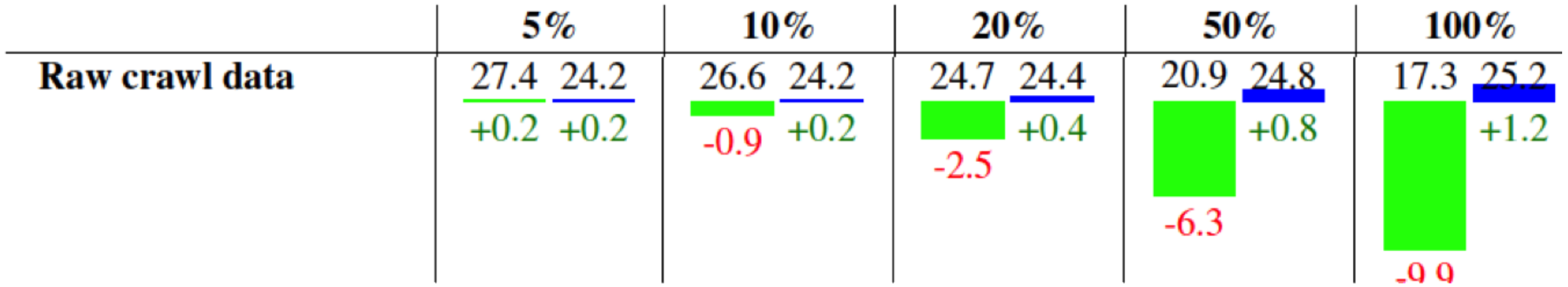
noisy data

Noise in Training Data

- Crawled parallel data from the web (very noisy)

	SMT	NMT
WMT17	24.0	27.2
+ Paracrawl	25.2 (+1.2)	17.3 (-9.9)

(German-English, 90m words each of WMT17 and Crawl data)



	5%		10%		20%		50%		100%	
Raw crawl data	27.4	24.2	26.6	24.2	24.7	24.4	20.9	24.8	17.3	25.2
	+0.2	+0.2	-0.9	+0.2	-2.5	+0.4	-6.3	+0.8	-9.9	+1.2

- Corpus cleaning methods [Xu and Koehn, EMNLP 2017] give improvements

Types of Noise

- Misaligned sentences
- Disfluent language (from MT, bad translations)
- Wrong language data (e.g., French in German–English corpus)
- Untranslated sentences
- Short segments (e.g., dictionaries)
- Mismatched domain

Mismatched Sentences

- Artificial created by randomly shuffling sentence order
- Added to existing parallel corpus in different amounts

5%		10%		20%		50%		100%	
	24.0		24.0		23.9	26.1	23.9	25.3	23.4
	-0.0		-0.0		-0.1	-1.1	-0.1	-1.9	-0.6

- Bigger impact on NMT (green, left) than SMT (blue, right)

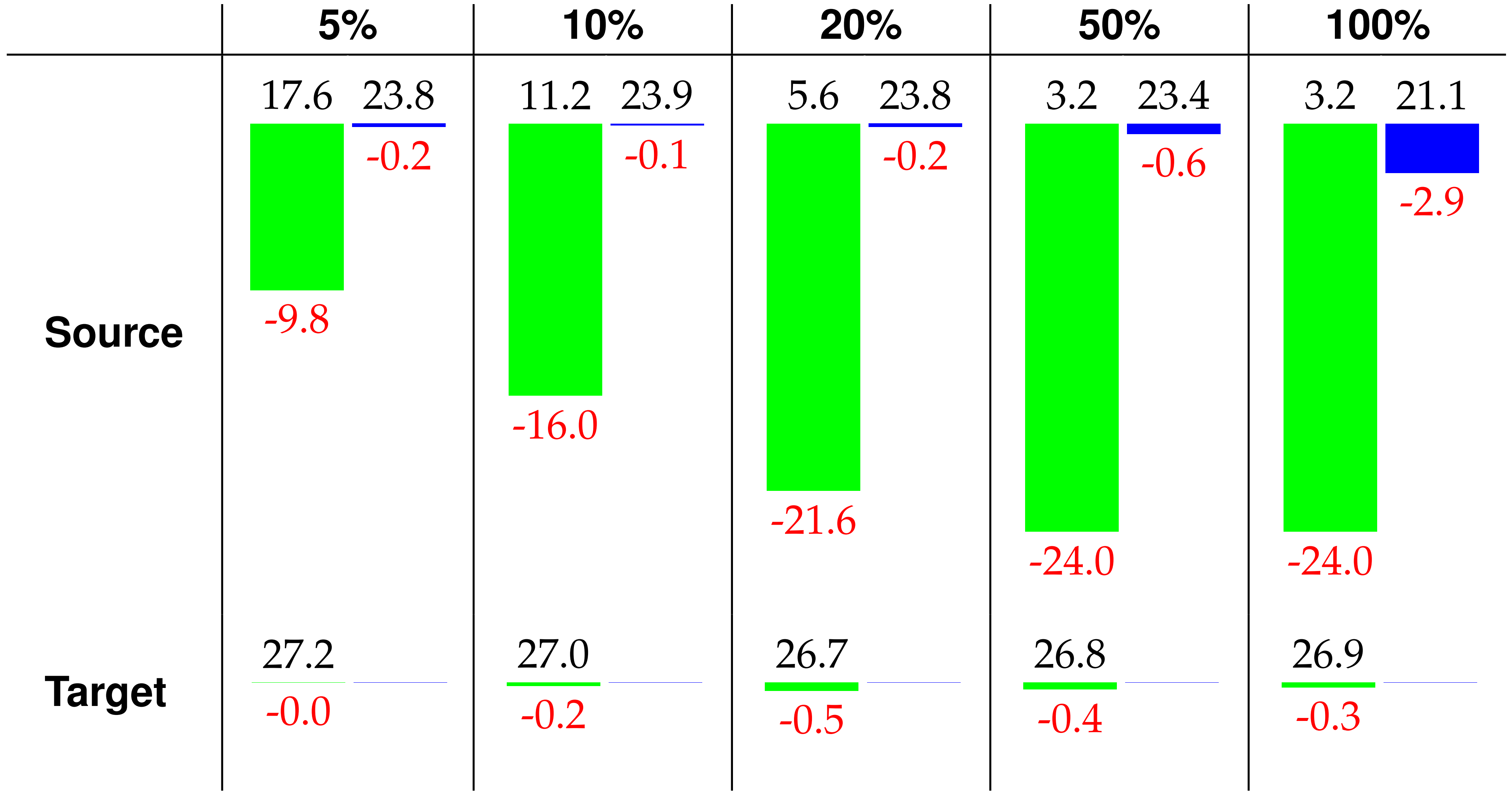
Misordered Words

- Artificial created by randomly shuffling words in each sentence

	5%		10%		20%		50%		100%	
Source		24.0 -0.0		23.6 -0.4		23.9 -0.1	26.6 -0.6	23.6 -0.4	25.5 -1.7	23.7 -0.3
Target		24.0 -0.0		24.0 -0.0		23.4 -0.6	26.7 -0.5	23.2 -0.8	26.1 -1.1	22.9 -1.1

- Similar impact on NMT than SMT, worse for source reshuffle

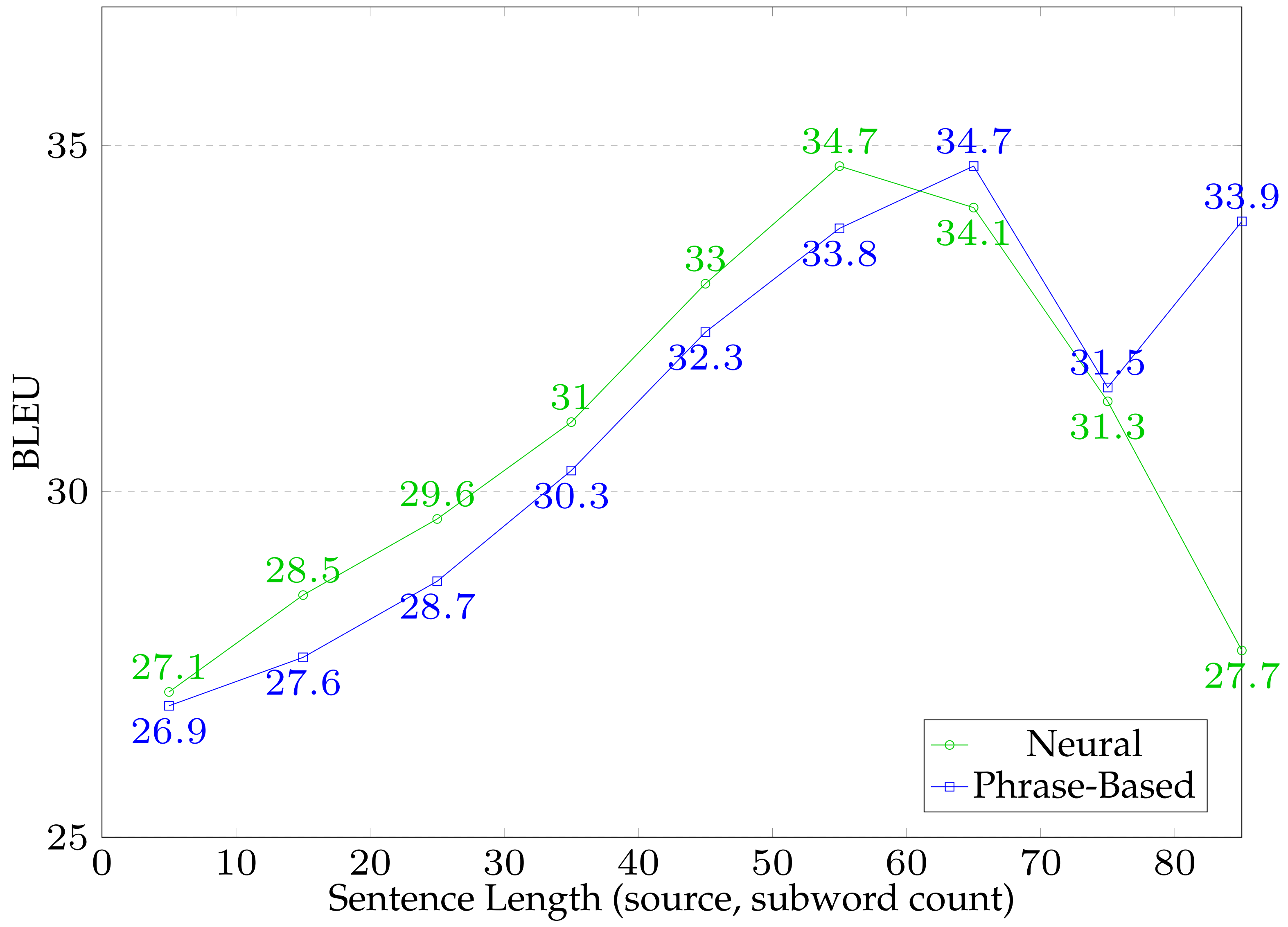
Untranslated Sentences



	5%		10%		20%		50%		100%	
Source	17.6 -9.8	23.8 -0.2	11.2 -16.0	23.9 -0.1	5.6 -21.6	23.8 -0.2	3.2 -24.0	23.4 -0.6	3.2 -24.0	21.1 -2.9
Target	27.2 -0.0		27.0 -0.2		26.7 -0.5		26.8 -0.4		26.9 -0.3	

sentence length

Sentence Length

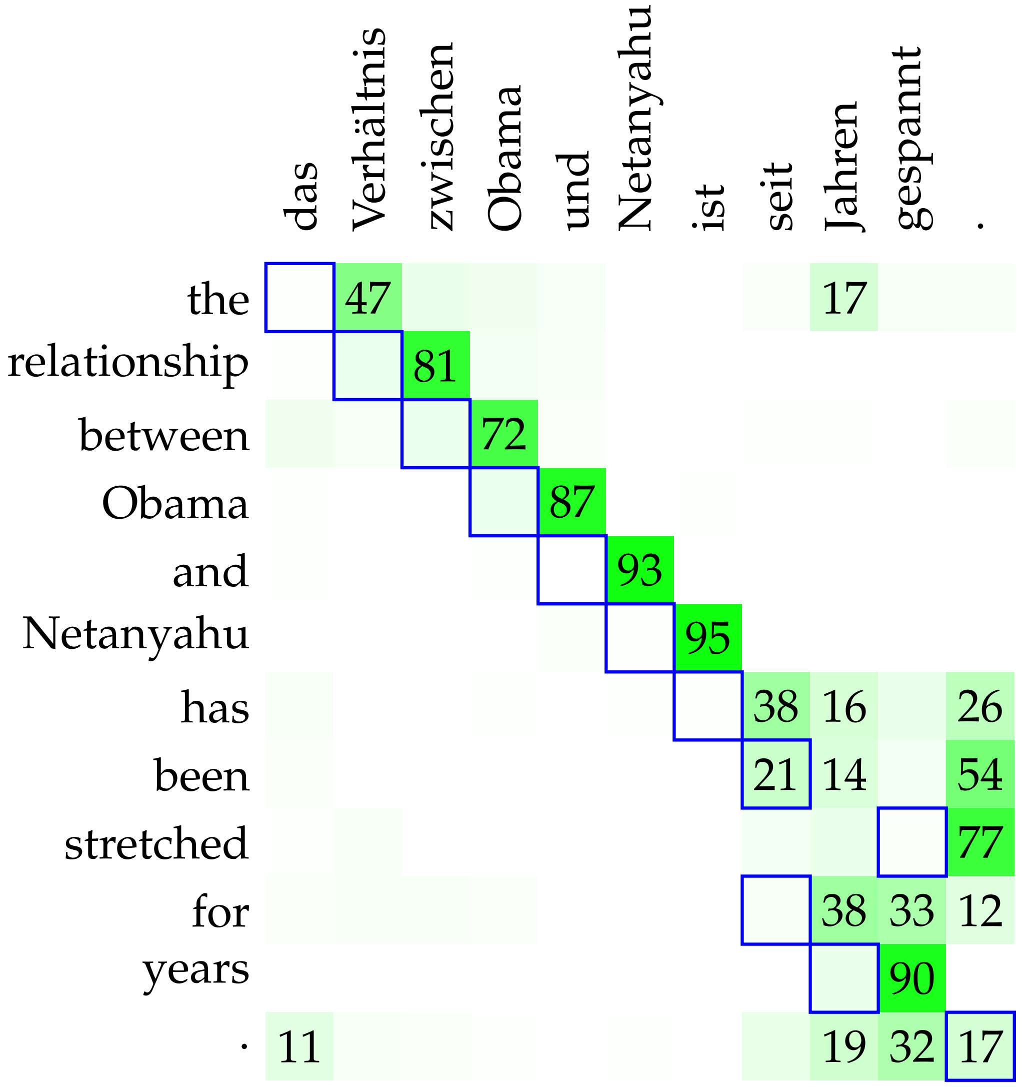


word alignment

Word Alignment

	relations	between	Obama	and	Netanyahu	have	been	strained	for	years	.
die	56		16								
Beziehungen	$\boxed{89}$										
zwischen		$\boxed{72}$	26								
Obama			$\boxed{96}$								
und				$\boxed{79}$							
Netanjahu					$\boxed{98}$						
sind						42	$\boxed{11}$	38			
seit						$\boxed{}$	$\boxed{}$	22	$\boxed{54}$	10	
Jahren										$\boxed{98}$	
angespannt								$\boxed{84}$			
.						11	14	23			$\boxed{49}$

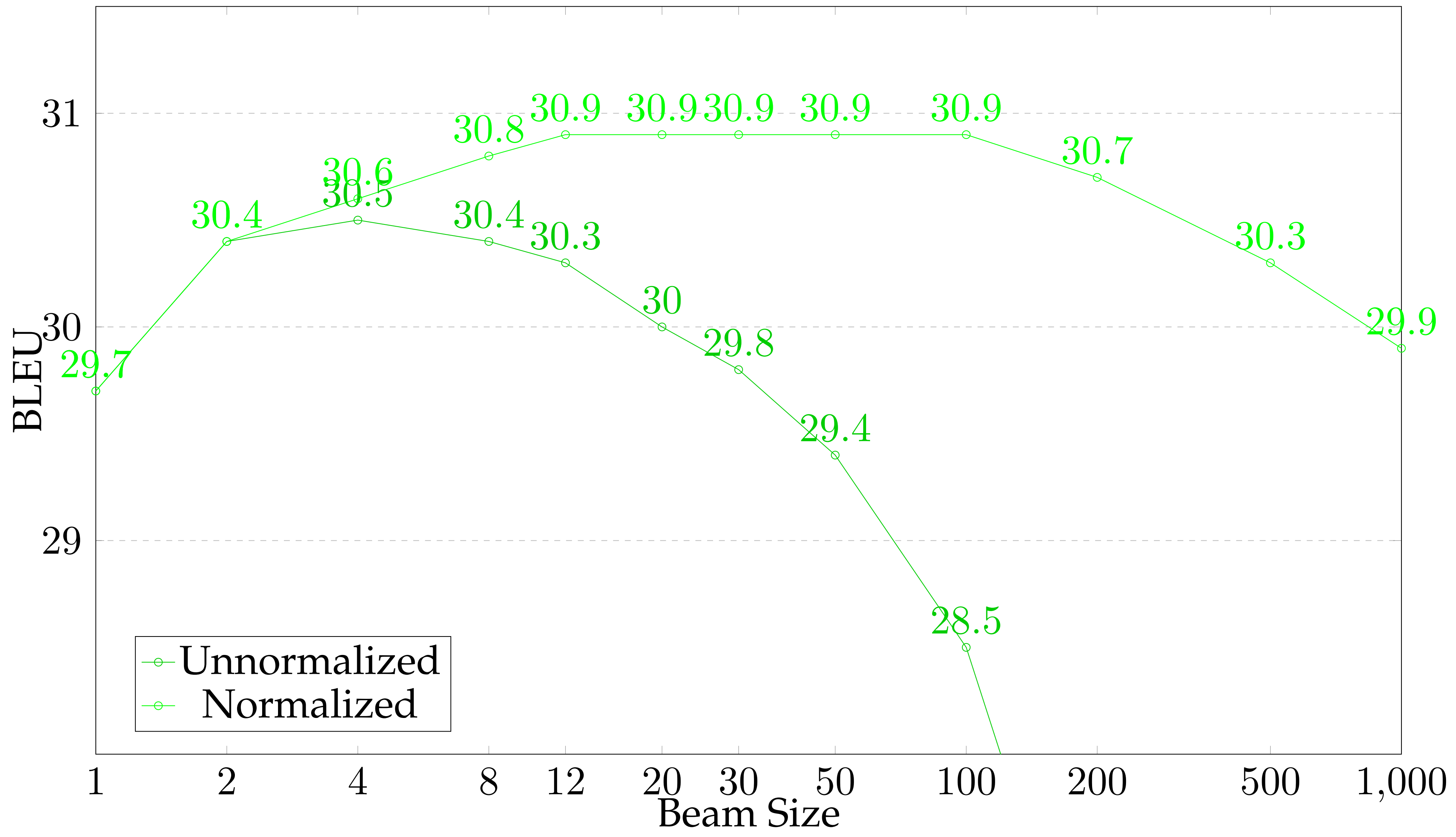
Word Alignment?



	das	Verhältnis	zwischen	Obama	und	Netanyahu	ist	seit	Jahren	gespannt	.
the		47							17		
relationship			81								
between				72							
Obama					87						
and						93					
Netanyahu							95				
has								38	16		26
been								21	14		54
stretched											77
for									38	33	12
years										90	
.	11								19	32	17

beam search

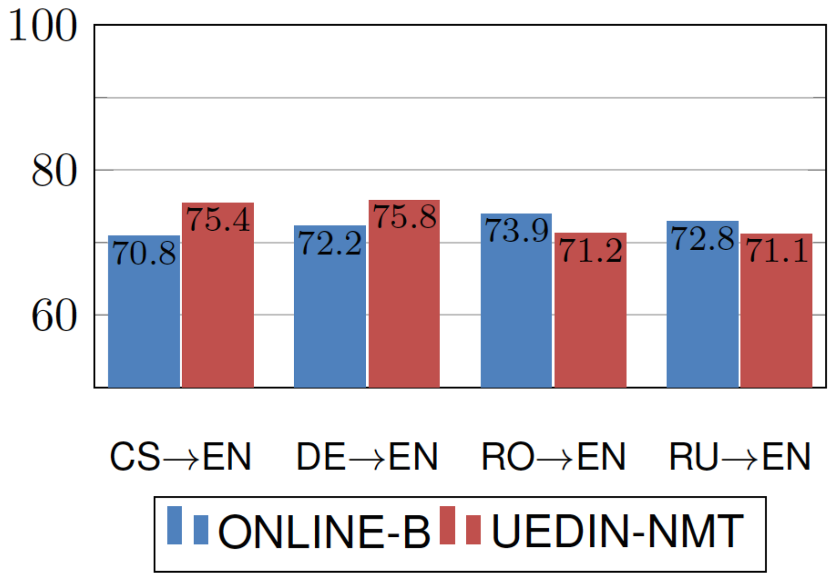
Beam Search



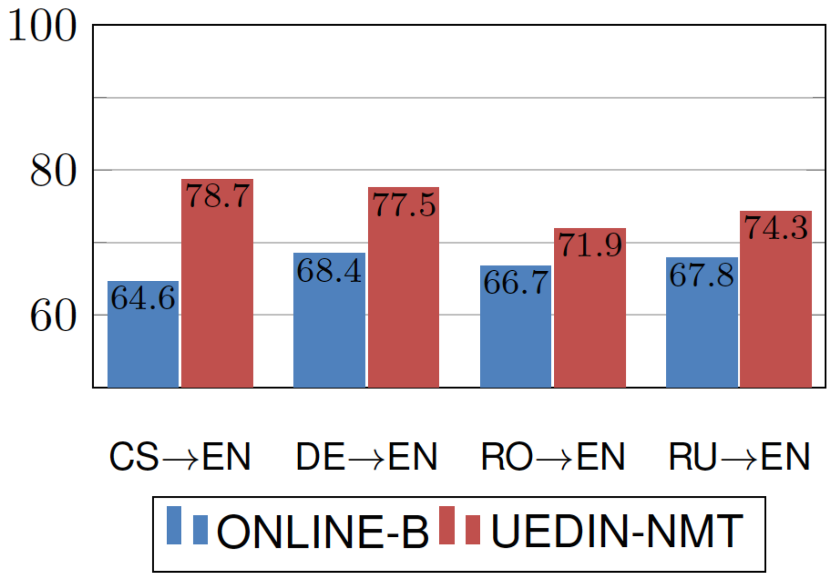
BLEU
31
30
29
Beam Size
1
2
4
8
12
20
30
50
100
200
500
1,000
Unnormalized
Normalized
29.7
30.4
30.5
30.4
30.3
30
29.8
29.4
28.5
30.6
30.8
30.9
30.9
30.9
30.9
30.9
30.7
30.3
29.9

Just Better Fluency?

Adequacy
+1%



Fluency
+13%



(from: Sennrich and Haddow, 2017)

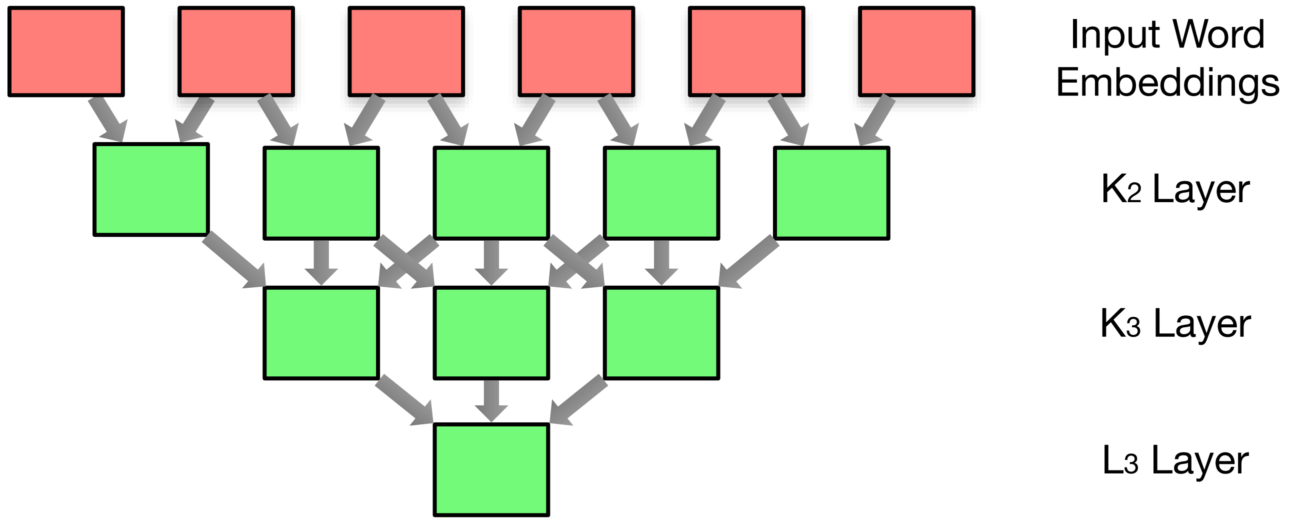
alternative architectures

Beyond Recurrent Neural Networks

- We presented the currently dominant model
  - recurrent neural networks for encoder and decoder
  - attention
- Convolutional neural networks
- Self attention

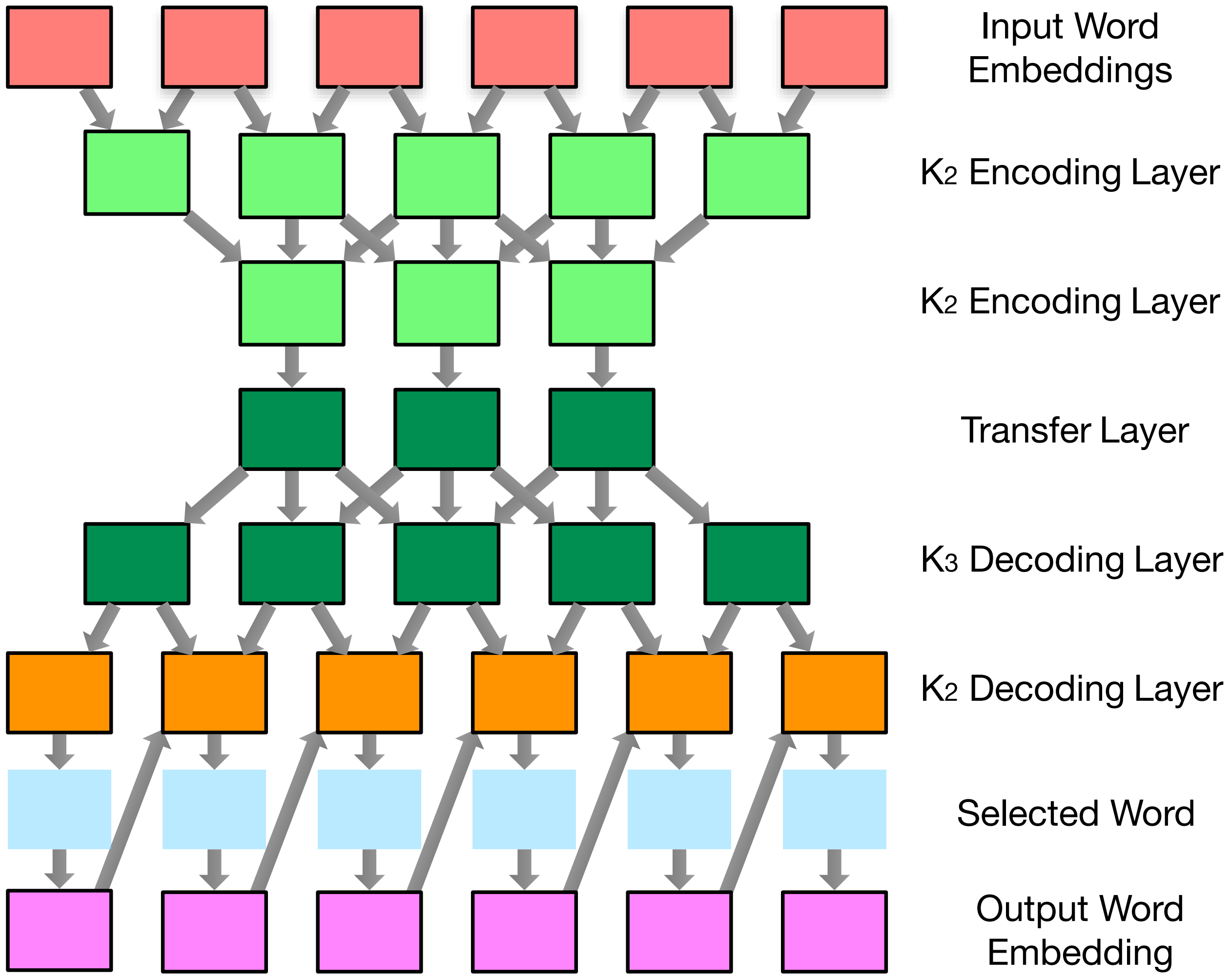
convolutional neural networks

Convolutional Neural Networks



- Build sentence representation bottom-up
  - merge any n neighboring nodes
  - n may be 2, 3, ...

Generation


Input Word Embeddings
K2 Encoding Layer
K2 Encoding Layer
Transfer Layer
K3 Decoding Layer
K2 Decoding Layer
Selected Word
Output Word Embedding

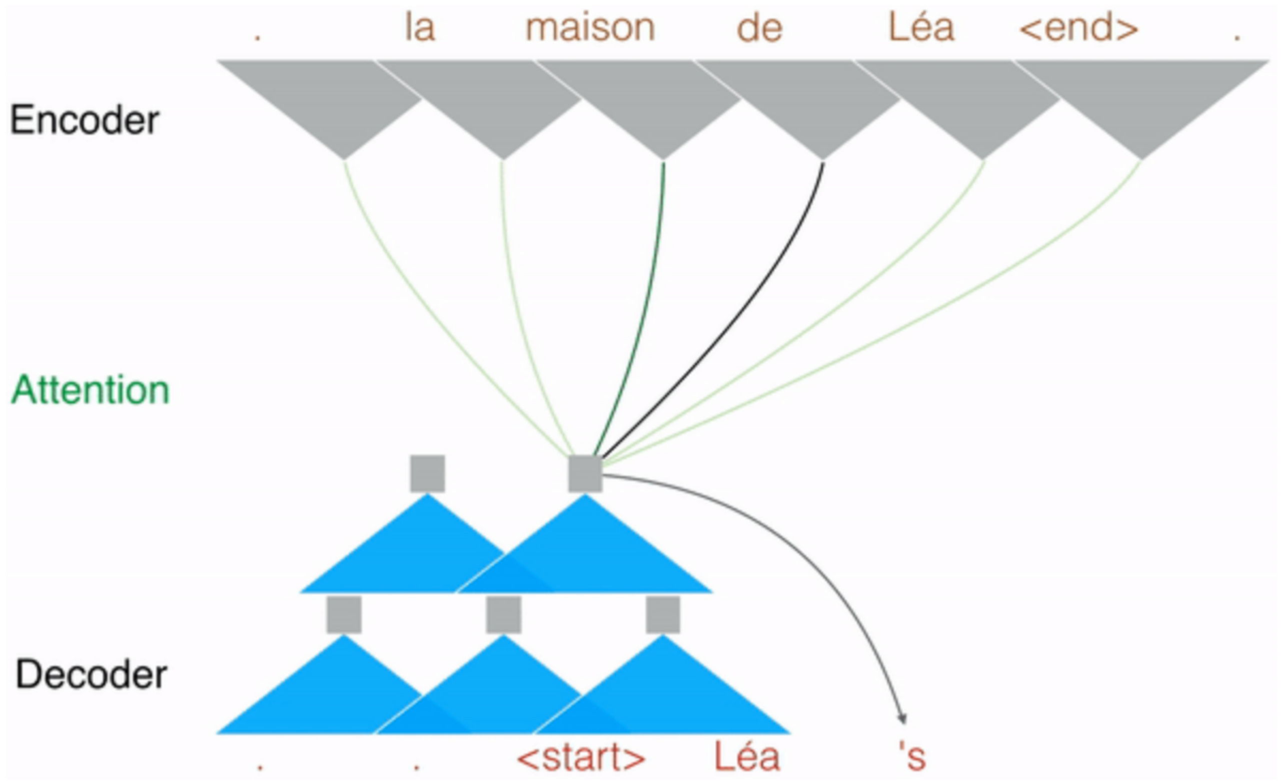
Generation

- Encode with convolutional neural network
- Decode with convolutional neural network
- Also include a linear recurrent neural network

- Important: predict length of output sentence

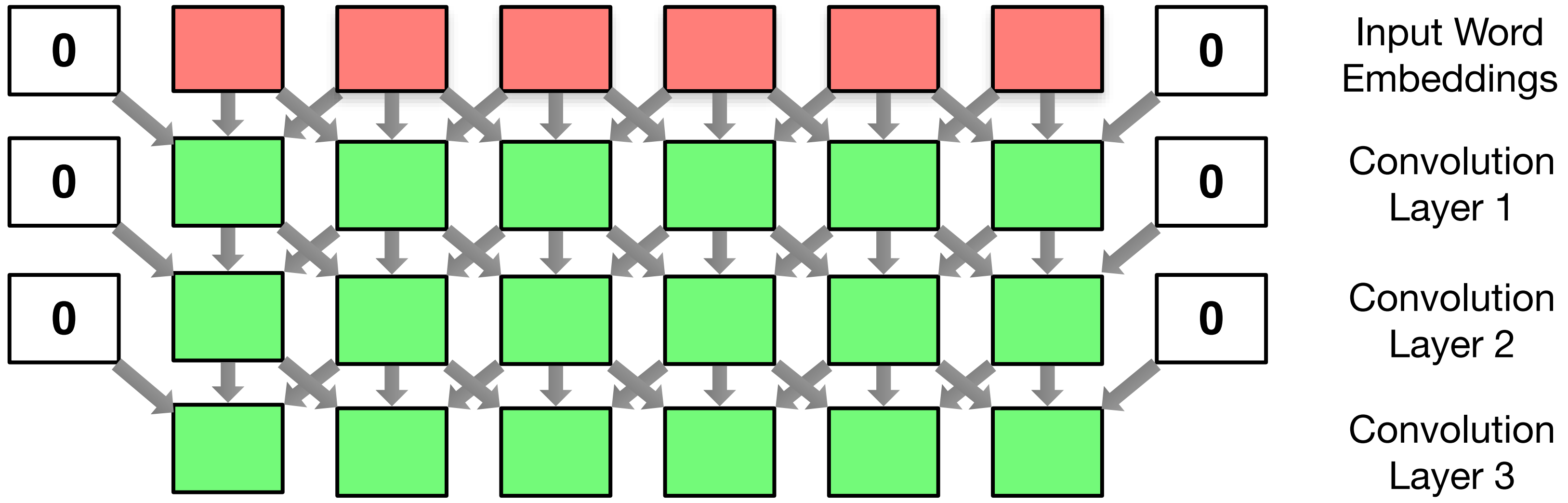
- Does it work?
  used successfully in re-ranking (Cho et al., 2014)

Convolutional Network with Attention



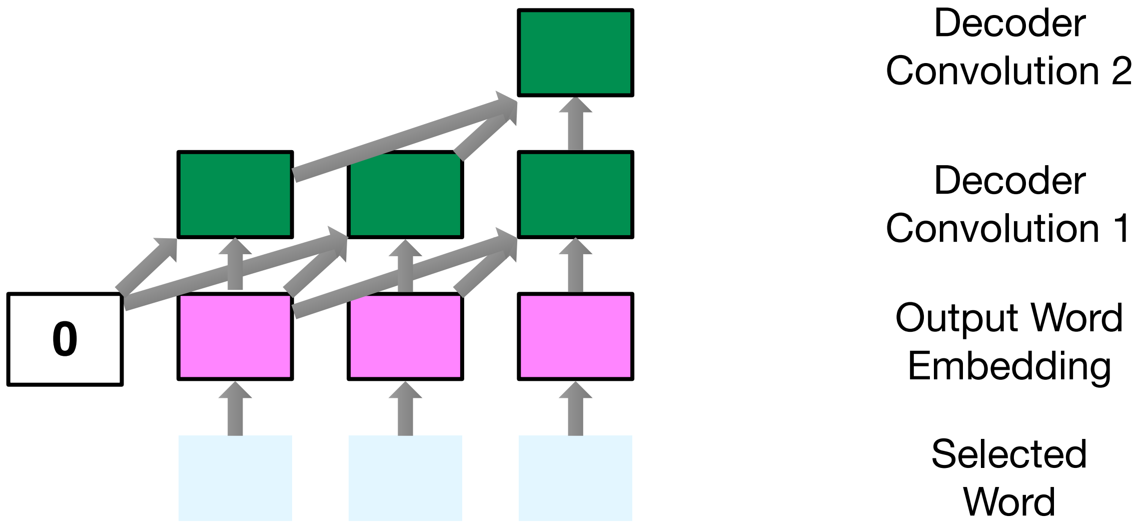
(Facebook, 2017)

Convolutional Encoder



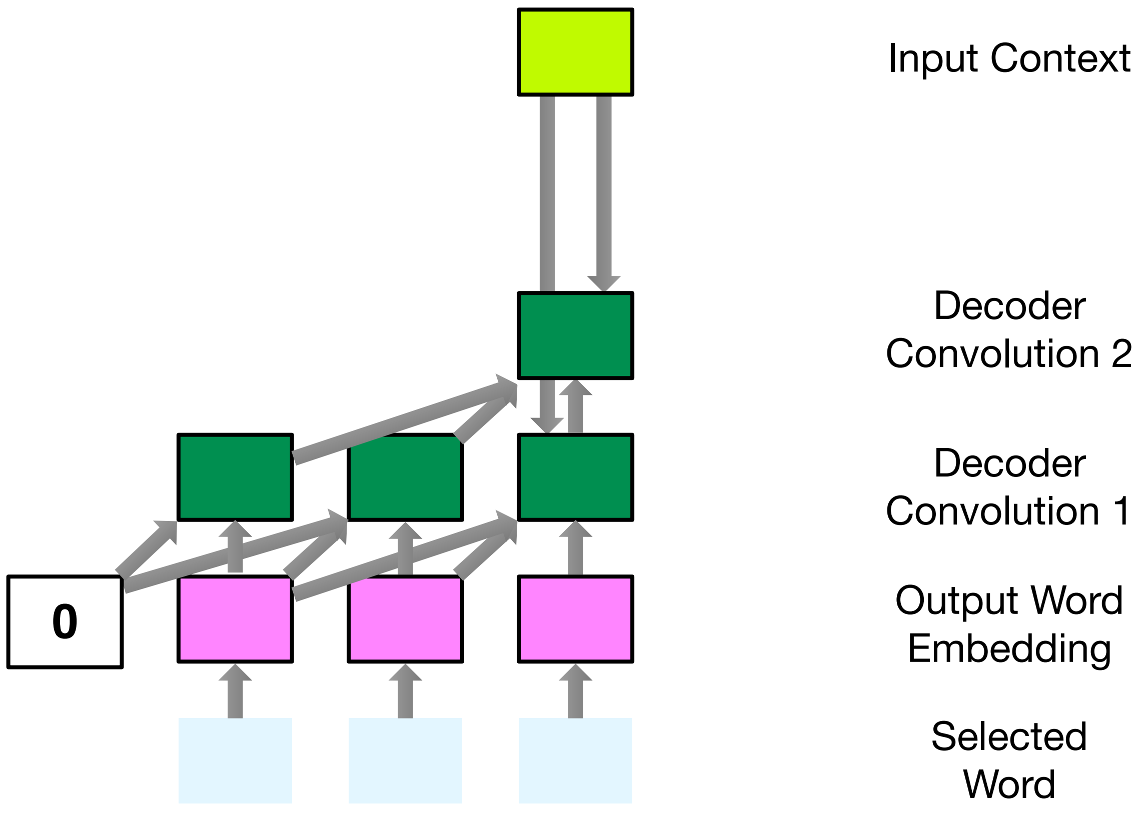
- Similar idea as deep recurrent neural networks
- Good: more parallelizable
- Bad: less context when refining representation of a word

Convolutional Decoder



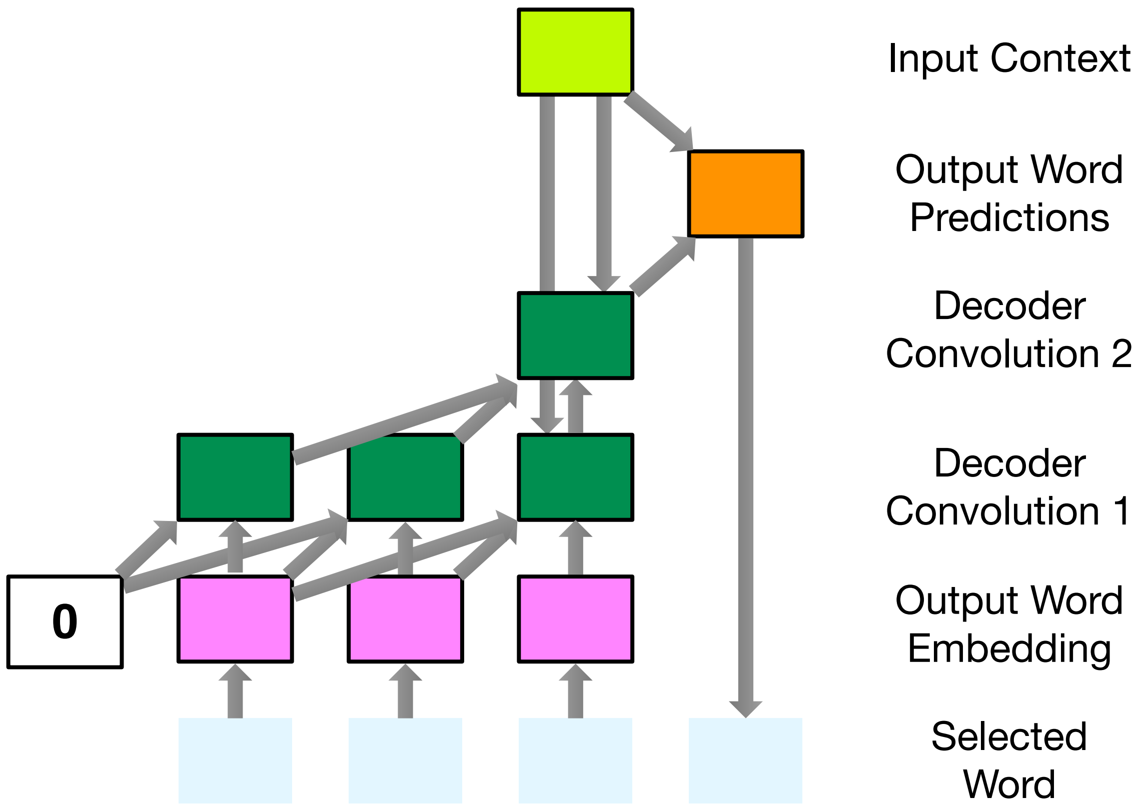
- Convolutions over output words
- Only previously produced output words (still left-to-right decoding)

Convolutional Decoder



- Inclusion of Input context
- Context result of attention mechanism (similar to previous)

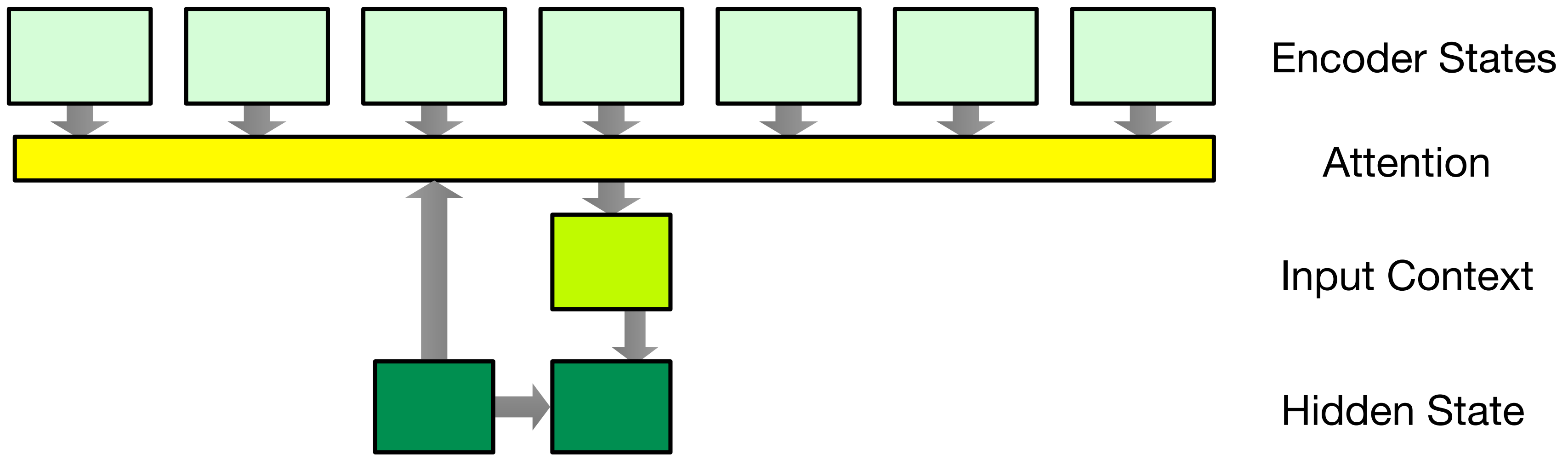
Convolutional Decoder



- Predict output word distribution
- Select output word

self-attention

Attention



- Compute association between last hidden state and encoder states

Attention Math

- Input word representation h_k
- Decoder state s_j
- Computations

$$a_{jk} = \frac{1}{|h|} s_j h_k^T \qquad \text{raw association}$$

$$\alpha_{jk} = \frac{\exp(a_{jk})}{\sum_\kappa \exp(a_{j\kappa})} \qquad \text{normalized association (softmax)}$$

$$\text{self-attention}(h_j) = \sum_k \alpha_{j\kappa} h_k \qquad \text{weighted sum}$$

Self-Attention

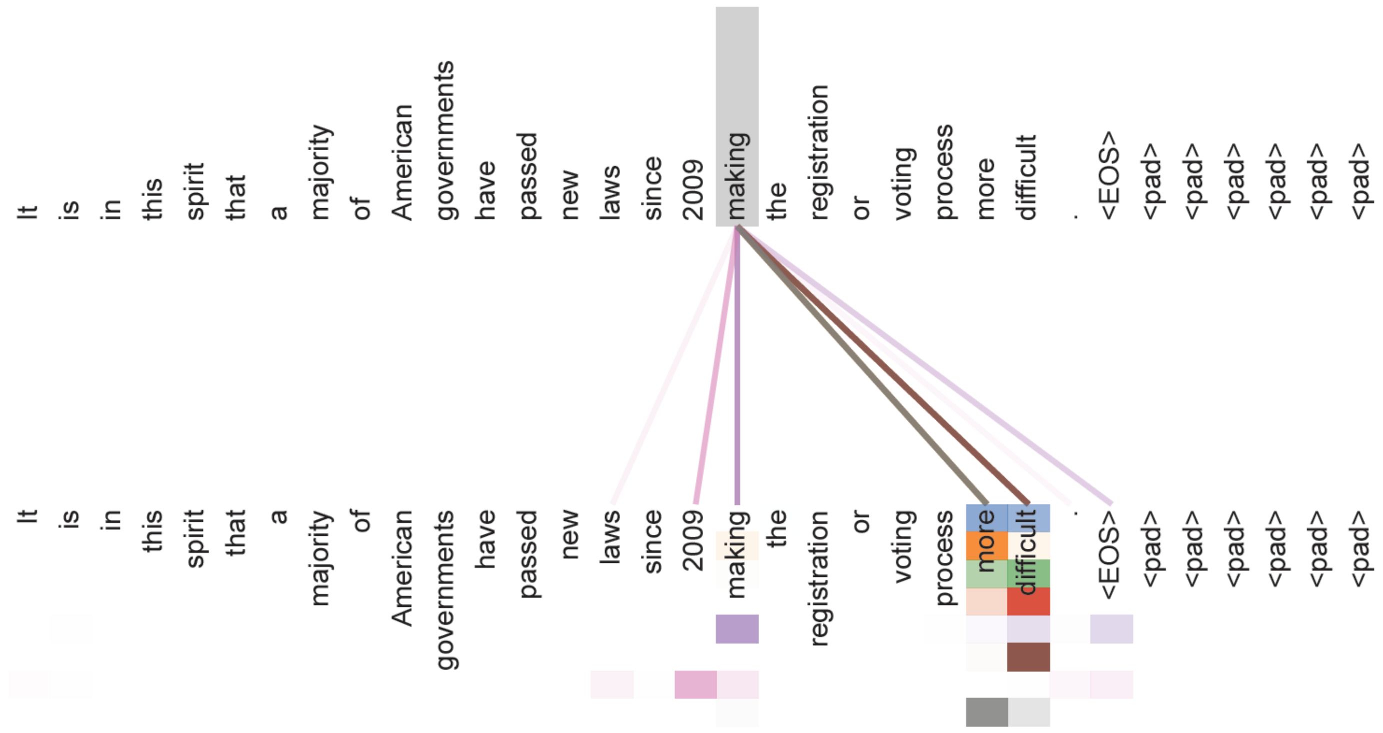
- Attention

$$a_{jk} = \frac{1}{|h|} s_j h_k^T$$

- Self-attention

$$a_{jk} = \frac{1}{|h|} h_j h_k^T$$

Why?

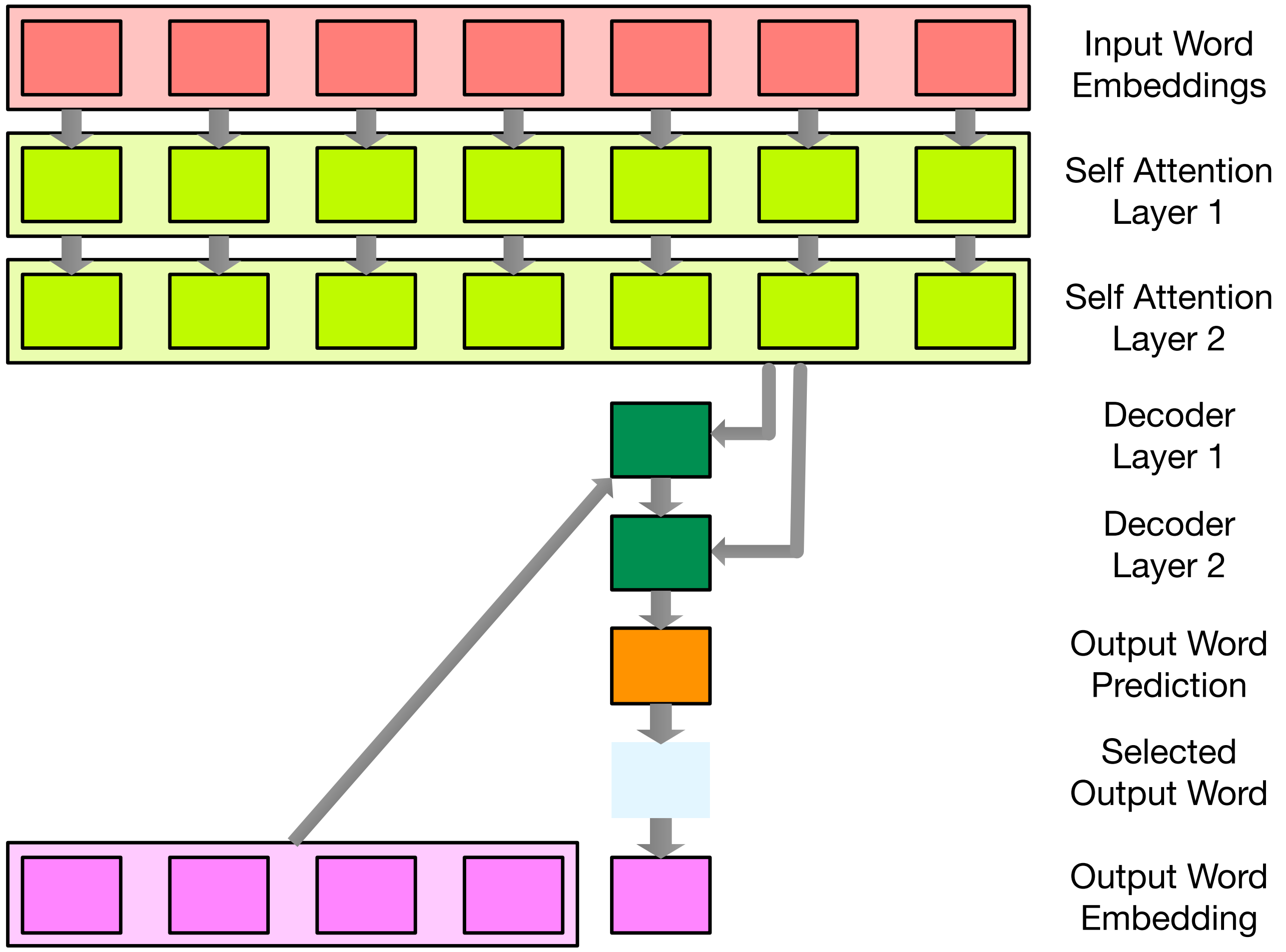


- Refine representation of word with related words

  making ... more difficult refines making

- Good: more parallelizable than recurrent neural network
- Good: wide context when refining representation of a word

Stacked Attention in Decoder



Where Are We Now?

- Recurrent neural network with attention currently dominant model
- Still many challenges
- New proposals in Spring 2017
  - convolutions (Facebook)
  - self-attention (Google)
- Self attention models very successful in WMT 2018
- Open source implementations are available

questions?