Administrative points

- If you have not done so yet, send me your topic interests
  - Missing names
- Student presentations starting on Tuesday
- Never too early to start thinking about your projects
  - I will schedule office hours for project questions as we get closer to the project proposal (starting week of Sept. 15)

CS 395T: LLM Crash Course

August 28th, 2025

Today: LLM Crash course

- What’s a language model?
- Transformers and self-attention
- Bit of pre-training history (BERT)
- GPT3, in-context learning
- CoT
- ReACT

Resources

- If you’re hungry for more, check out:
  - Greg Durrett’s CS388 slides (some of which I stole)
  - Daniel Khashabi’s CS 601.471/671 NLP: Self-supervised Models slides

Assumed background

- Probability
- Multi-layer perceptron
- NN basics (softmax, classification)
- Word embeddings and distributional semantics

What is language modeling?

- Predicting the next word

The students sat at _____

- Modeling a probability distribution

$$P(w_i|w_1, w_2, \ldots, w_{i-1})$$

Where w_i is a word in a fixed vocabulary.

What is language modeling?

- Assigning probabilities to sequences

$$P(w_1, w_2, \ldots, w_i)$$

- This can tell you useful information:
  - What's the probability of "*The students sat at desks"* vs. *"The students ate at desks"* vs. *"The students sat at bananas"?*

Why do language modeling?

- Practical applications

- Large language models

How to learn a language model?

- Q: How do you learn a language model from data?
- One approach: n-gram language model
- N-gram = *n* consecutive tokens

  The students sat at desks
  - Unigram: *The, students, sat, at, desks*
  - Bigram: *The students, students sat, sat at, at desks*
  - Trigram: *The students sat, students sat at, sat at desks*
  - Four-gram…
- General idea: estimate probability via count

Estimating probabilities

- Estimate the probability of a word by looking at the count
- Markov assumption: w_i depends only on the preceding $n-1$ words

$$P(w_i|w_1, w_2, \ldots, w_{i-2}, w_{i-1}) = P(w_i|\underbrace{w_{i-n+1}, \ldots, w_{i-1}}_{n-1 \text{ words}})$$

- Conditional probability

$$P(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \frac{P(\overbrace{w_i, w_{i-n+1}, \ldots, w_{i-1}}^{n \text{ gram}})}{P(\underbrace{w_{i-n+1}, \ldots, w_{i-1}}_{n-1 \text{ gram}})}$$

Estimating probabilities

- Estimating via counts

$$P(w_i|w_{i-n+1},\ldots,w_{i-1}) = \frac{P(w_i,w_{i-n+1},\ldots,w_{i-1})}{P(w_{i-n+1},\ldots,w_{i-1})}$$

$$\approx \frac{Count(w_i,w_{i-n+1},\ldots,w_{i-1})}{Count(w_{i-n+1},\ldots,w_{i-1})}$$

The students sat at desks
The students sat at tables
The parents sat at tables
The students ate at desks

$$n = 3$$

$$P(\text{tables}|\text{sat at}) = \frac{Count(\text{sat at tables})}{Count(\text{sat at})}$$

What's wrong with n-grams?

- Markov assumption discards useful information

$$P(\text{tables}|\text{sat at}) \neq P(\text{tables}|\text{students sat at})$$

- Sparsity
  - Longer n-grams = better model (higher-order Markov assumption)
  - Longer n-grams = more sparsity, zero probability

The students sat at desks
The students sat at tables
The parents sat at tables
The students ate at desks

$$P(\text{counters}|\text{sat at})$$
$$Count(\text{sat at counters}) = 0$$

But we know $P(\text{counters}|\text{sat at}) > 0$

Why is having zero probability especially bad?

Smoothing: add a small value to all words in vocab so we don't get zeros

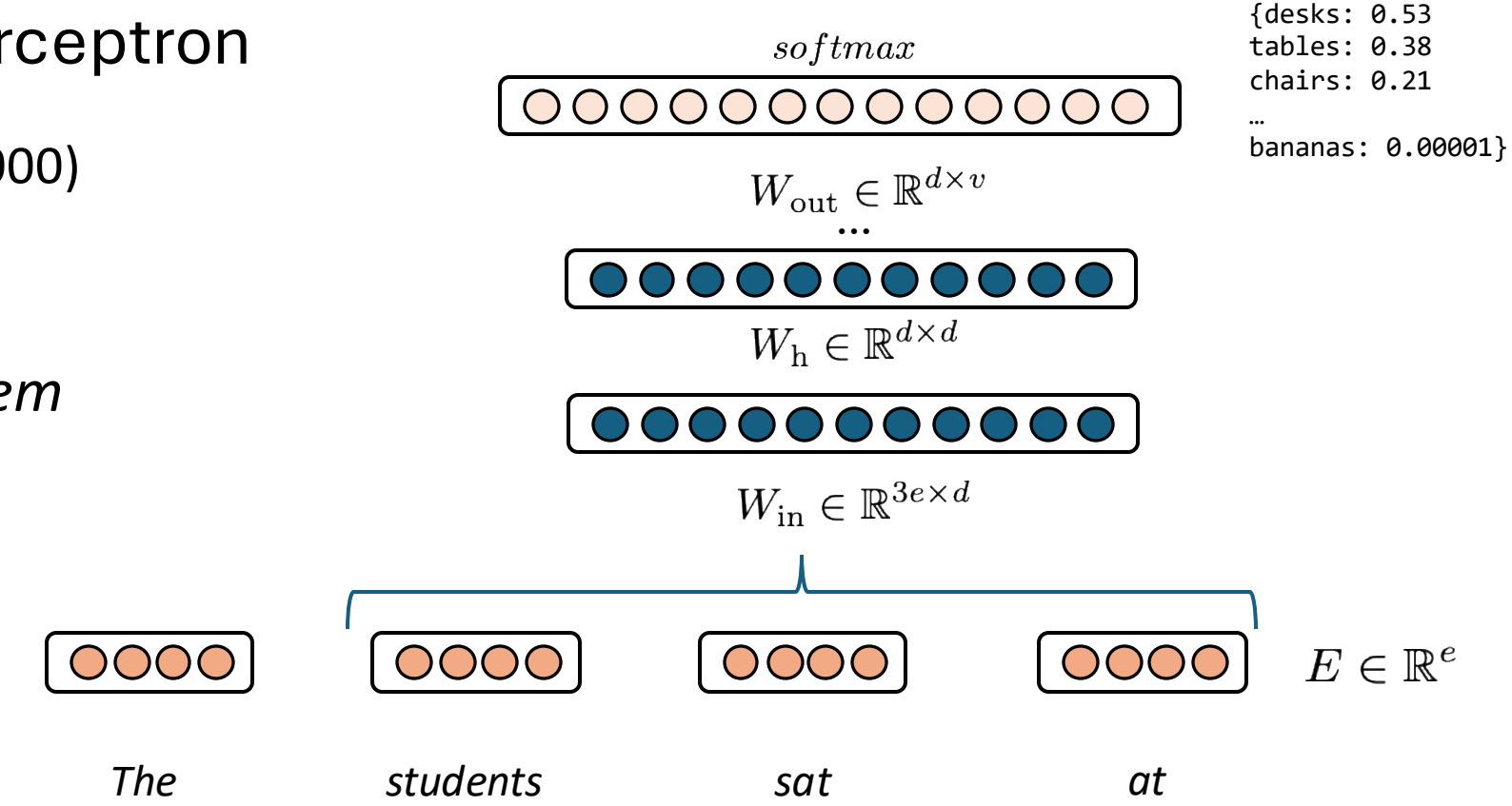
What about $P(\text{tables}|\text{rested at})$?

Backoff: If the full conditioner does not occur, use a smaller n-gram (*at*)

Neural LM

- Word embeddings: continuous vector representations of words
  - Fewer sparsity issues in continuous space!
- Multi-layer perceptron

Bengio et al. (2000)



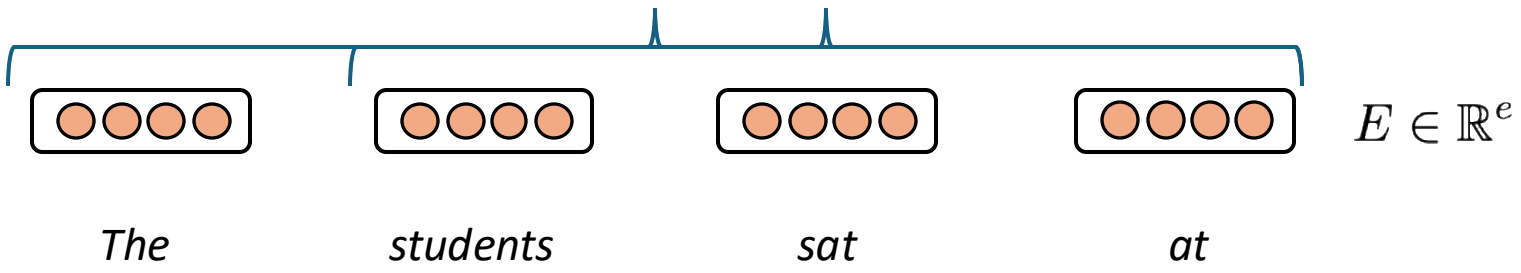
What is one key problem with this approach?

Basic neural LM still has a fixed window!

- We need to define a dimensionality for the input weight matrix

  $W_{\text{in}} \in \mathbb{R}^{3e \times d}$

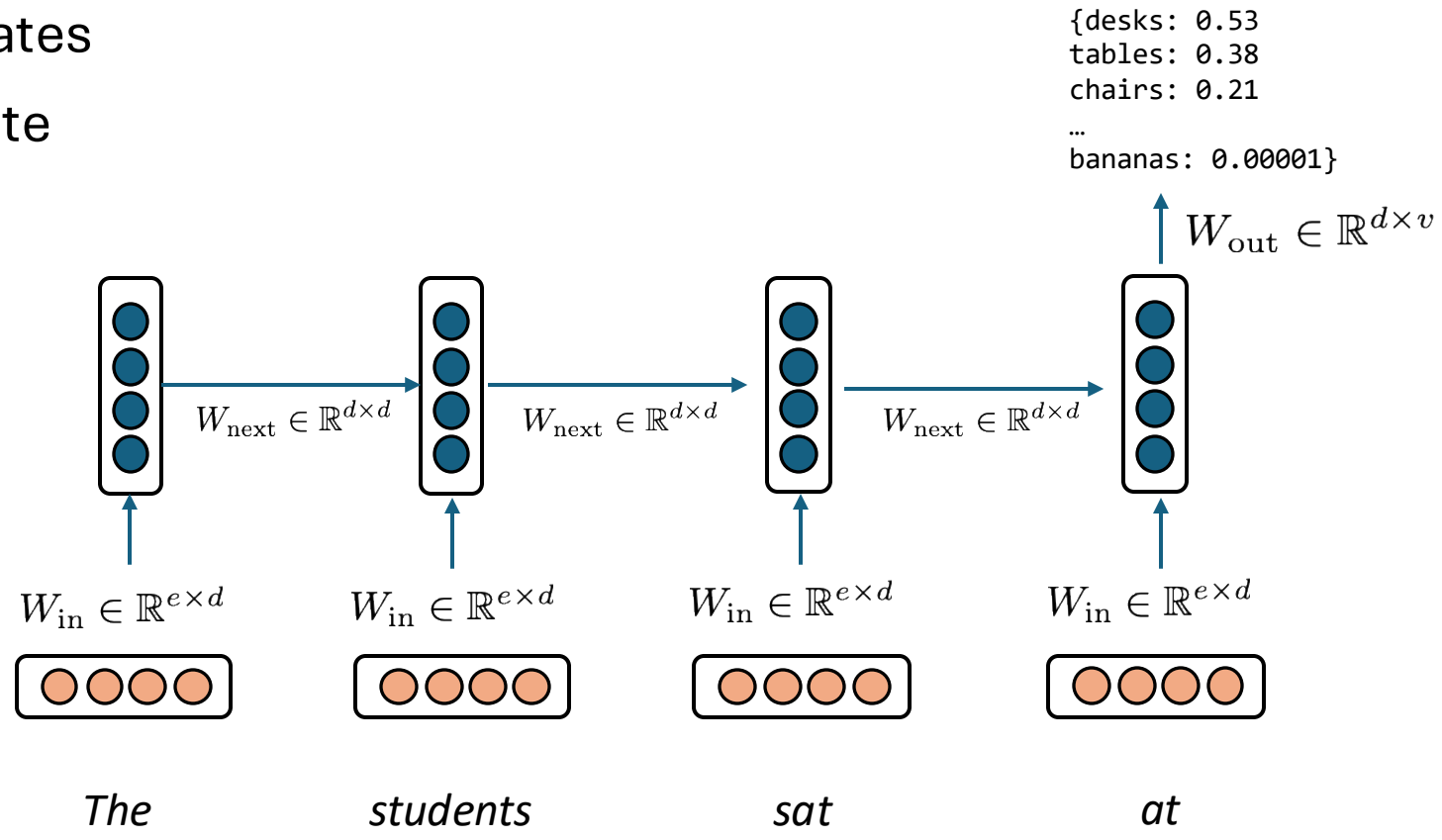
- If we want more context → bigger weight matrix, more expensive computation
- Language involves dealing with variable-length inputs!



Recurrent neural LM

- Apply same weights repeatedly
- Weight to map inputs to hidden states
- Weight to bring forward hidden state

Elman, 1990
Bengio et al.,1994
Hochreiter and Schmidhuber, 1997

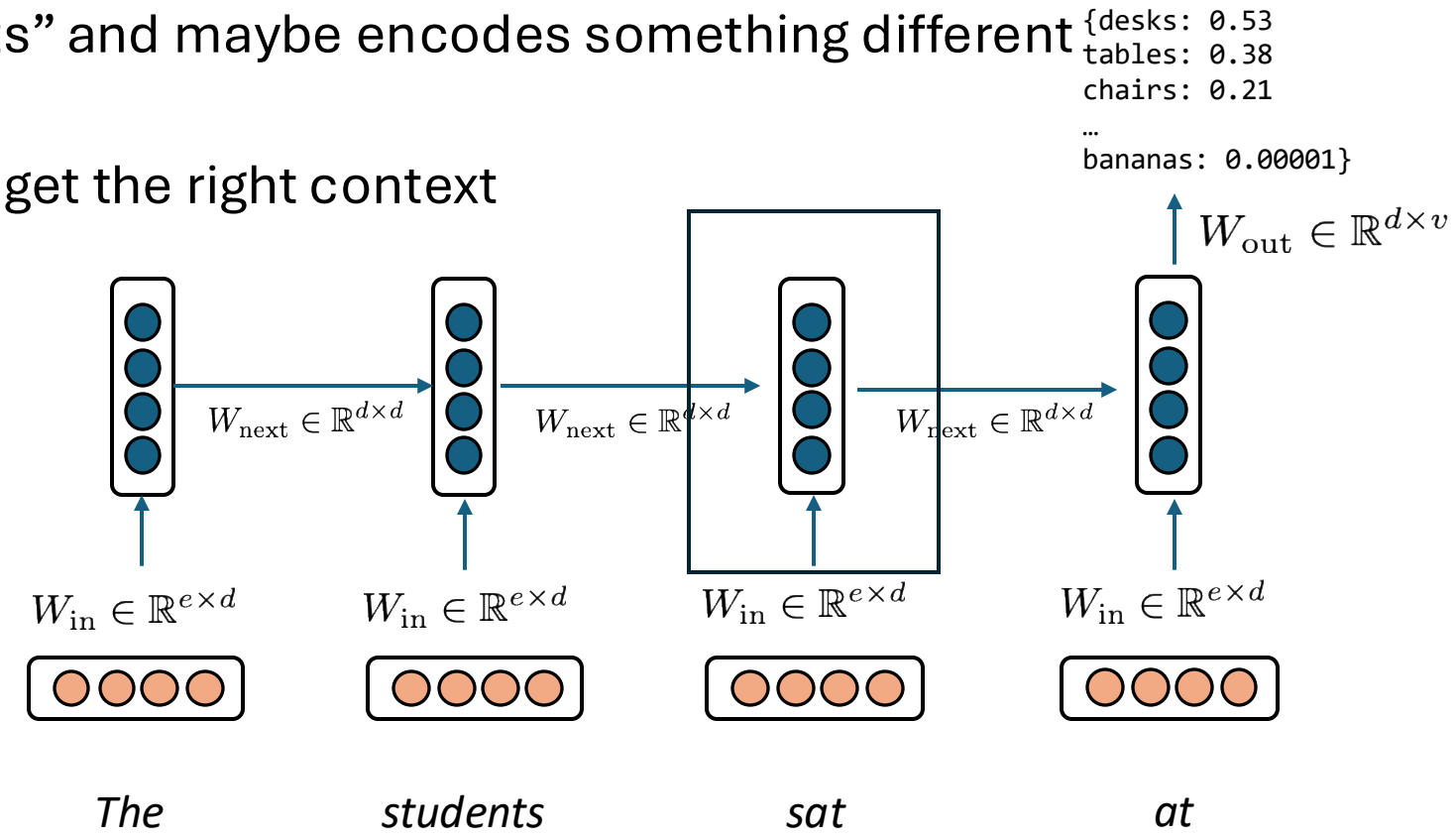


Detour: Contextualized embeddings

- In contrast to something like GLOVE (Pennington et al., 2014) or word2vec (Mikolov et al., 2013)
  - These embeddings were fixed per word: “apple” means the same thing no matter the context
- Context matters
  - “*I ate an apple”* is (hopefully) talking about a different *apple* than *“Apple stock rose by 0.5%”*
- We want our embeddings to be “contextualized”, i.e. different given different contexts
- RNNs can actually give us “contextualized” embeddings

Recurrent neural LM

- What if we just take the hidden state instead of the embedding?
- Now *sat* depends on “The students” and maybe encodes something different from “The elephants sat”
- We can make this bidirectional to get the right context
- This was done pre-BERT
  - ELMO (Peters et al., 2018)
    - Feb. 2018
  - BERT (Devlin et al., 2018)
    - **Oct. 2018**
- We’ll come back to this



Why not RNNs?

- Slow: They do not parallelize and there are O(n) non-parallel operations to encode n items
- Sequential dependency/lost context: Even modifications like LSTMs still don’t enable learning over very long sequences. Transformers can scale to thousands of words!
  - Meaning gets washed out across steps

Attention and Self-Attention

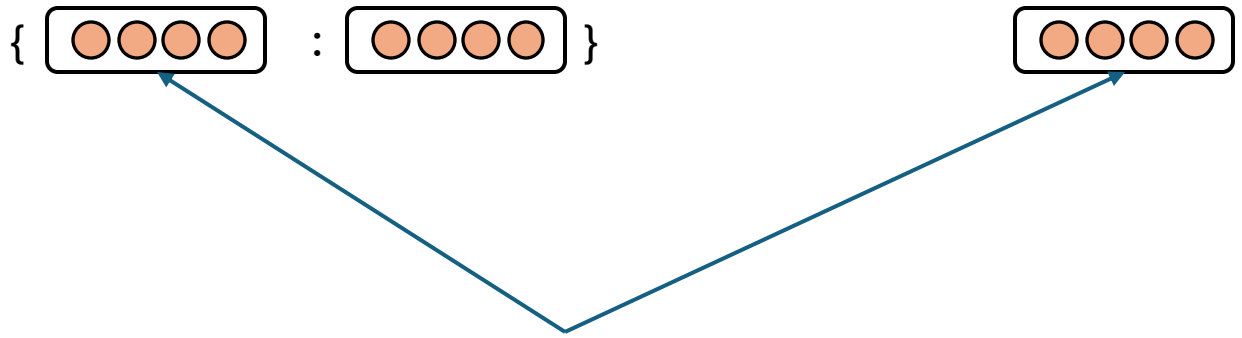
- Think of this as a kind of “soft” dictionary lookup table
  - Keys, Queries, Values

{“apple”: “A reddish fruit…”, …}

Key Value

“apple”

Query



Self-Attention Transformer (Vaswani et al.)

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

$$Q = EW^Q,\ K = EW^K,\ V = EW^V$$

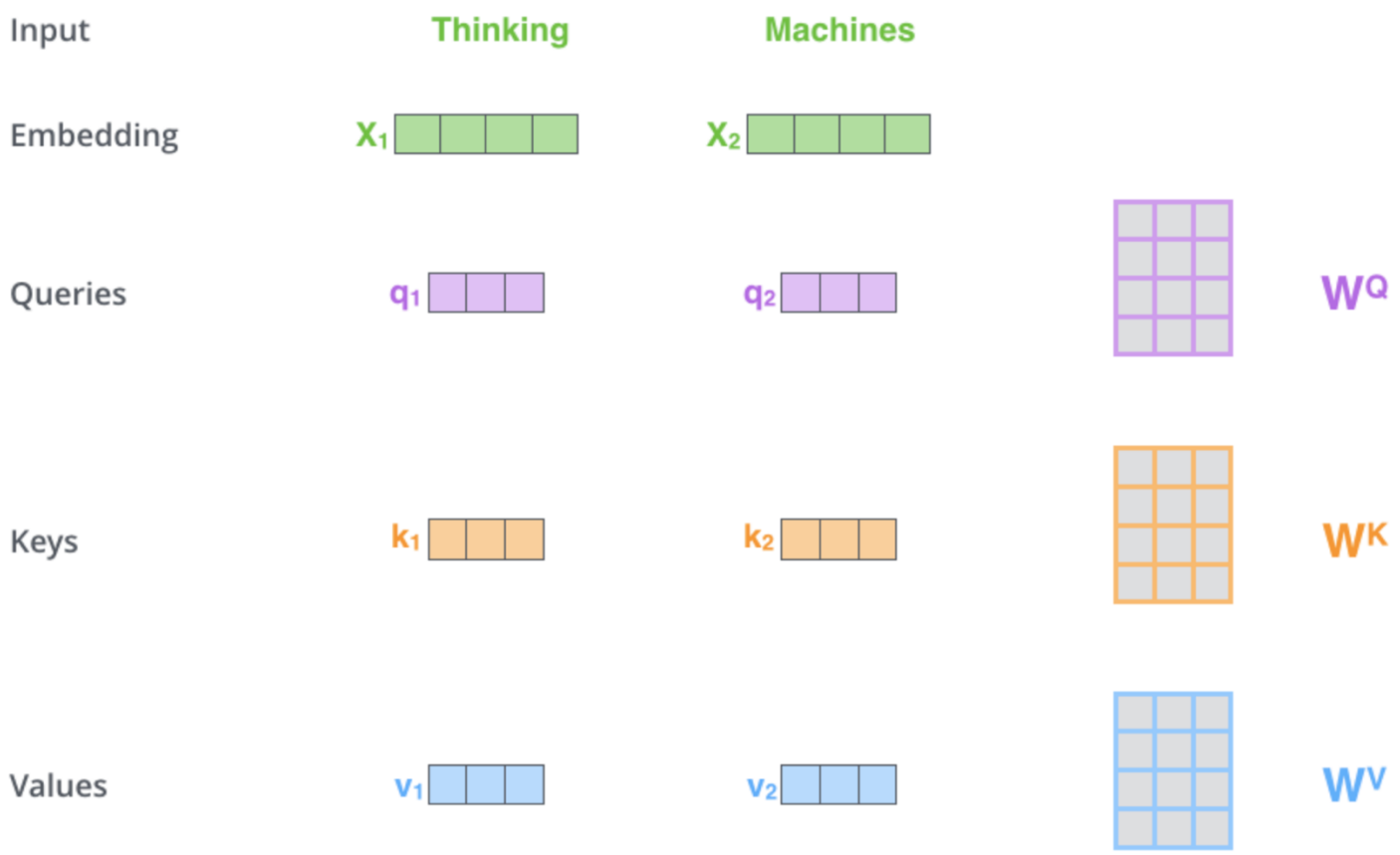
Average values together, weighted by how well they match

Get soft match between query and keys

- E is our word embedding (non-contextual for now)
- W's are weights (projecting embedding to latent space)
- Normalization to control scale
- This is just one head of attention

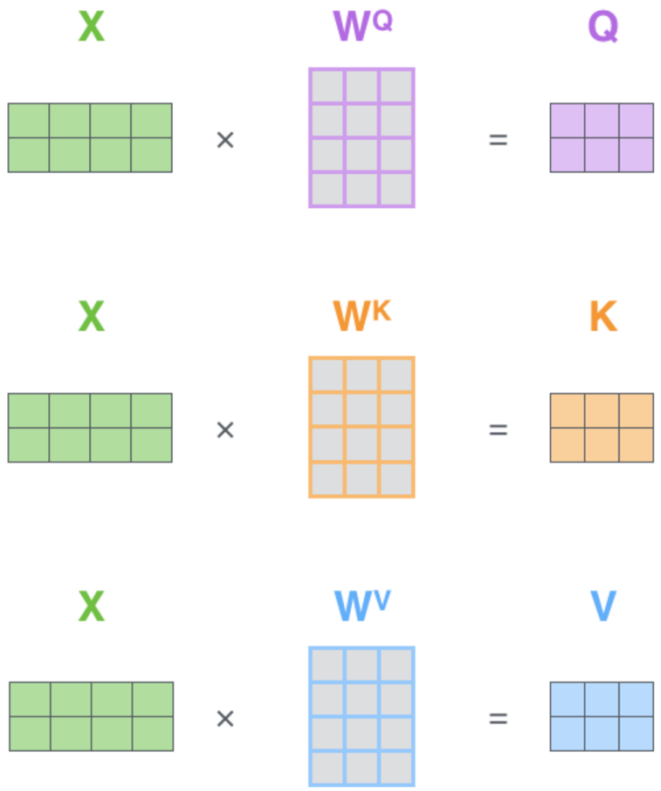
Self-Attention

Alammar, *The Illustrated Transformer*



Self-Attention

Alammar, *The Illustrated Transformer*



sent len x sent len (attn for each word to each other)

$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

sent len x hidden dim

Z is a weighted combination of V rows

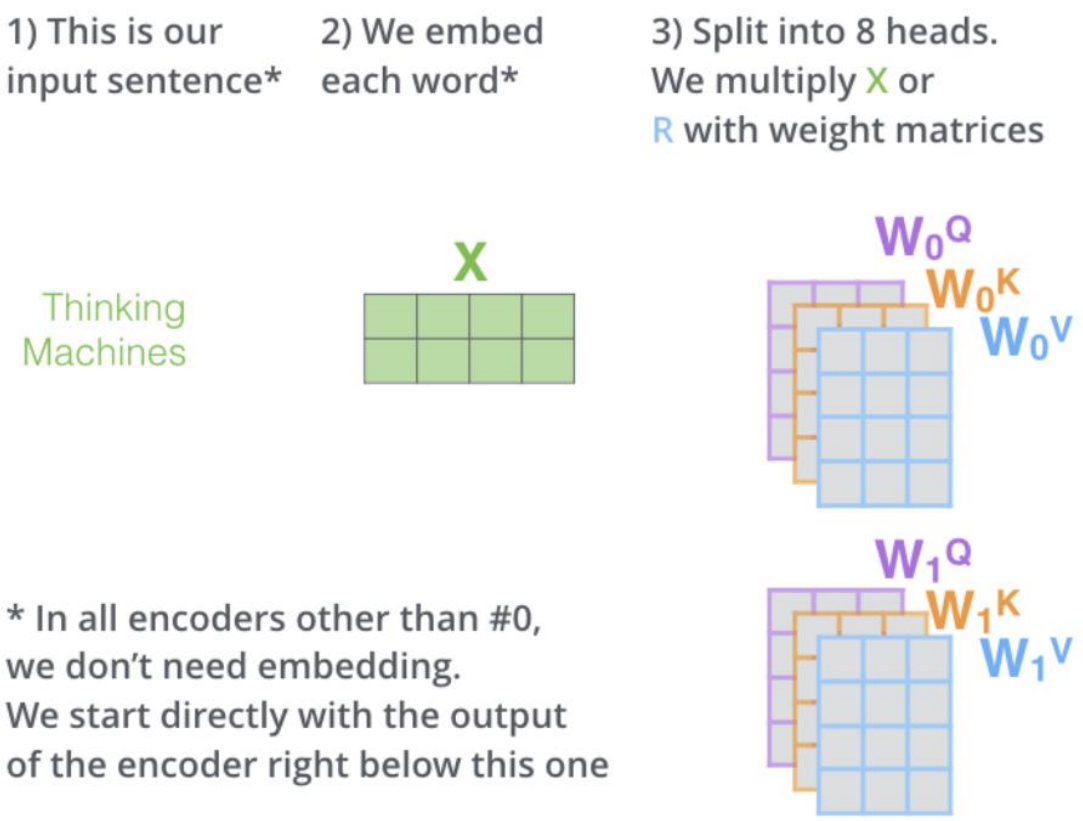
Why is this better?

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

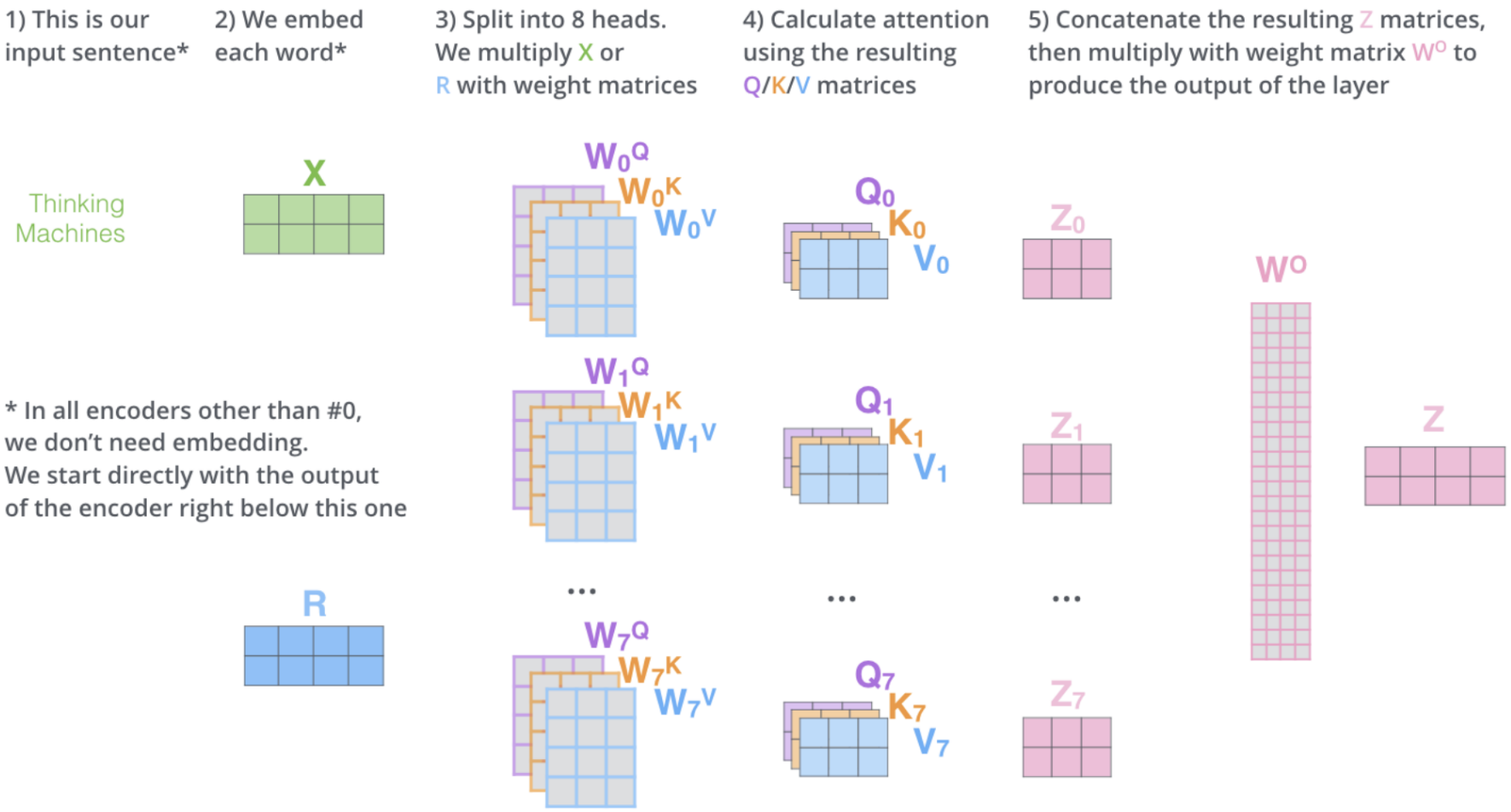
- n = sentence length, d = hidden dim, k = kernel size, r = restricted neighborhood size
- Quadratic complexity, but O(1) sequential operations (not linear like in RNNs) and O(1) “path” for words to inform each other
- So very parallelizable!

Multi-head Self-Attention

- In practice, Transformers have many copies (heads) of self-attention
- Also parallelizable

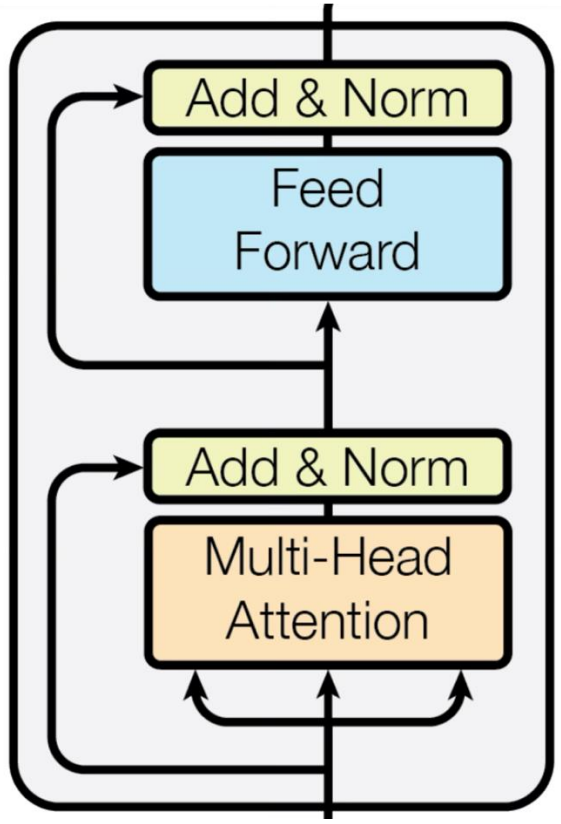


Multi-head Self-attention



Full Transformer stack

- Repeated layers with the same substructure
- Multi-head attention
- Residual + Layernorm
- FFN (most of the parameters)
- Residual + Layernorm
- Note: there are many variations on this basic theme
  - Most prominently: Pre-norm



BERT (Devlin et al., 2018)

- Pre-trained Transformer model
- A bit of historical context
  - The last big NLP revolution before this was ~2013/2014 with word embeddings
  - Before that: words = 1-hot vectors
  - Word embeddings: learned vector representations of word distributions
  - Brought in a crucial idea: these vectors are something that can be shared across the community
  - Word2vec and Glove were made publicly available and easy to download and use
- BERT follows this paradigm but for a whole model
  - Instead of just sharing the embeddings, share a model that can embed words on-the-fly

Masked language modeling (MLM)

- Normally when we train a language model
  - Treat it like an N-gram model (P(token | prefix))
- BERT introduced a different paradigm, enabled by Transformer's architecture

This is a sentence we are modeling

This is a _____ we are modeling

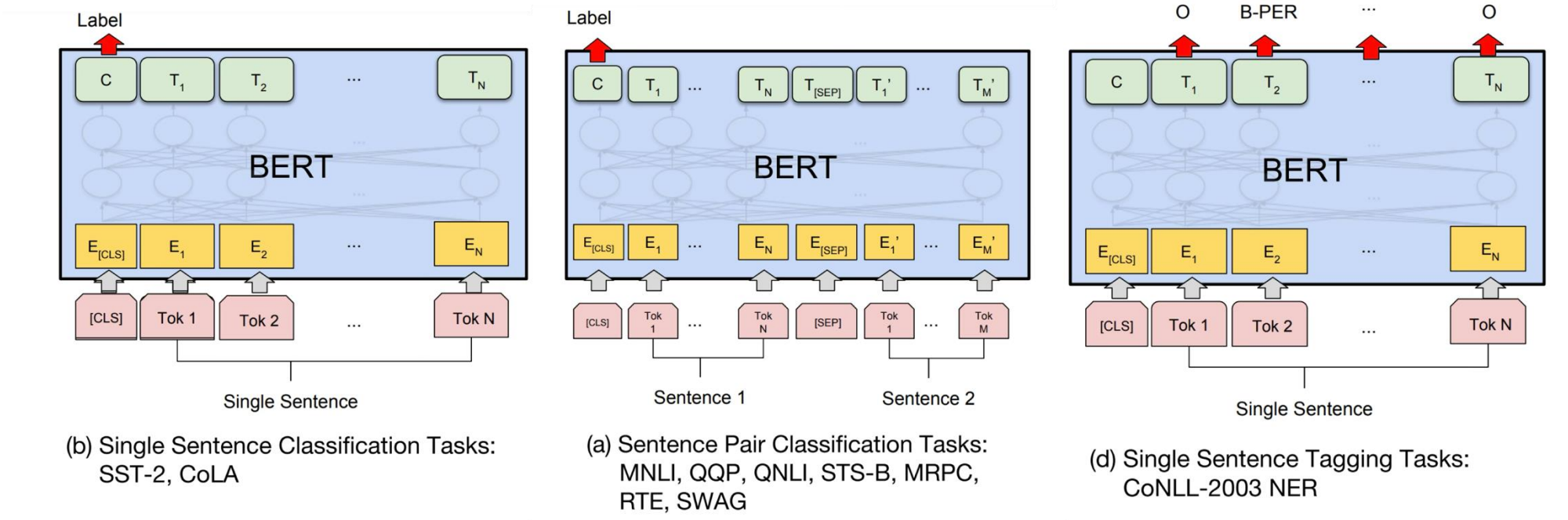
Model: P(____ | This is a … we are modeling)

- Questions for you:
  - Why is this something that's easier to do with a Transformer than an RNN?

Special tokens

- BERT also introduced a few “special” tokens
- [MASK]: masking tokens (for filling in blanks)
- [CLS]: Classification token (pooled representation of the whole sentence/input)
- [SEP]: separating sentences/lines/documents

A few things BERT can do:

- 

- Classification with [CLS]
- Similarity
- Token-level tagging

Question: What is missing here?

- What can’t BERT do (at least, not easily)?

Results

- For the sake of time, we will skip detailed results
- TLDR: BERT vastly outperformed most prior solutions across most tasks
  - To the point where the tasks NLP cared about fundamentally changed
  - For most applications, BERT + Finetuning became standard approach
- In general, LLMs do better across tasks now (and the tasks we care about have changed)
  - Exception: retrieval (e.g. ModernBERT)
- Main innovation/takeaway:
  - Shift in NLP towards using shared models
  - Transformers library introduced (which would later become Huggingface)

GPT-3 and Prompting/Few-shot Learning

- Between BERT and GPT-3: GPT-2
  - Big news
  - Encoder-only vs. Decoder-only
- Encoder-only
  - All the tokens come in at once (like BERT)
  - Can’t generate
- Decoder-only
  - ”Autoregressive”, one token at a time
  - Generates, but not as good as an embedding
- SCALE!
  - BERT-Large (340M) → GPT2 (774M) → GPT3 (175 **B**)

Zero-shot Prompting

- You have a single datapoint *x* that you want to predict a label *y* for
- E.g. if you're doing sentiment analysis

> ***x*** = *The movie's acting could've been better, but the visuals and directing were top-notch*.

- You add a "prompt" which tells the model what you want it to do

> Review: *The movie's acting could've been better, but the visuals and directing were top-notch*.
> On a scale from 1 to 5, this review is a

- For models like GPT3, this needed to be an autoregressive prompt

Few-shot prompting

Brown et al. (2020)

- Add multiple examples of the task you want to the prompt

Review: *The movie was a complete disaster*
On a scale from 1 to 5, this review is a 1

Review: *I loved this movie, it's my favorite*
On a scale from 1 to 5, this review is a 5

Review: *The movie was average, I don't feel strongly about it*
On a scale from 1 to 5, this review is a 3

Review: *The movie's acting could've been better, but the visuals and directing were top-notch*.
On a scale from 1 to 5, this review is a

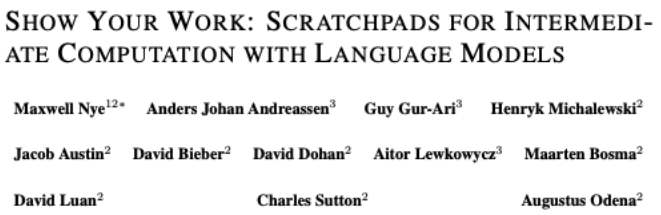
Few-shot prompting

- Powerful way to shape model responses
  - Teach the task you want the model to do
  - No training needed!
  - Not much data needed (5-10 examples are common, instead of 1000s for training)
- Open questions
  - How does ICL work?
  - Can ICL teach models new information?

Chain-of-Thought

Wei et al. (2022)

- Most well-known paper in line of work now thought of as "reasoning"
  - Actually comes after "scratchpad" work
- Seemingly-simple idea:
  - Ask model to reason "step-by-step" in the prompt



SHOW YOUR WORK: SCRATCHPADS FOR INTERMEDIATE COMPUTATION WITH LANGUAGE MODELS

Maxwell Nye[1,2*] Anders Johan Andreassen[3] Guy Gur-Ari[3] Henryk Michalewski[2]
Jacob Austin[2] David Bieber[2] David Dohan[2] Aitor Lewkowycz[3] Maarten Bosma[2]
David Luan[2] Charles Sutton[2] Augustus Odena[2]

Nye et al. (2021)

Review: *The movie's acting could've been better, but the visuals and directing were top-notch*.
Give the score for this review on a scale from 1 to 5. First, let's reason step-by-step…

1. It seems like the author wasn't satisfied with the acting
2. The author highlights "top-notch" visuals and directing

Based on this, I would give the movie a 4

Putting things together

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao[*,1], Jeffrey Zhao[2], Dian Yu[2], Nan Du[2], Izhak Shafran[2], Karthik Narasimhan[1], Yuan Cao[2]

[1]Department of Computer Science, Princeton University
[2]Google Research, Brain team
[1]{shunyuy,karthikn}@princeton.edu
[2]{jeffreyzhao,dianyu,dunan,izhak,yuancao}@google.com

About the paper

- ICLR 2023
- Coming from a group working generally on agents
- Last author was on GPT2 paper
- We’ll see other datasets/papers from the same group

Background

- CoT: improves heavily on reasoning
  - LLMs applied to reasoning tasks like GSM8K (math reasoning)

Problem: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?
Solution: Beth bakes 4 2 dozen batches of cookies for a total of 4*2 = <<4*2=8>>8 dozen cookies
There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of 12*8 = <<12*8=96>>96 cookies
She splits the 96 cookies equally amongst 16 people so they each eat 96/16 = <<96/16=6>>6 cookies
Final Answer: 6

Problem: Mrs. Lim milks her cows twice a day. Yesterday morning, she got 68 gallons of milk and in the evening, she got 82 gallons. This morning, she got 18 gallons fewer than she had yesterday morning. After selling some gallons of milk in the afternoon, Mrs. Lim has only 24 gallons left. How much was her revenue for the milk if each gallon costs $3.50?
Mrs. Lim got 68 gallons - 18 gallons = <<68-18=50>>50 gallons this morning.
So she was able to get a total of 68 gallons + 82 gallons + 50 gallons = <<68+82+50=200>>200 gallons.
She was able to sell 200 gallons - 24 gallons = <<200-24=176>>176 gallons.
Thus, her total revenue for the milk is $3.50/gallon x 176 gallons = $<<3.50*176=616>>616.
Final Answer: 616

Problem: Tina buys 3 12-packs of soda for a party. Including Tina, 6 people are at the party. Half of the people at the party have 3 sodas each, 2 of the people have 4, and 1 person has 5. How many sodas are left over when the party is over?
Solution: Tina buys 3 12-packs of soda, for 3*12= <<3*12=36>>36 sodas
6 people attend the party, so half of them is 6/2= <<6/2=3>>3 people
Each of those people drinks 3 sodas, so they drink 3*3=<<3*3=9>>9 sodas
Two people drink 4 sodas, which means they drink 2*4=<<4*2=8>>8 sodas
With one person drinking 5, that brings the total drank to 5+9+8+3= <<5+9+8+3=25>>25 sodas
As Tina started off with 36 sodas, that means there are 36-25=<<36-25=11>>11 sodas left
Final Answer: 11

OpenAI (2021)

Background

- We'll see later in the course: LLMs applied to action

Huang et al. (2022)

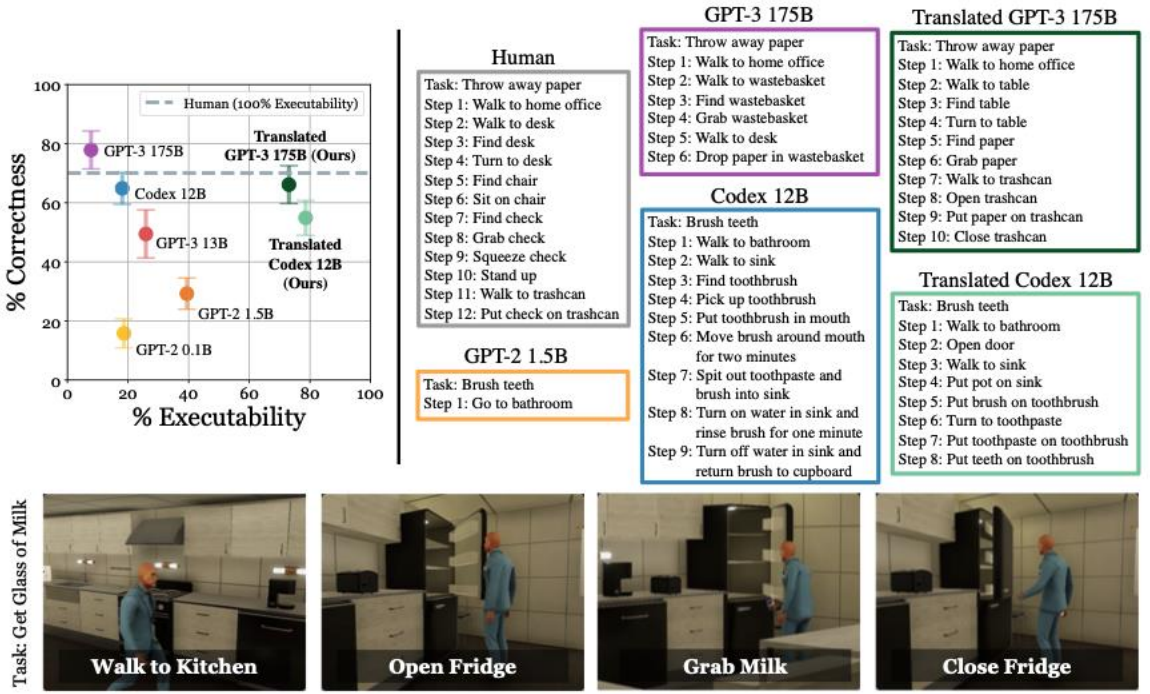
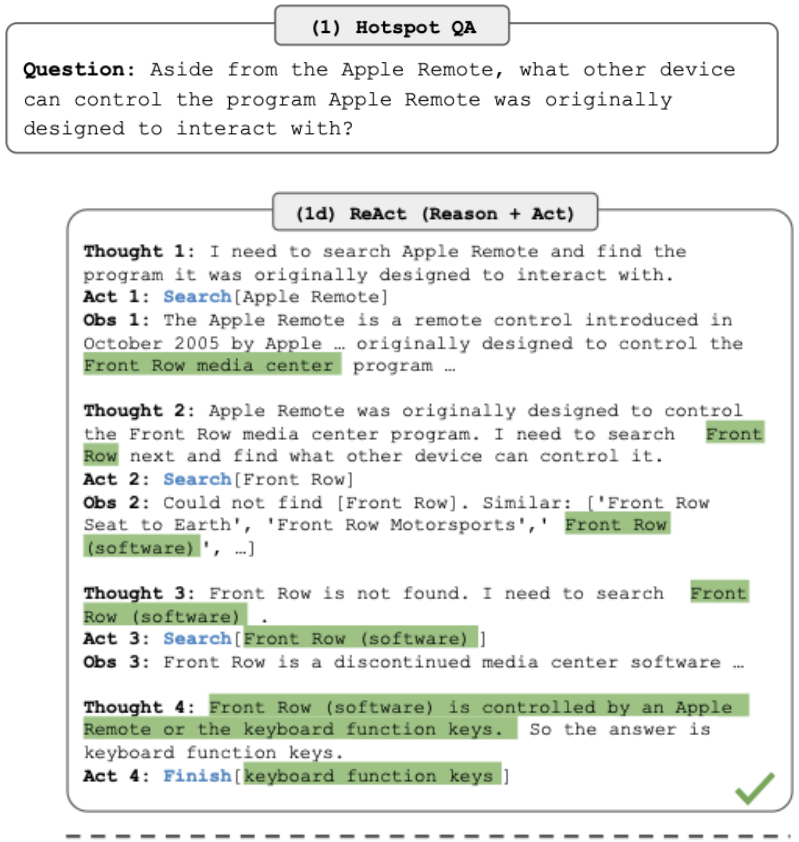


Figure 1: Executability v.s. semantic correctness of generated plans **(left)**, sample plans by different models **(right)**, and example environment execution **(bottom)**. Large models can produce action plans indistinguishable from those by humans, but frequently are not executable in the environment. Using our techniques, we can significantly improve executability, albeit at the cost of correctness. More samples can be found in Appendix A.5.

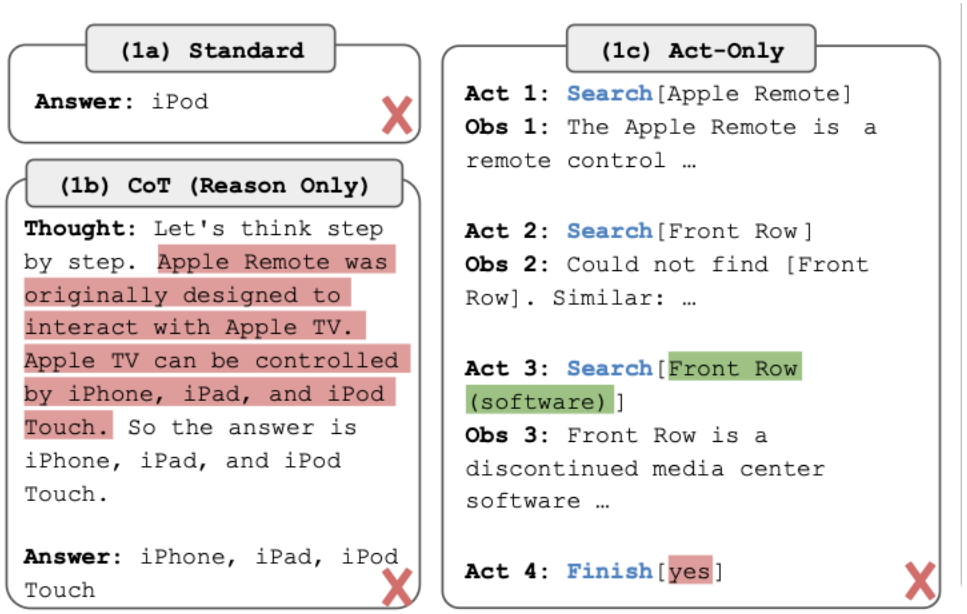
ReACT: Combining these two directions

- First reason (produce a CoT), then produce an action
- ReACT is effectively a prompt format
  - Encourages model to produce “thoughts” before acting



Problem with CoT and action only

- CoT only: good reasoning, but actions don’t correspond to anything that the agent can actually do
- Act-only: actions are real, but they are not correct



ReACT approach

- “augment the agent’s action space to… the space of language”
  - Sounds complicated
- Easier to understand by just looking at the method (in appendix)

ReAct

Question	What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?
Thought 1	I need to search Colorado orogeny, find the area that the eastern sector of the Colorado orogeny extends into, then find the elevation range of the area.
Action 1	Search[Colorado orogeny]
Observation 1	The Colorado orogeny was an episode of mountain building (an orogeny) in Colorado and surrounding areas.
Thought 2	It does not mention the eastern sector. So I need to look up eastern sector.
Action 2	Lookup[eastern sector]
Observation 2	(Result 1 / 1) The eastern sector extends into the High Plains and is called the Central Plains orogeny.
Thought 3	The eastern sector of Colorado orogeny extends into the High Plains. So I need to search High Plains and find its elevation range.
Action 3	Search[High Plains]
Observation 3	High Plains refers to one of two distinct land regions
Thought 4	I need to instead search High Plains (United States).
Action 4	Search[High Plains (United States)]

ICL example

Human-written thought (2 purposes)

Human-written action

Observation from env. after executing action

ReACT is using ICL

- 6 ICL questions
  - Each has a sequence of <thought, action, observation>
- These sequences instruct the model on how to respond to a given question/observation

Datasets

- QA
  - HotpotQA: requires multi-step reasoning over multiple Wikipedia passages
  - FEVER: fact verification with claims that are supported, refuted, or neither (not enough info) based on Wikipedia
  - Actions: search, command+F, finish

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

Yang et al. (2018)

Claim: The Rodney King riots took place in the most populous county in the USA.

[wiki/Los_Angeles_Riots]
The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, arsons, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.

[wiki/Los_Angeles_County]
Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.

Verdict: Supported

Figure 1: Manually verified claim requiring evidence from multiple Wikipedia pages.

Thorne et al. (2018)

Datatsets

- Decision-Making
  - ALFWorld: text game version of ALFRED
  - WebShop: simulated online shopping environment
  - Will see both of these again later on

```
You are in the middle of a room. Looking quickly around you, you see a drawer 2,
a shelf 5, a drawer 1, a shelf 4, a sidetable 1, a drawer 5, a shelf 6, a shelf
1, a shelf 9, a cabinet 2, a sofa 1, a cabinet 1, a shelf 3, a cabinet 3, a
drawer 3, a shelf 11, a shelf 2, a shelf 10, a dresser 1, a shelf 12, a
garbagecan 1, a armchair 1, a cabinet 4, a shelf 7, a shelf 8, a safe 1, and a
drawer 4.

Your task is to: put some vase in safe.

> go to shelf 6
You arrive at loc 4. On the shelf 6, you see a vase 2.

> take vase 2 from shelf 6
You pick up the vase 2 from the shelf 6.
```

Shridhar et al. (2020)

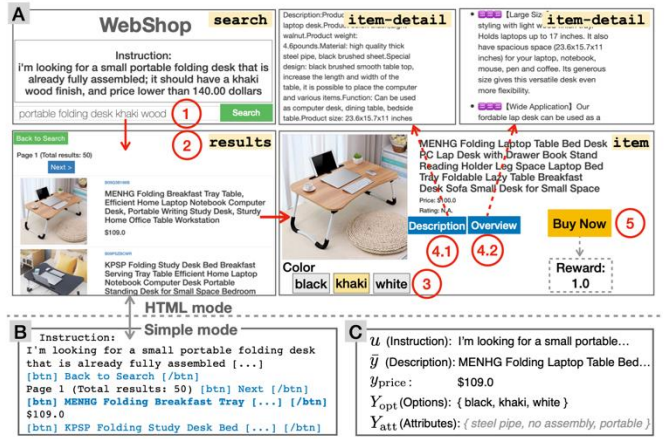


Figure 1: The WebShop environment. **A**: An example task trajectory in HTML mode, where a user can (1) search a query in a search page, (2) click a product item in a results page, (3) choose a color option in a item page, (4) check item-detail pages and go back to the item page, and (5) finally buy the product to end the episode and receive a reward $r \in [0, 1]$ (§3.2). **B**: the results page in simple mode for agent training and evaluation. The blue text indicates clickable actions and bold text indicates an action selected by the agent. **C**: The product notation used in §3 with corresponding examples from the product in **A**. The attributes Y_{att} are hidden from the task performer.

Yao et al. (2023)

QA Baselines

- Standard prompting (just provide action directly, no thoughts)
- CoT (allow thoughts but no observations/actions in ICL)
- Acting-only (ACT): no thoughts
- Another method
  - ReACT → CoT-SC
  - If ReACT fails to provide an answer, back off to CoT with self-consistency
  - Self-Consistency: Vote over multiple answers
    - Sample N CoTs and then take the majority answer

QA Results

- ReACT > ACT
  - On PALM-540B (Google model from a few years ago, has been supplanted)
- ReACT alone:
  - Not much better than Standard
  - Not much better than CoT
  - Worse/same as CoT-SC
- ReACT + CoT-SC
  - Consistently better than baselines
- Note: still far below full finetuning SOTA

Prompt Method[a]	HotpotQA (EM)	Fever (Acc)
Standard	28.7	57.1
CoT (Wei et al., 2022)	29.4	56.3
CoT-SC (Wang et al., 2022a)	33.4	60.4
Act	25.7	58.9
ReAct	27.4	60.9
CoT-SC → ReAct	34.2	**64.6**
ReAct → CoT-SC	**35.1**	62.0
Supervised SoTA[b]	67.5	89.5

Table 1: PaLM-540B prompting results on HotpotQA and Fever.

Where is performance/error coming from?

- Error analysis
  - ReACT suffers from repetitive steps but has fewer hallucinations

	Type	Definition	ReAct	CoT
Success	True positive	Correct reasoning trace and facts	94%	86%
	False positive	Hallucinated reasoning trace or facts	6%	14%
Failure	Reasoning error	Wrong reasoning trace (including failing to recover from repetitive steps)	47%	16%
	Search result error	Search return empty or does not contain useful information	23%	-
	Hallucination	Hallucinated reasoning trace or facts	0%	56%
	Label ambiguity	Right prediction but did not match the label precisely	29%	28%

Table 2: Types of success and failure modes of ReAct and CoT on HotpotQA, as well as their percentages in randomly selected examples studied by human.

Decision-making Baselines

- ALFWorld:
  - ACT
  - BUTLER (supervised system from Shridhar et al. (2020))
    - Trained to copy successful trajectories, no thoughts
    - Not exactly fair comparison since this is trained, ReACT is not
- WebShop
  - ACT
  - Their own supervised (imitation learning) baseline
  - Imitation learning + reinforcement learning

Decision-Making results

- Metrics
  - Best of N trials
  - Success rate / Score

Method	Pick	Clean	Heat	Cool	Look	Pick 2	All
`Act` (best of 6)	88	42	74	67	72	**41**	45
`ReAct` (avg)	65	39	83	76	55	24	57
`ReAct` (best of 6)	**92**	58	**96**	86	**78**	**41**	**71**
`ReAct-IM` (avg)	55	59	60	55	23	24	48
`ReAct-IM` (best of 6)	62	**68**	87	57	39	33	53
BUTLER$_g$ (best of 8)	33	26	70	76	17	12	22
BUTLER (best of 8)	46	39	74	**100**	22	24	37

Table 3: AlfWorld task-specific success rates (%). BUTLER and BUTLER$_g$ results are from Table 4 of Shridhar et al. (2020b). All methods use greedy decoding, except that BUTLER uses beam search.

Method	Score	SR
`Act`	62.3	30.1
`ReAct`	**66.6**	**40.0**
IL	59.9	29.1
IL+RL	62.4	28.7
Human Expert	82.1	59.6

Table 4: Score and success rate (SR) on Webshop. IL/IL+RL taken from Yao et al. (2022).

Decision-Making results

- ReACT > ACT on ALFRED ReACT > BUTLER on ALFRED
  - Is this a fair comparison?
    - Hard to say
    - BUTLER was a custom architecture, trained largely from scratch (so some advantages, som disadvantages)
    - BUTLER is trained, but has far fewer parameter (used BERT embeddings)
    - ReACT uses PALM-540B, but not trained

Method	Pick	Clean	Heat	Cool	Look	Pick 2	All
`Act` (best of 6)	88	42	74	67	72	**41**	45
`ReAct` (avg)	65	39	83	76	55	24	57
`ReAct` (best of 6)	**92**	58	**96**	86	**78**	**41**	**71**
`ReAct-IM` (avg)	55	59	60	55	23	24	48
`ReAct-IM` (best of 6)	62	**68**	87	57	39	33	53
BUTLER$_g$ (best of 8)	33	26	70	76	17	12	22
BUTLER (best of 8)	46	39	74	**100**	22	24	37

Table 3: AlfWorld task-specific success rates (%). BUTLER and BUTLER$_g$ results are from Table 4 of Shridhar et al. (2020b). All methods use greedy decoding, except that BUTLER uses beam search.

Decision-Making results

- On WebShop, ReACT > ACT
- Also better than IL and IL+RL
- Here, comparison is a bit fairer
  - IL and IL+RL models are more recent, Transformer-based
  - But still smaller

Method	Score	SR
Act	62.3	30.1
ReAct	**66.6**	**40.0**
IL	59.9	29.1
IL+RL	62.4	28.7
Human Expert	82.1	59.6

Table 4: Score and success rate (SR) on Webshop. IL/IL+RL taken from Yao et al. (2022).

Takeaways

- Remember, all of this is just conditioning
  - Ultimately the model is just a conditional distribution P(w | prefix)
  - All of these methods boil down to one thing: getting the prefix right s.t. the next word is right
  - This can mean 2 things
    - Next word is the right answer (e.g. the right action)
    - Next word is something that gets added to the prefix and leads eventually to the right answer (CoT, ReACT)
- A lot of prompting comes down to this insight
  - Can you give your model the right conditioner (e.g. ICL examples)?
  - Can you get your model to produce the right conditioner (e.g. ReACT)?

Next class

- Semantic parsing
- Much more complex action spaces
- Tool use
  - Interleave reasoning and action even more