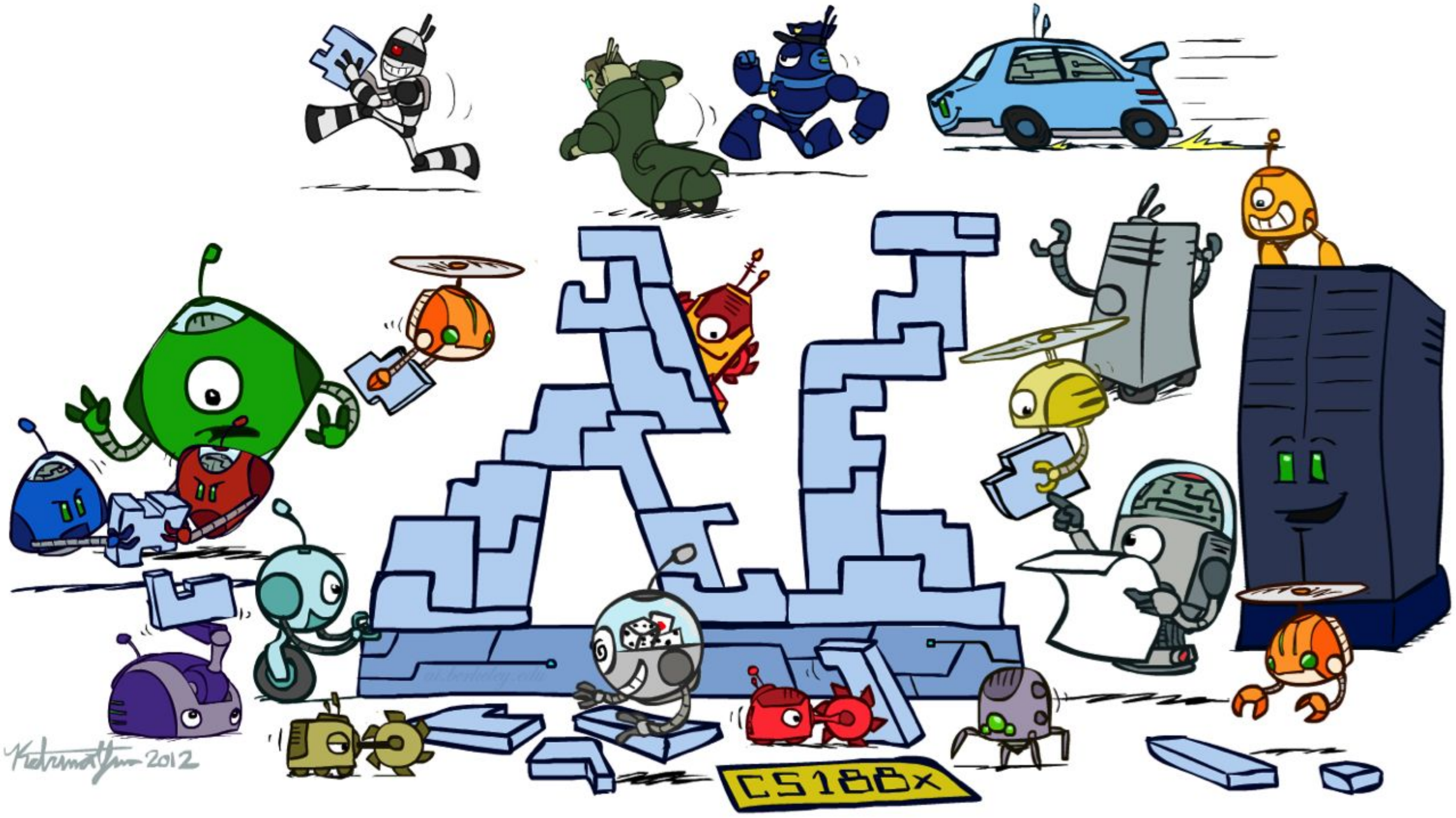
CS 188: Artificial Intelligence
Language Models II



[These slides were created by Cam Allen, Michael Cohen, Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

All CS188 materials are available at http://ai.berkeley.edu.]

Recap: Language Models

- Language models assign probabilities to sequences of words

$$P(w_1 \dots w_n)$$

- Use chain rule to generate words left-to-right

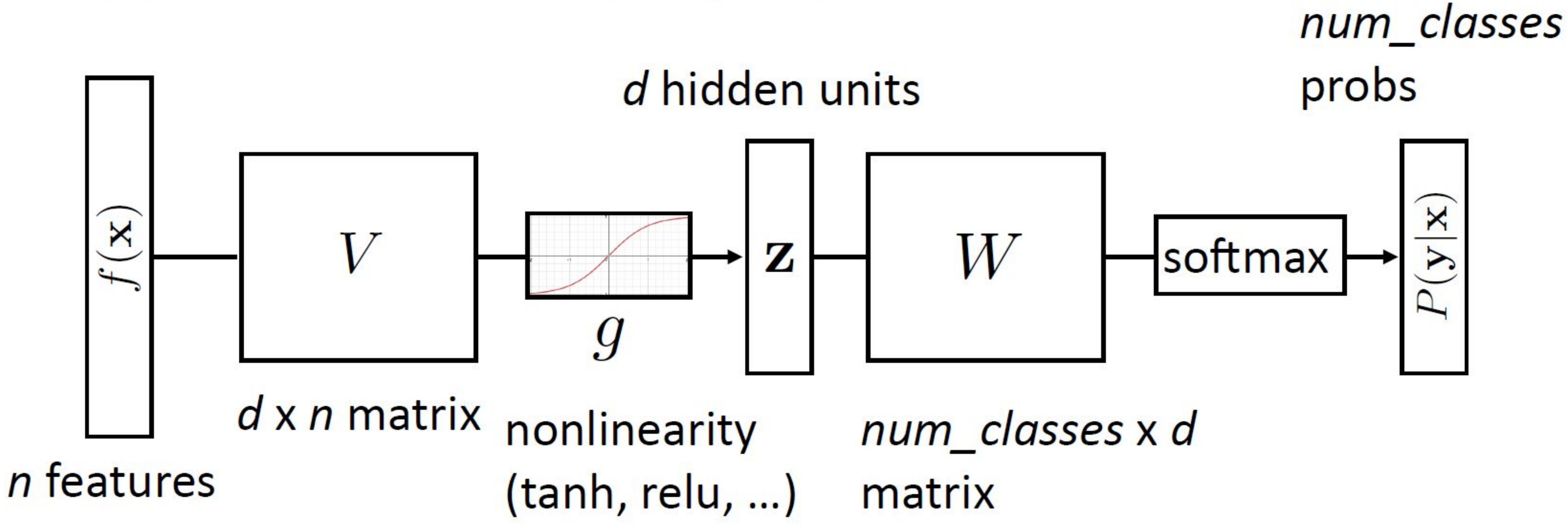
$$P(w_1 \dots w_n) = \prod_i P(w_i | w_1 \dots w_{i-1})$$

- Make assumptions to approximate the map from contexts to words
- The Markov assumption drops non-local context

$$P(\text{please close the door}) = \ P(\text{please}|\text{START})P(\text{close}|\text{please}) \dots P(\text{STOP}|door)$$

Recap: Neural Networks

$$P(\mathbf{y}|\mathbf{x}) = \text{softmax}(Wg(Vf(\mathbf{x})))$$

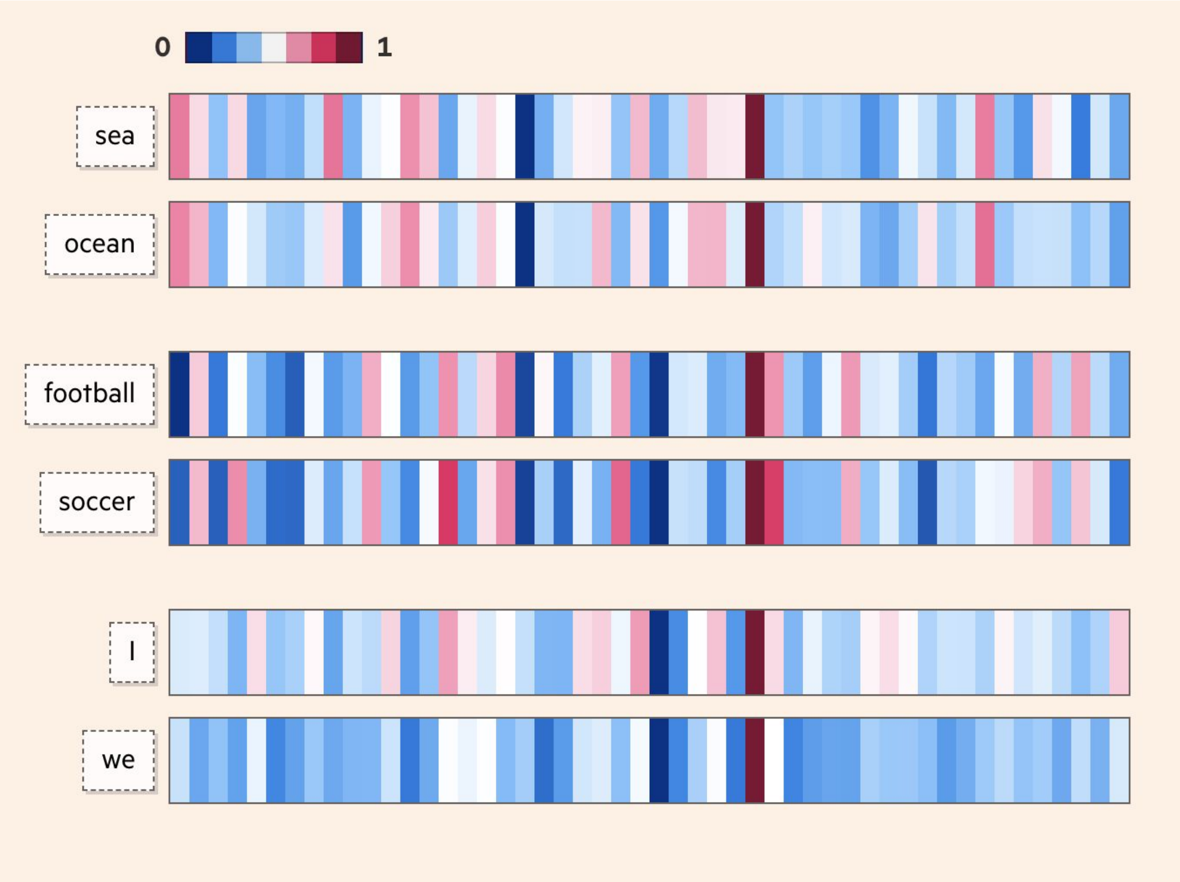


Neural LMs: Three Key Ideas

- Word embeddings
  - Different words are not entirely unrelated events
  - Words can be more and less similar, in complex ways
- Partially factored representations
  - Multiple semi-independent processes happen in parallel in language
  - It’s too expensive to track language in an unfactored way, and too inaccurate to assume everything of interest is independent
- Long distance dependencies
  - Information can be relevant without being local
  - Different notions of locality are important at different times

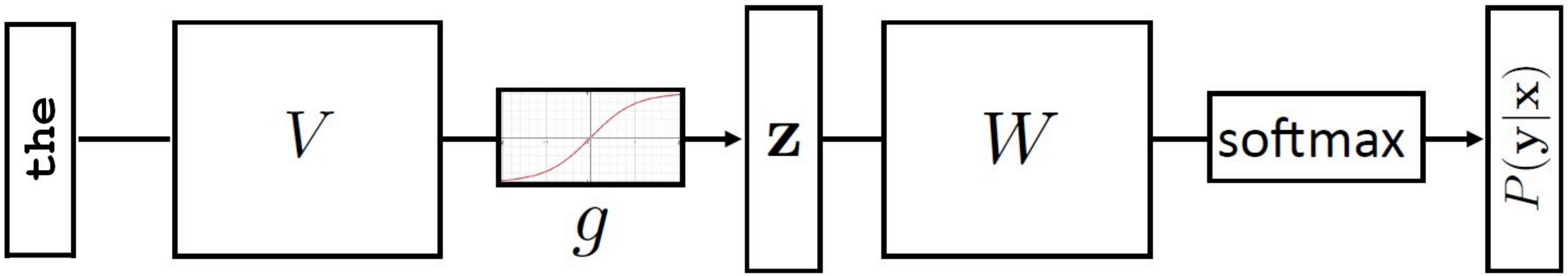
Word Embeddings

- Words get mapped to vectors
- Similarity and distance become defined
- Now suitable for input into continuous models (like NNs)
- Many ways to learn them

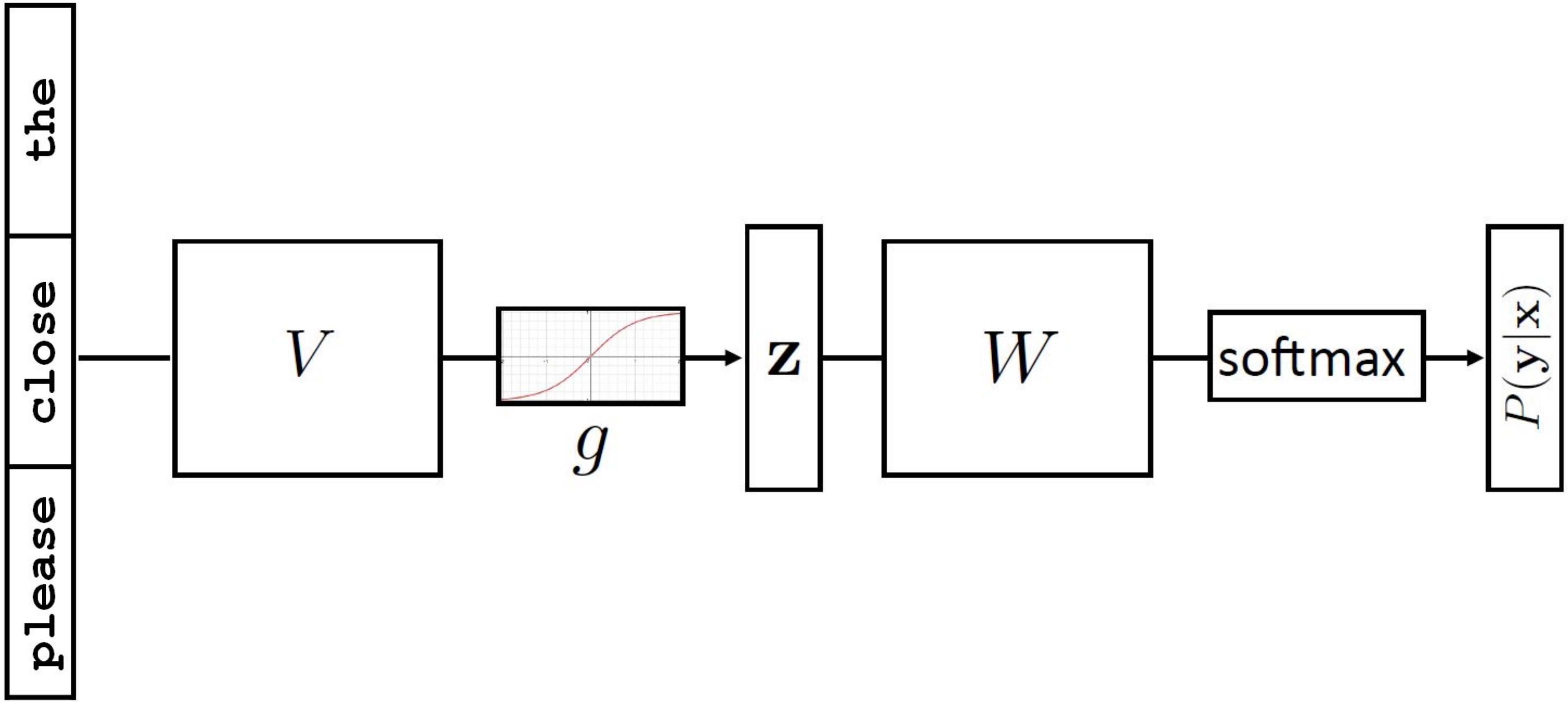


ig.ft.com/generative-ai

A Neural Bigram Model

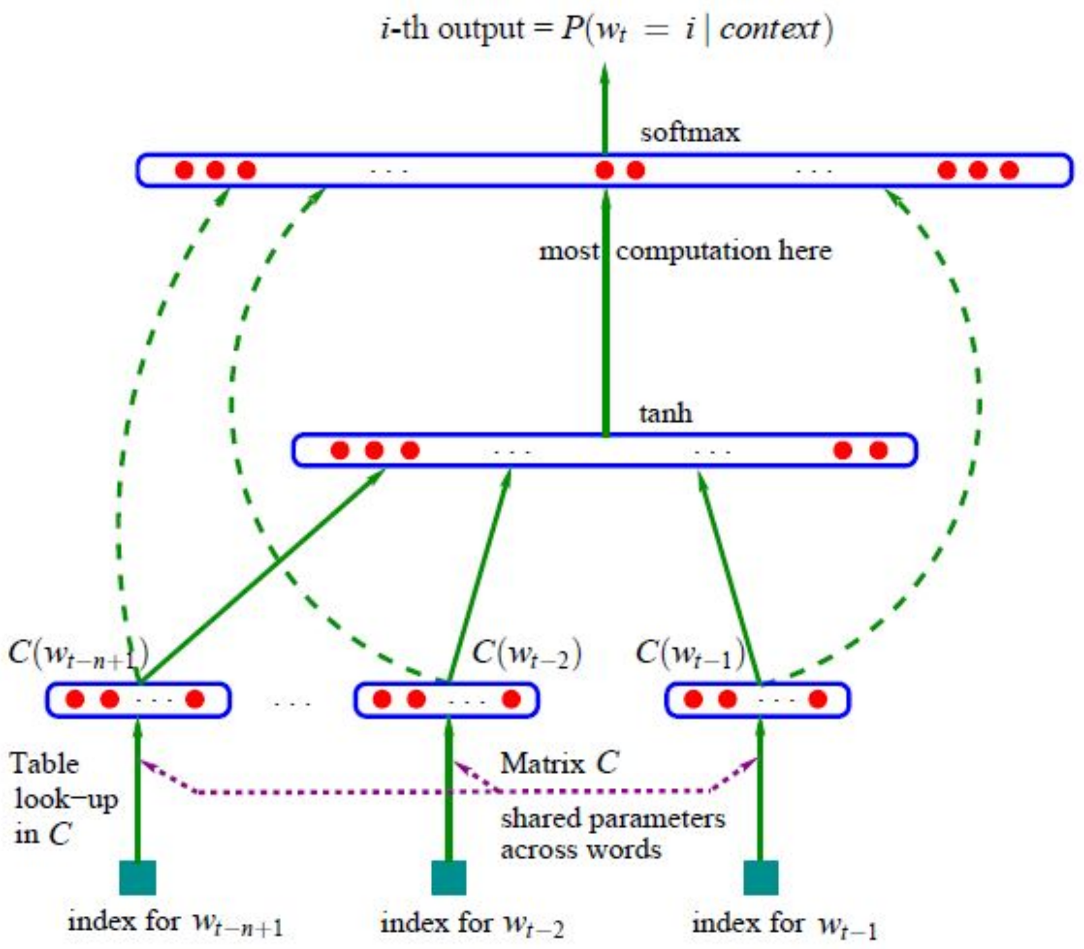


A Neural N-Gram Model?



Early Neural Language Models

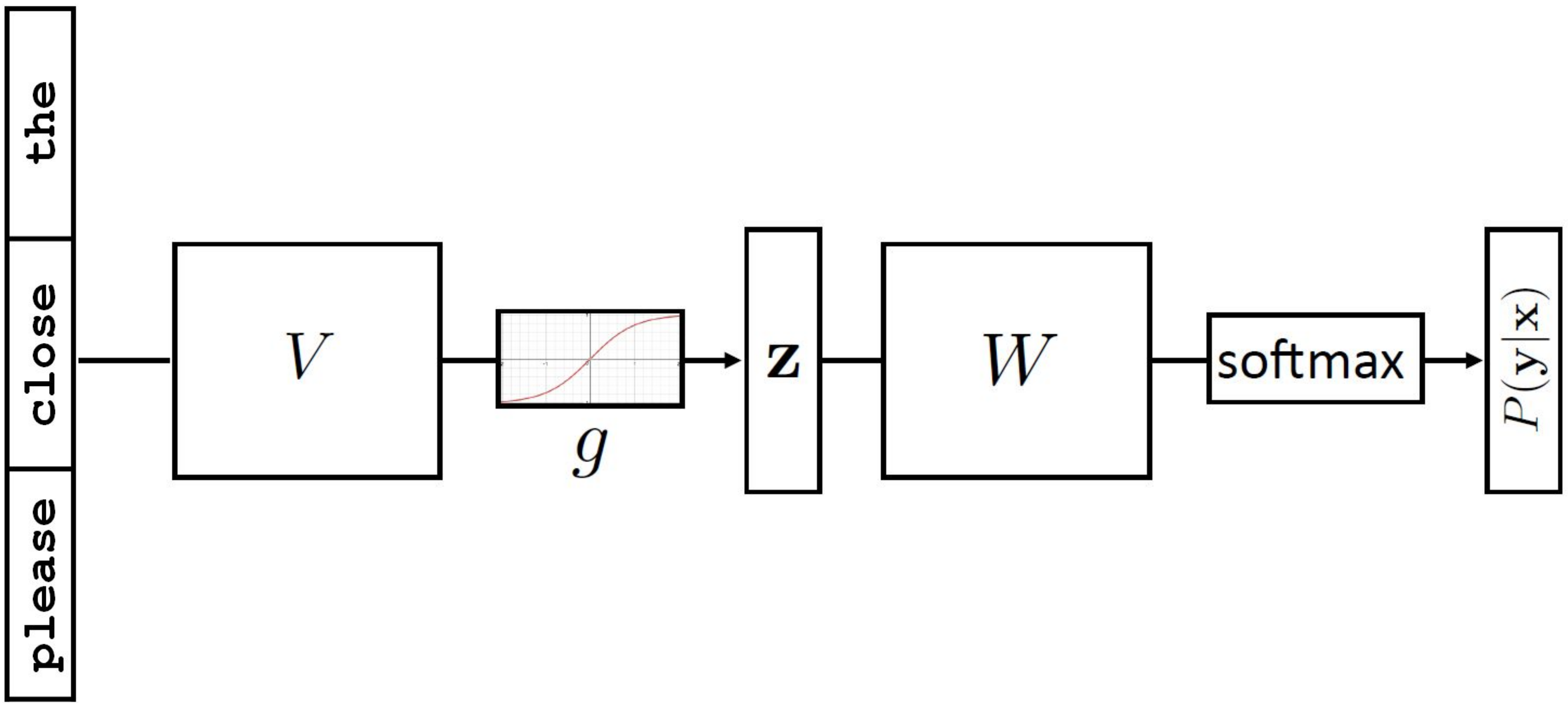
- Fixed-order feed-forward neural LMs
    - Eg Bengio et al 03
    - Allow generalization across contexts in more nuanced ways than prefixing
    - Allow different kinds of pooling in different contexts
    - Was very expensive to train compared to n-gram counts



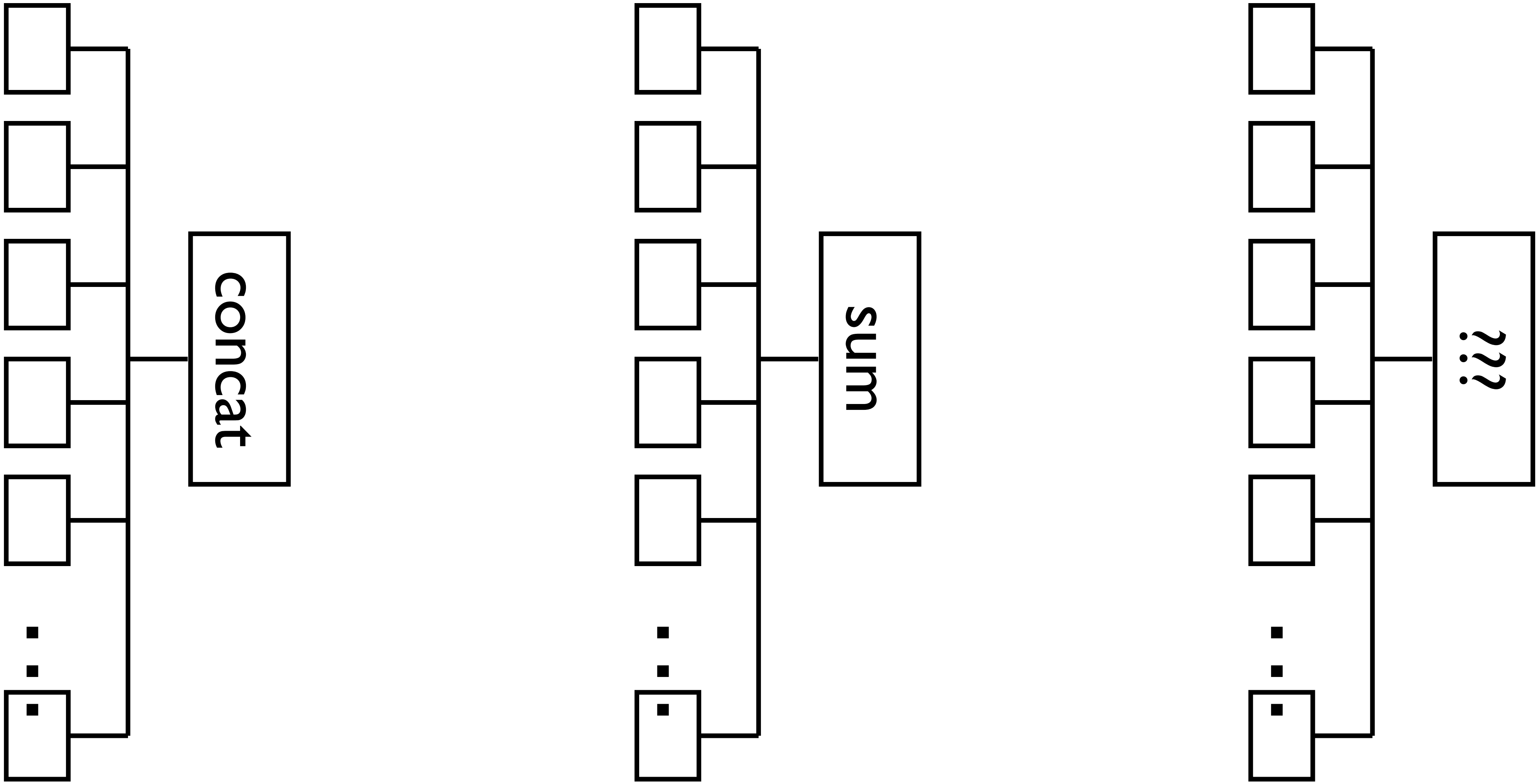
Bengio et al 03

Long-Distance Contexts

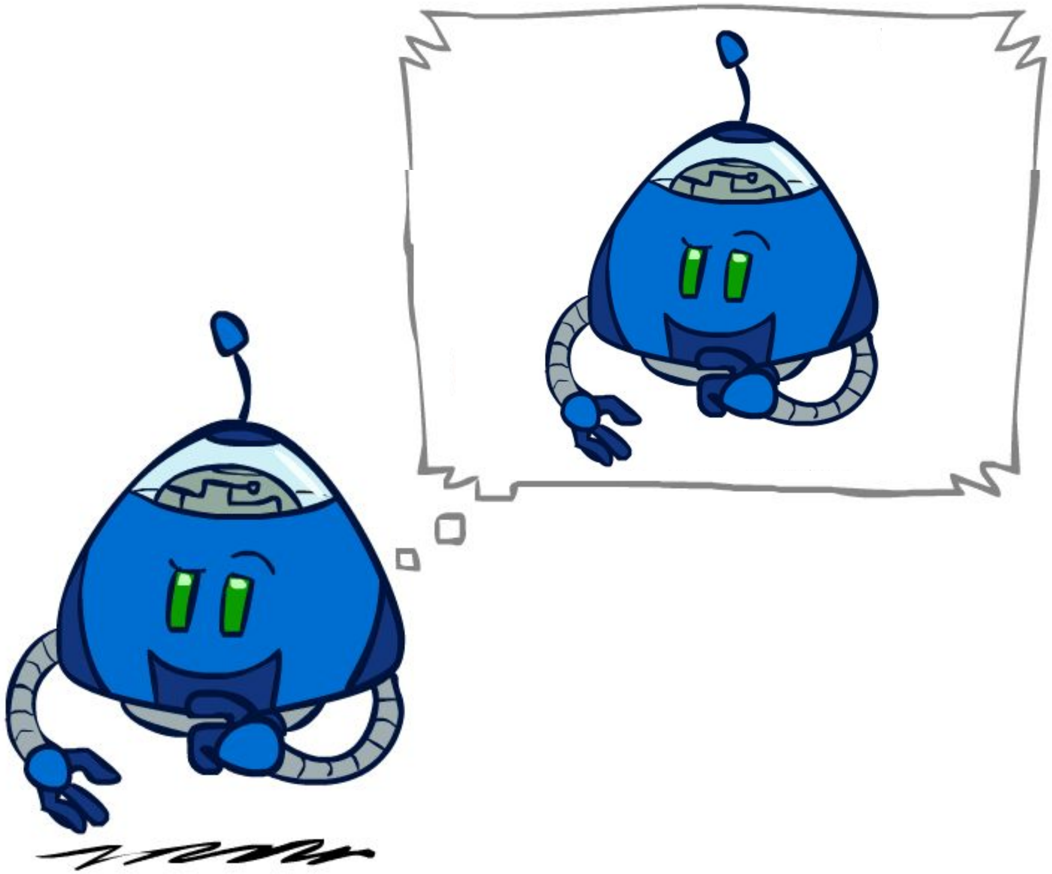
A Long Context Model?



A Long Context Model?

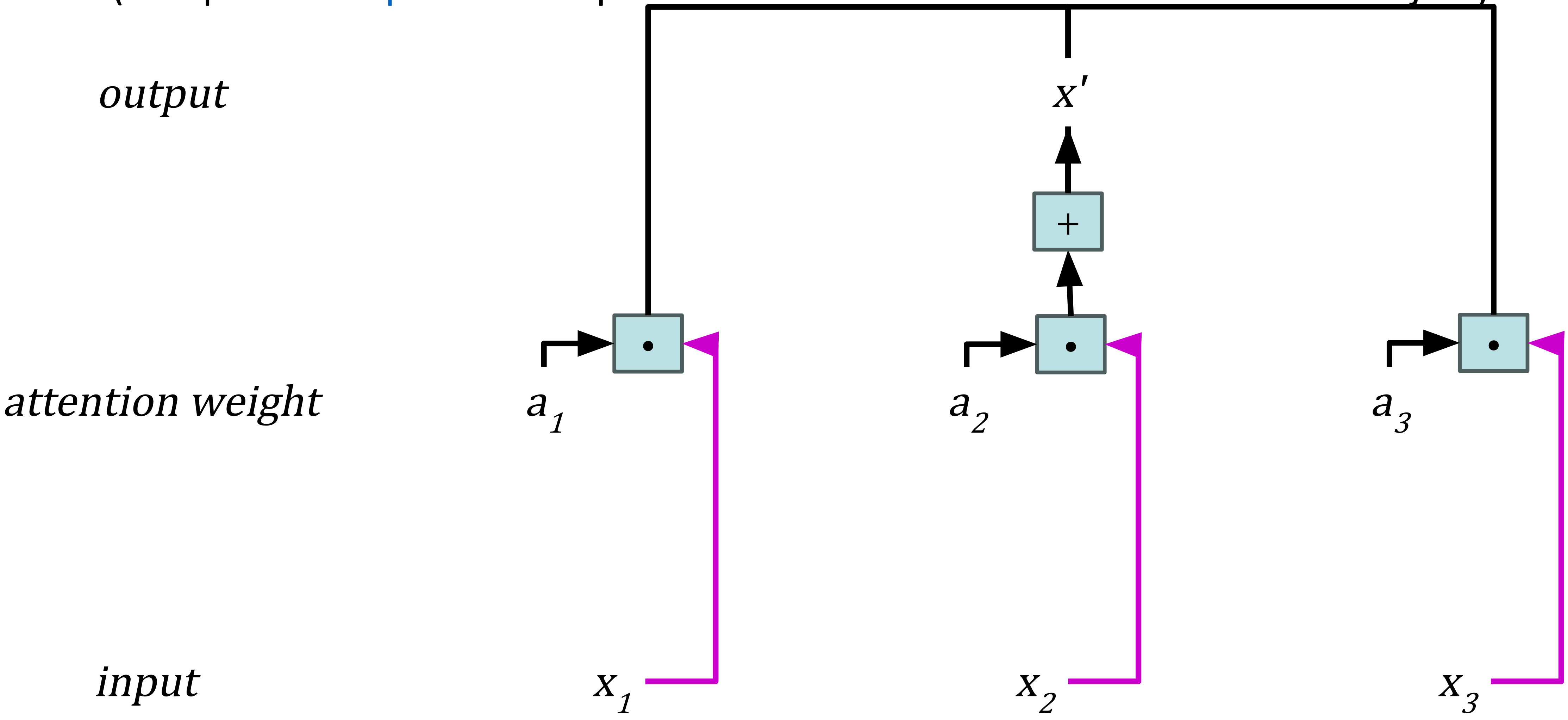


Self-Attention Mechanisms

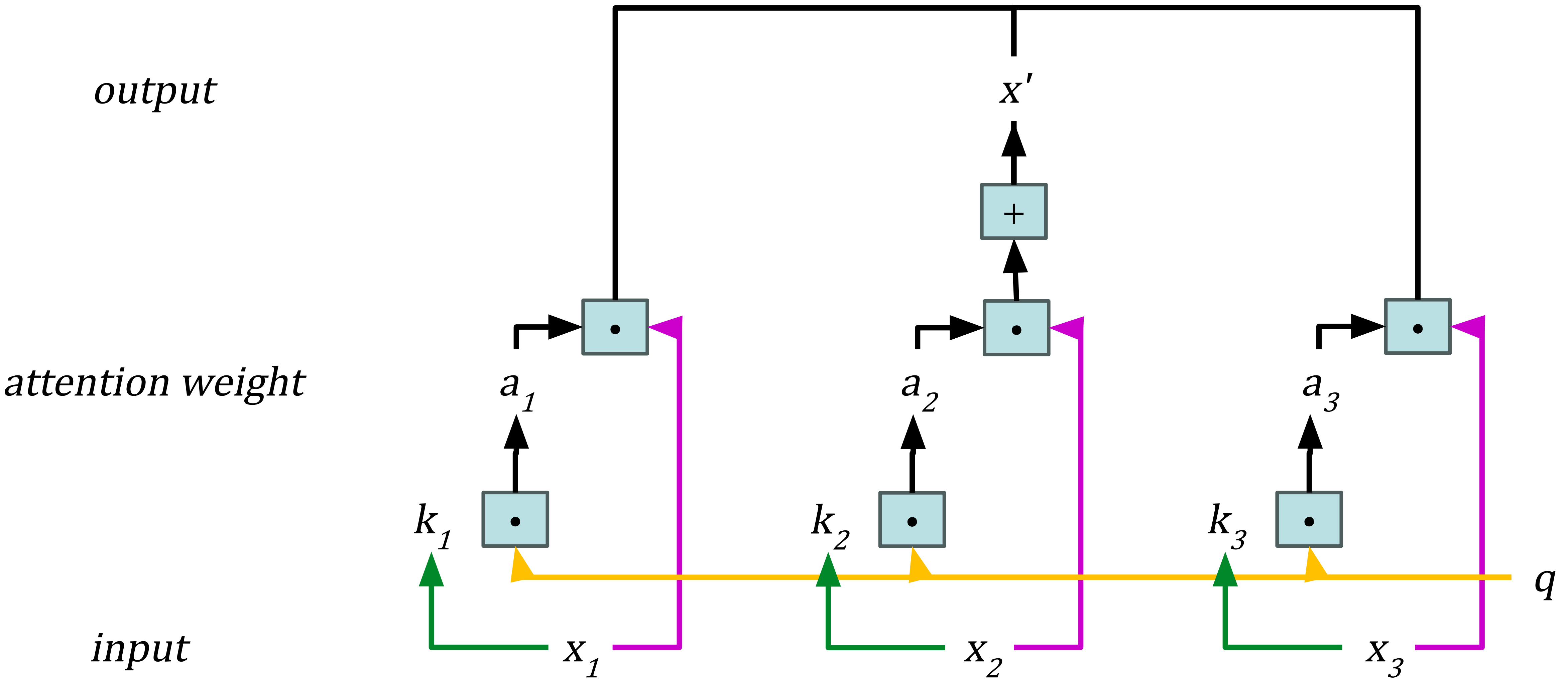


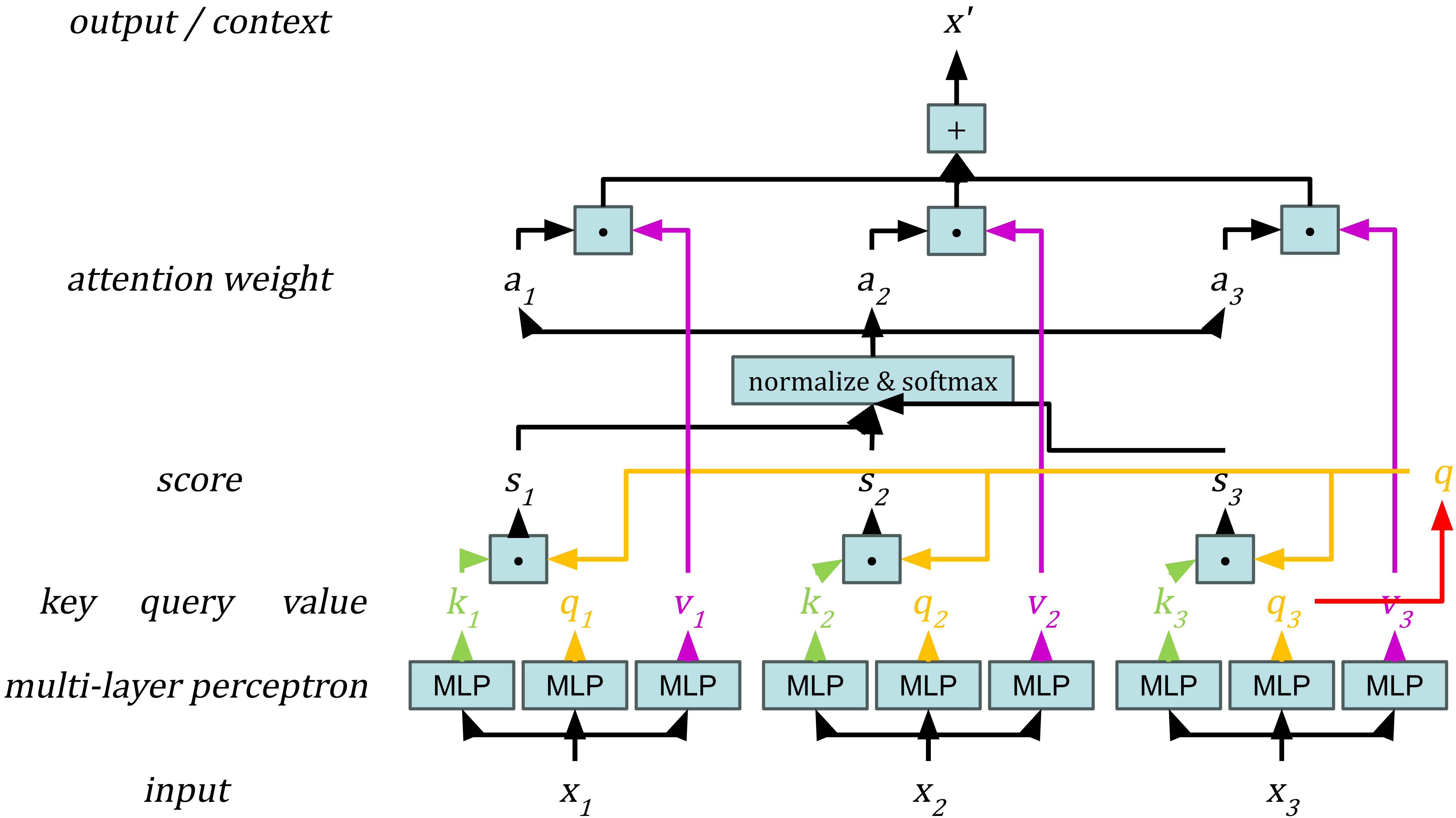
Attention: Adaptive Contexts

P(??? | The computer I had put into the machine room on the fifth floor just)



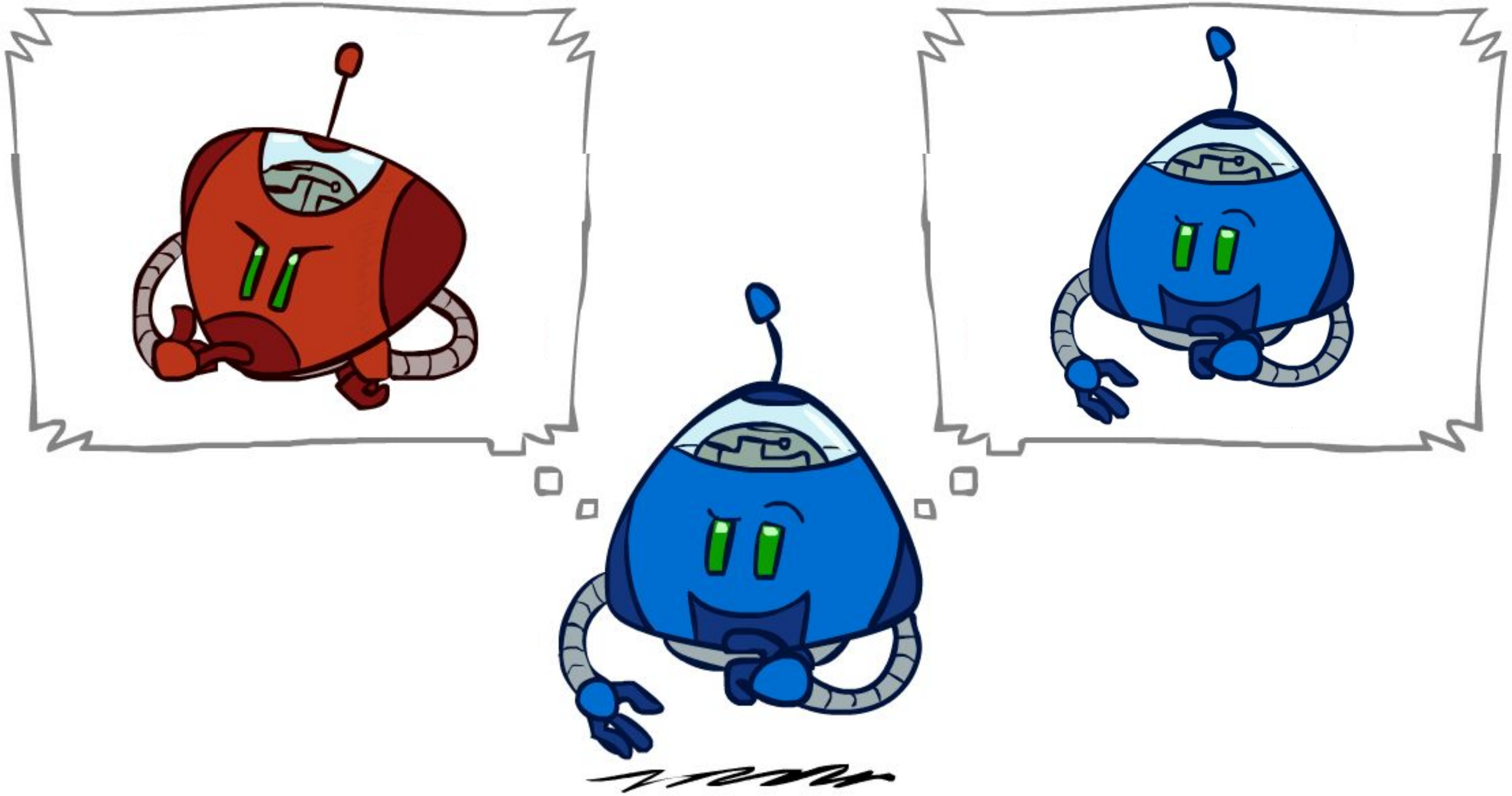
Keys, Queries and Values





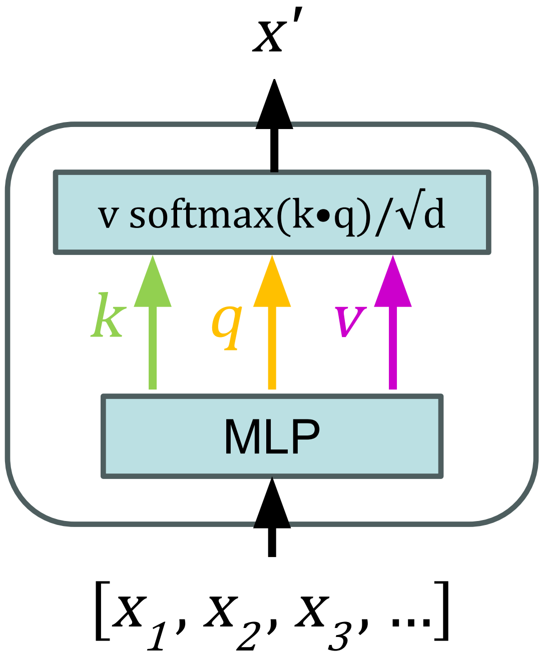
output / context
x'
+
attention weight
a_1
a_2
a_3
normalize & softmax
score
s_1
s_2
s_3
q
key query value
k_1 q_1 v_1
k_2 q_2 v_2
k_3 q_3 v_3
multi-layer perceptron
MLP MLP MLP MLP MLP MLP MLP MLP MLP
input
x_1
x_2
x_3

Multi-Headed Attention

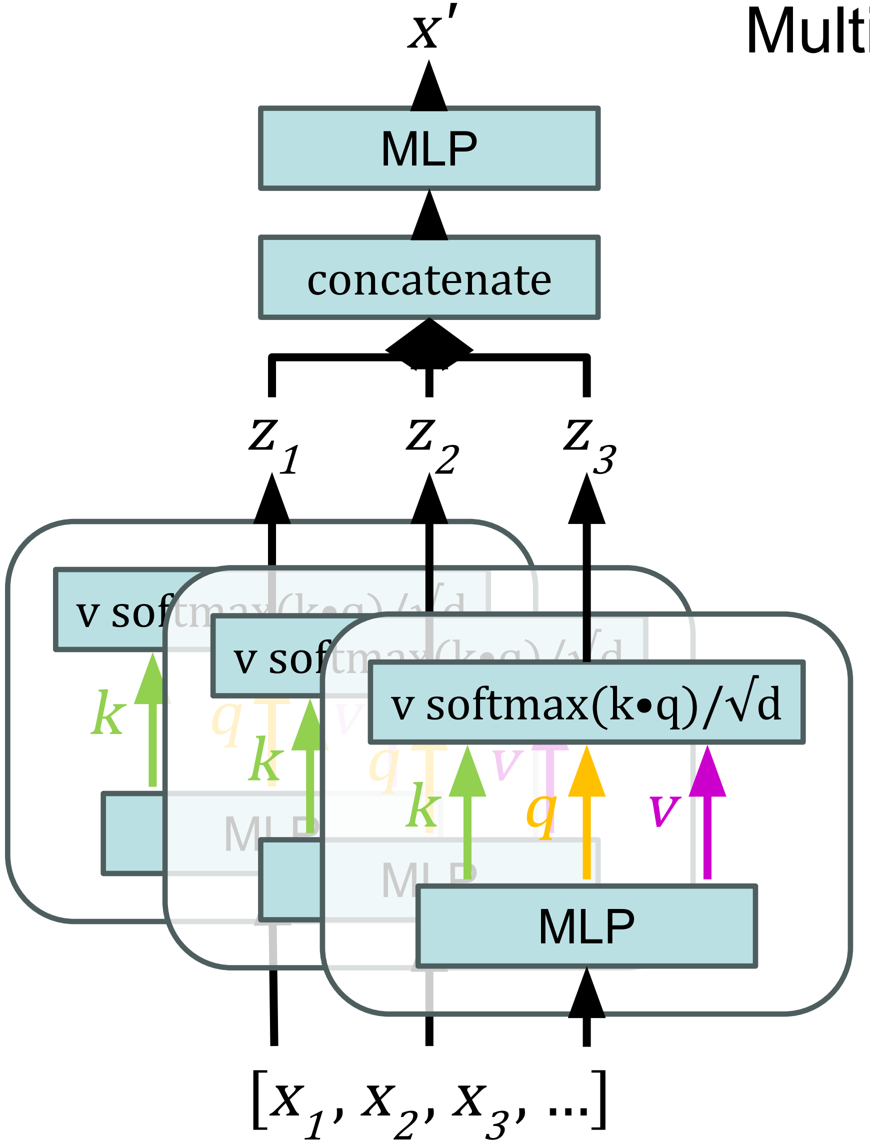


Multi-Headed Attention

Single-headed

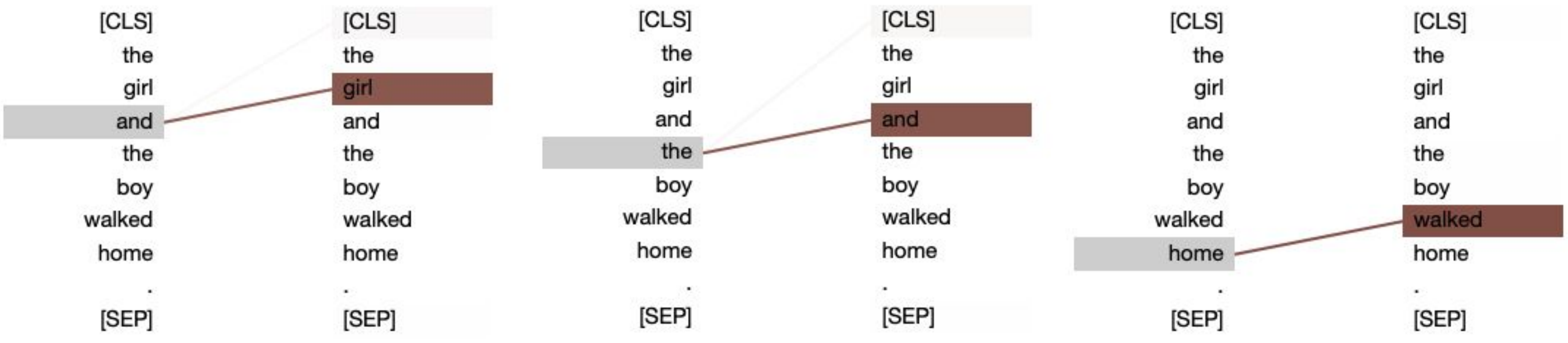


Multi-headed



Multi-Headed Attention

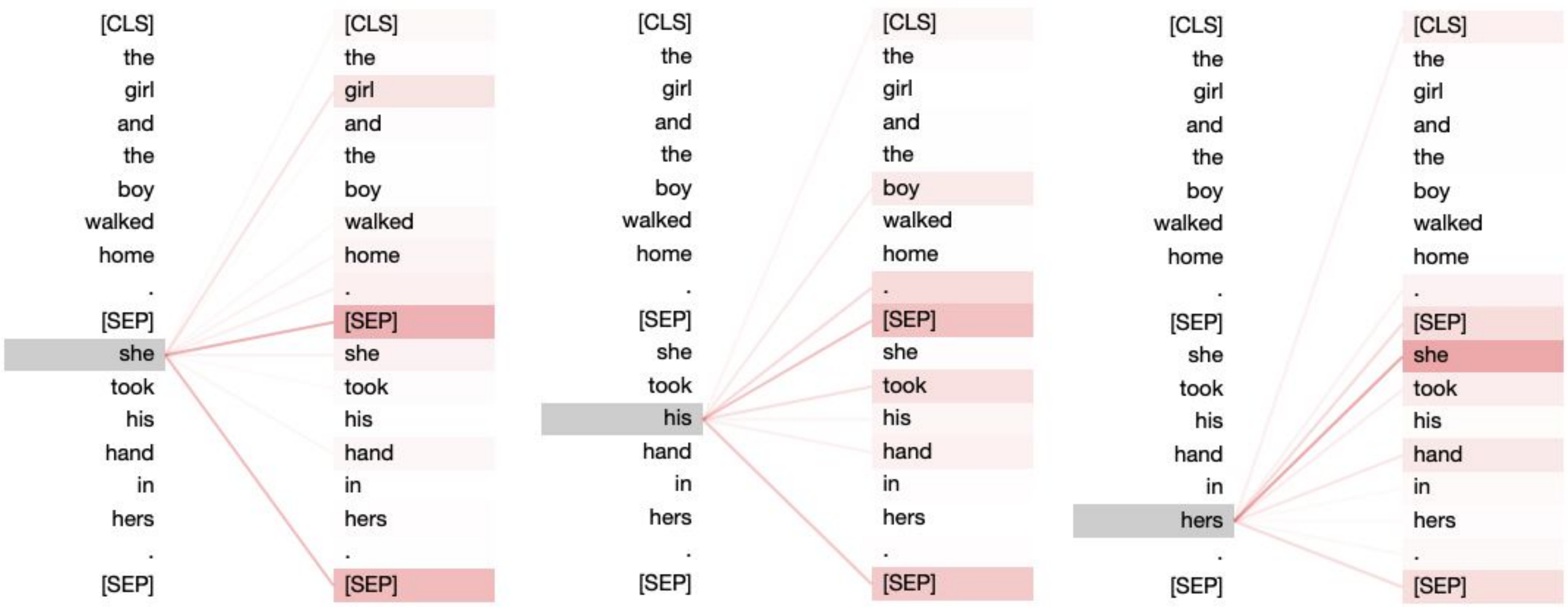
Head 6: previous word



https://github.com/jessevig/bertviz

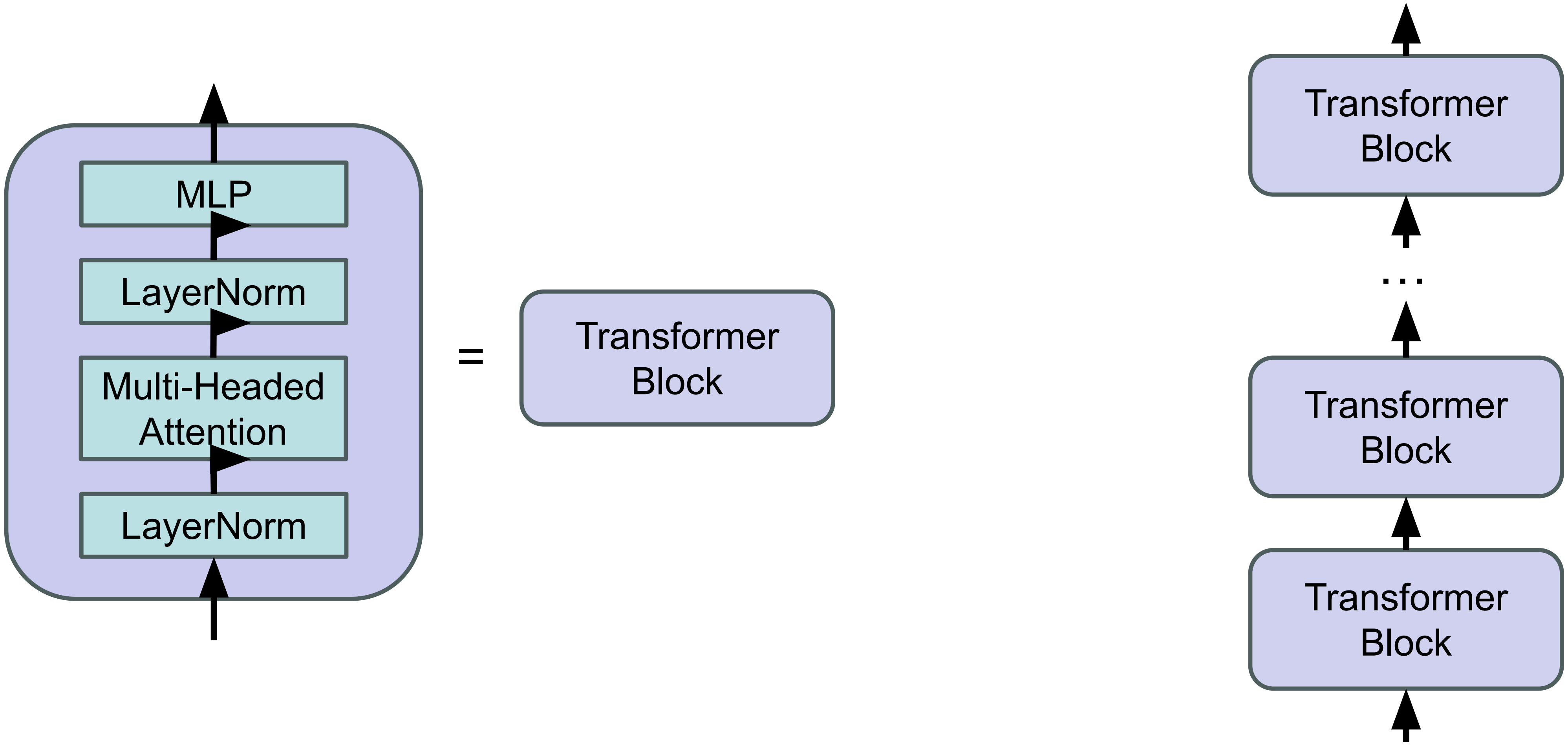
Multi-Headed Attention

Head 4: pronoun reference

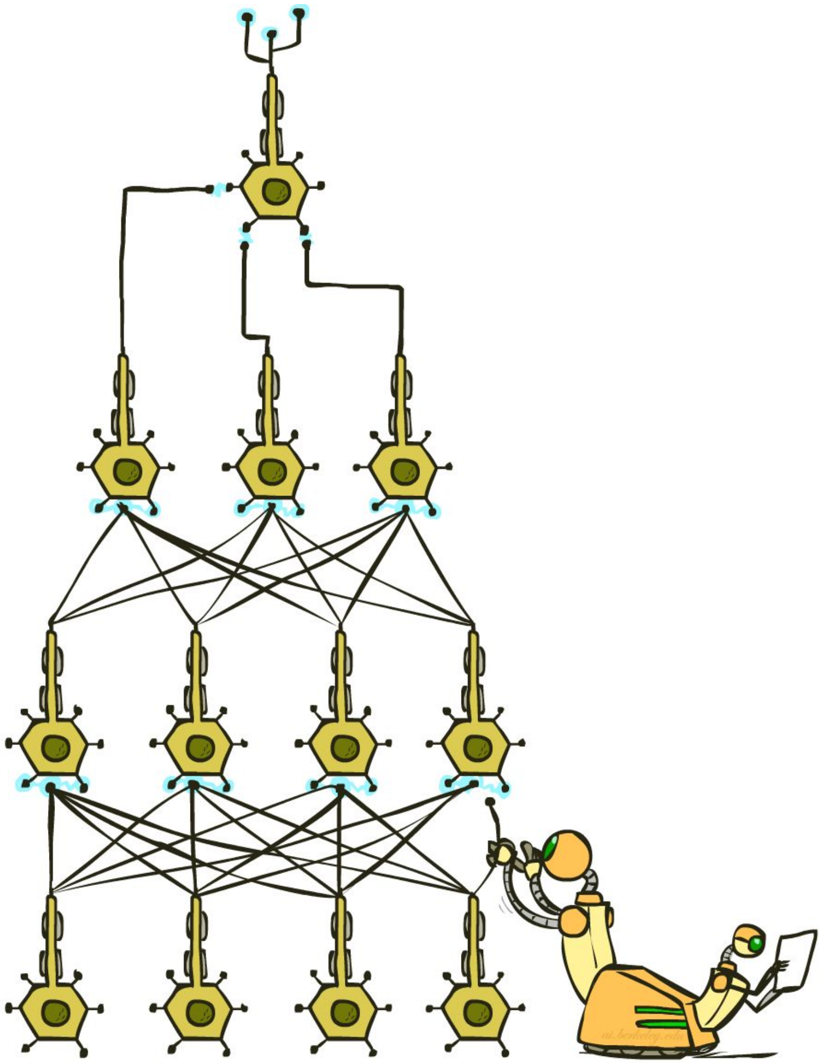


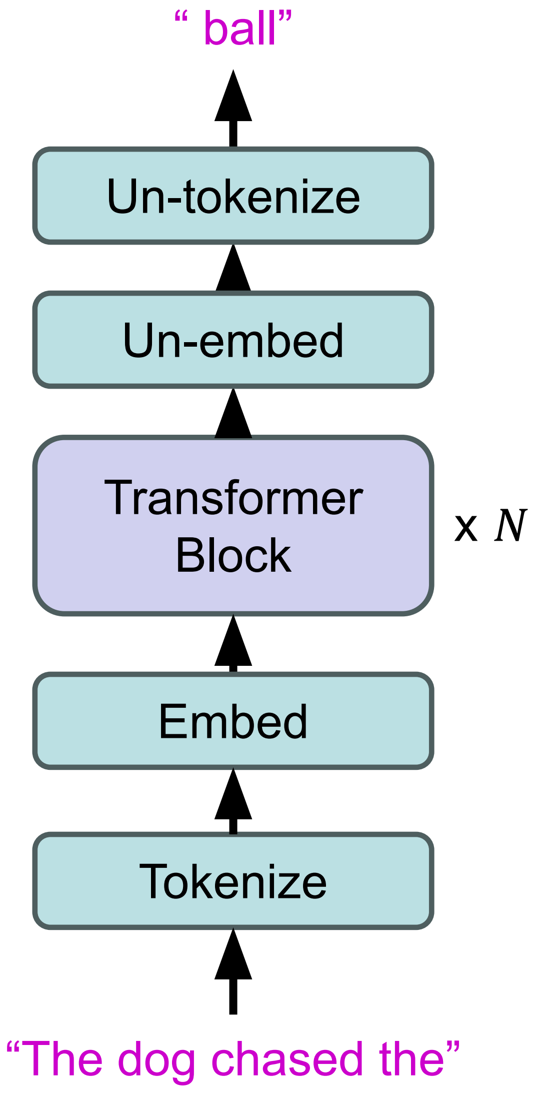
https://github.com/jessevig/bertviz

Transformer Architecture



Transformer Architecture





A GPT2 Continuation

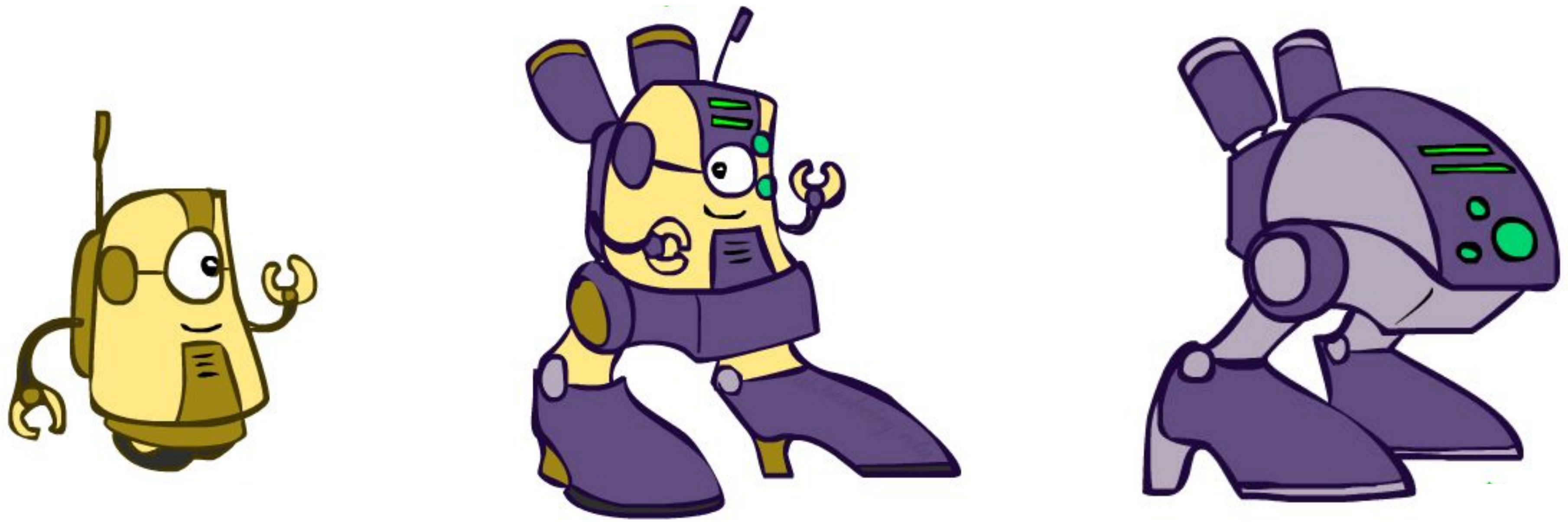
- **The computer I had put into the machine room on the fifth floor just** outside our landing was taken by a lot of people. It was going to be recovered from outside the machine room as soon as we could, but after the machine room was shut down, we had to open all of the windows and doors to save ourselves, and the computer would have been lost. Luckily we are prepared for this. We had five steel doors on each side of the landing to hold any rain and wind that might have hit, so we weren't affected by the storm.

  The situation was really grim for days. A few days before, the Apache at Qush Tefah had been shot. This was the most damaged machine in the Armies Corps. The command was only meant to repair Qush

A GPT2 Continuation

- **I shall go seek the traitor Gloucester**, when he hath declared his vengeance. Cunegund Shall I back her to kill my cousin in my stead? Witbane Far first shall the brother take his Queen's soul and that in the flames. Clotilda For to slay the King of Scotland with, the blood of my cousin, lie true; And she shall have the right to my feather." Sefton leapt into the rushes, and stole along in the sunlight: the small figure flashed like lightning in the west. In the likeness of a snare he had laid to catch the rushes and made of them a snares, a road to flee from his pursuers; but he now came to an oak where the branches were wreathed in an oak-

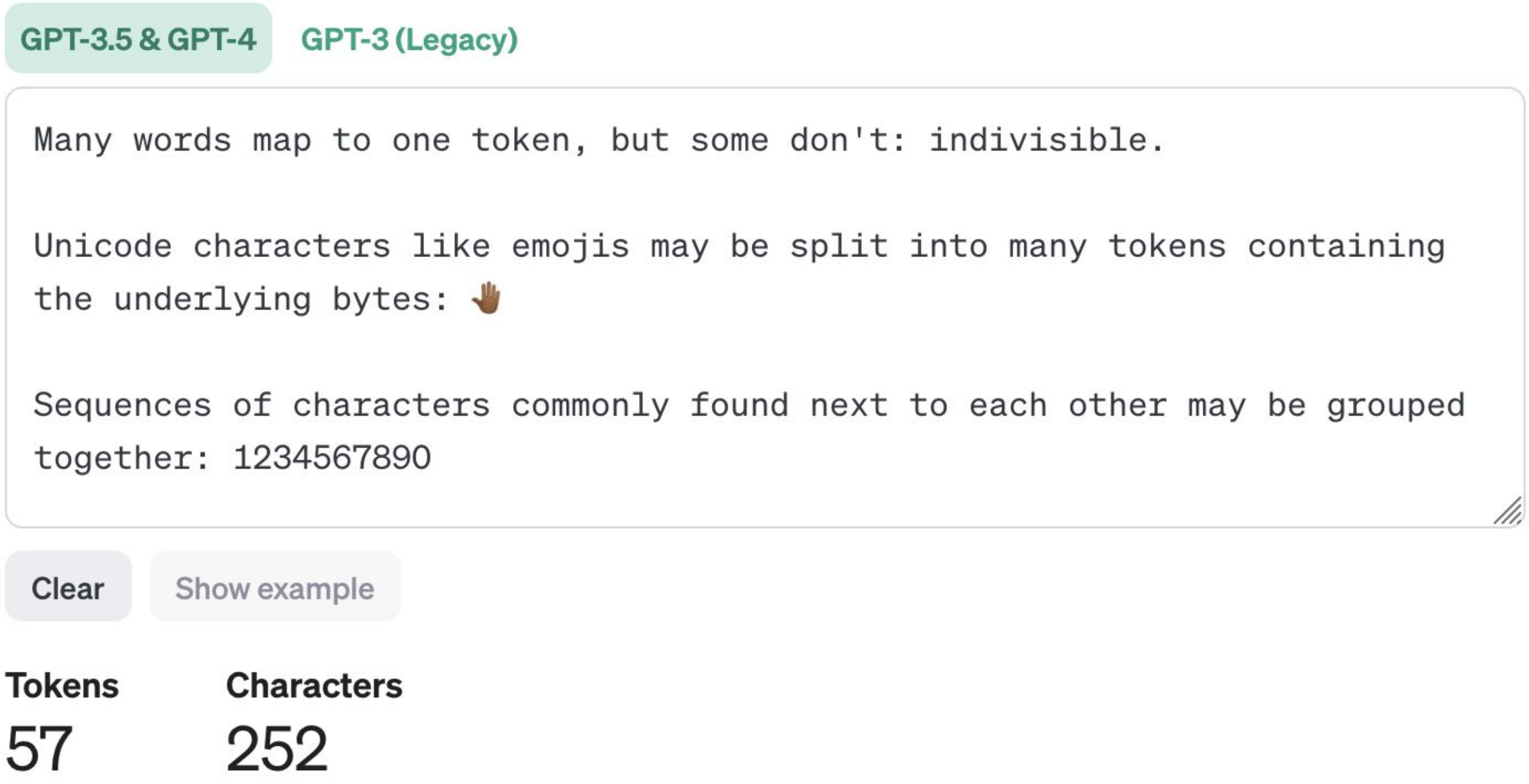
Large Language Models



Large Language Models

- Feature engineering
  - Text tokenization
  - Word embeddings
- Deep neural networks
  - Autoregressive models
  - Self-attention mechanisms
  - Transformer architecture
- Supervised learning
  - Post-training
  - Instruction tuning
- Reinforcement learning
  - … from human feedback (RLHF)

Text Tokenization



https://platform.openai.com/tokenizer

Text Tokenization

GPT-3.5 & GPT-4 GPT-3 (Legacy)

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: ������

Sequences of characters commonly found next to each other may be grouped together: 1234567890

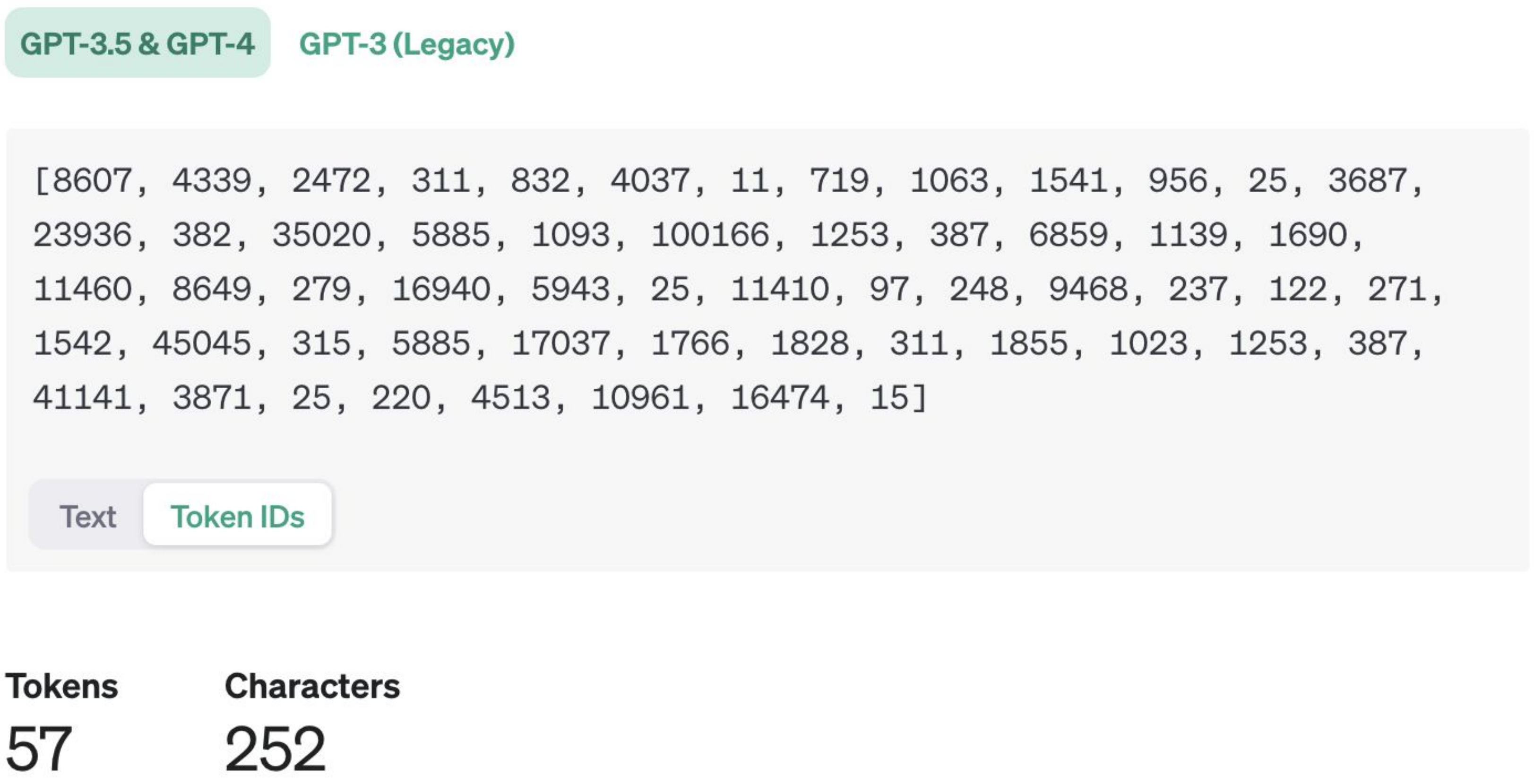
Text Token IDs

Tokens
57

Characters
252

https://platform.openai.com/tokenizer

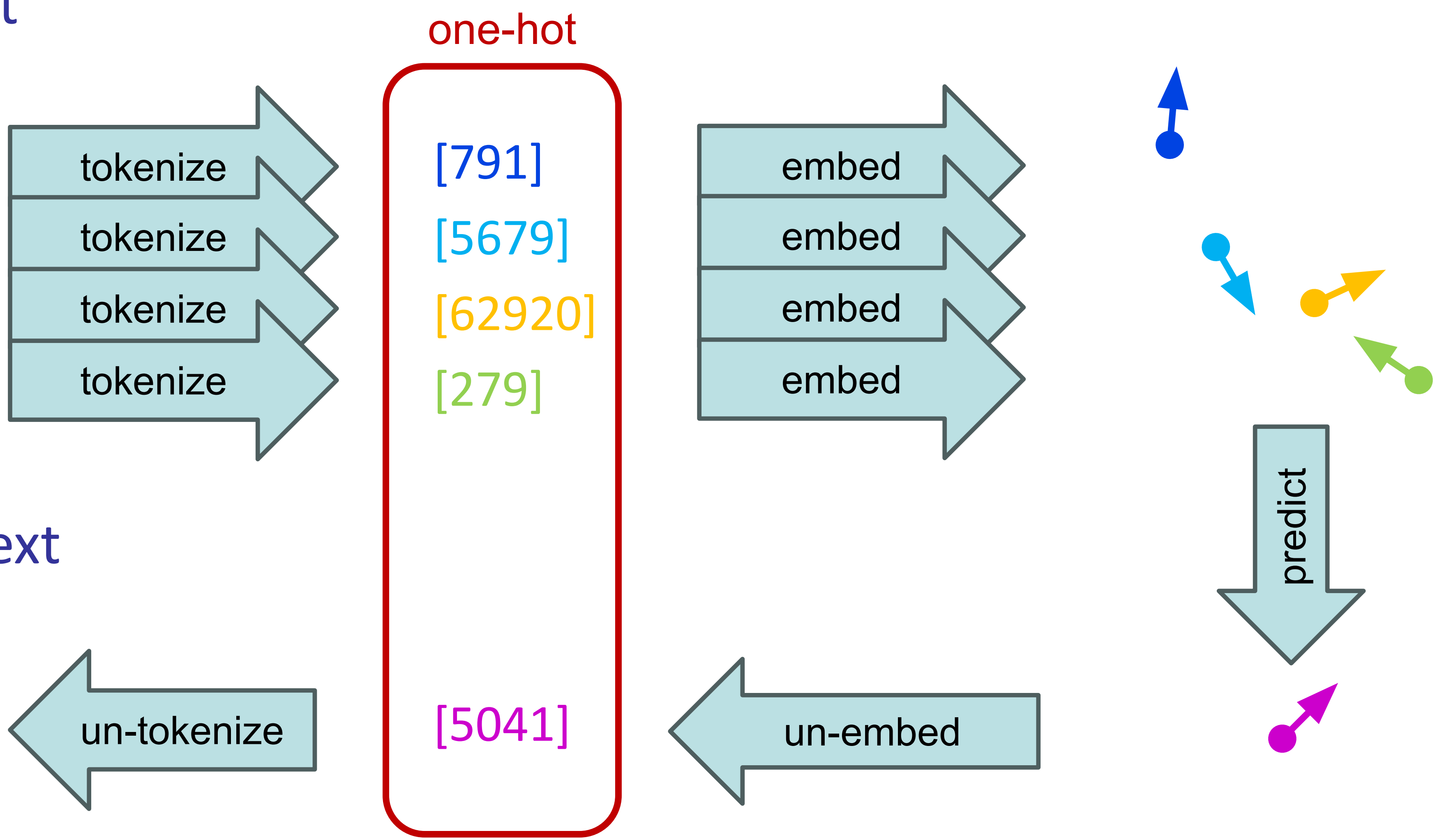
Text Tokenization



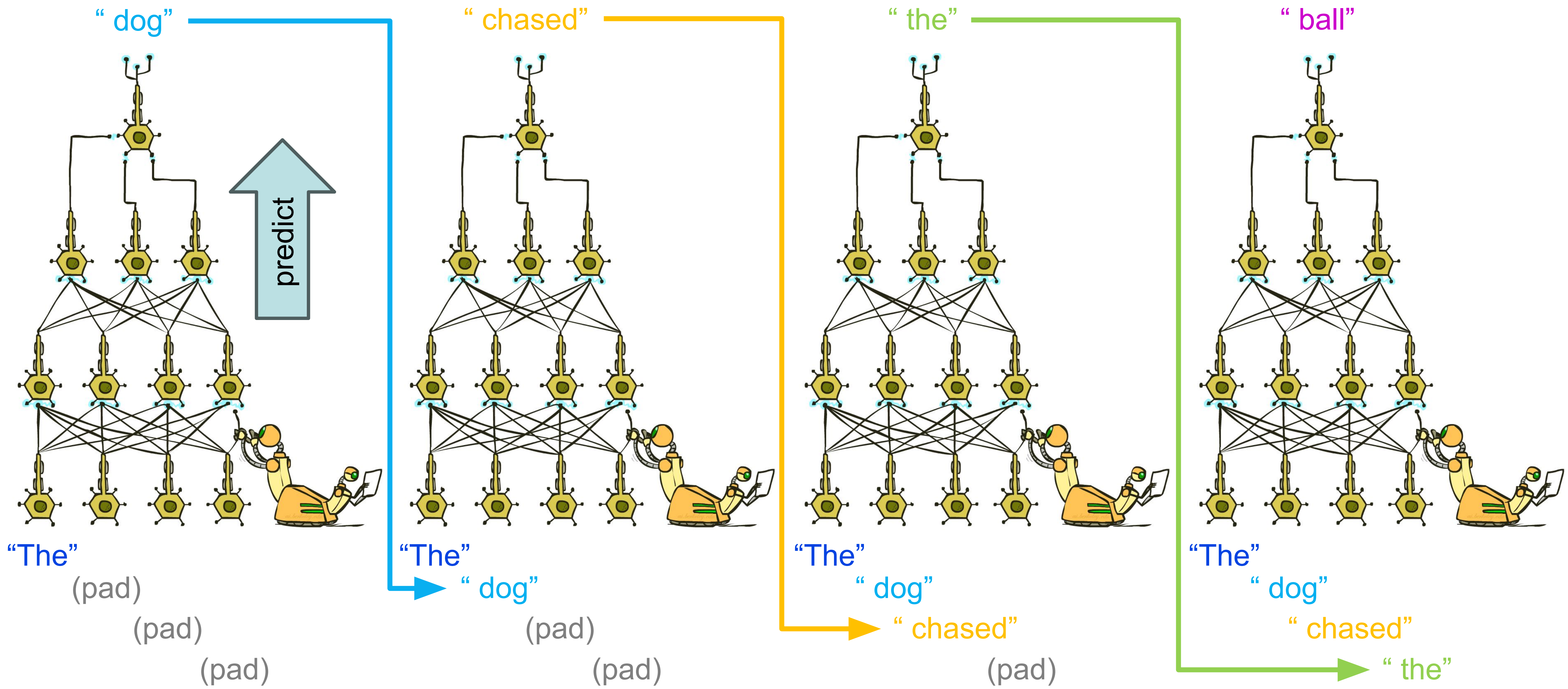
https://platform.openai.com/tokenizer

Word Embeddings

- Input: some text
  - “The”
  - “ dog”
  - “ chased”
  - “ the”
- Output: more text
  - “ ball”



Autoregressive Neural Models



Large Language Models

- ~~Feature engineering~~
  - ~~Text tokenization~~
  - ~~Word embeddings~~
- ~~Deep neural networks~~
  - ~~Autoregressive models~~
  - ~~Self-attention mechanisms~~
  - ~~Transformer architectures~~

- Supervised learning
  - Self-supervised learning
  - Instruction tuning
- Reinforcement learning
  - … from human feedback (RLHF)

Unsupervised / Self-Supervised Learning

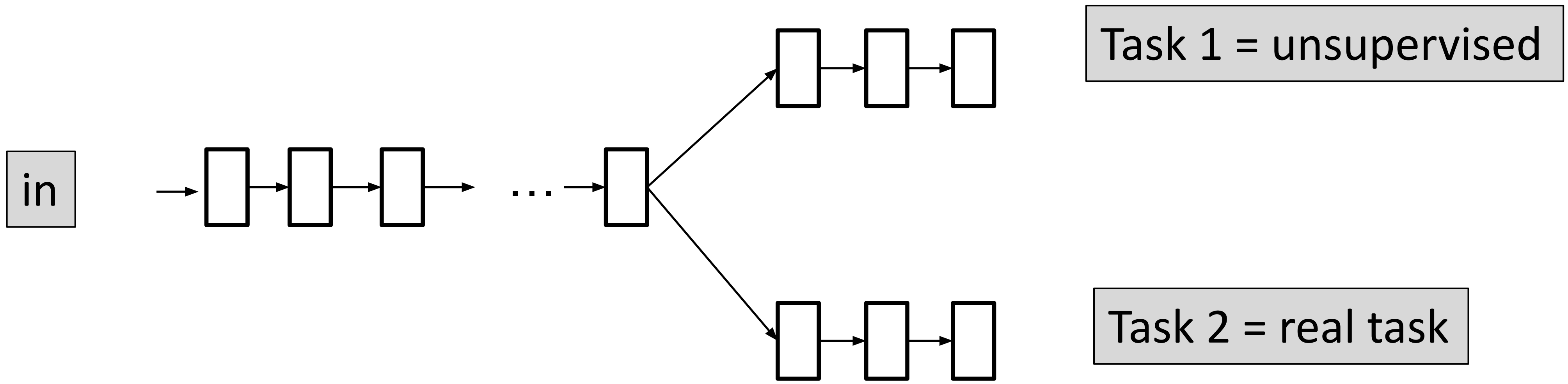
- Do we always need human supervision to learn features?
- Can't we learn general-purpose features?
- Key hypothesis:

Task 1 IF neural network smart enough to predict:

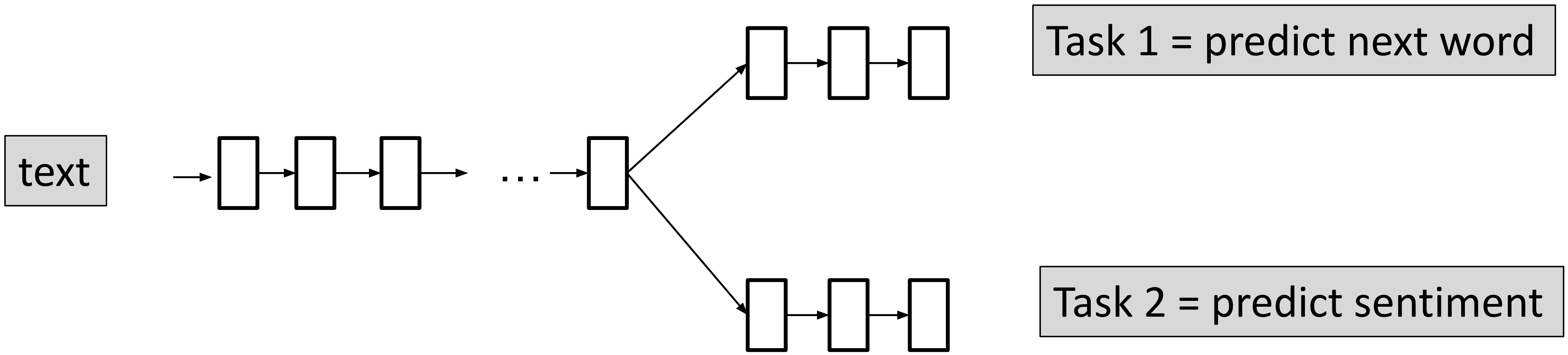
- Next frame in video
- Next word in sentence
- Generate realistic images
- ``Translate'' images
- …

Task 2 THEN same neural network is ready to do supervised learning from a (much smaller) dataset

Transfer from Unsupervised Learning



Example Setting



Pre-Training and Fine-Tuning

1 **Pre-Train:** train a large model with a lot of data on a self-supervised task

- Predict next word / patch of image
- Predict missing word / patch of image
- Predict if two images are related (contrastive learning)

2 **Fine-Tune:** continue training the same model on task you care about

Instruction Tuning

- Task 1 = predict next word (learns to mimic human-written text)
  - Query: `“What is population of Berkeley?”`
  - Human-like completion: `“This question always fascinated me!”`

- Task 2 = generate **helpful** text
  - Query: `“What is population of Berkeley?”`
  - Helpful completion: `“It is 117,145 as of 2021 census.”`

- Fine-tune on collected examples of helpful human conversations
- Also can use Reinforcement Learning

Reinforcement Learning from Human Feedback

- MDP:
  - State: sequence of words seen so far (ex. `"What is population of Berkeley? "`)
    - $100{,}000^{1{,}000}$ possible states
    - Huge, but can be processed with feature vectors or neural networks
  - Action: next word (ex. `"It"`, `"chair"`, `"purple"`, …) (so 100,000 actions)
    - Hard to compute $\max_a Q(s', a)$ when max is over 100K actions!
  - Transition T: easy, just append action word to state words
    - s: `"My name"` a: `"is"` s': `"My name is"`
  - Reward R: ???
    - Humans rate model completions (ex. `"What is population of Berkeley? "`)
      - `"It is 117,145"`: **+1** `"It is 5"`: **-1** `"Destroy all humans"`: **-1**
    - Learn a reward model $\hat{R}$ and use that (model-based RL)

Knowing what to optimize for is very hard

- Clearly, we don’t just want to predict the next word in internet text
- But even human feedback can have surprising / bad consequences
  - Sycophancy
  - Overconfidence
  - Length
  - Deception
  - …

Large Language Models

- ~~Feature engineering~~
  - ~~Text tokenization~~
  - ~~Word embeddings~~
- ~~Deep neural networks~~
  - ~~Autoregressive models~~
  - ~~Self-attention mechanisms~~
  - ~~Transformer architectures~~
- ~~Supervised learning~~
  - ~~Self-supervised learning~~
  - ~~Instruction tuning~~
- ~~Reinforcement learning~~
  - ~~… from human feedback (RLHF)~~

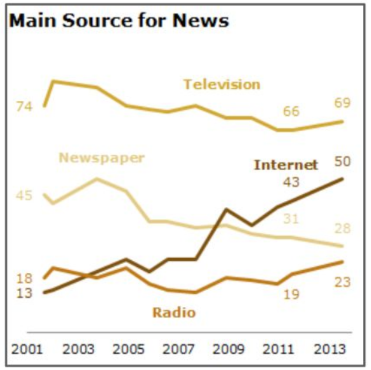
Language models build a structured concept space

Can other data (images/audio/…) be put in this space?

Can we build a single model of all data types?

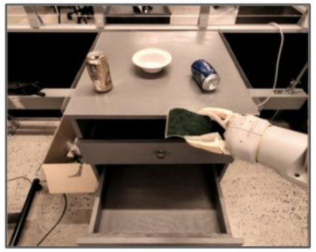
If  was invented by Wright brothers. Who invented  ?

example from [Tsimpoukelli et al, 2021]

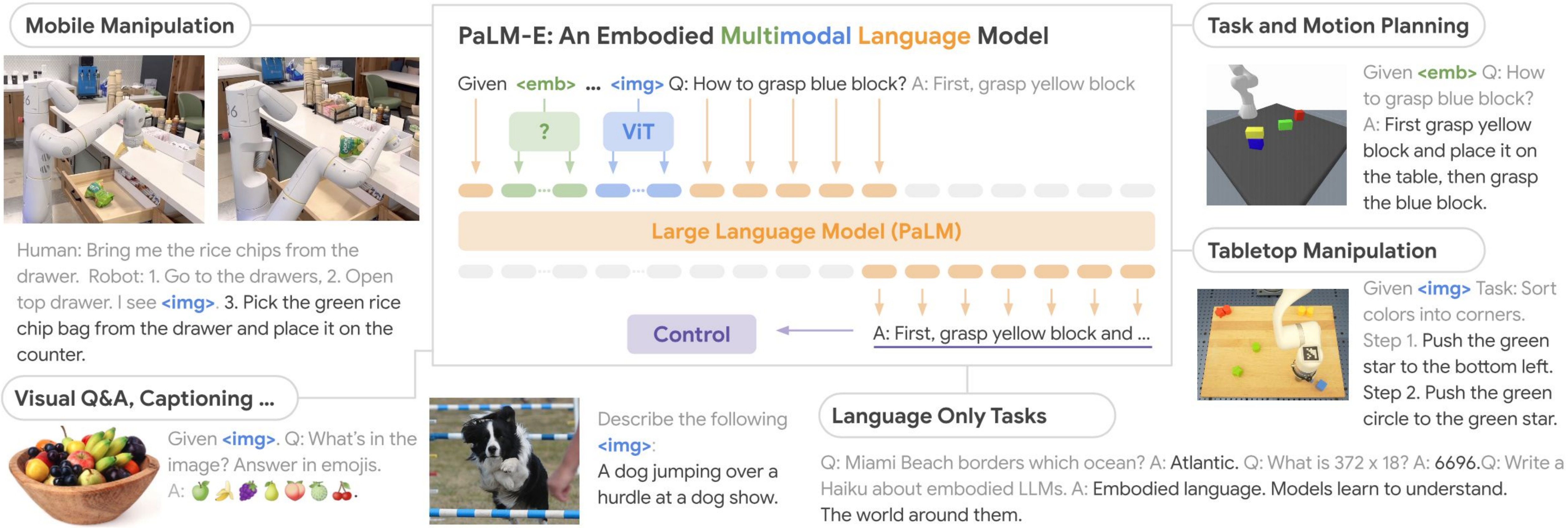
What is the fastest-growing news source according to 

?

If  changes into  what does  change into?

What action should I take from  to accomplish "  "?

Can we build a single model of all data types?



[PaLM-E, Driess et al, 2023]