
Language Models

Philipp Koehn

22 April 2025



Language models



- **Language models** answer the question:

How likely is a string of English words good English?

- Help with word order

$$p_{\text{LM}}(\text{the house is small}) > p_{\text{LM}}(\text{small the is house})$$

- Help with word choice

$$p_{\text{LM}}(\text{I am going home}) > p_{\text{LM}}(\text{I am going house})$$

N-Gram Language Models



2

- Given: a string of English words $W = w_1, w_2, w_3, \dots, w_n$
- Question: what is $p(W)$?
- Sparse data: Many good English sentences will not have been seen before

→ Decomposing $p(W)$ using the chain rule:

$$p(w_1, w_2, w_3, \dots, w_n) = p(w_1) p(w_2|w_1) p(w_3|w_1, w_2) \dots p(w_n|w_1, w_2, \dots, w_{n-1})$$

(not much gained yet, $p(w_n|w_1, w_2, \dots, w_{n-1})$ is equally sparse)

- **Markov assumption:**

- only previous history matters
 - limited memory: only last k words are included in history (older words less relevant)
- **k th order Markov model**

- For instance 2-gram language model:

$$p(w_1, w_2, w_3, \dots, w_n) \simeq p(w_1) p(w_2|w_1) p(w_3|w_2) \dots p(w_n|w_{n-1})$$

- What is conditioned on, here w_{i-1} is called the **history**

Estimating N-Gram Probabilities



4

- Maximum likelihood estimation

$$p(w_2|w_1) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)}$$

- Collect counts over a large text corpus
- Millions to billions of words are easy to get
(trillions of English words available on the web)

Example: 3-Gram

- Counts for trigrams and estimated word probabilities

the green (total: 1748)			the red (total: 225)			the blue (total: 54)		
word	c.	prob.	word	c.	prob.	word	c.	prob.
paper	801	0.458	cross	123	0.547	box	16	0.296
group	640	0.367	tape	31	0.138	.	6	0.111
light	110	0.063	army	9	0.040	flag	6	0.111
party	27	0.015	card	7	0.031	,	3	0.056
ecu	21	0.012	,	5	0.022	angel	3	0.056

- 225 trigrams in the Europarl corpus start with the red
 - 123 of them end with cross
- maximum likelihood probability is $\frac{123}{225} = 0.547$.

How good is the LM?



6

- A good model assigns a text of real English W a high probability
- This can be also measured with cross entropy:

$$H(W) = -\frac{1}{n} \log_2 p(W_1^n)$$

- Or, **perplexity**

$$\text{perplexity}(W) = 2^{H(W)}$$

Example: 3-Gram

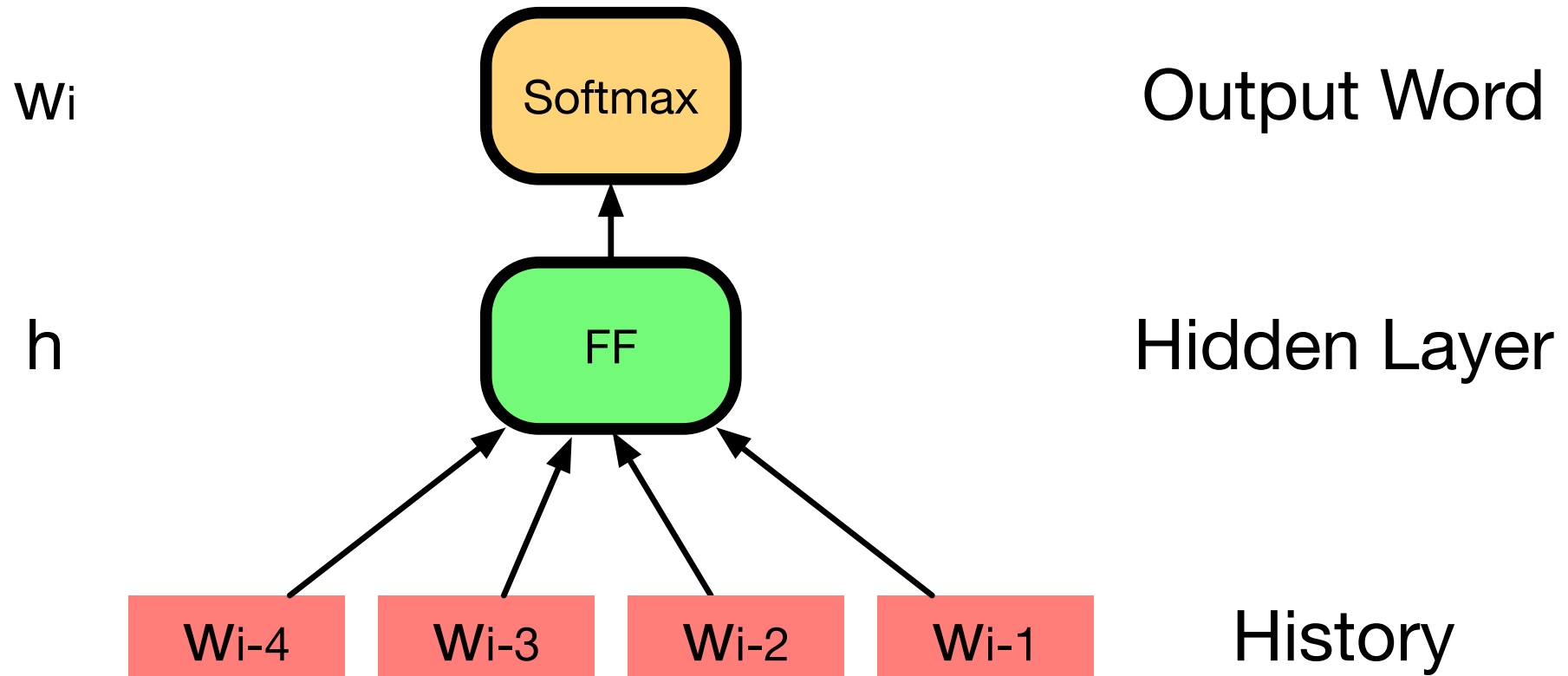
prediction	p_{LM}	$-\log_2 p_{\text{LM}}$
$p_{\text{LM}}(\text{i} \text{</s><s>})$	0.109	3.197
$p_{\text{LM}}(\text{would} \text{<s>i})$	0.144	2.791
$p_{\text{LM}}(\text{like} \text{i would})$	0.489	1.031
$p_{\text{LM}}(\text{to} \text{would like})$	0.905	0.144
$p_{\text{LM}}(\text{commend} \text{like to})$	0.002	8.794
$p_{\text{LM}}(\text{the} \text{to commend})$	0.472	1.084
$p_{\text{LM}}(\text{rapporteur} \text{commend the})$	0.147	2.763
$p_{\text{LM}}(\text{on} \text{the rapporteur})$	0.056	4.150
$p_{\text{LM}}(\text{his} \text{rapporteur on})$	0.194	2.367
$p_{\text{LM}}(\text{work} \text{on his})$	0.089	3.498
$p_{\text{LM}}(. \text{his work})$	0.290	1.785
$p_{\text{LM}}(\text{</s>} \text{work .})$	0.99999	0.000014
average		2.634

Comparison 1–4-Gram

word	unigram	bigram	trigram	4-gram
i	6.684	3.197	3.197	3.197
would	8.342	2.884	2.791	2.791
like	9.129	2.026	1.031	1.290
to	5.081	0.402	0.144	0.113
commend	15.487	12.335	8.794	8.633
the	3.885	1.402	1.084	0.880
rapporteur	10.840	7.319	2.763	2.350
on	6.765	4.140	4.150	1.862
his	10.678	7.316	2.367	1.978
work	9.993	4.816	3.498	2.394
.	4.896	3.020	1.785	1.510
</s>	4.828	0.005	0.000	0.000
average	8.051	4.072	2.634	2.251
perplexity	265.136	16.817	6.206	4.758

neural language models

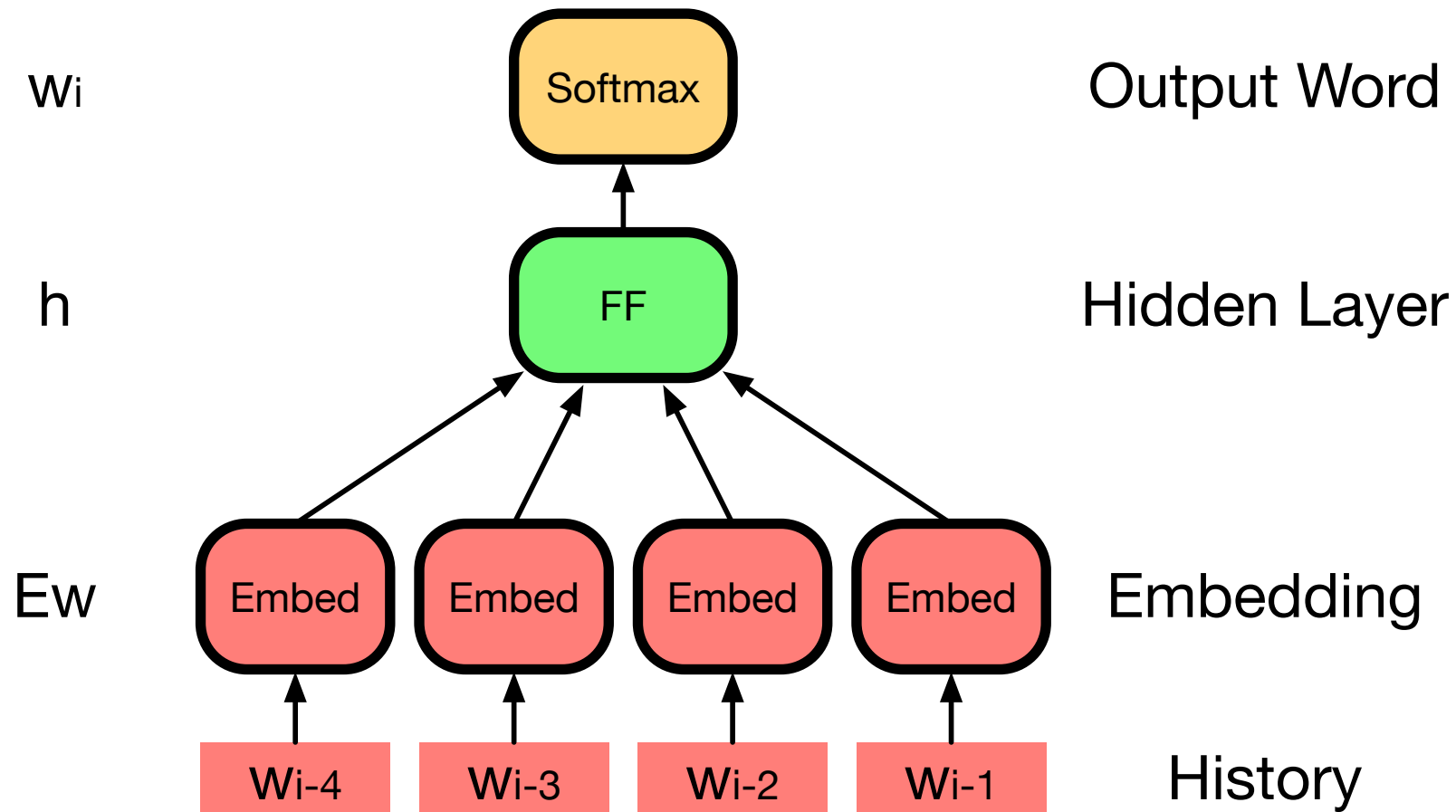
First Sketch



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
 - splitting rare words into subwords
 - character-based models

Add an Embedding Layer

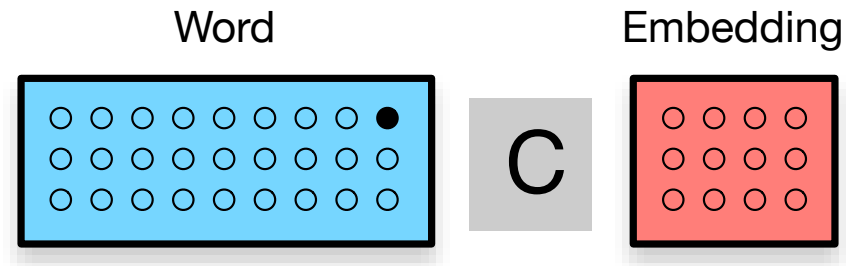


- Map each word first into a lower-dimensional real-valued space
- Shared weight matrix E

Details (Bengio et al., 2003)

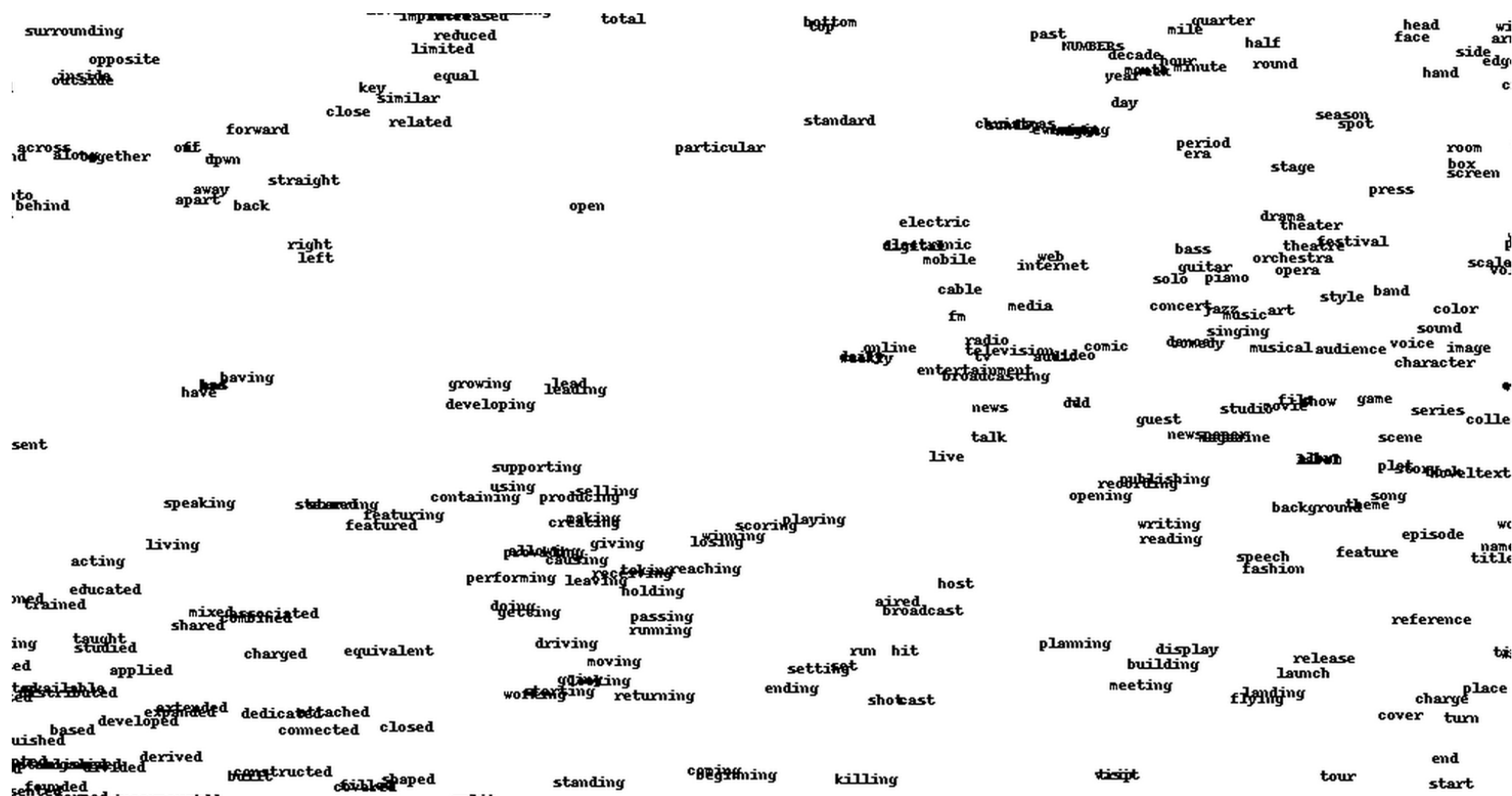
- Add direct connections from embedding layer to output layer
- Activation functions
 - input→embedding: none
 - embedding→hidden: tanh
 - hidden→output: softmax
- Training
 - loop through the entire corpus
 - update between predicted probabilities and 1-hot vector for output word

Word Embeddings



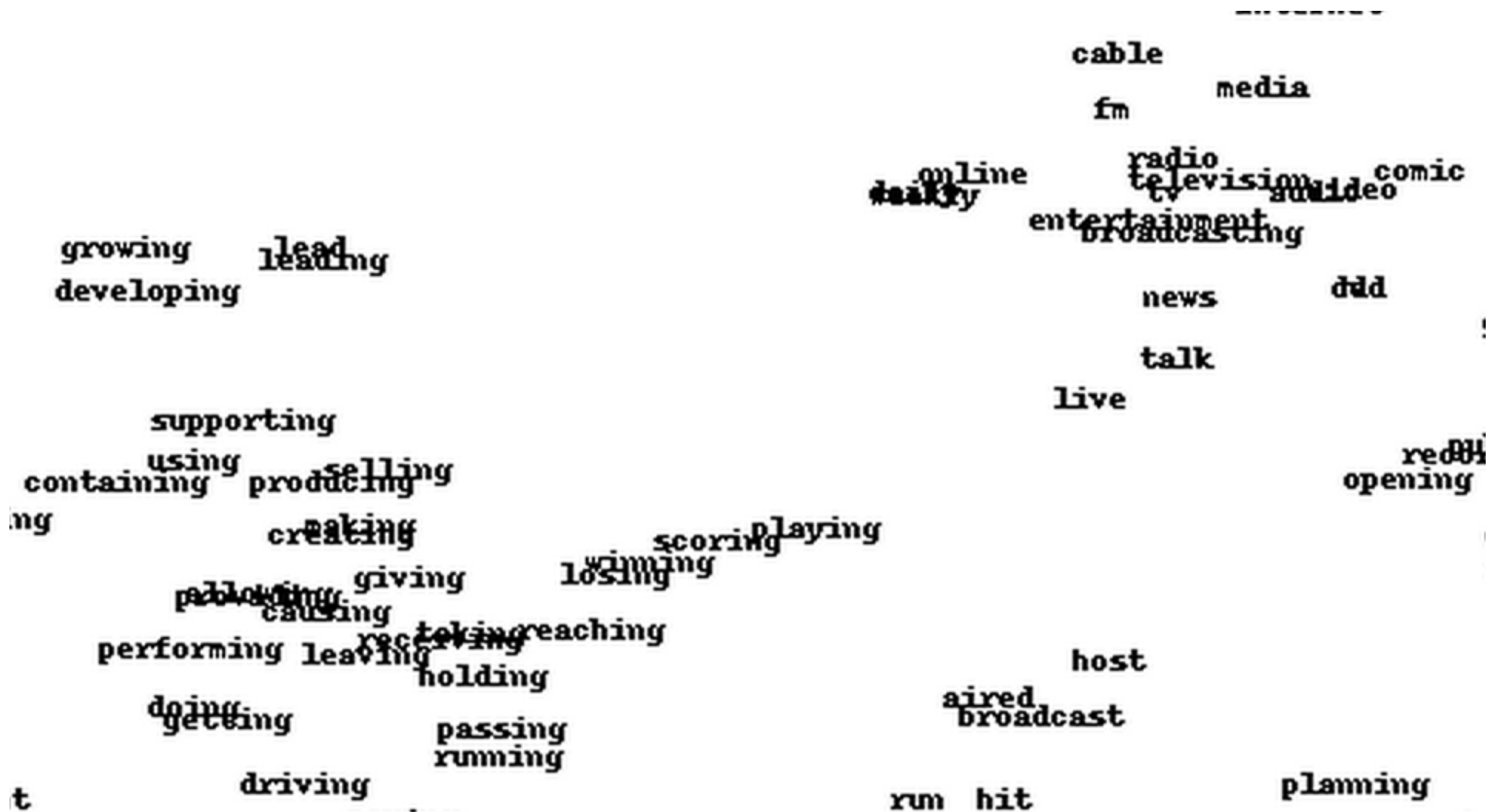
- By-product: embedding of word into continuous space
- Similar contexts \rightarrow similar embedding
- Recall: distributional semantics

Word Embeddings

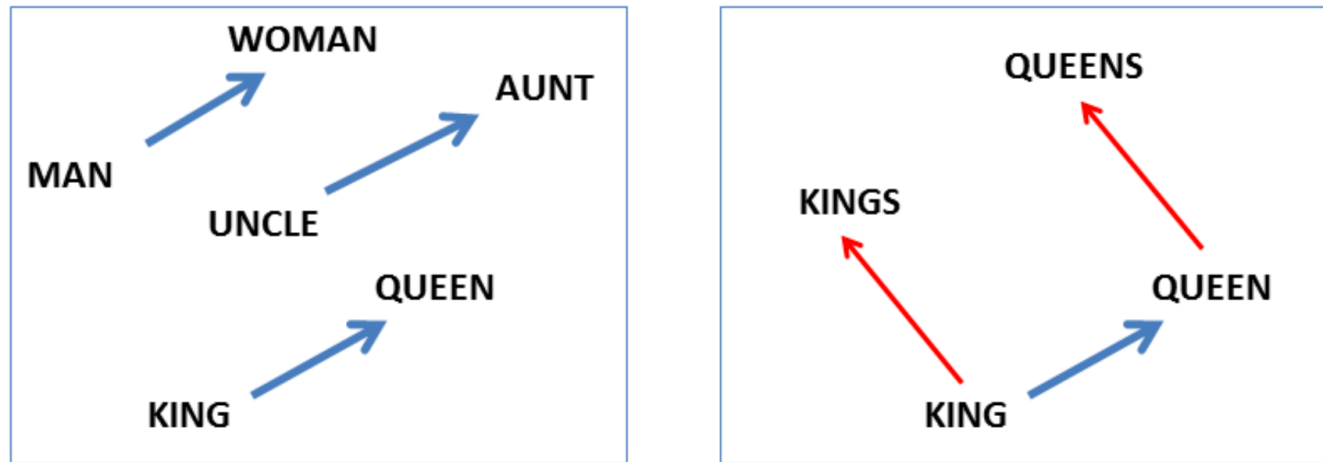


Word Embeddings

16



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

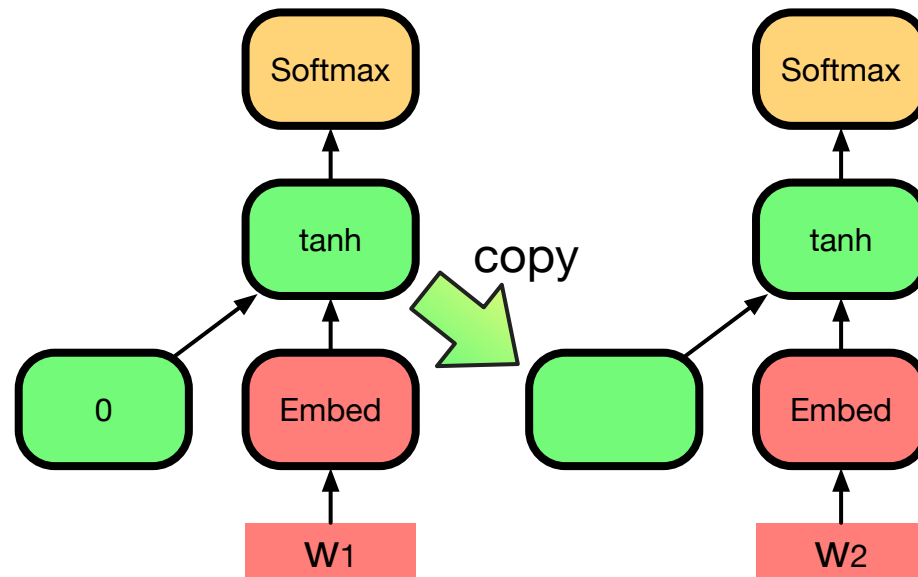
recurrent neural networks

Recurrent Neural Networks



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

Recurrent Neural Networks



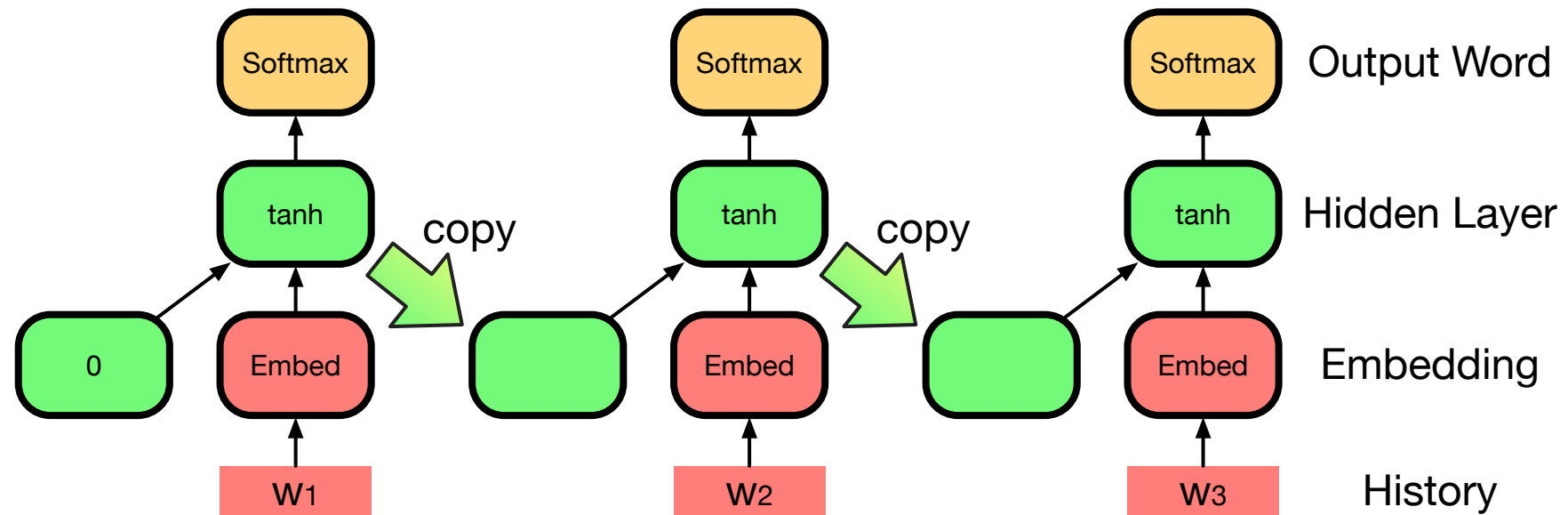
Output Word

Hidden Layer

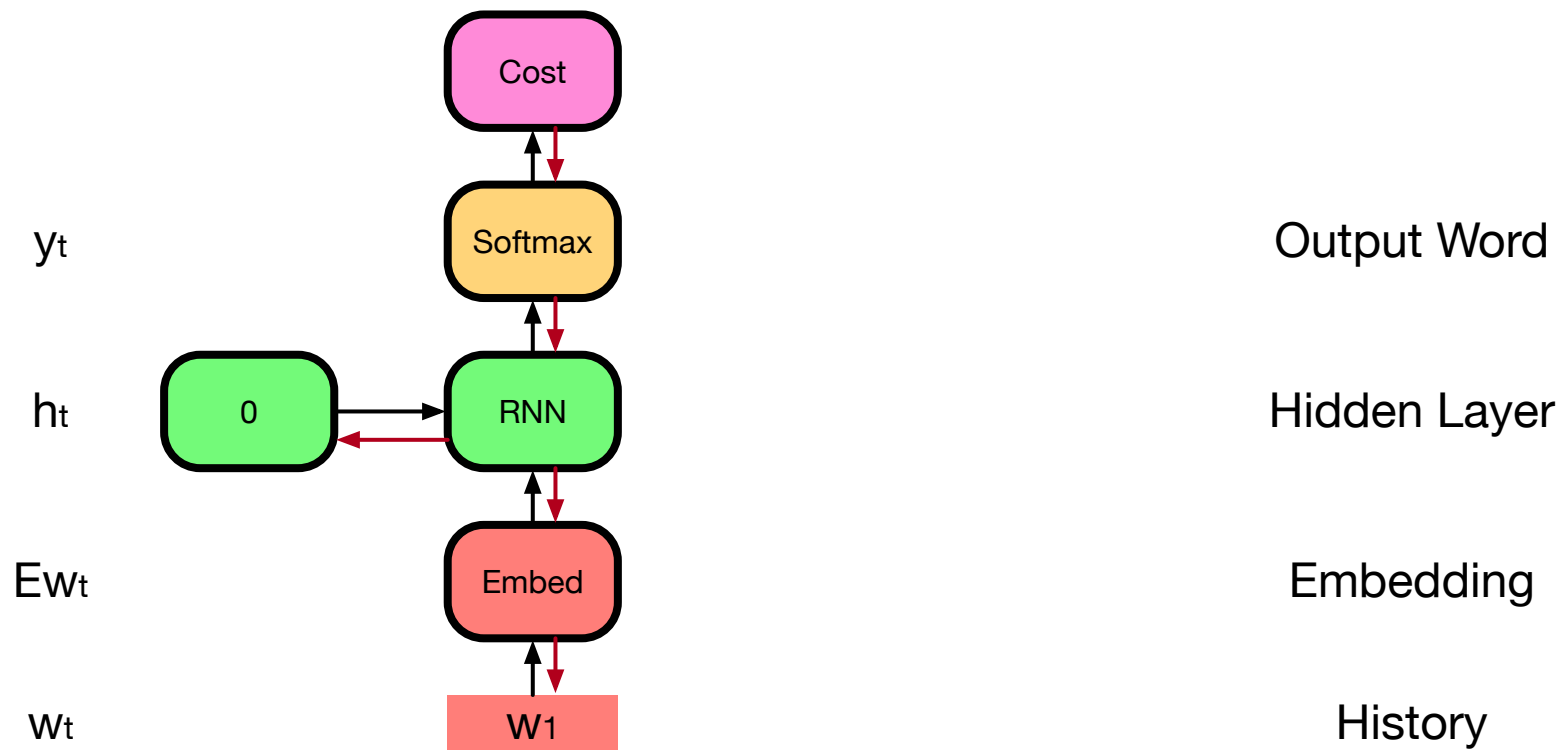
Embedding

History

Recurrent Neural Networks

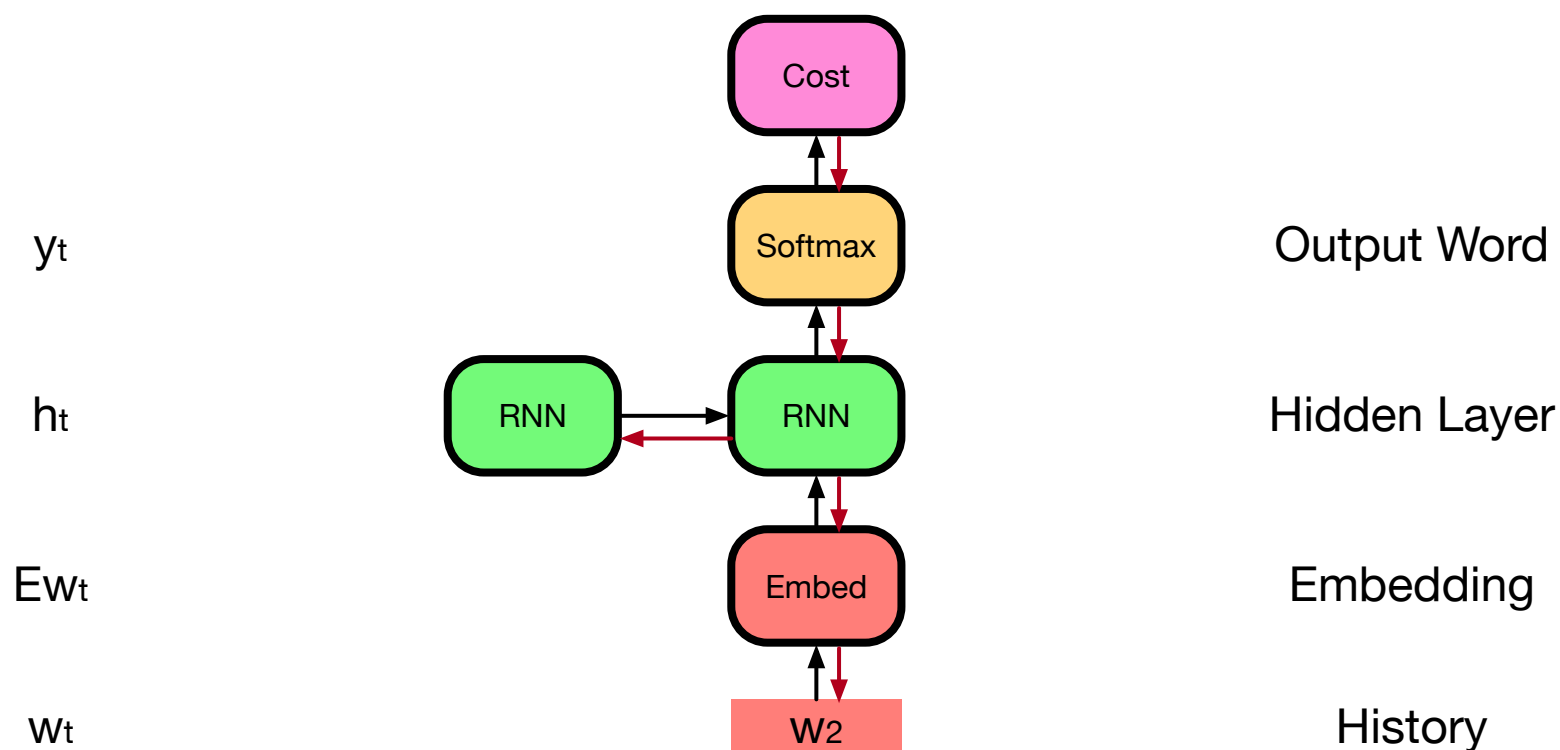


Training



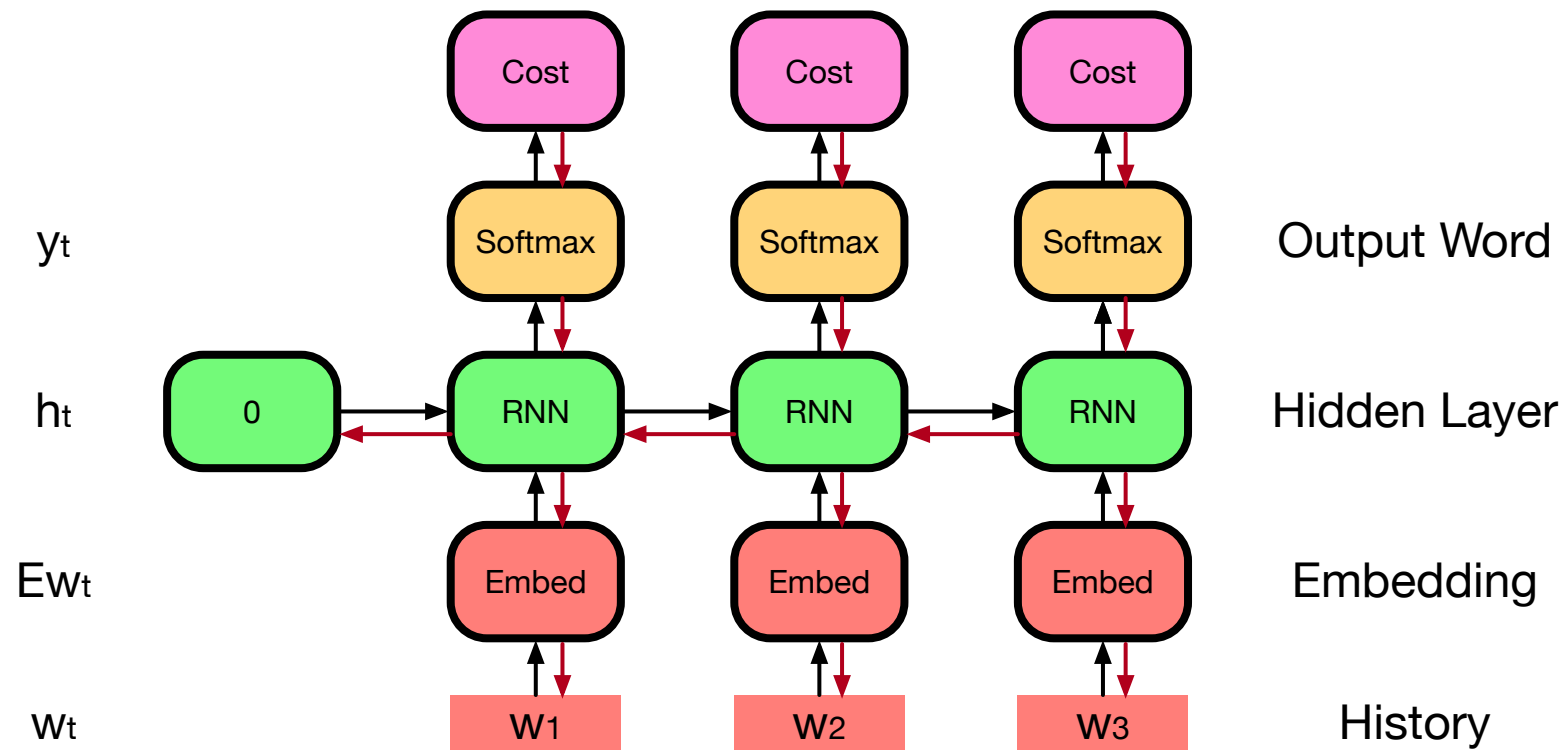
- Process first training example
- Update weights with back-propagation

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

Visualizing Individual Cells

25



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

26



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.
     */
}
```

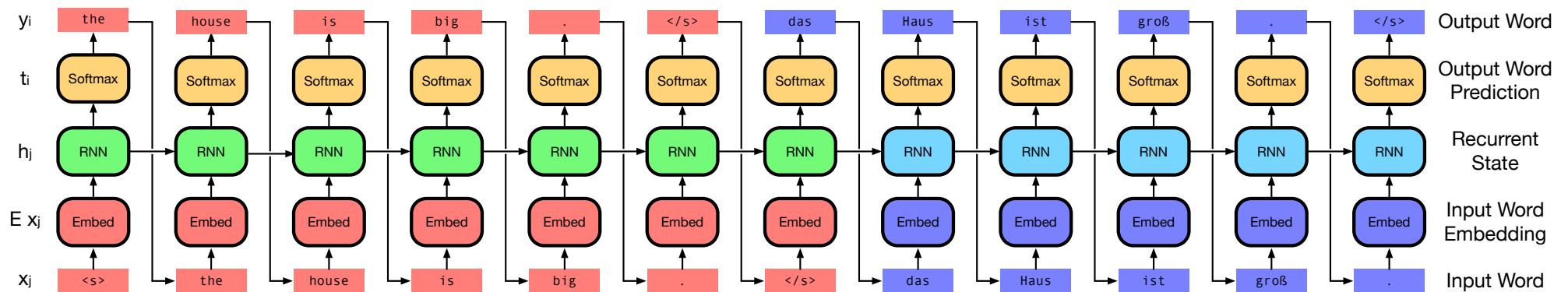
encoder-decoder models

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)

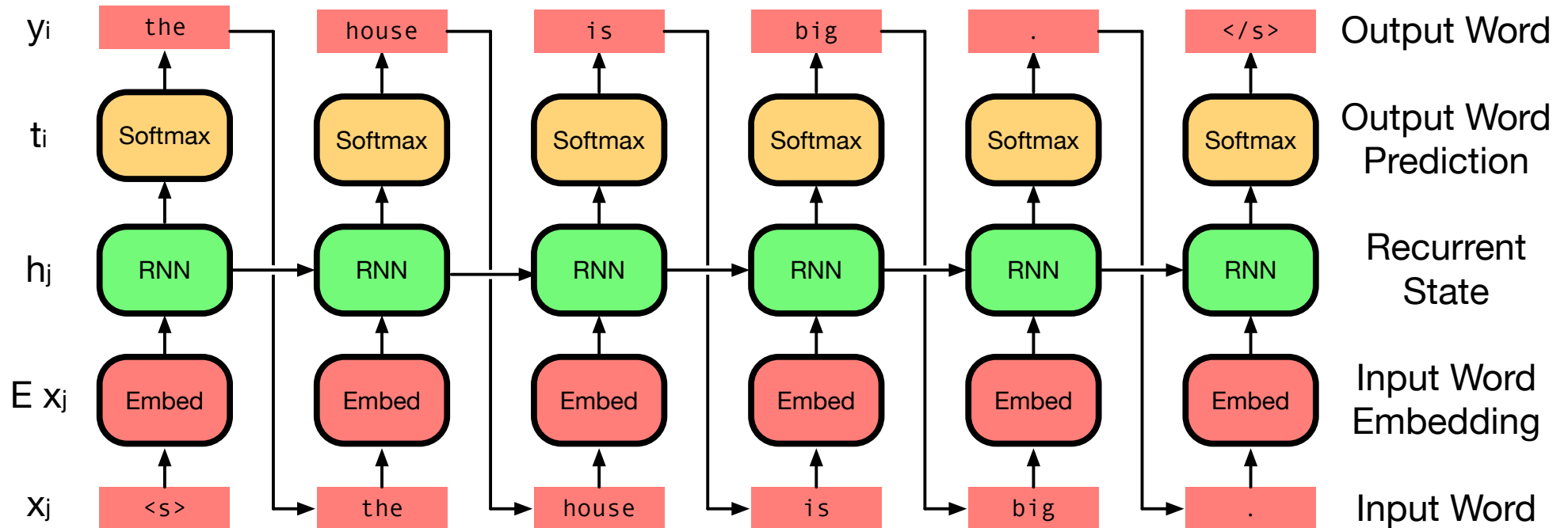
What is Missing?

- Alignment of input words to output words

⇒ Solution: attention mechanism

neural translation model with attention

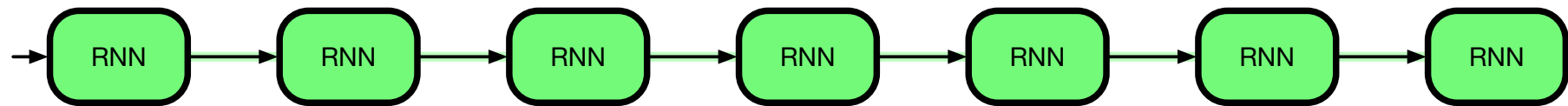
Input Encoding



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

Hidden Language Model States

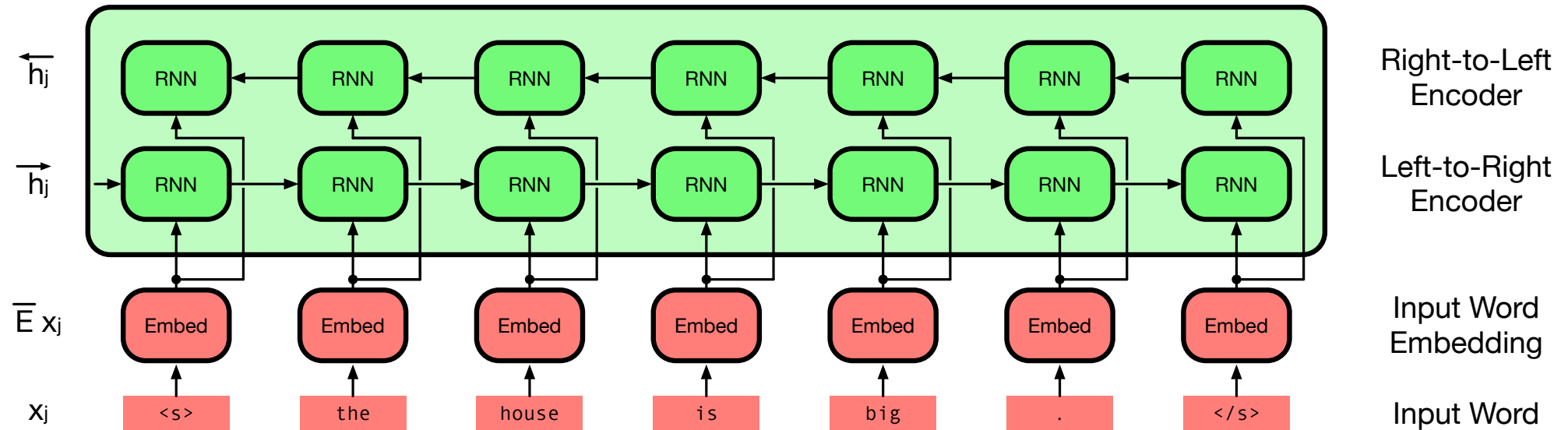
- This gives us the hidden states



- These encode left context for each word
- Same process in reverse: right context for each word

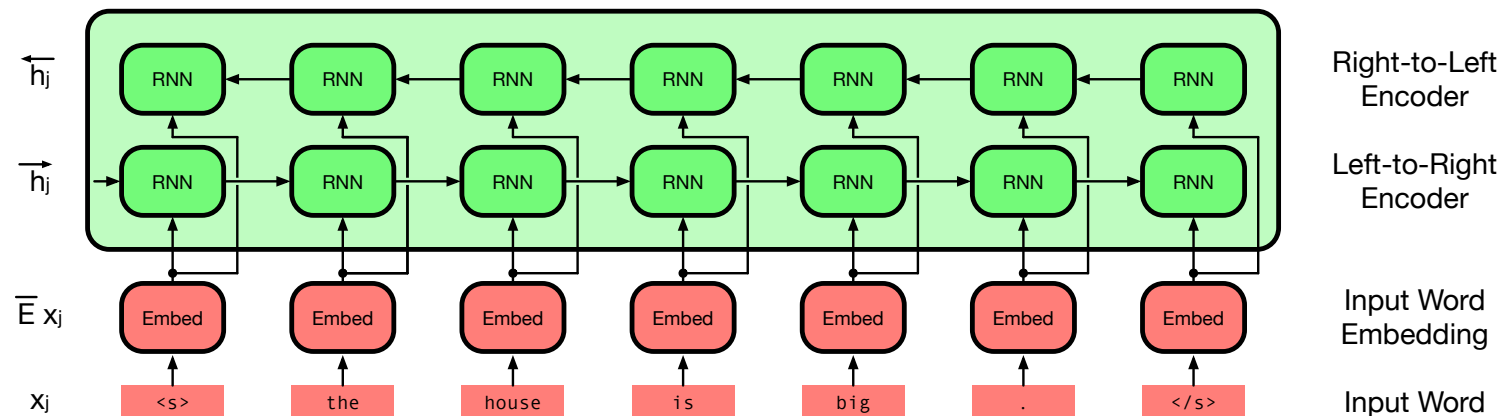


Input Encoder



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

Encoder: Math



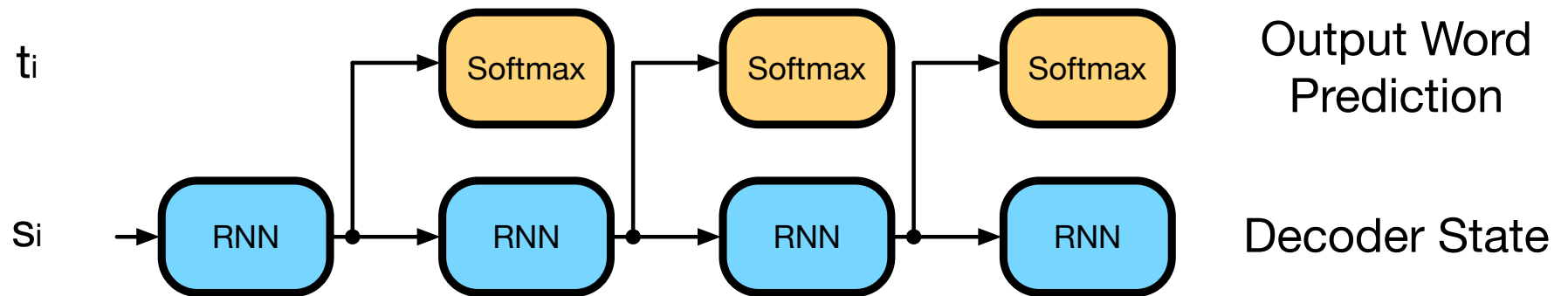
- Input is sequence of words x_j , mapped into embedding space $\bar{E} x_j$
- Bidirectional recurrent neural networks

$$\overleftarrow{h}_j = f(\overleftarrow{h}_{j+1}, \bar{E} x_j)$$
$$\vec{h}_j = f(\vec{h}_{j-1}, \bar{E} x_j)$$

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

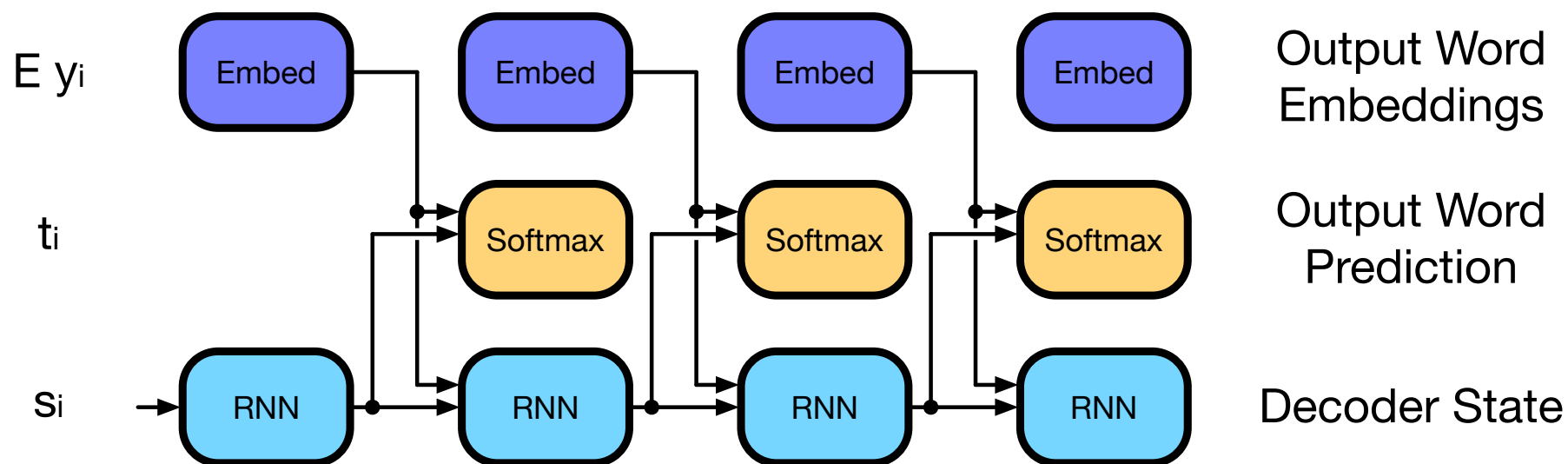
Decoder

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



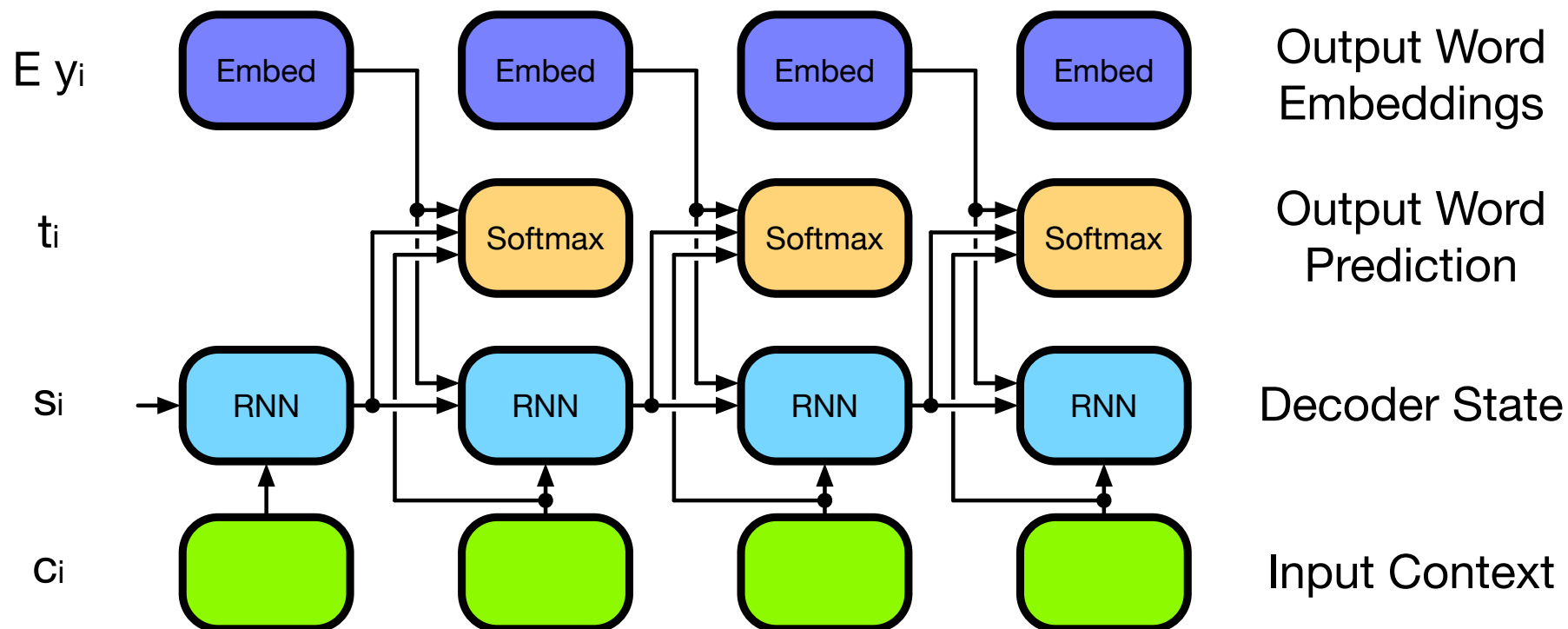
Decoder

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



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

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



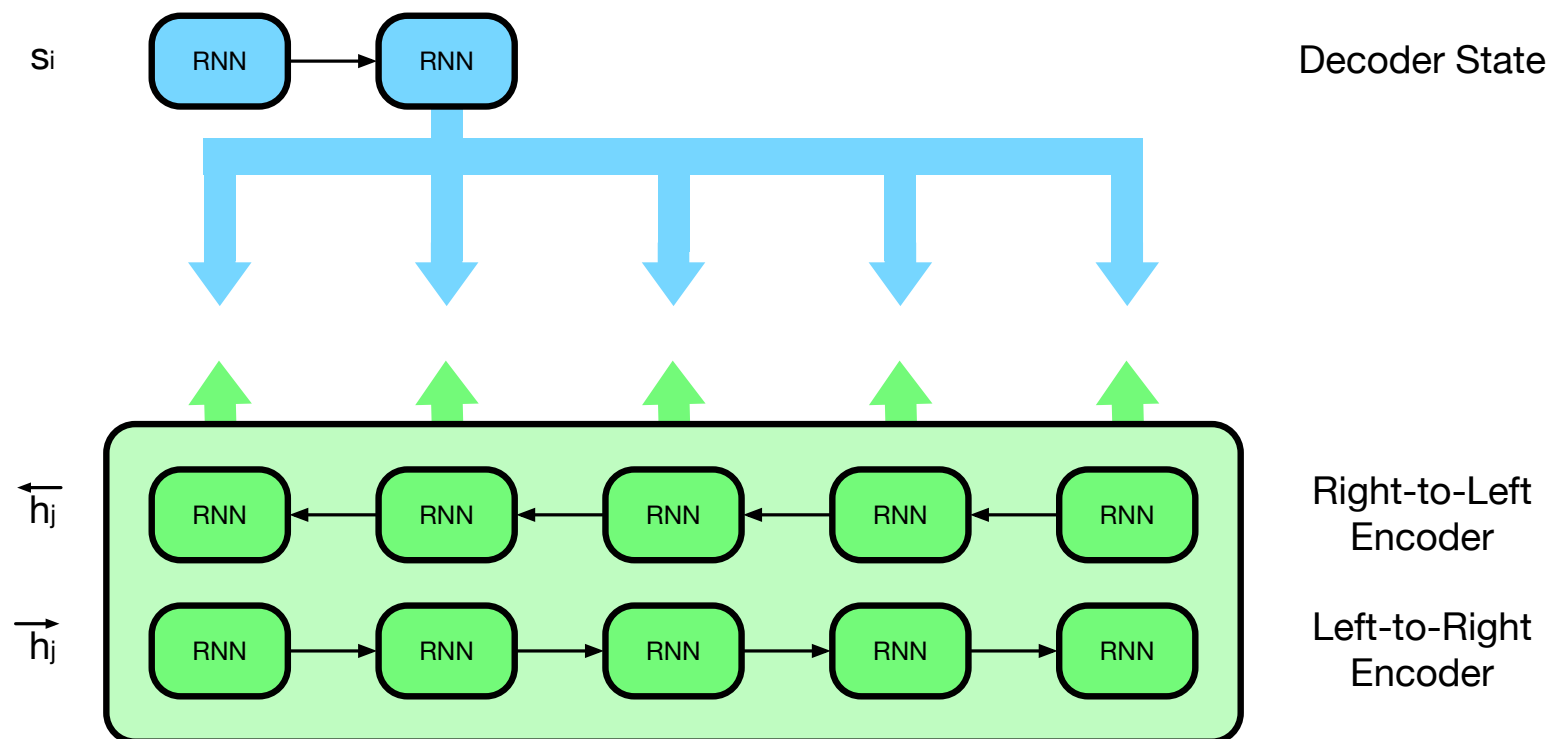
- We feed decisions on output words back into the decoder state
- Decoder state is also informed by the input context

attention

- Machine translation is a structured prediction task
 - output is not a single label
 - output structure needs to be built, word by word
- Relevant information for each word prediction varies
- Human translators pay attention to different parts of the input sentence when translating

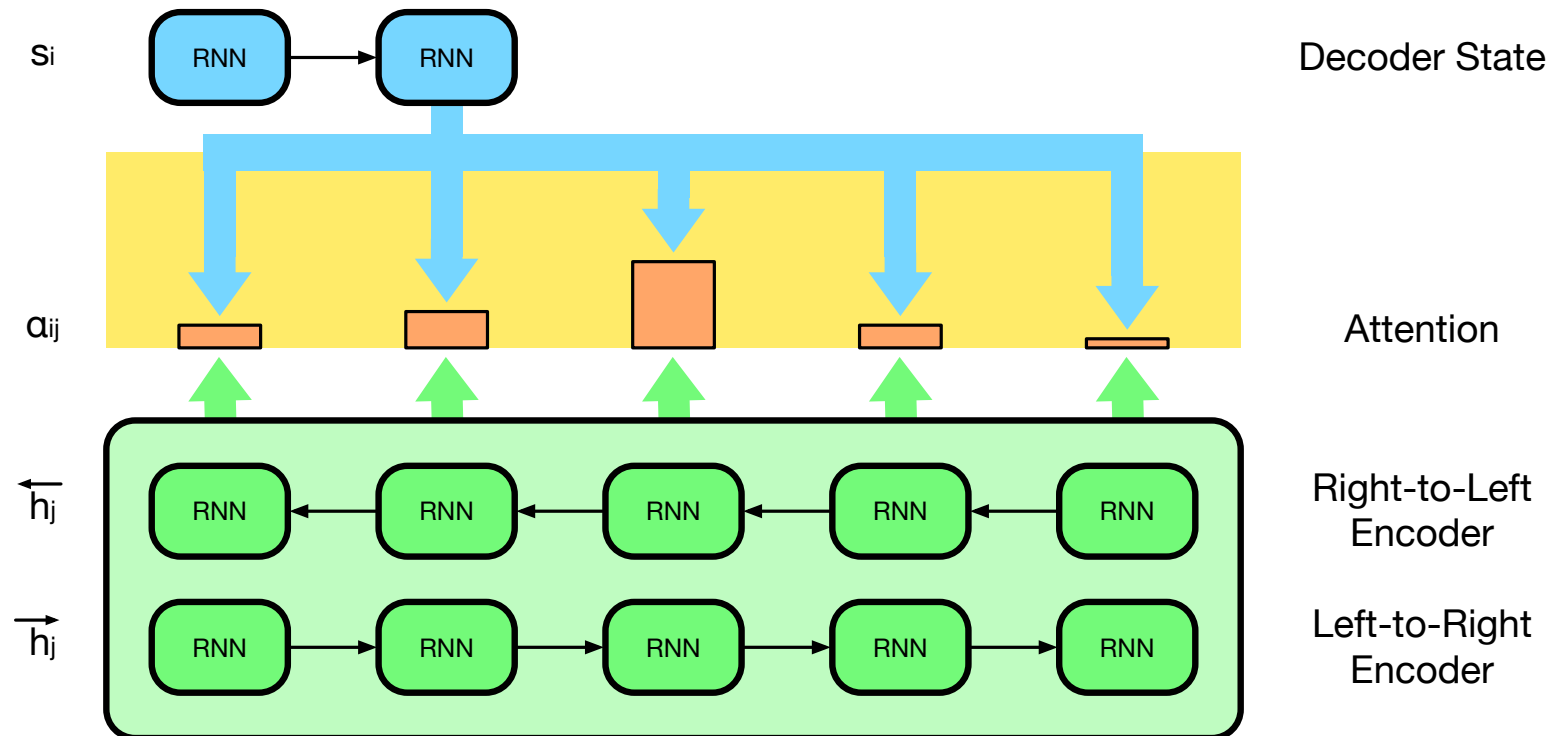
⇒ Attention mechanism

Attention



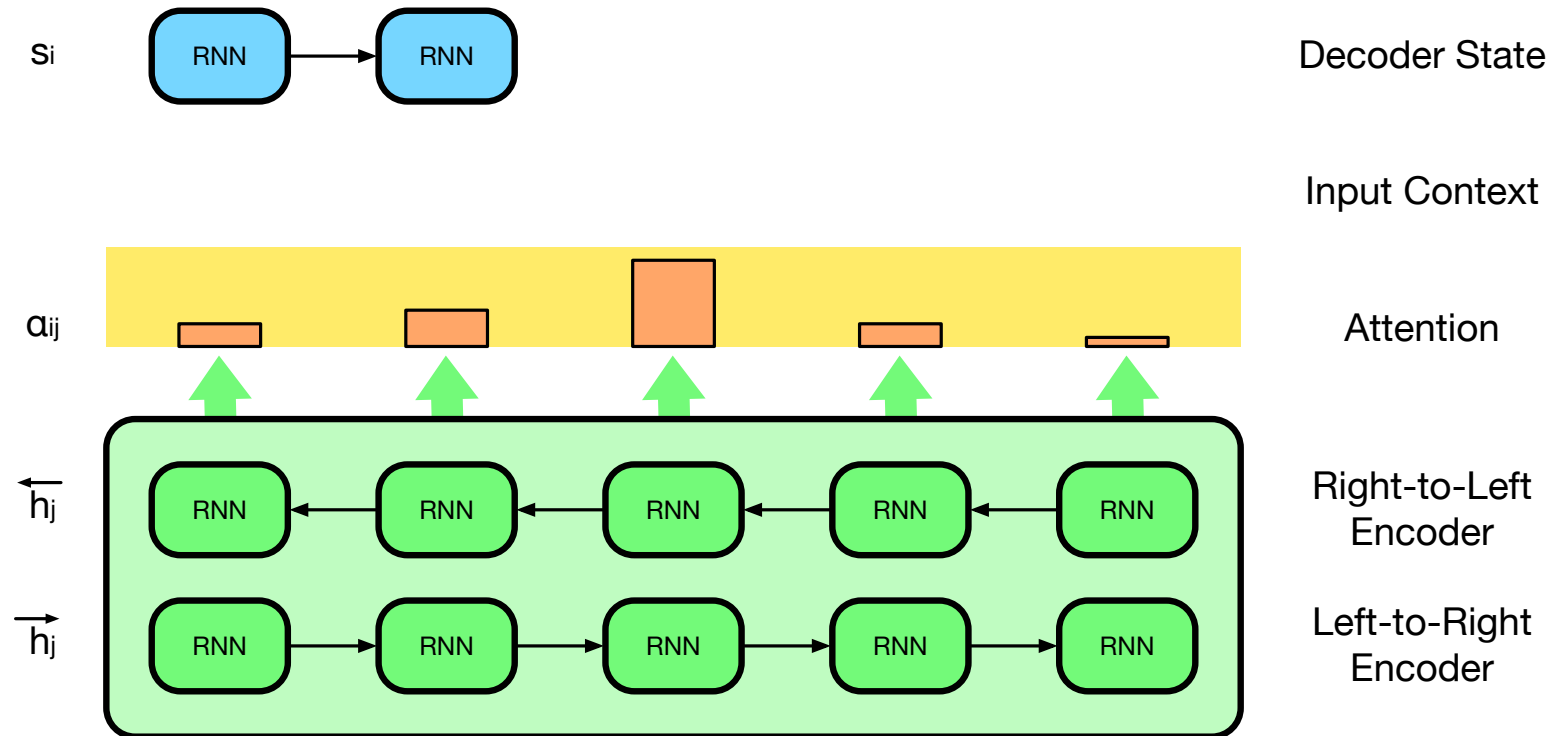
- Given what we have generated so far (decoder hidden state)
- ... which words in the input should we pay attention to (encoder states)?

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)

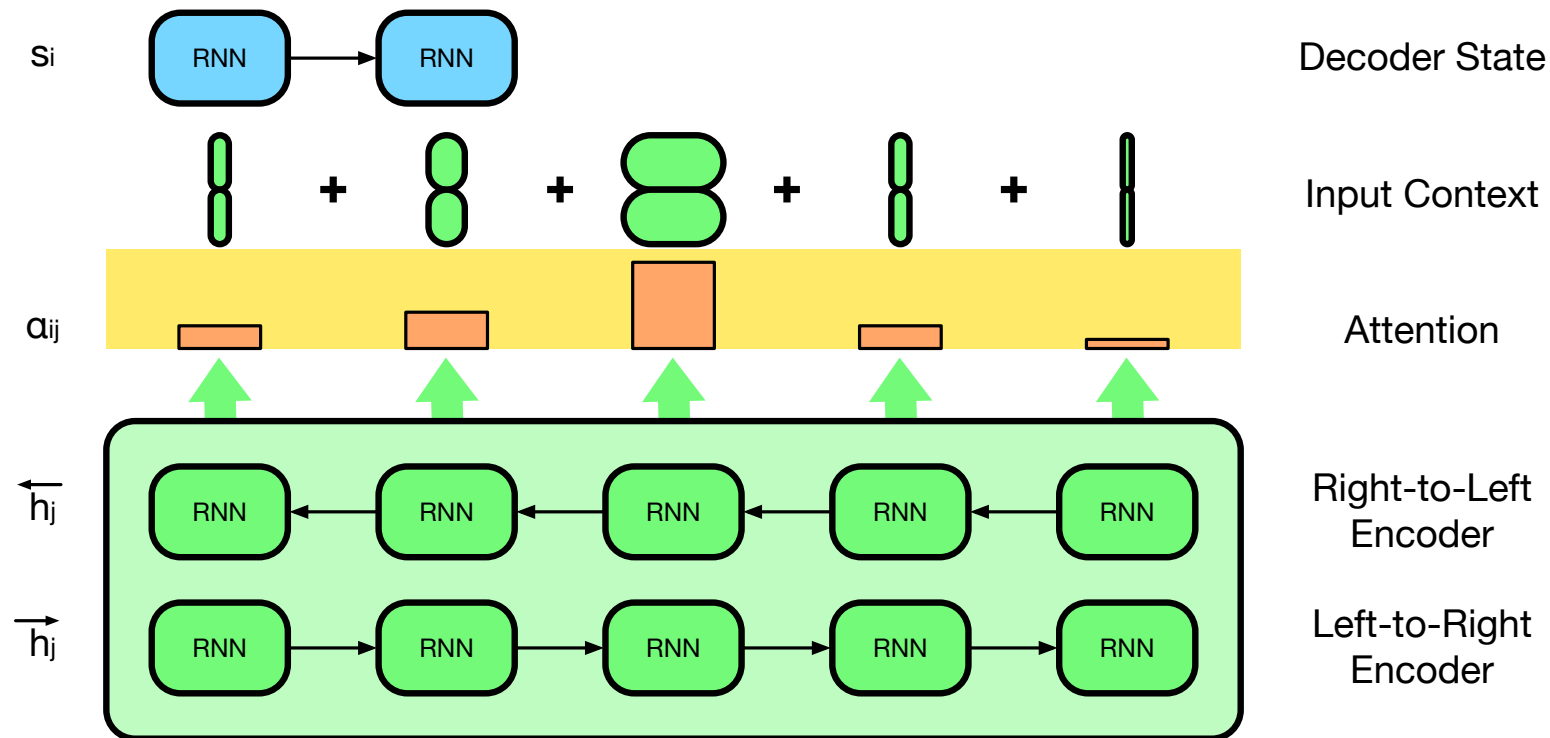
Attention



- Normalize attention (softmax)

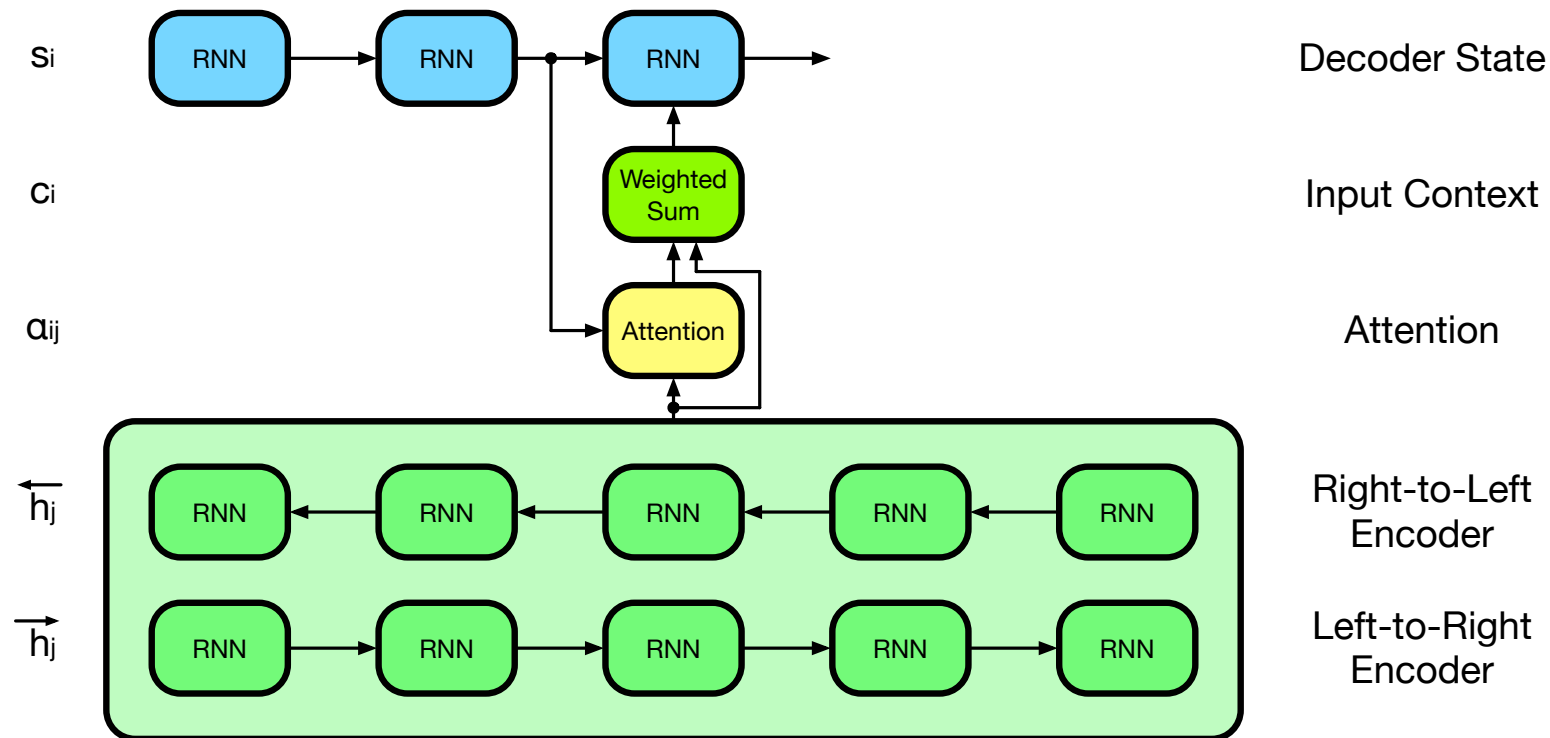
$$\alpha_{ij} = \frac{\exp(a(s_{i-1}, h_j))}{\sum_k \exp(a(s_{i-1}, h_k))}$$

Attention



- 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

- Attention mechanism in neural translation model (Bahdanau et al., 2015)
 - previous hidden state s_{i-1}
 - input word embedding h_j
 - trainable parameters b, W_a, U_a, v_a

$$a(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} + U_a h_j + b)$$

- Other ways to compute attention
 - Dot product: $a(s_{i-1}, h_j) = s_{i-1}^T h_j$
 - Scaled dot product: $a(s_{i-1}, h_j) = \frac{1}{\sqrt{|h_j|}} s_{i-1}^T h_j$
 - General: $a(s_{i-1}, h_j) = s_{i-1}^T W_a h_j$
 - Local: $a(s_{i-1}) = W_a s_{i-1}$

Attention of Luong et al. (2015)



- Luong et al. (2015) demonstrate good results with the dot product

$$a(s_{i-1}, h_j) = s_{i-1}^T h_j$$

- No trainable parameters
- Additional changes
- Currently more popular

- Three element

Query : decoder state

Key : encoder state

Value : encoder state

- Intuition

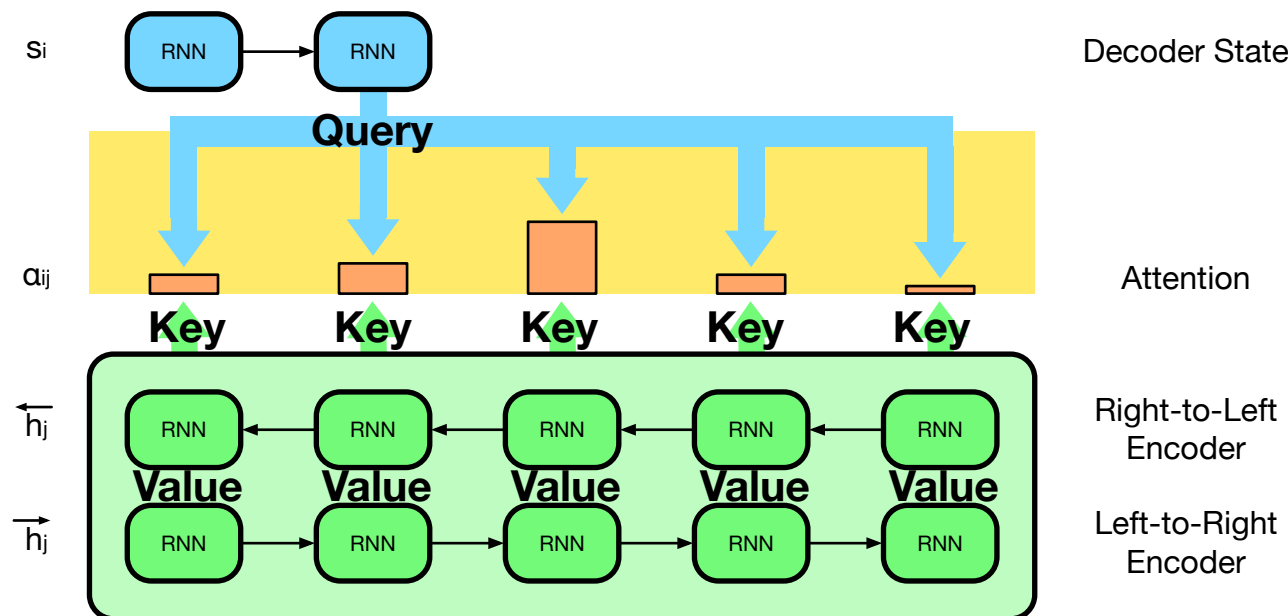
- given a query (the decoder state)
- we check how well it matches keys in the database (the encoder states)
- and then use the matching score to scale the retrieved value (also the encoder state)

- Computation

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

General View of Dot-Product Attention

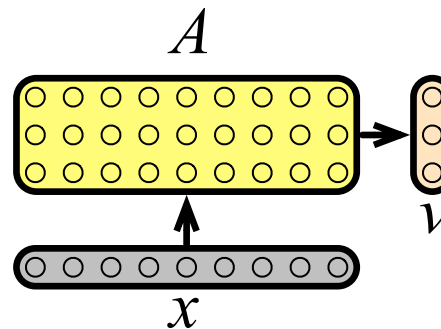
$$\text{Attention}(Q, K, V)$$



- Query: encoder state, Key and Value: decoder state

Dimensionality Reduction

- Instead of simple dot product of query and key vectors QK^T ...
- First multiply with weight matrices W^Q and W^K
- Scales down the size of the vectors

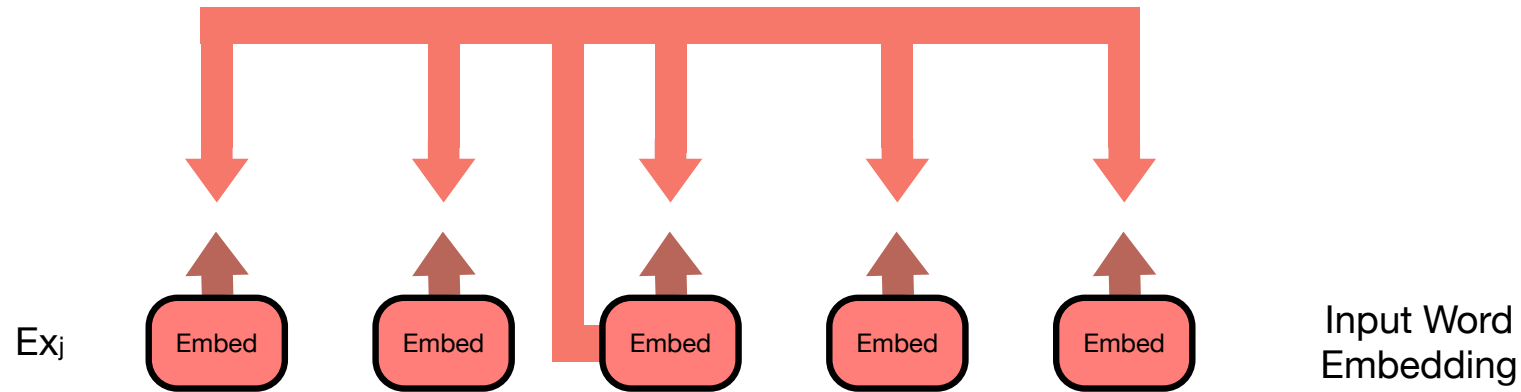


- New computation: $\text{Attention}(QW^Q, KW^K, V)$

self attention

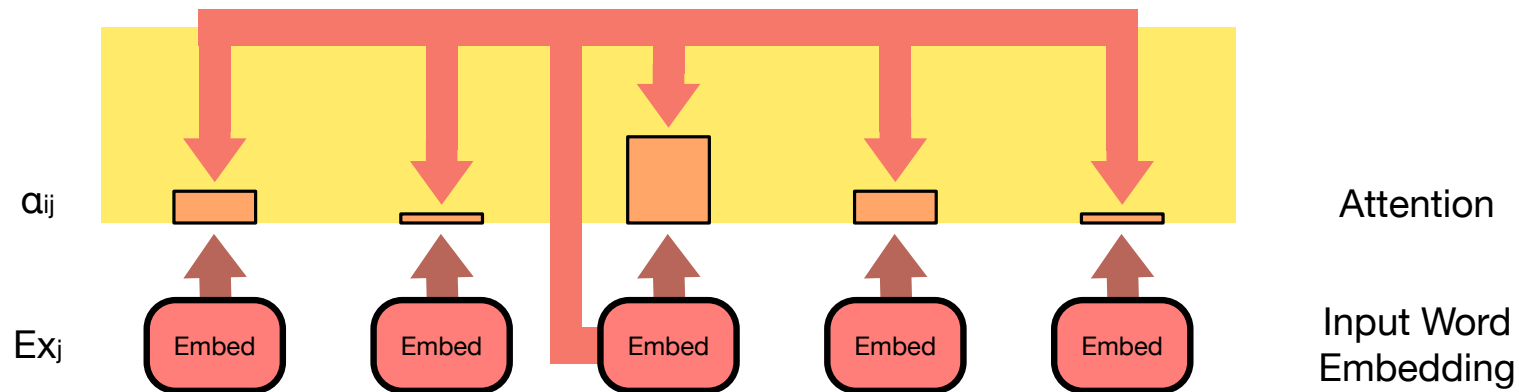
- Finally, a very different take at attention
- Motivation so far: need for alignment between input words and output words
- Now: refine representation of input words in the encoder
 - representation of an input word mostly depends on itself
 - but also informed by the surrounding context
 - previously: recurrent neural networks (considers left or right context)
 - now: attention mechanism
- Self attention:
Which of the surrounding words is most relevant to refine representation?

Self Attention



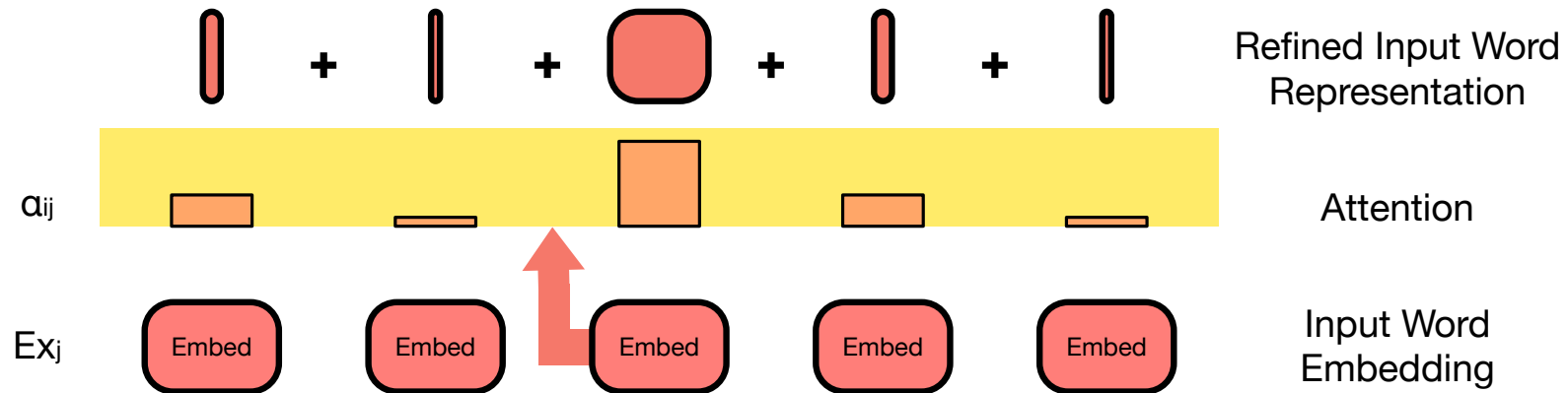
- Given: input word embeddings
- Task: consider how each should be refined in view of others
- Needed: how much attention to pay to others

Self Attention



- Computation of attention weights as before
 - Key: word embedding (or generally: encoder state for word H)
 - Query: word embedding (or generally: encoder state for word H)
- Again, multiple with weight matrices: $Q=HW^Q$ and $K=HW^K$
- Attention weights: QK^T

Self Attention



- Full self attention

$$\text{self-attention}(H) = \text{Attention}(HW^Q, HW^K, H)$$

- Resulting vector uses weighted context words

Multi-Head Attention

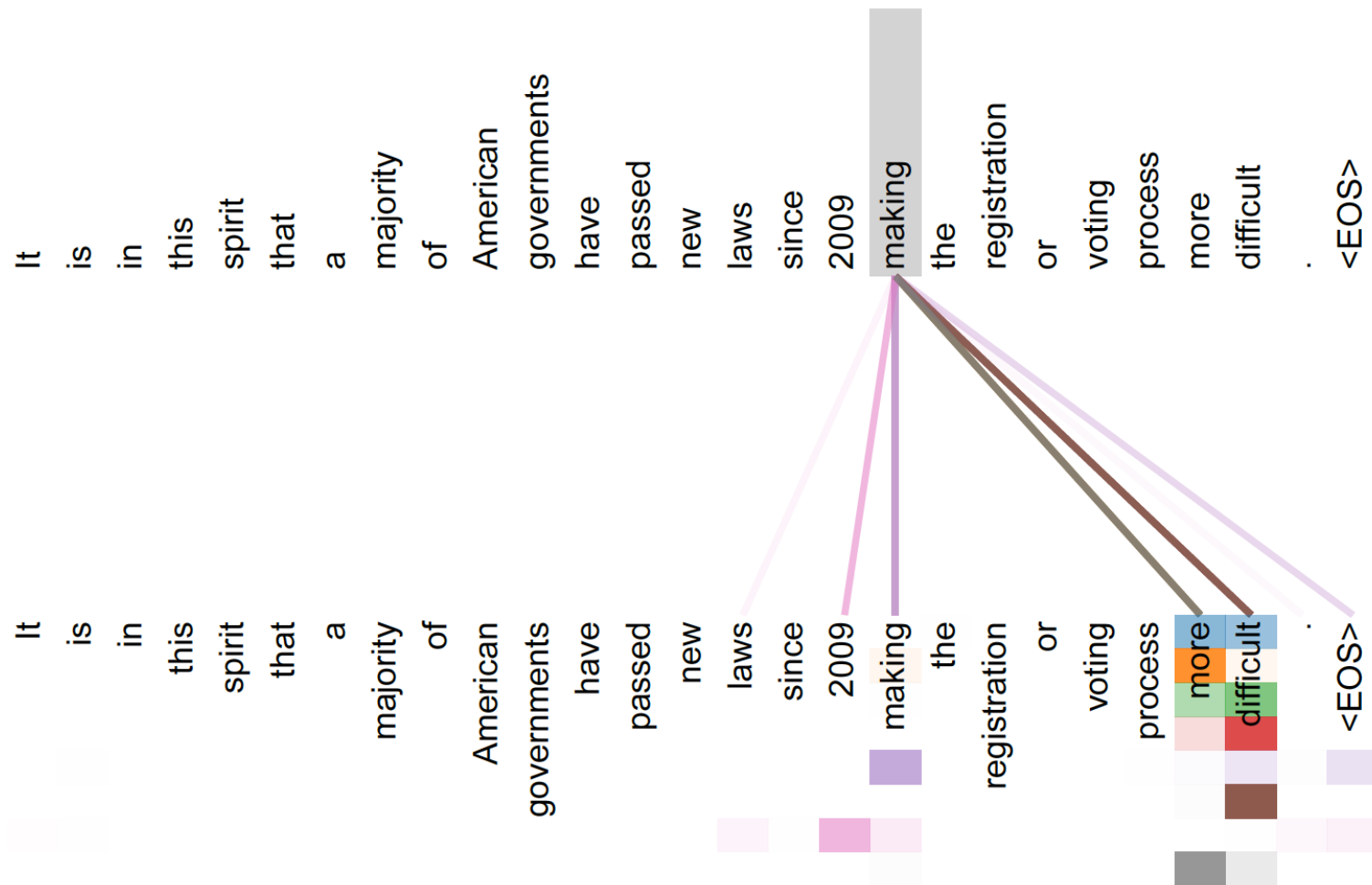
- Add redundancy
 - say, 16 attention weights
 - each based on its own parameters W_i^Q, W_i^K, W_i^V ■

- Formally:

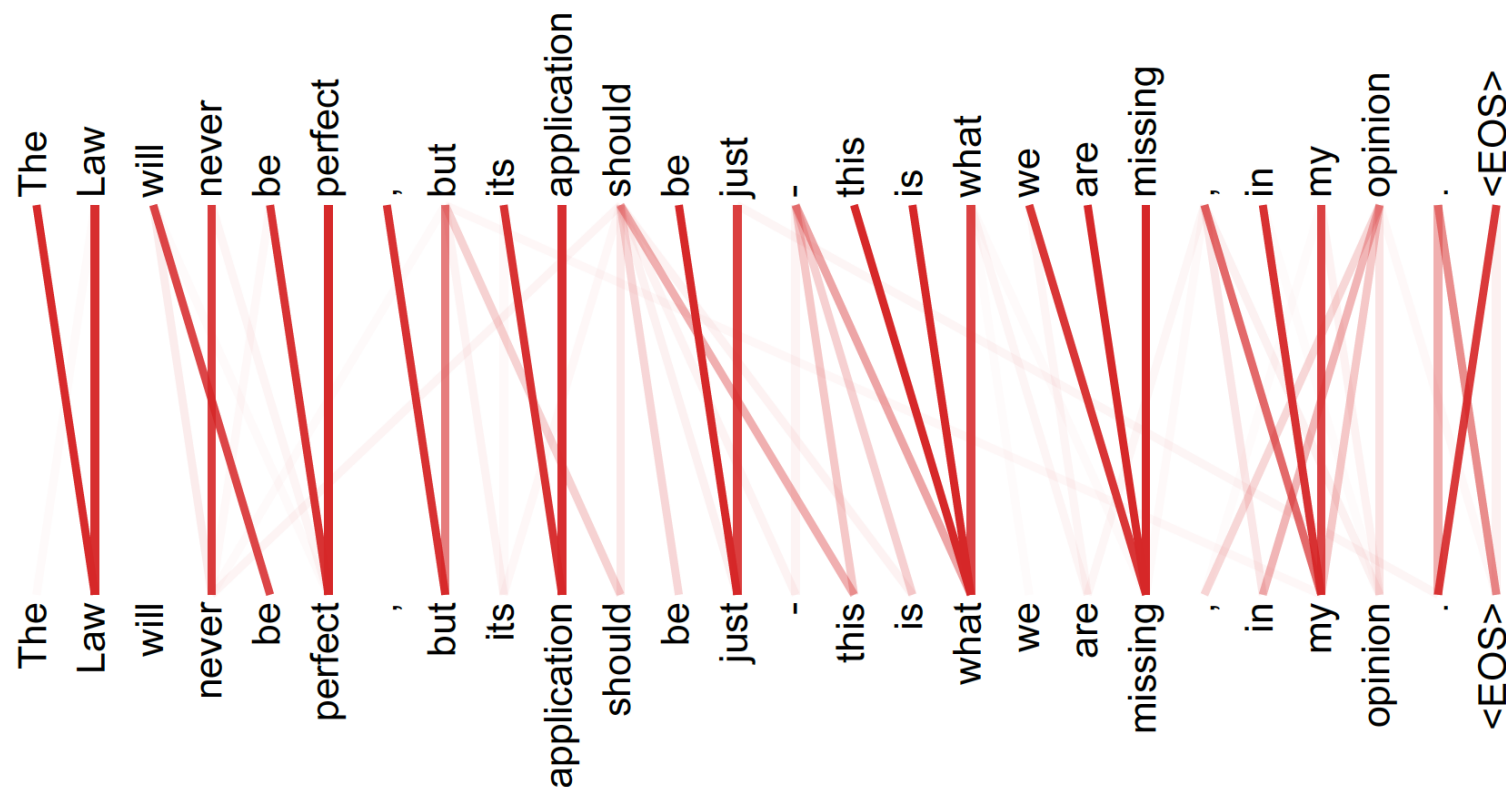
$$\begin{aligned}\text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\ \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O\end{aligned}$$

- Multi-head attention is a form of ensembling

Multi-Head Attention



Multi-Head Attention



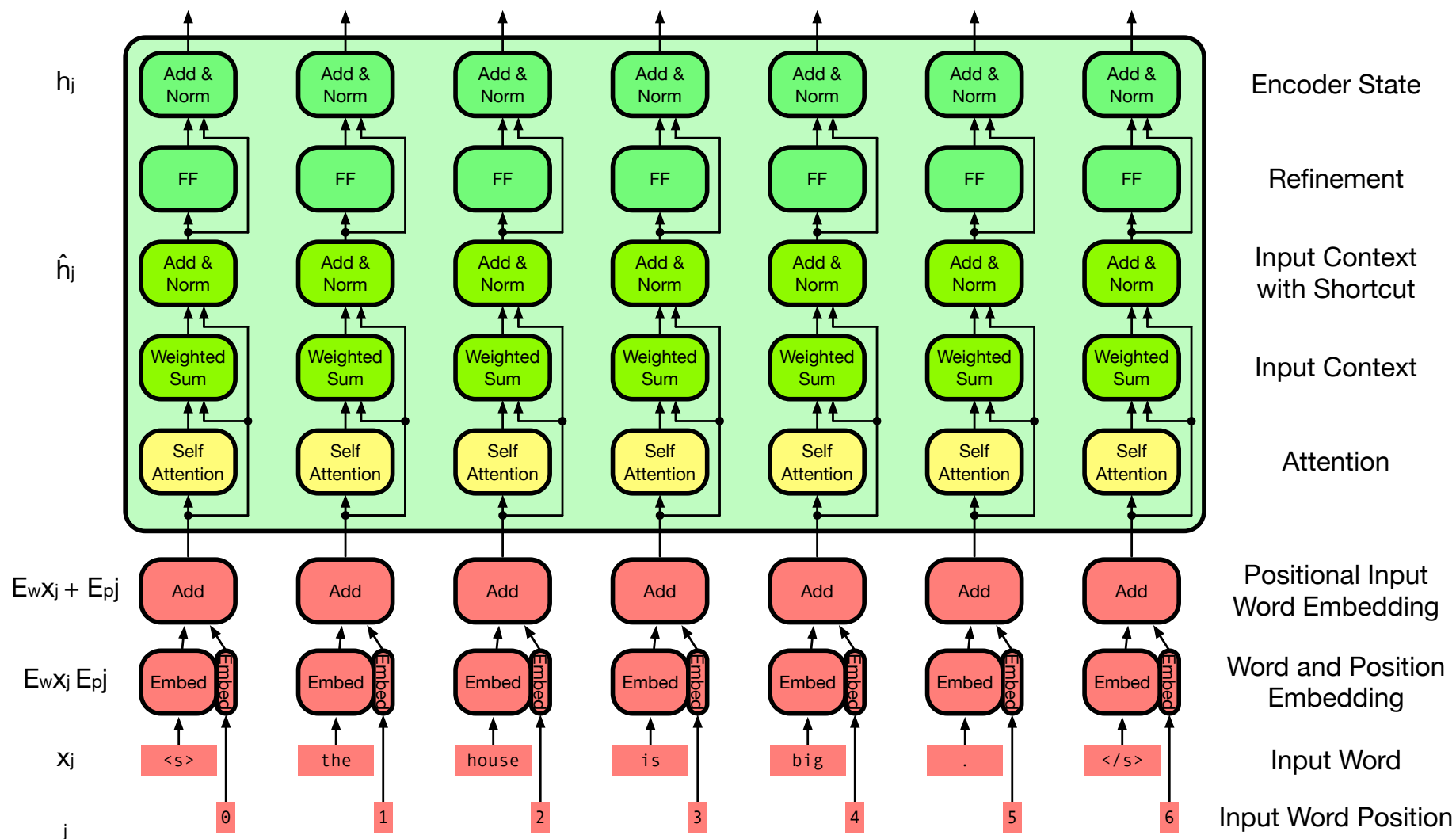
“Many of the attention heads exhibit behaviour that seems related to the structure of the sentence.”

transformer

Self Attention: Transformer

- Self-attention in encoder
 - refine word representation based on relevant context words
 - relevance determined by self attention
- Self-attention in decoder
 - refine output word predictions based on relevant previous output words
 - relevance determined by self attention
- Also regular attention to encoder states in decoder
- Currently most successful model
(maybe only with self attention in decoder, but regular recurrent decoder)

Encoder



Sequence of self-attention layers

Self Attention Layer

- Given: input word representations h_j , packed into a matrix $H = (h_1, \dots, h_j)$
- Self attention
 $\text{self-attention}(H) = \text{MultiHead}(H, H, H)$
- Shortcut connection
 $\text{self-attention}(h_j) + h_j$
- Layer normalization
- Feed-forward step with ReLU activation function and large intermediate vector
 $\text{relu}(W_1 \hat{h}_j + b_1)W_2 + b_2$
- Again, shortcut connection and layer normalization

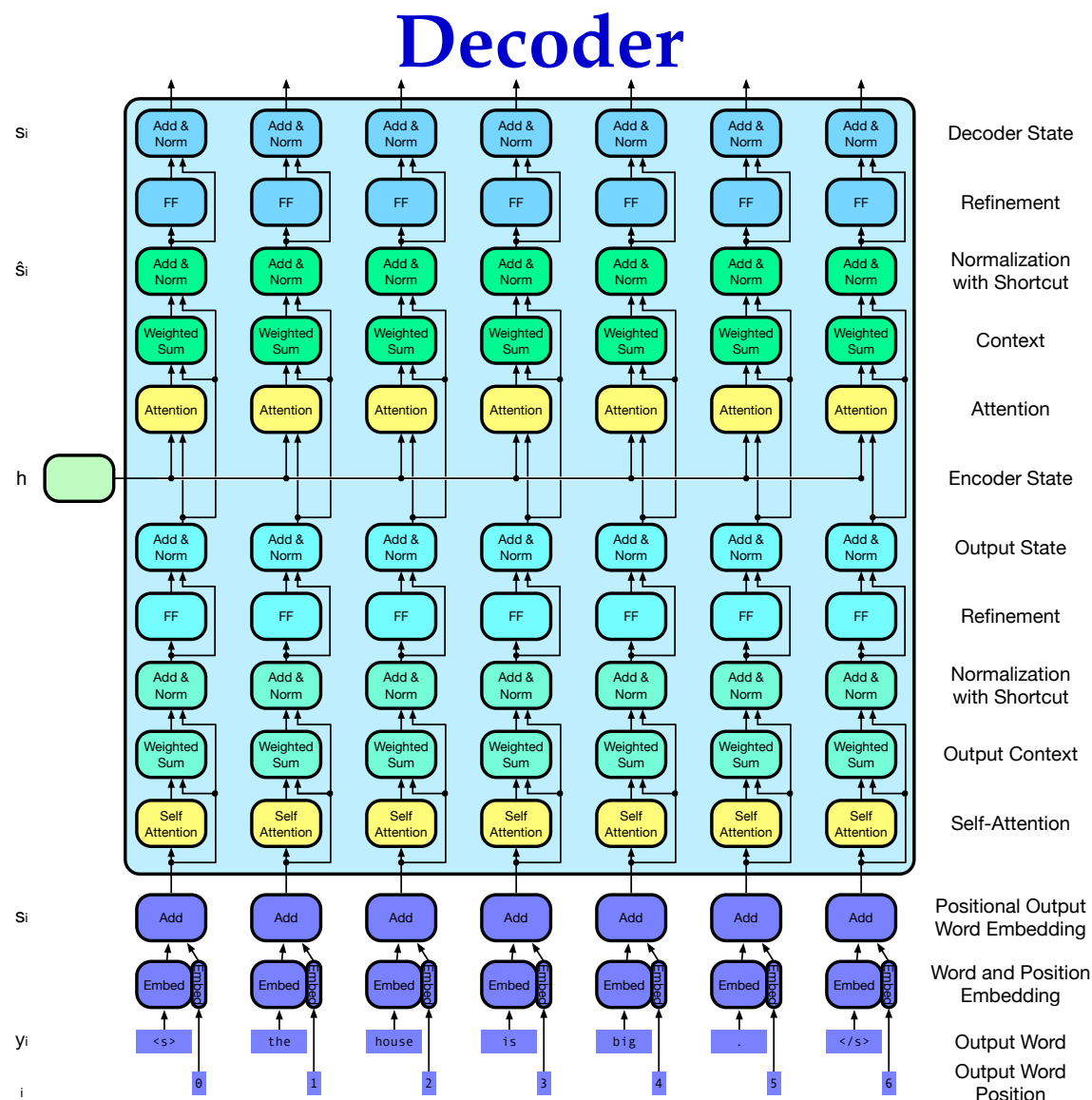
Stacked Self Attention Layers

- Stack several such layers (say, $D = 6$)
- Start with input word embedding

$$h_{0,j} = Ex_j$$

- Stacked layers

$$h_{d,j} = \text{self-attention-layer}(h_{d-1,j})$$



Decoder computes attention-based representations of the output in several layers, initialized with the embeddings of the previous output words

Self-Attention in the Decoder

- Same idea as in the encoder
- Output words are initially encoded by word embeddings $s_i = Ey_i$.
- Self attention is computed over previous output words
 - association of a word s_i is limited to words s_k ($k \leq i$)
 - resulting representation \tilde{s}_i

$$\text{self-attention}(\tilde{S}) = \text{MultiHead}(\tilde{S}, \tilde{S}, \tilde{S})$$

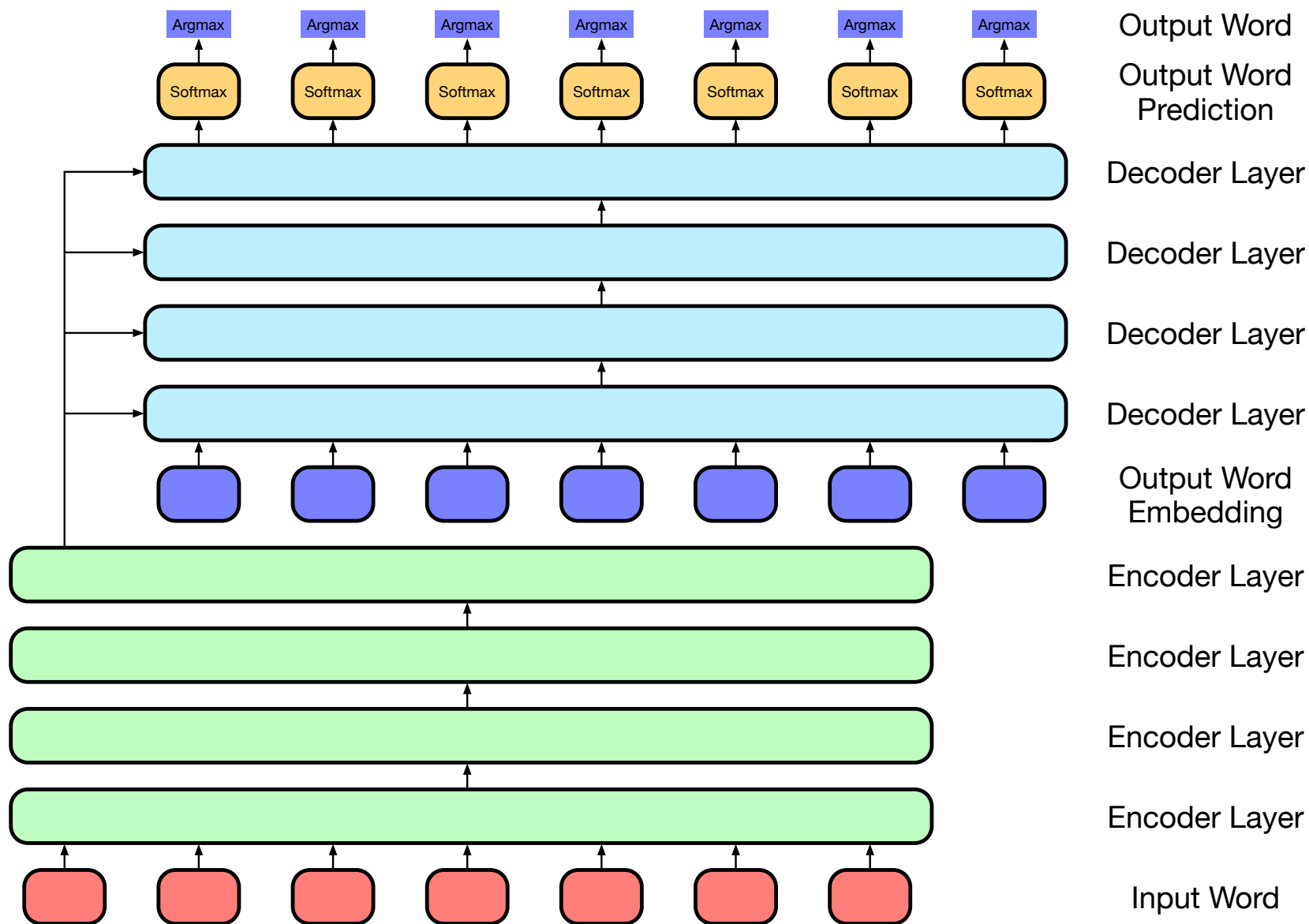
Attention in the Decoder

- Original intuition of attention mechanism: focus on relevant input words
- Compute attention between the decoder states \tilde{S} and the final encoder states H

$$\text{attention}(\tilde{S}, H) = \text{MultiHead}(\tilde{S}, H, H)$$

- Note: attention mechanism formally mirrors self-attention

Full Decoder

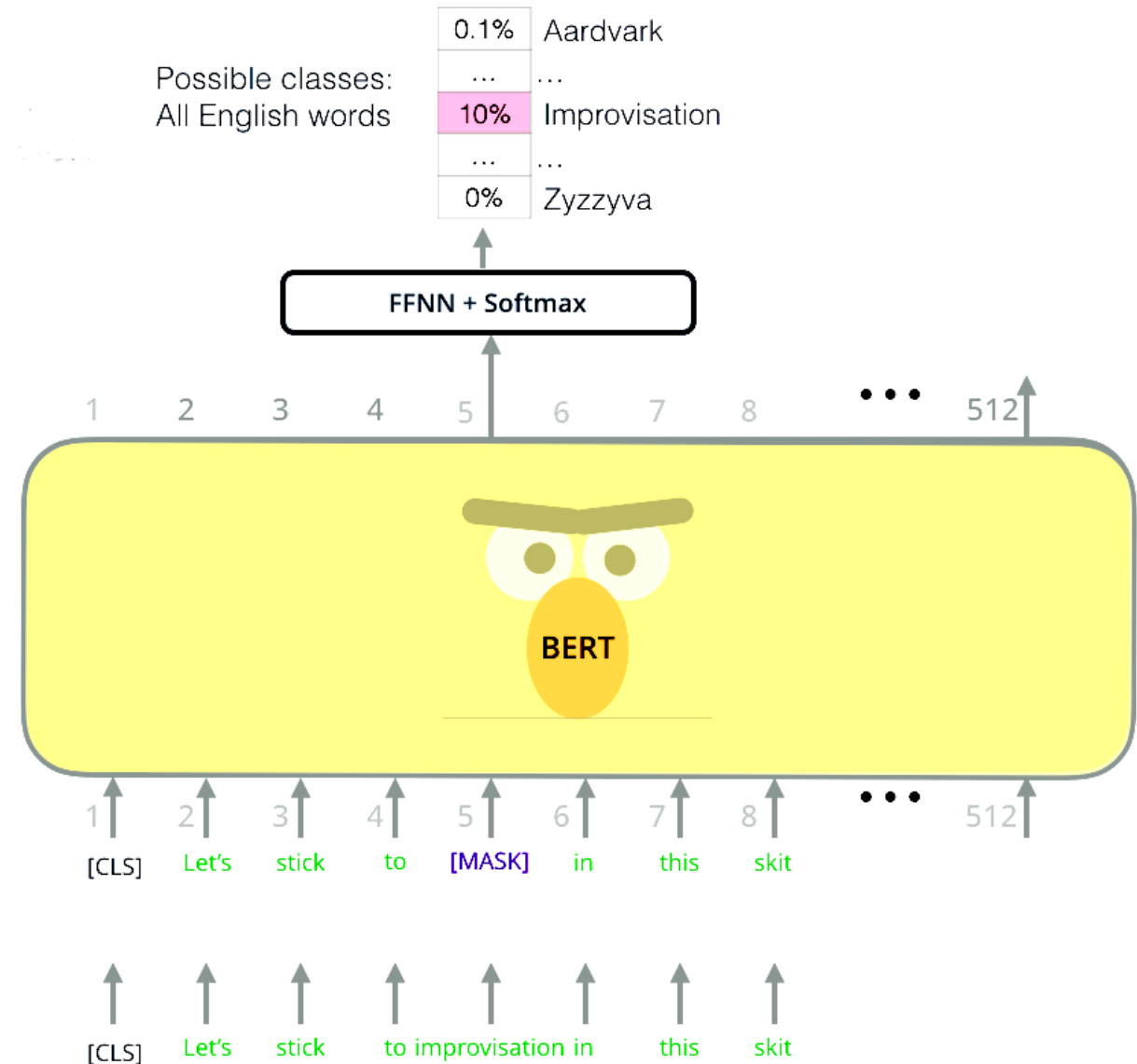


- Self-attention $\text{self-attention}(\tilde{S}) = \text{MultiHead}(\tilde{S}, \tilde{S}, \tilde{S})$
 - shortcut connections
 - layer normalization
 - feed-forward layer
- Attention $\text{attention}(\tilde{S}, H) = \text{softmaxMultiHead}(\tilde{S}, H, H)$
 - shortcut connections
 - layer normalization
 - feed-forward layer
- Multiple stacked layers

large language models

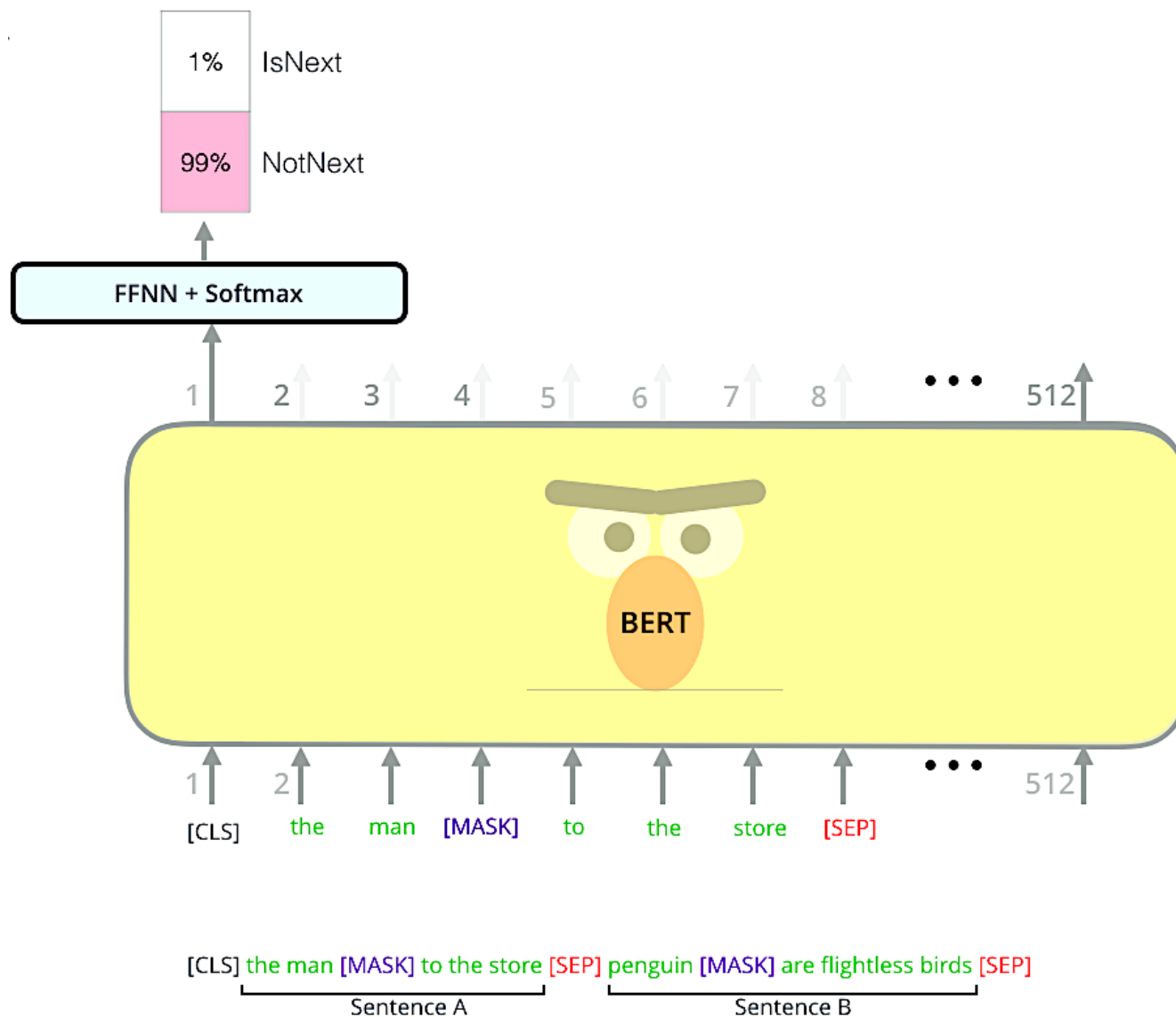
Masked Language Model Training

- Transformer expects an input and an output sequence
- Masked training
 - output sequence: one sentence of text
 - input sequence: same sentence, with some words masked out

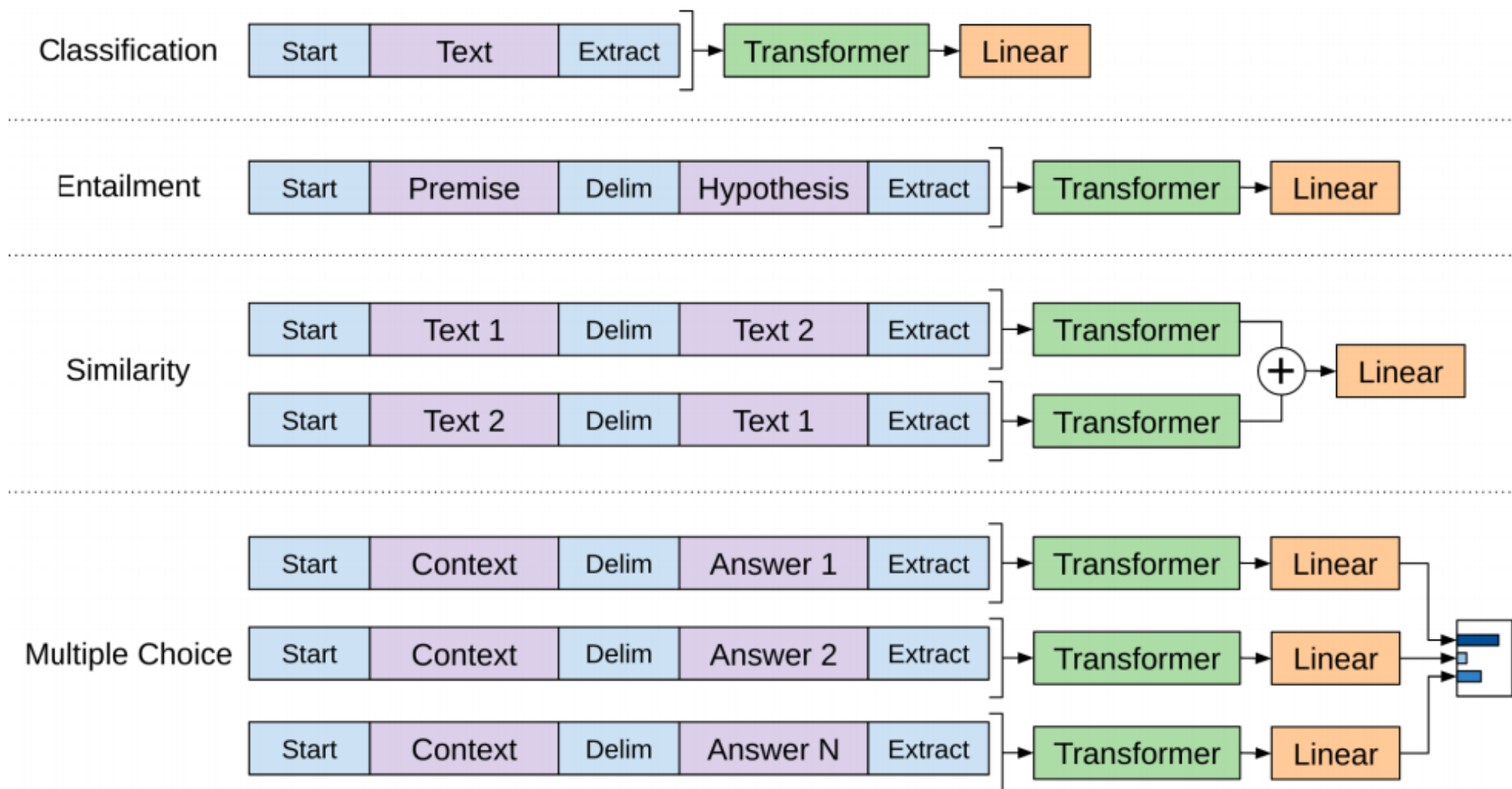


Next Sentence Prediction

- Next sentence prediction
- Input: two sentences
- Output: prediction that they were in sequence



Using Sentence Representations



- Train language models on relatively clean text data (GPT-2)
- Such text contains **naturally occurring demonstrations** of many tasks
- Convert any NLP problem into a text continuation problem

"I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbécile** [I'm not a fool].

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "**Mentez mentez, il en restera toujours quelque chose,**" which translates as, "**Lie lie and something will always remain.**"

"I hate the word '**perfume,**'" Burr says. 'It's somewhat better in French: '**parfum.**'

If listened carefully at 29:55, a conversation can be heard between two guys in French: "**Comment on fait pour aller de l'autre cote? Quel autre coté?**" ", which means "**How do you get to the other side? What side?**".

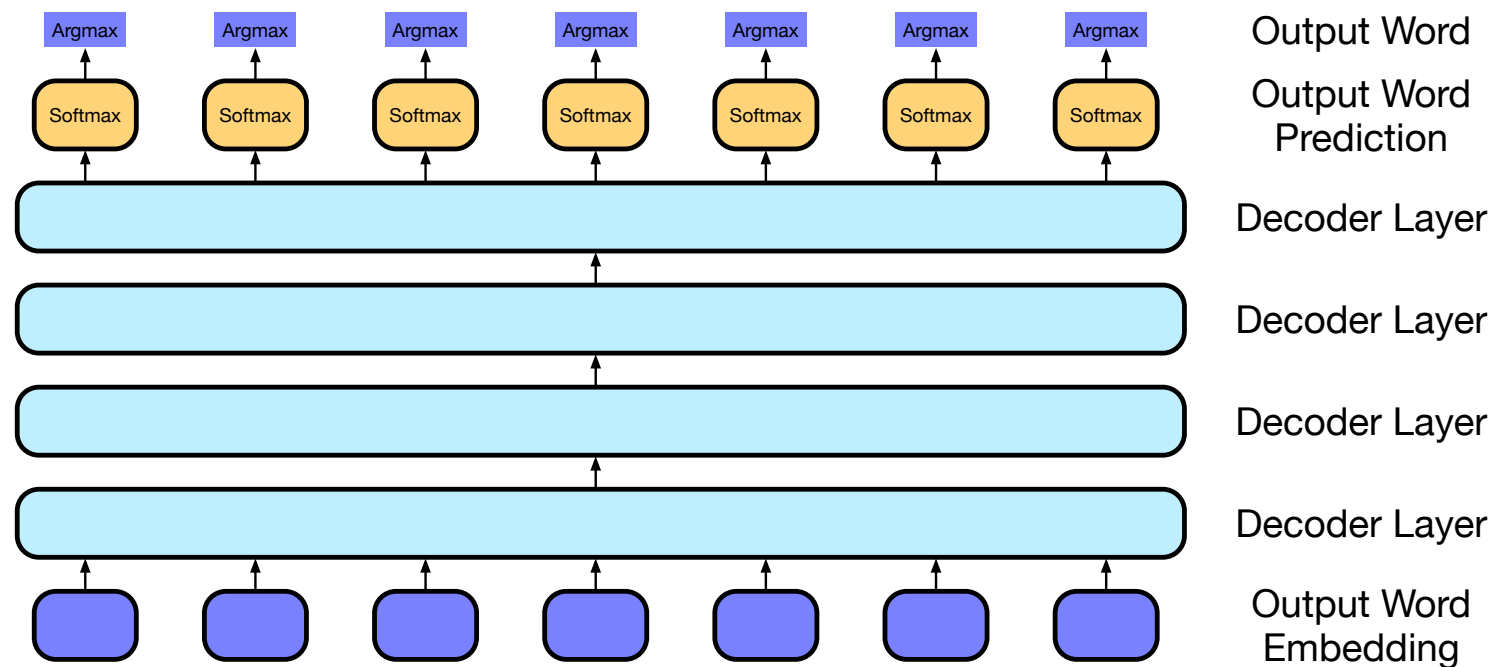
If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinema?** , or **Did you go to the movies?**, which literally translates as **Have-you to go to movies/theater?**

Decoder-Only Models

- Alternative architecture: Just decoder of Transformer model

⇒ no input, only self-attention

- Trained with next-word prediction



evaluation



- Reading comprehension
- Given: a short text, questions
- Expected answer: span of words in text
- SQuAD V2: Also added unanswerable questions

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?

gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

graupel

Where do water droplets collide with ice crystals to form precipitation?

within a cloud

```
def solution(lst):  
    """Given a non-empty list of integers, return the sum of all of the odd elements  
    that are in even positions.  
  
    Examples  
    solution([5, 8, 7, 1]) ==>12  
    solution([3, 3, 3, 3, 3]) ==>9  
    solution([30, 13, 24, 321]) ==>0  
    """  
    return sum(lst[i] for i in range(0, len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

- Generation of computer code from a textual description of the task
- Example HumanEval [Chen et al., 2021]: Hand-written evaluation set
- Evaluation: run the code and see if the answers are correct (unit tests)
- Very similar test set: Mostly Basic Programming Problems (MBPP) [Austin et al., 2021]

Relation	Formulated question example
AtLocation	<i>Where would I not want a fox? A. hen house, B. england, C. mountains, D. ...</i>
Causes	<i>What is the hopeful result of going to see a play? A. being entertained, B. meet, C. sit, D. ...</i>
CapableOf	<i>Why would a person put flowers in a room with dirty gym socks? A. smell good, B. many colors, C. continue to grow , D. ...</i>
Antonym	<i>Someone who had a very bad flight might be given a trip in this to make up for it? A. first class, B. reputable, C. propitious , D. ...</i>
HasSubevent	<i>How does a person begin to attract another person for reproducing? A. kiss, B. genetic mutation, C. have sex , D. ...</i>
HasPrerequisite	<i>If I am tilting a drink toward my face, what should I do before the liquid spills over? A. open mouth, B. eat first, C. use glass , D. ...</i>
CausesDesire	<i>What do parents encourage kids to do when they experience boredom? A. read book, B. sleep, C. travel , D. ...</i>
Desires	<i>What do all humans want to experience in their own home? A. feel comfortable, B. work hard, C. fall in love , D. ...</i>
PartOf	<i>What would someone wear to protect themselves from a cannon? A. body armor, B. tank, C. hat , D. ...</i>
HasProperty	<i>What is a reason to pay your television bill? A. legal, B. obsolete, C. entertaining , D. ...</i>

- Questions about commonsense knowledge
- Example COMMONSENSEQA [Talmor et al., 2019]: Questions derived from CONCEPTNET
- Evaluation: multiple choice, highest probability assigned to A, B, C or D
- Similar test set obtained with crowdsourcing: WinoGrande [Sakaguchi et al., 2021]

The trophy doesn't fit into the brown suitcase because it's too large.

The trophy doesn't fit into the brown suitcase because it's too small.

trophy / suitcase

trophy / **suitcase**

- Ability to work through mathematical problems
- Example: MATH [Hendrycks et al., 2021]
- Taken from math competition problem sets
- Evaluation of the final solution (box in figure)

Problem: Tom has a red marble, a green marble, a blue marble, and three identical yellow marbles. How many different groups of two marbles can Tom choose?

Solution: There are two cases here: either Tom chooses two yellow marbles (1 result), or he chooses two marbles of different colors ($\binom{4}{2} = 6$ results). The total number of distinct pairs of marbles Tom can choose is $1 + 6 = \boxed{7}$.

Problem: The equation $x^2 + 2x = i$ has two complex solutions. Determine the product of their real parts.

Solution: Complete the square by adding 1 to each side. Then $(x + 1)^2 = 1 + i = e^{\frac{i\pi}{4}} \sqrt{2}$, so $x + 1 = \pm e^{\frac{i\pi}{8}} \sqrt[4]{2}$. The desired product is then $(-1 + \cos(\frac{\pi}{8}) \sqrt[4]{2})(-1 - \cos(\frac{\pi}{8}) \sqrt[4]{2}) = 1 - \cos^2(\frac{\pi}{8}) \sqrt{2} = 1 - \frac{(1 + \cos(\frac{\pi}{4}))}{2} \sqrt{2} = \boxed{\frac{1 - \sqrt{2}}{2}}$.

One of the reasons that the government discourages and regulates monopolies is that

- (A) producer surplus is lost and consumer surplus is gained.
- (B) monopoly prices ensure productive efficiency but cost society allocative efficiency.
- (C) monopoly firms do not engage in significant research and development.
- (D) consumer surplus is lost with higher prices and lower levels of output.



- Questions about humanities, social science, STEM
- Example: MMLU [\[https://arxiv.org/pdf/2009.03300\]](https://arxiv.org/pdf/2009.03300)
- Multiple choice questions
- Collected from practice questions for college or certification exams
- Multilingual version MMMLU: human translated into 14 languages

MMLU Reasoning Example

As Seller, an encyclopedia salesman, approached the grounds on which Hermit's house was situated, he saw a sign that said, "No salesmen. Trespassers will be prosecuted. Proceed at your own risk."

Although Seller had not been invited to enter, he ignored the sign and drove up the driveway toward the house. As he rounded a curve, a powerful explosive charge buried in the driveway exploded, and Seller was injured. Can Seller recover damages from Hermit for his injuries?

- (A) Yes, unless Hermit, when he planted the charge, intended only to deter, not harm, intruders. ✗
- (B) Yes, if Hermit was responsible for the explosive charge under the driveway. ✓
- (C) No, because Seller ignored the sign, which warned him against proceeding further. ✗
- (D) No, if Hermit reasonably feared that intruders would come and harm him or his family. ✗

Test on Train?



- Grave concerns about training data contamination
- If test sets are built on web data → very likely in the training data
- Larger models → higher capacity to memorize

Question Answering Without the Question 83



- Can LLMs answer multiple-choice questions without the question? [Balepur et al., 2024]

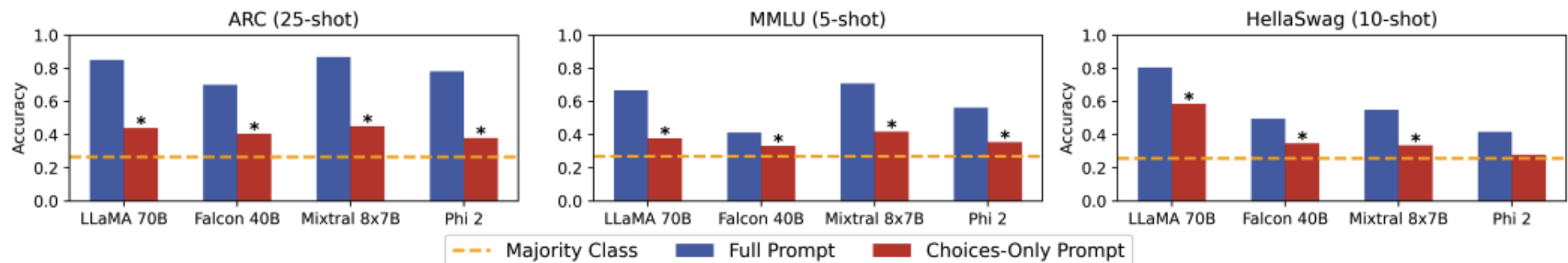
No Choices

Question: Which of these contains only a solution?
Answer: (B)

Empty Choices

Question: Which of these contains only a solution?
Choices: (A) \n (B) \n (C) \n (D) \n
Answer: (B)

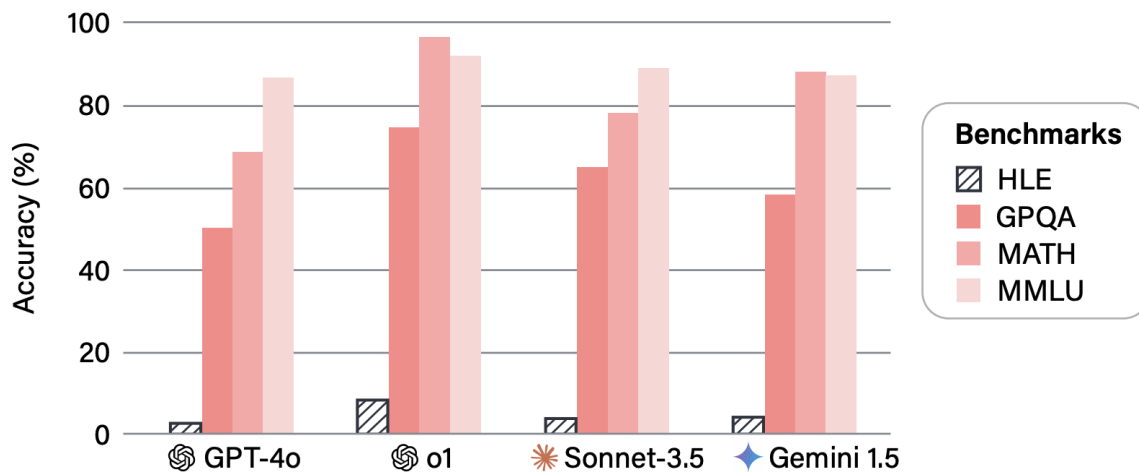
- Results



- Note that none of these require generation of long fluent text
- Why? Evaluation of responses is difficult
Write a story about a cow who wants to be a pig.■
- Response similarity metrics (akin to BLEU) exist
 - e.g., ROUGE for summarization
 - ... but not very reliable■
- A currently popular solution:
ask a language model to score against reference response

Humanity's Last Exam

- 2,700 questions across dozens of subjects, including mathematics, humanities, and the natural sciences
- Written by subject-matter experts
- Multiple-choice and short-answer questions → suitable for automated grading



√x Mathematics

Question:

The set of natural transformations between two functors $F, G : C \rightarrow D$ can be expressed as the end

$$\text{Nat}(F, G) \cong \int_A \text{Hom}_D(F(A), G(A)).$$

Define set of natural cotransformations from F to G to be the coend

$$\text{CoNat}(F, G) \cong \int^A \text{Hom}_D(F(A), G(A)).$$

Let:

- $F = B_*(\Sigma_4)_*$ be the under ∞ -category of the nerve of the delooping of the symmetric group Σ_4 on 4 letters under the unique 0-simplex $*$ of $B_*\Sigma_4$.
- $G = B_*(\Sigma_7)_*$ be the under ∞ -category nerve of the delooping of the symmetric group Σ_7 on 7 letters under the unique 0-simplex $*$ of $B_*\Sigma_7$.

How many natural cotransformations are there between F and G ?

👤 Emily S
🏢 University of São Paulo

training

Three Stages of Training

- Stage 1: Train on massive amounts of text (up to a trillion words)■
- Stage 2: Instruction training■
- Stage 3: Reinforcement learning from human feedback

pretraining

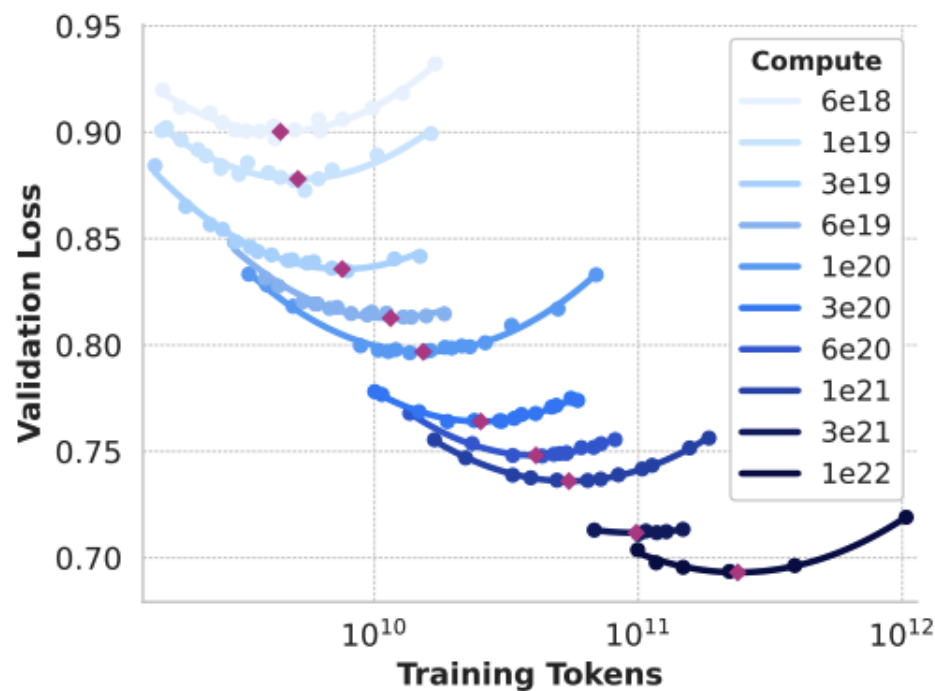
Massive Amounts of Text

- Web crawls
 - publicly available raw data: CommonCrawl
 - filtered and cleaned data: Fineweb
- eBooks
- Compute code (from github)
- Trillions of words

Common Crawl



Training



- Scaling laws: more data \rightarrow bigger models \rightarrow better performance
- Today: trillions of words \rightarrow 10s to 100s of billions of parameters
- Llama3 405B: trained on 16,384 GPUs — available open source

Massive Engineering Effort

- Example: Llama3
 - 16K H100 GPUs
 - 54 days
- Example: Deepseek V3
 - 2048 H800 GPUs
 - 2.8 million H800 GPU hours
- Rarely done — even big company do this maybe once a year



instruction tuning

- Long-running research efforts into specific NLP tasks
 - machine translation
 - questions answering (about a provided document / open-ended)
 - summarization (given one or more documents)
 - named entity detection (finding company, person, etc. names)
 - sentiment detection (positive review?)
 - grammar correction
 - ... and many more
- For all these training data and test sets were created

⇒ These can be converted into LLM training data

- We expect the model to answer user questions

⇒ Text needs to be adapted

Instruction

User: "Write a summary of the movie Interstellar."

Model: "Interstellar is a sci-fi movie directed by Christopher Nolan. It follows a group of astronauts who travel through a wormhole in search of a new home for humanity."

Chat

User: "Okay, let's say I'm writing a story. Can you help me brainstorm some ideas for a fantasy setting?"

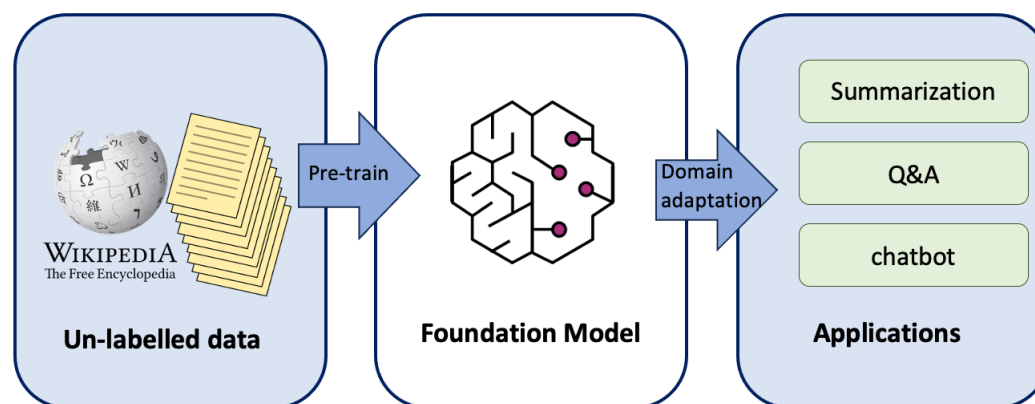
Model: "Sure, I can help with that. What kind of fantasy setting are you interested in? Do you have any specific elements in mind, like a particular type of magic system or a specific type of monster?"

Human Answers to Prompts

- Developers of LLMs collect a lot user questions
- These are evaluated for quality control
- **Good responses** \Rightarrow use as instruction training data
- **Bad responses** \Rightarrow (expert) human create acceptable answers
use as instruction training data

Supervised Fine-Tuning

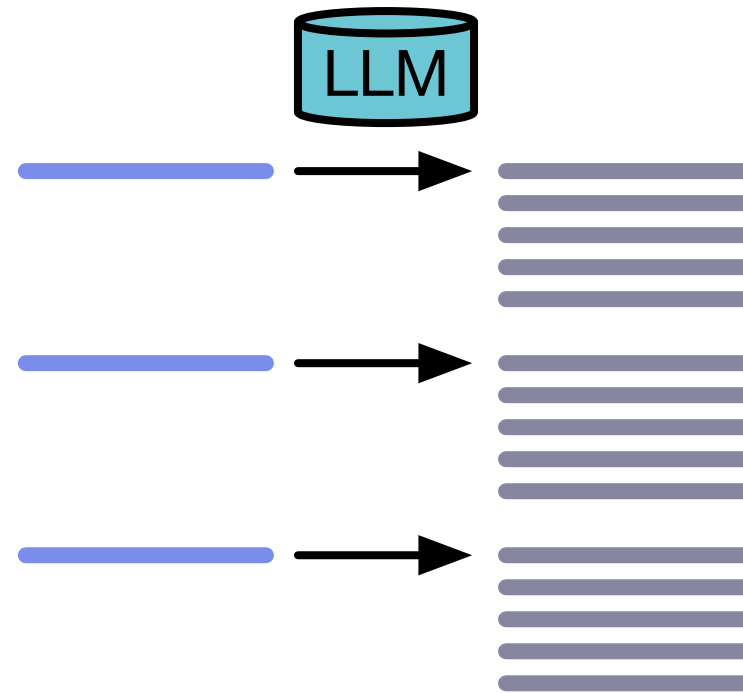
- Take pre-training model
- Continue training with instruction data



- This can be done many times, in different ways
... even by application builders with modest resources

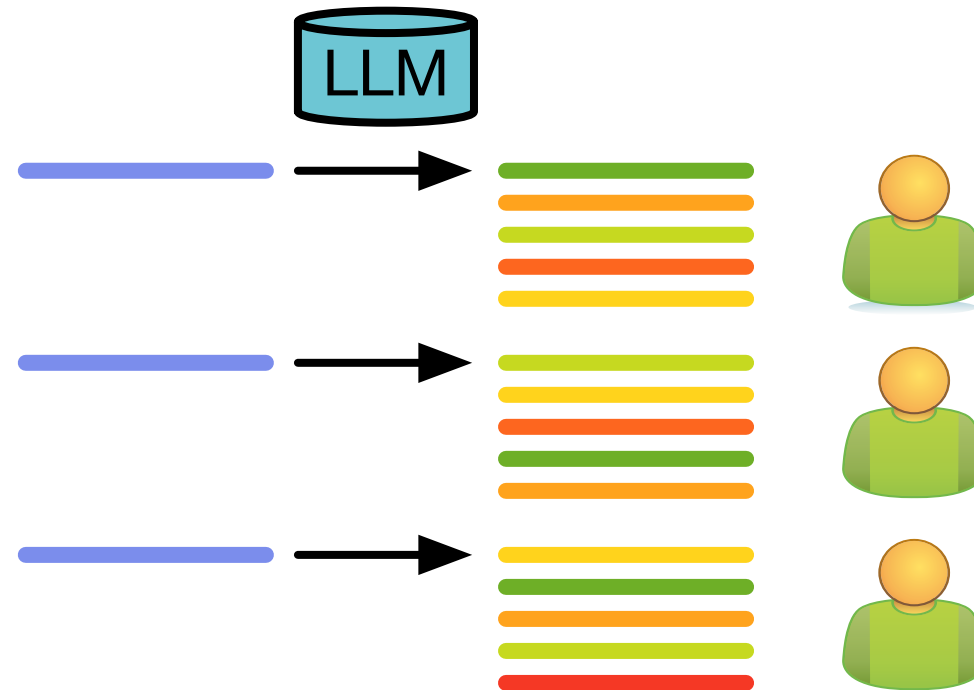
preference training

Learning from Human Preferences



- Generate responses from a prompt by sampling
 - greedy decoding: always choose word prediction with 80% probability
 - Monte Carlo decoding: choose it 80% of the time

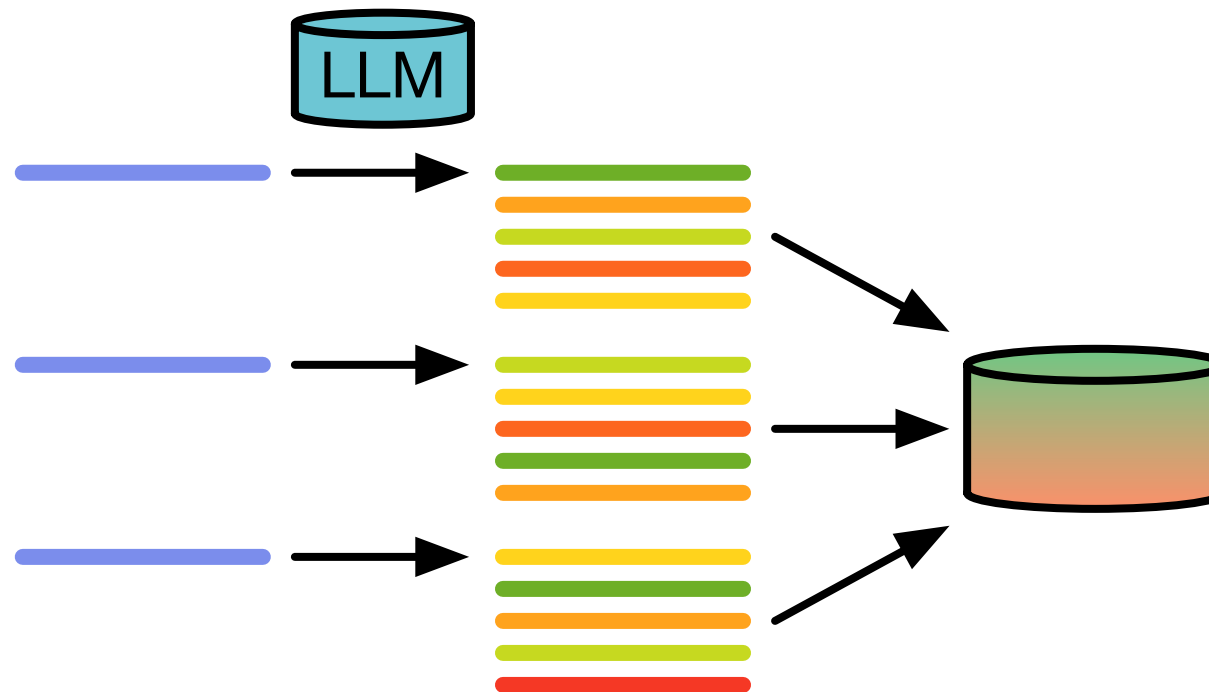
Learning from Human Preferences



- Human annotators rank the responses
- This is easier to do than authoring responses but still expensive

Learning from Human Preferences

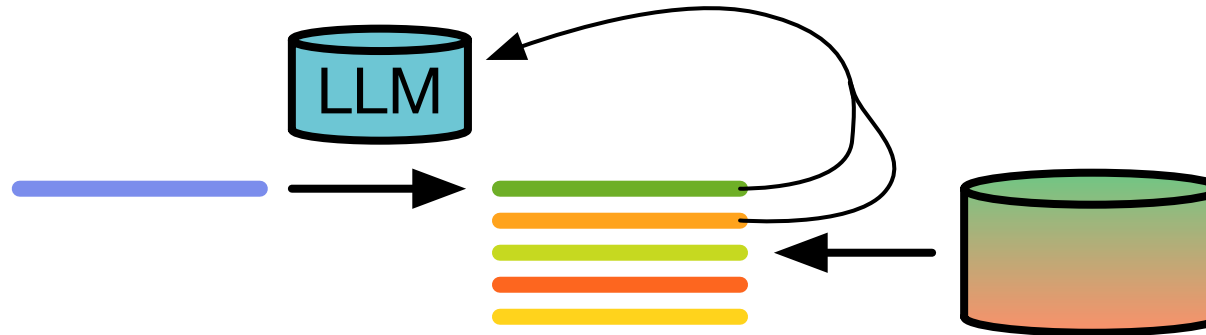
100



- Train a preference model
- Typically based on sequence representations from language models

Learning from Human Preferences

101



- Use the preference model during training original model
 - for a prompt, generate responses
 - score the responses with the preference model
 - update model to
 - * promote higher-scoring responses (winner)
 - * demote lower-scoring responses (loser)

Reinforcement Learning from Human Feedback

- This idea was originally introduced as a form of reinforcement learning
- The idea of a reward model stems from reinforcement learning
- Method: Proximal Policy Optimization (PPO)
- Recently, simpler methods are more common

Direct Preference Optimization (DPO)

103



- First train a reward model r^*
- Sample two possible responses for an input x
- Score them with the reward model
 - higher scoring translation is the winner y_w
 - higher scoring translation is the loser y_l
- Train a new model π_θ from an original model π_{ref} (using a hyper parameter β)

$$\text{Loss}(x, y_w, y_l) = \log \text{sigmoid} \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right)$$

[from Rafailov et al., 2023]



compact models

Large Language Models Very Costly

105

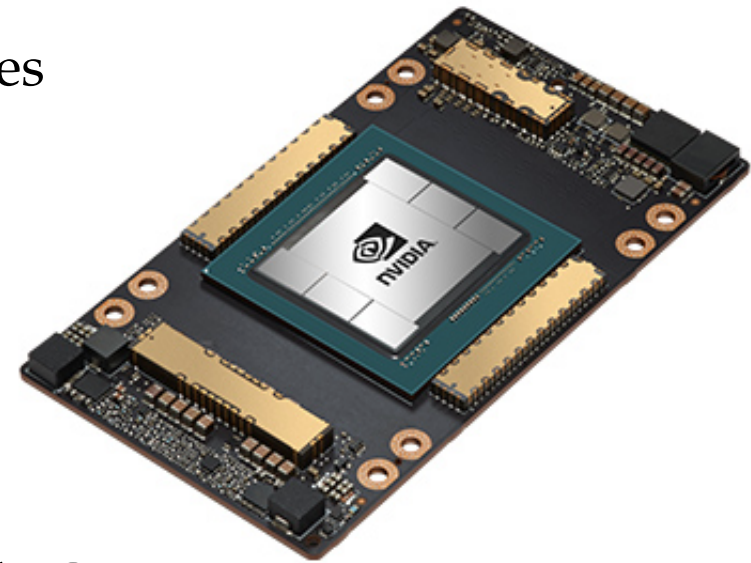


- 10-1000 times as many parameters as dedicated MT models
- More powerful machines needed (with multiple \$20,000 GPUs)
- Slower, each translation request more expensive
- Very costly to adapt to particular user cases

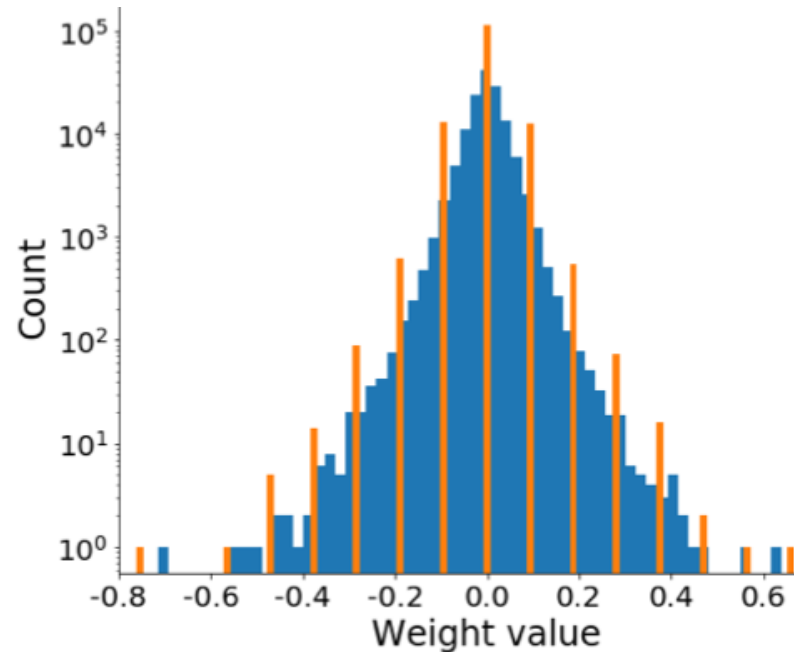




- Considering the size of language models
 - parameters are typically stored as 16-bit floats
 - during training also gradients and optimizer states need to be stored
 - ⇒ 6 bytes per parameter
 - Also need to store the state of training examples (depends on sequence length and batch size)
- Size of GPUs
 - A100: 40-80GB RAM (\$15,000)
 - RTX2080ti: 11GB RAM (\$800)
- Only a few billion parameters models fit on single GPU



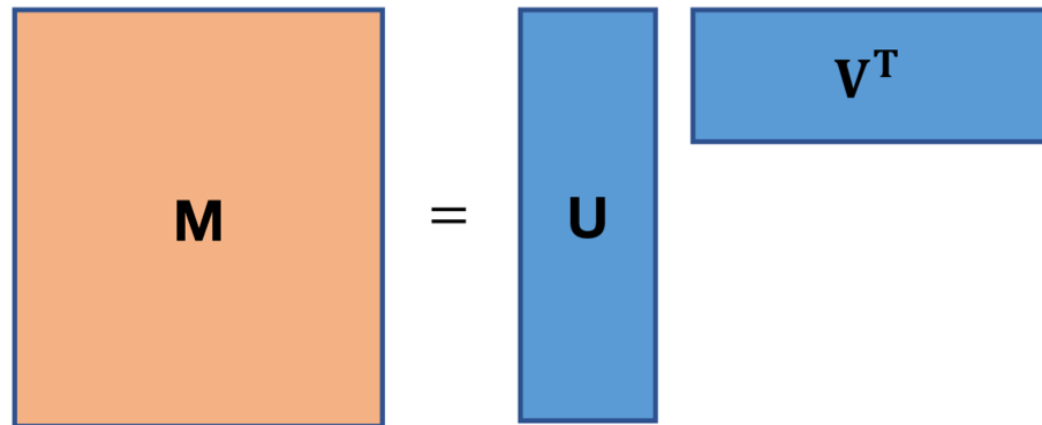
Solution 1: Quantization



- Store values in 4 bit floats (or less)
- Computation still in 16 bits
- Additional tricks: double quantization, paged optimizers

Solution 2: Low-Rank Adaptation (LoRA)

108



- Keep original model intact during adaptation
- Add adaptation parameters in form of low-rank matrices
 - original: n^2 parameter matrix M
 - adaptation:
 - * nr and rn matrices U, V
 - * with $r \ll n$
 - * e.g., $n=2048, r=16$

Solution 3: Knowledge Distillation

- Large language model as Teacher
- Small language model as Student
- Data distillation
 - process task-relevant data with Teacher model → good responses
 - use this synthetic data to train Student model
- Model distillation
 - Train Student model directly on predictions of Teacher model



using large language models

- Basic usage: Query existing language model
- In context learning: Include description of task in prompt
- Few-shot prompting: Include examples of task in prompt
- Retrieval: collect relevant information, include in prompt
- Fine-tune model to task
- Tool use / agentic systems
- Reasoning



- Create training data ... with a language model
- Solve problem ... with a language model
- Evaluate results ... with a language model



prompt engineering

- How a task is presented to the language model matters
- Black art: often unclear what is in the training data■
- *Think step by step.*
 - language model has very limited working memory
 - complex reasoning may require several inference steps
 - by allowing the model to produce intermediate steps, working memory is created■
- *Imagine you are...*
 - creates a style or sets a context for generation



- How a task is presented to the language model matters
- Black art: often unclear what is in the training data
- “Imagine you are...”: creates a style or sets a context for generation
- Detailed instructions
 - “Translate every word.”
 - “Do not provide any explanations.”
 - “Do not drop or add any information.”



- Language model has very limited working memory
- Complex reasoning may require several inference steps

What is 56 times 204? Think step by step.

I first multiply 50 and 200, which is 10,000.

Then I multiply 6 and 200, which is 1,200.

I first multiply 50 and 4, which is 200.

Then I multiply 6 and 4, which is 24.

Adding together 10,000 and 1,200 and 200 and 24 gives 11,424.

- Intermediate steps create working memory

Decomposition of Tasks

Write an outline for an argument that LLMs are great.

Outline:

1. LLMs have emerged as general problem solvers
2. LLMs have become publicly available
3. Many people find LLMs useful
4. ...

Write a paragraph about "LLMs have emerged as general problem solvers".

Language model (LLMs) have rapidly emerged as versatile general problem solvers across various domains. Leveraging vast datasets and sophisticated algorithms, LLMs exhibit remarkable capabilities in understanding, generating,...



in-context learning

In-Context Learning

- Explain the task in the prompt
- Provide examples of the task in the prompt

I want you to assess the sentiment of statements as either positive or negative.■

Here are some examples:

It's a beautiful day. Assessment: positive

We did not play well in the game. Assessment: negative

That's not bad. Assessment: positive■

Now assess the following statement:

Yeah, right. Assessment:



- Problem
 - language models are trained on very diverse language usage
 - it may be confused on what it is expected to do
- Solution: provide examples (“shots”) of the task in the prompt
- This has been shown to be successful even for new tasks



- Provide examples in the prompt

Translate from German to English. Here are some examples.

German: Ein Hund bellt. English: A dog barks.

German: Ein Schwein grunzt. English: A pig grunts.

German: Eine Katze miaut. English: A cat meows.

German: Ein Wolf heult. English: A wolf howls.

Now translate the following sentence.

German: Ein Vogel singt. English:

- This is the standard approach when prompting language models



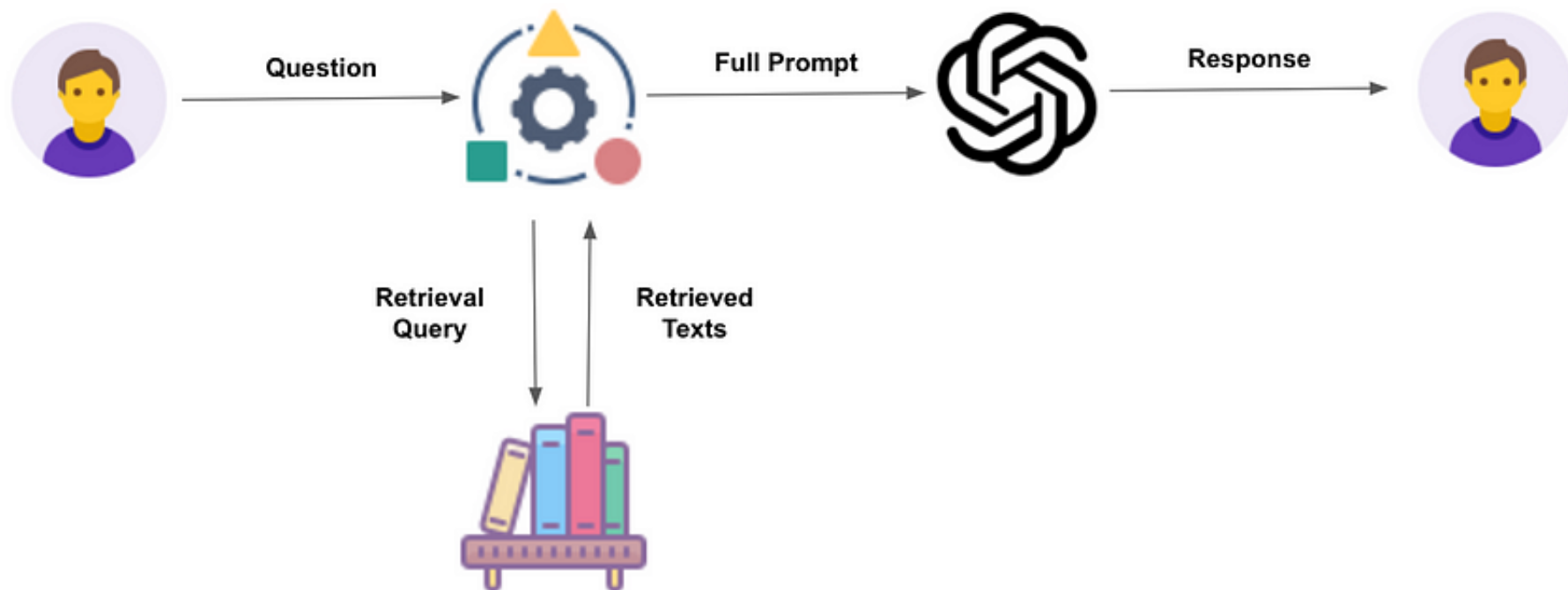
retrieval augmented generation

Retrieval Augmented Generation

123



- Example: question answering task
 - may be aided by specific documents
 - these documents may not even be part of LLM training





- Query for information

Who won the FIFA World Cup in 2014?

- Relevant document

The 2014 FIFA World Cup was the 20th FIFA World Cup, the quadrennial world championship for men's national football teams organised by FIFA. It took place in Brazil from 12 June to 13 July 2014, after the country was awarded the hosting rights in 2007. It was the second time that Brazil staged the competition, the first being in 1950, and the fifth time that it was held in South America.

In the final, Germany defeated Argentina 1–0 after extra time thanks to a Mario Götze half-volley in the 113th minute of the final to win the tournament and secure the country's fourth world title, their first major tournament win since UEFA Euro 1996, the first after German reunification in 1990, when as West Germany they also beat Argentina by the same score in 90 minutes in the World Cup final.

Prompts with Retrieved Content

125



- Insert retrieved content into the prompt
- Cannot be too long \Rightarrow break up into text chunks (say, 500 tokens)
- Could include multiple chunks
 - robust against imperfect retrieval
 - relevant information may be spread out across multiple documents



neurosymbolic ai

Two Kinds of Intelligence

- Neural approaches
 - encode information with vectors
 - learn associations to make inference
- Symbolic approaches
 - encode information with symbols
 - traditional computer science methods
 - databases, key words
 - logical inference



- LLM generates call to APIs
- Executes call
- Receives result back
- Uses result to continue text generation

The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

(Toolformer, Schick et al., 2023)



- If LLMs are universal problem solvers, use them for any task

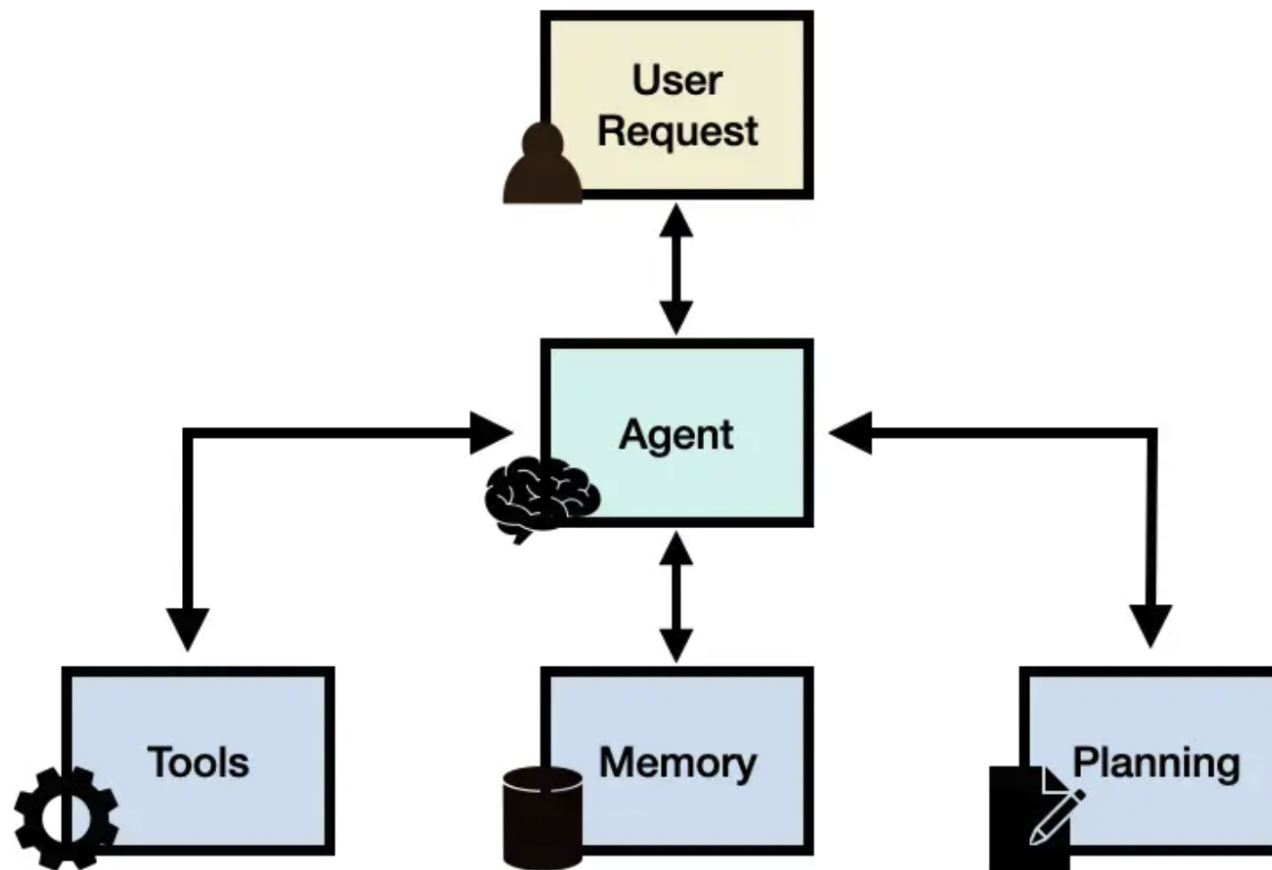
```
a = 1.6
b_text = "What is the distance to the moon in miles?"
c_text = LLM(b_text)
c = process_response(c_text)
print("Distance to moon in km:", a * c)
```

Agent Systems

130



- LLM agent creates and revises plan, calls multiple components
- May also take actions, such as filling out web forms



Example

131



(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: `Search[Apple Remote]`

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.

Act 2: `Search[Front Row]`

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search Front Row (software) .

Act 3: `Search[Front Row (software)]`

Obs 3: Front Row is a discontinued media center software ...

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

Act 4: `Finish[keyboard function keys]`



- Handle complex tasks that require assembling diverse sets of information
- Mix of LLM, search, and browse actions

Prompt

I am a researcher interested in bringing mixed-gas sorption capabilities to my lab. Please discuss the differences between pure- and mixed-gas sorption for glassy polymers, how the dual-mode sorption model can be used to predict mixed-gas sorption behavior in glassy polymers (include equations where applicable), and what challenges there are in accurately predicting pure- and mixed-gas sorption using the dual-mode sorption model. Only include open-source information as of October 1, 2024.

Processing Steps

Understanding sorption models
[search action]
Assessing open access
[browse action]
Examining sections
[browse action]
Clarifying key properties
[browse action]
Piecing together
[browse action]
Confirming title relevance
[search action]

Example from <https://openai.com/index/introducing-deep-research/>



reasoning

Think Step by Step

134



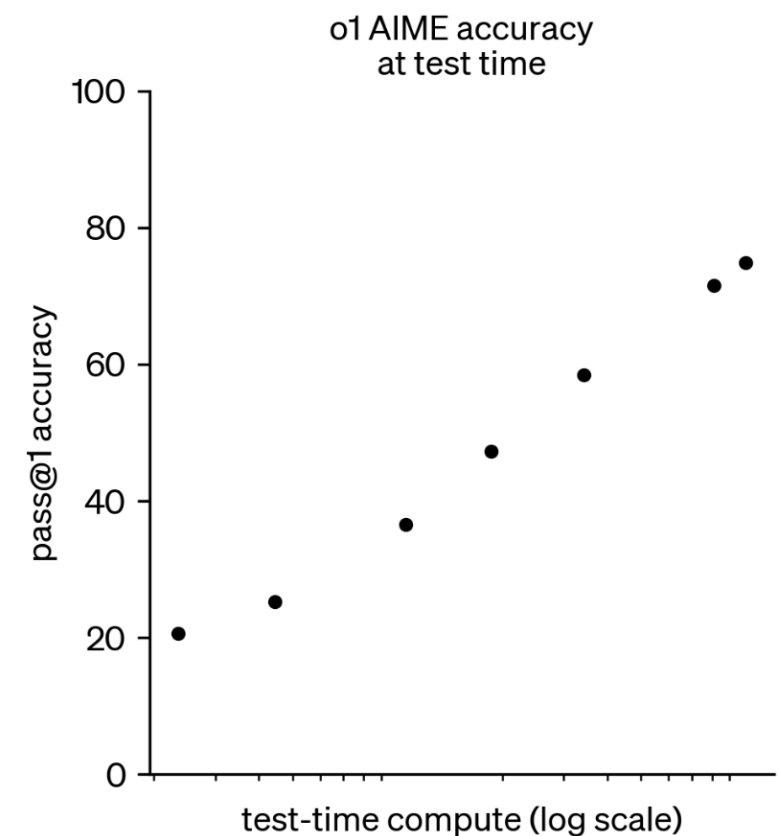
- Questions that require some consideration cannot be directly answered by LLM

What is $12 \times (19 - 3)$?

- Prompt model to first think about the problem (chain-of-thought)

Let's think step by step.■

- Observation: longer responses more accurate (test time compute scaling)



- Explicitly separate reasoning from answering
- Example: DeepSeek r1 prompt

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`.
User: prompt. Assistant:

- The reasoning process may be hidden from the user

Example

136



Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both \dots

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

\dots

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be \dots

We started with the equation:

$$\sqrt{a - \sqrt{a + x}} = x$$

First, let's square both sides:

$$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$$

Next, I could square both sides again, treating the equation: \dots

\dots



- There is no training data of good reasoning processes
⇒ cannot do instruction training
 - But: Only correctness of final answer matters
- ⇒ Reinforcement learning — only score final answer
- Reward model
 - measure of correctness of answer
(may be a rule-based binary score)
 - formal correctness of reasoning process
(coherent language, uses provided tags, produces an answer)



- Hallucinations
- Interpretability
- HHH: Helpful, Honest, Harmless
- Preserving privacy of training data



questions?