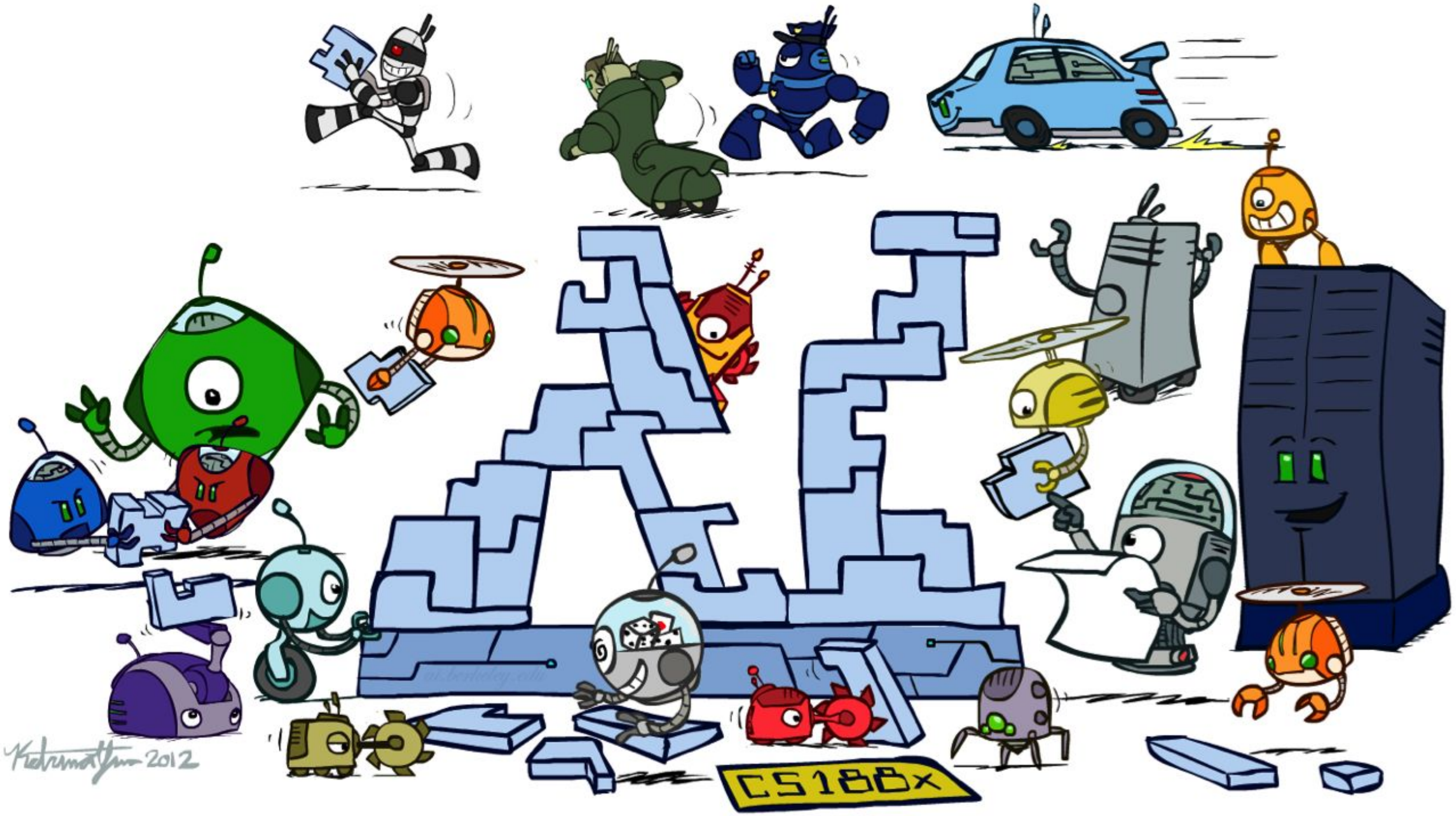
CS 188: Artificial Intelligence
Large Language Models



Instructors: Peyrin Kao and Emma Pierson --- University of California, Berkeley

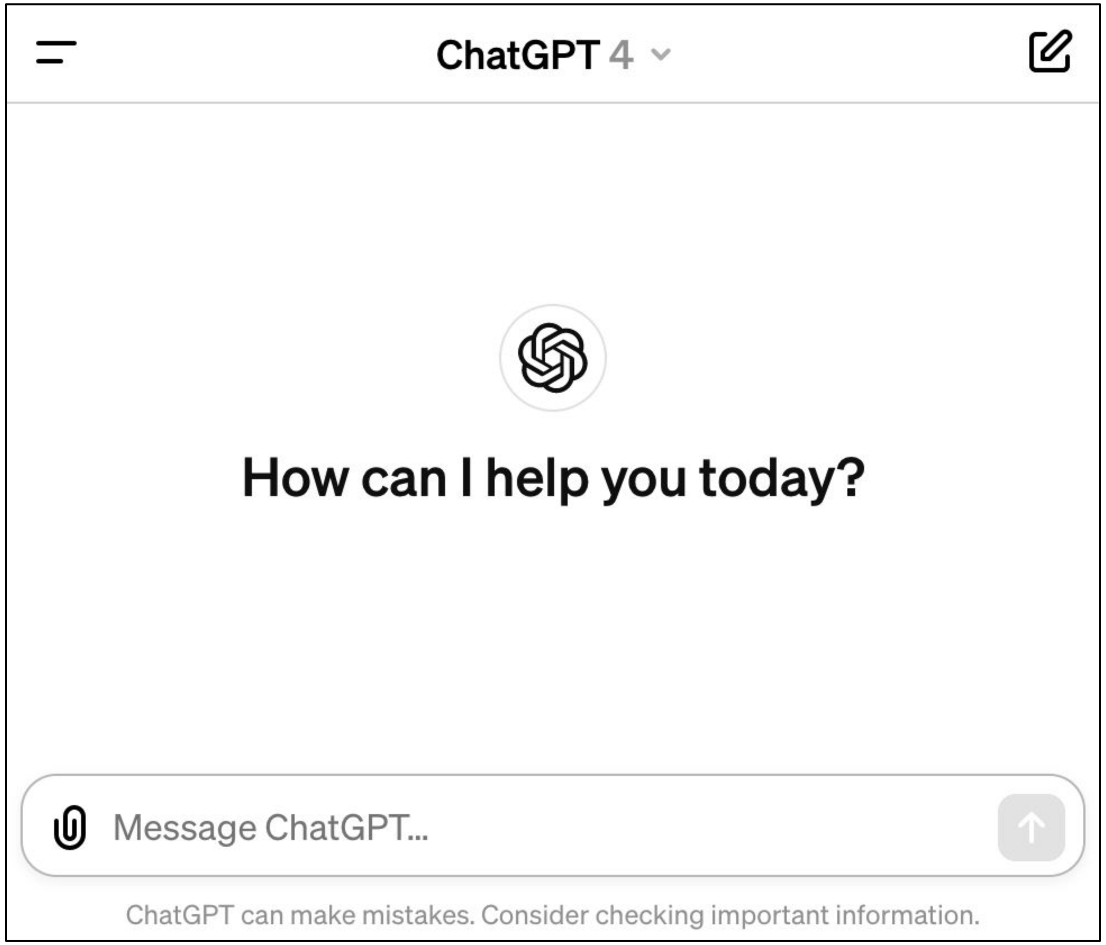
[These slides were created by Cam Allen, Michael Cohen, Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley.

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

Next few classes

- Today: LLMs
- Tuesday (11/25): Applications: AI for Healthcare (and extra credit!)
- Thursday (11/27): No class (Thanksgiving!)
- Tuesday (12/02): AI for Equality and Course Wrap-Up

Today's AI

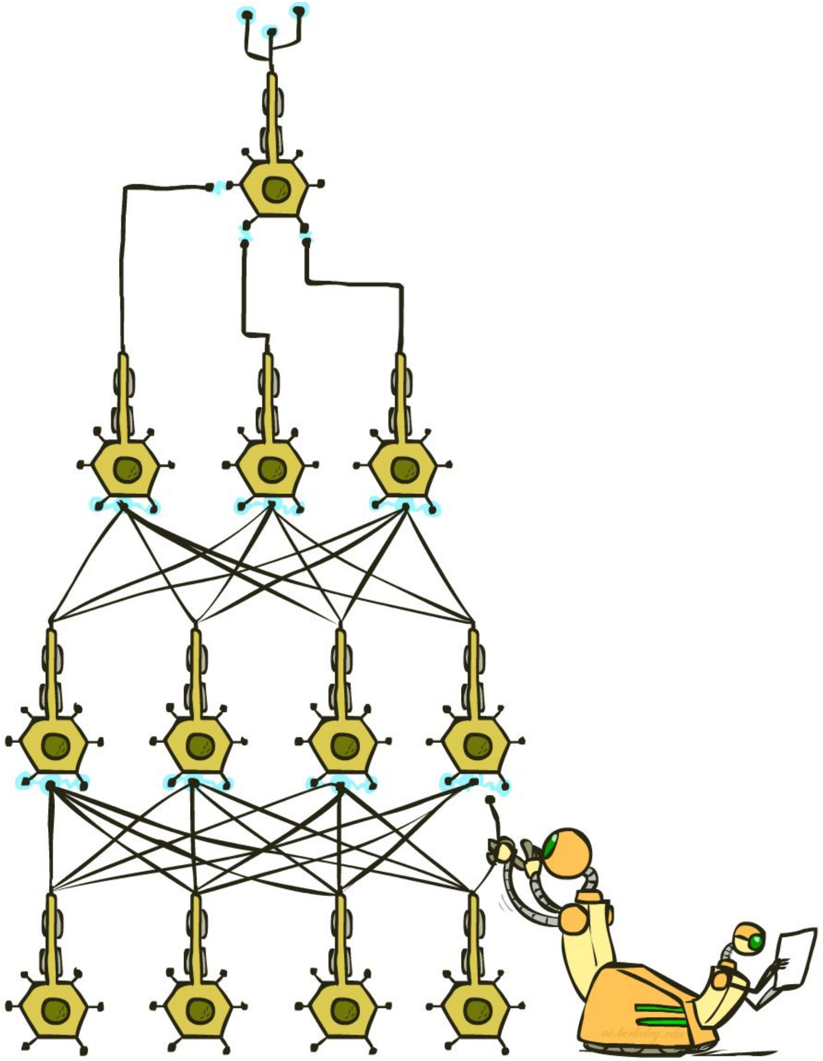

ChatGPT 4
How can I help you today?
Message ChatGPT...
ChatGPT can make mistakes. Consider checking important information.


Untitled
What can I help you with today?
Message Claude...
Claude 3 Opus

Large Language Models

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

Deep Neural Networks



- Input: some text
  - “The dog chased the”
- Output: more text
  - … “ ball”
- Implementation:
  - Linear algebra
  - How??

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
```

Clear   Show example

Tokens	Characters
57	252

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

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

[8607, 4339, 2472, 311, 832, 4037, 11, 719, 1063, 1541, 956, 25, 3687, 23936, 382, 35020, 5885, 1093, 100166, 1253, 387, 6859, 1139, 1690, 11460, 8649, 279, 16940, 5943, 25, 11410, 97, 248, 9468, 237, 122, 271, 1542, 45045, 315, 5885, 17037, 1766, 1828, 311, 1855, 1023, 1253, 387, 41141, 3871, 25, 220, 4513, 10961, 16474, 15]

Text Token IDs

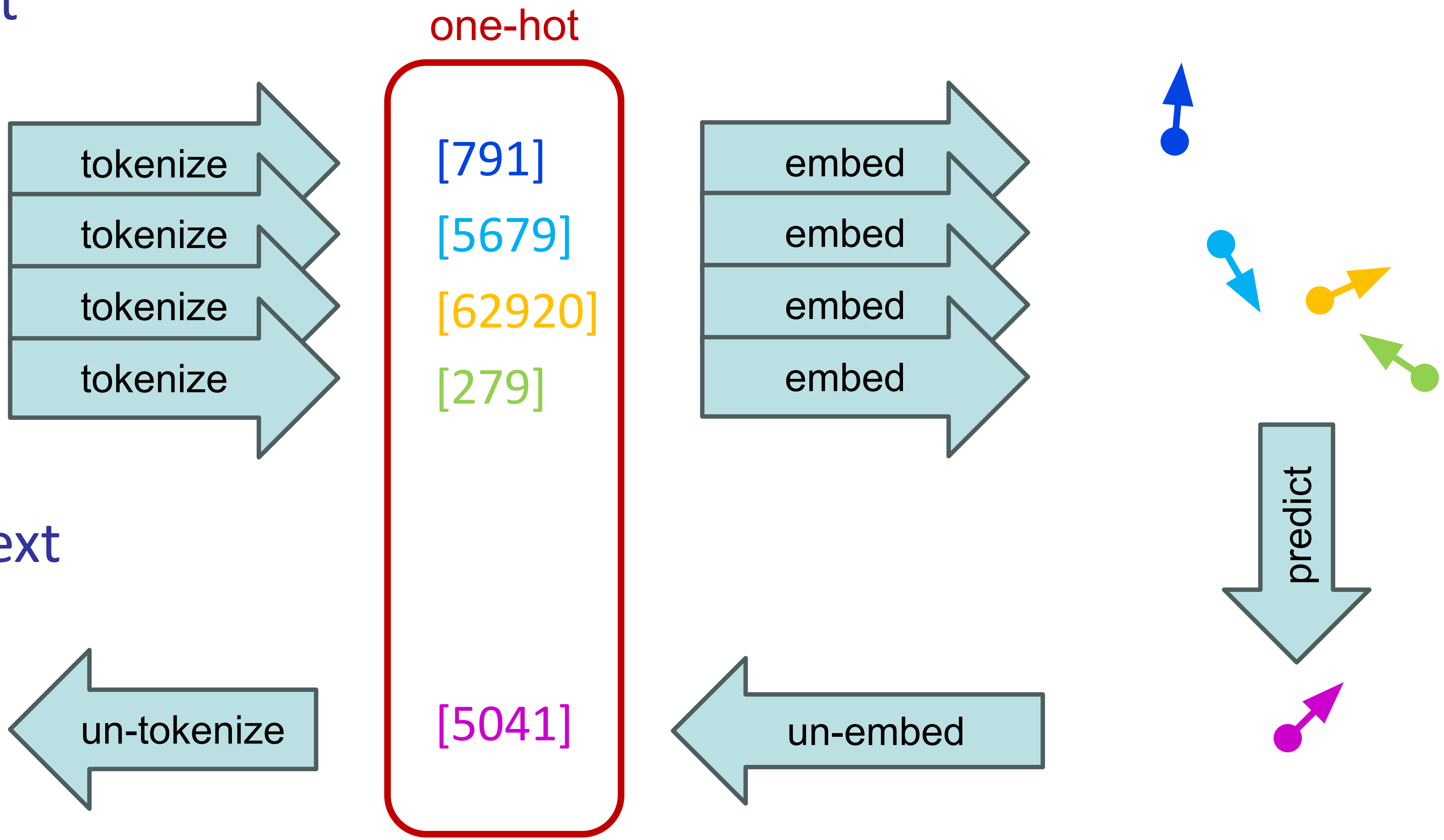
Tokens
57

Characters
252

https://platform.openai.com/tokenizer

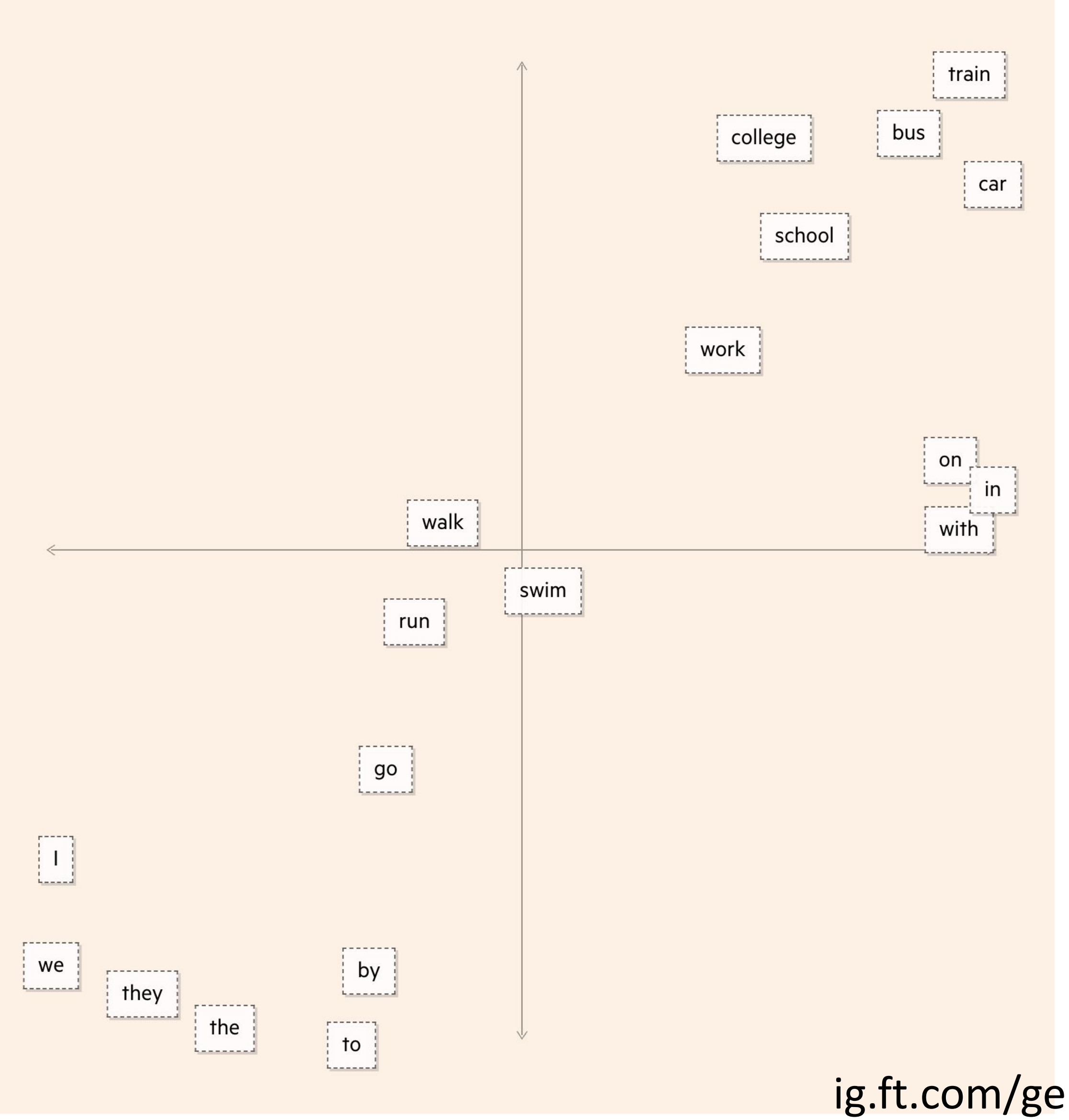
Word Embeddings

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



What do word embeddings look like?

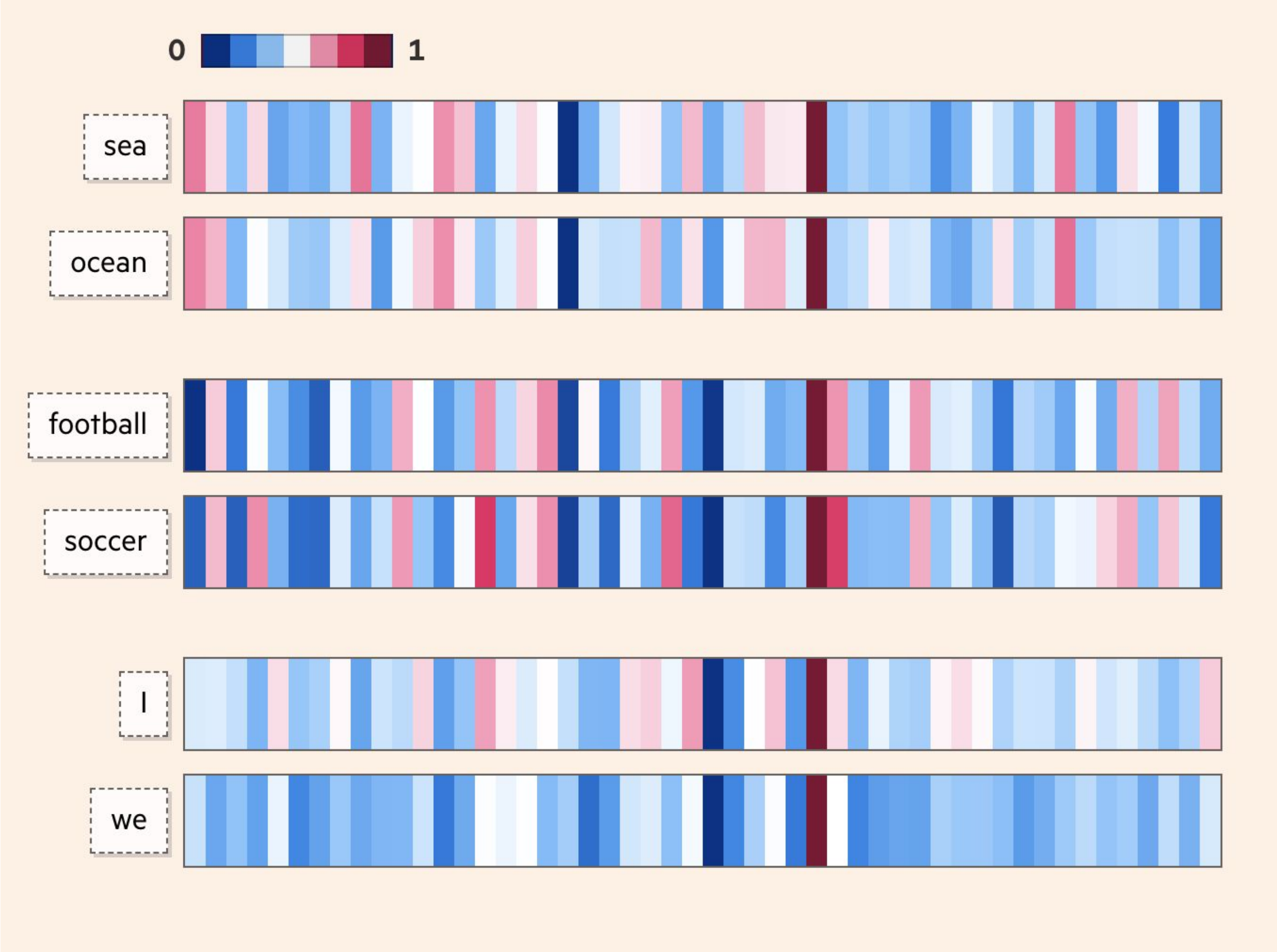
- Words cluster by similarity:



ig.ft.com/generative-ai

What do word embeddings look like?

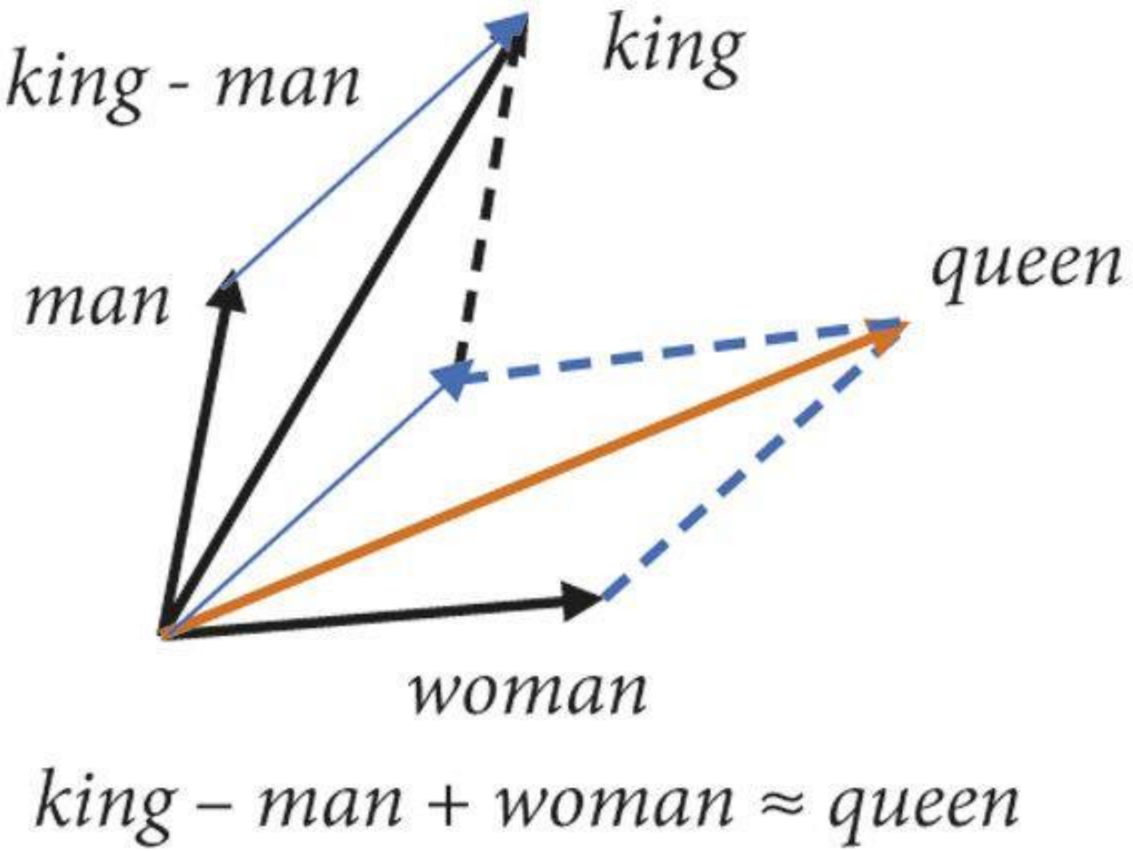
- Features learned in language models:



ig.ft.com/generative-ai

What do word embeddings look like?

- Signs of sensible algebra in embedding space:



[Efficient estimation of word representations in vector space, Mikolov et al, 2013]

What do word embeddings look like?

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, Adam Kalai

Aside: interactive explainer of modern language models

ig.ft.com/generative-ai

Artificial Intelligence

Generative AI exists because of the transformer

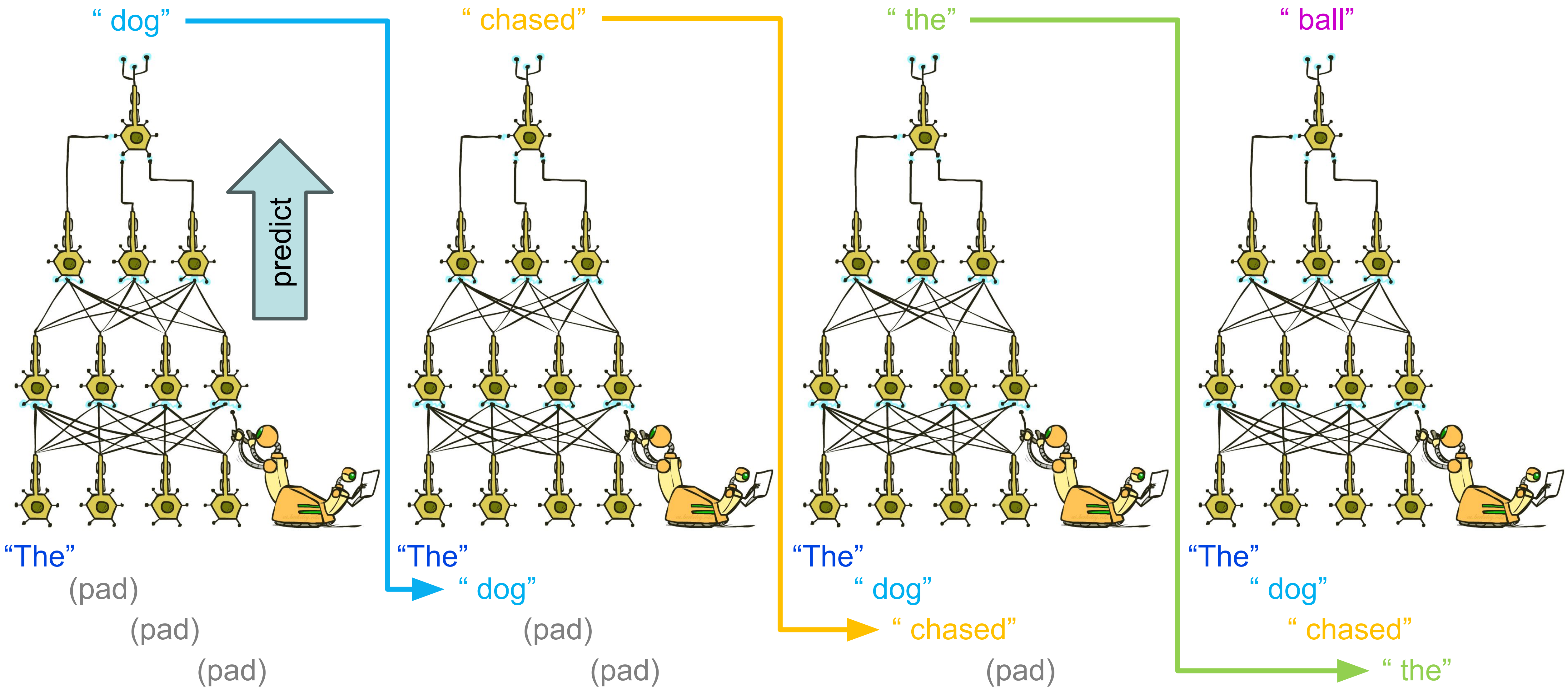
This is how it works

By **Visual Storytelling Team** and **Madhumita Murgia** in London SEPTEMBER 11 2023

Large Language Models

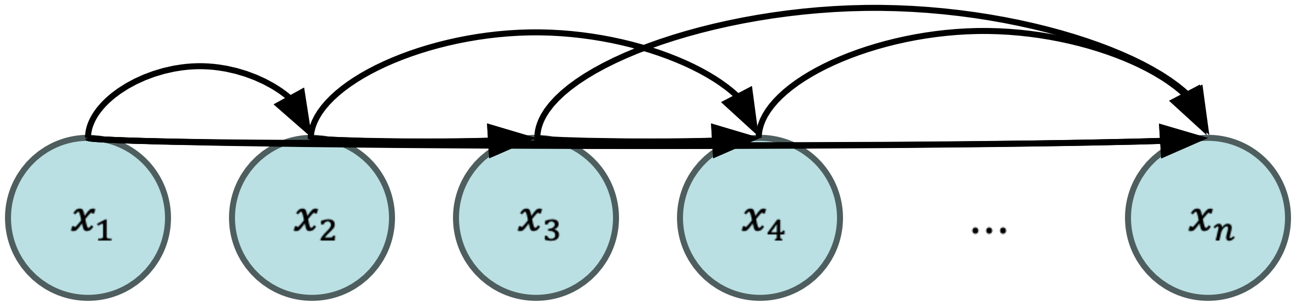
- ~~Feature engineering~~
  - ~~Text tokenization~~
  - ~~Word embeddings~~
- Deep neural networks
  - Autoregressive models
  - Self-attention mechanisms
  - Transformer architectures
- Multi-class classification
- Supervised learning
  - Self-supervised learning
  - Instruction tuning
- Reinforcement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

Autoregressive Models

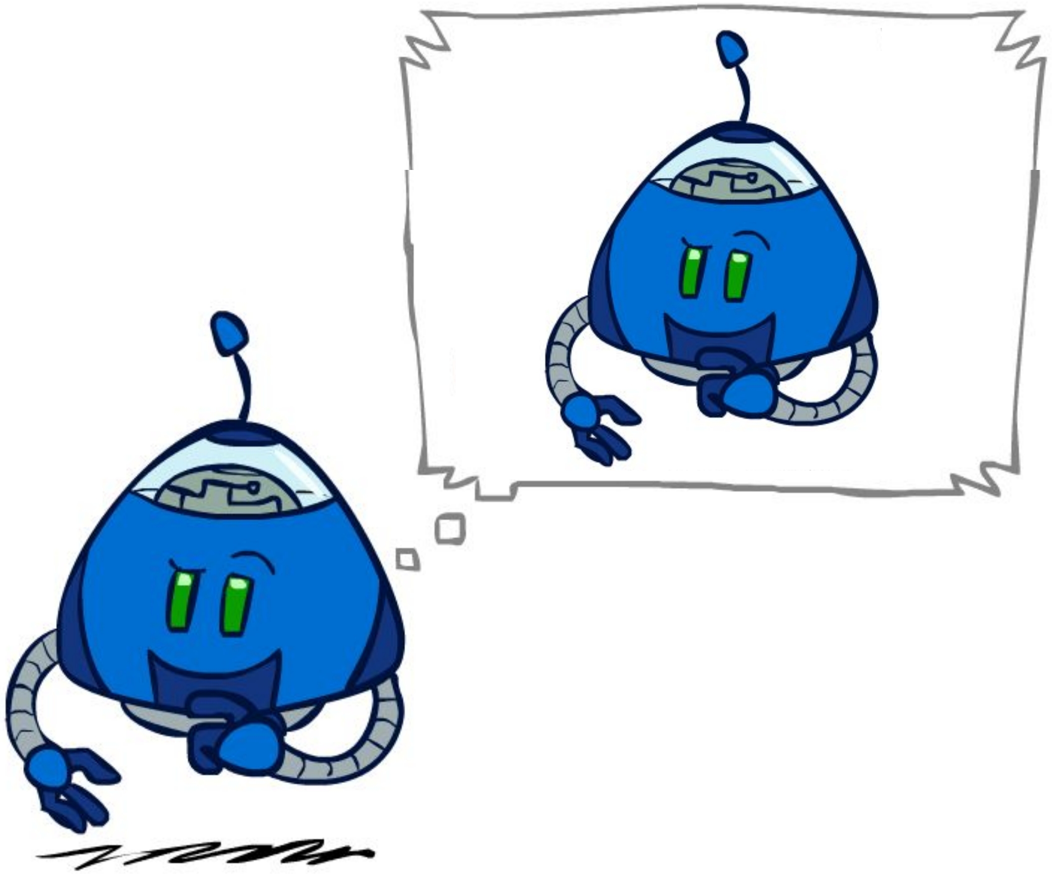


Autoregressive Models

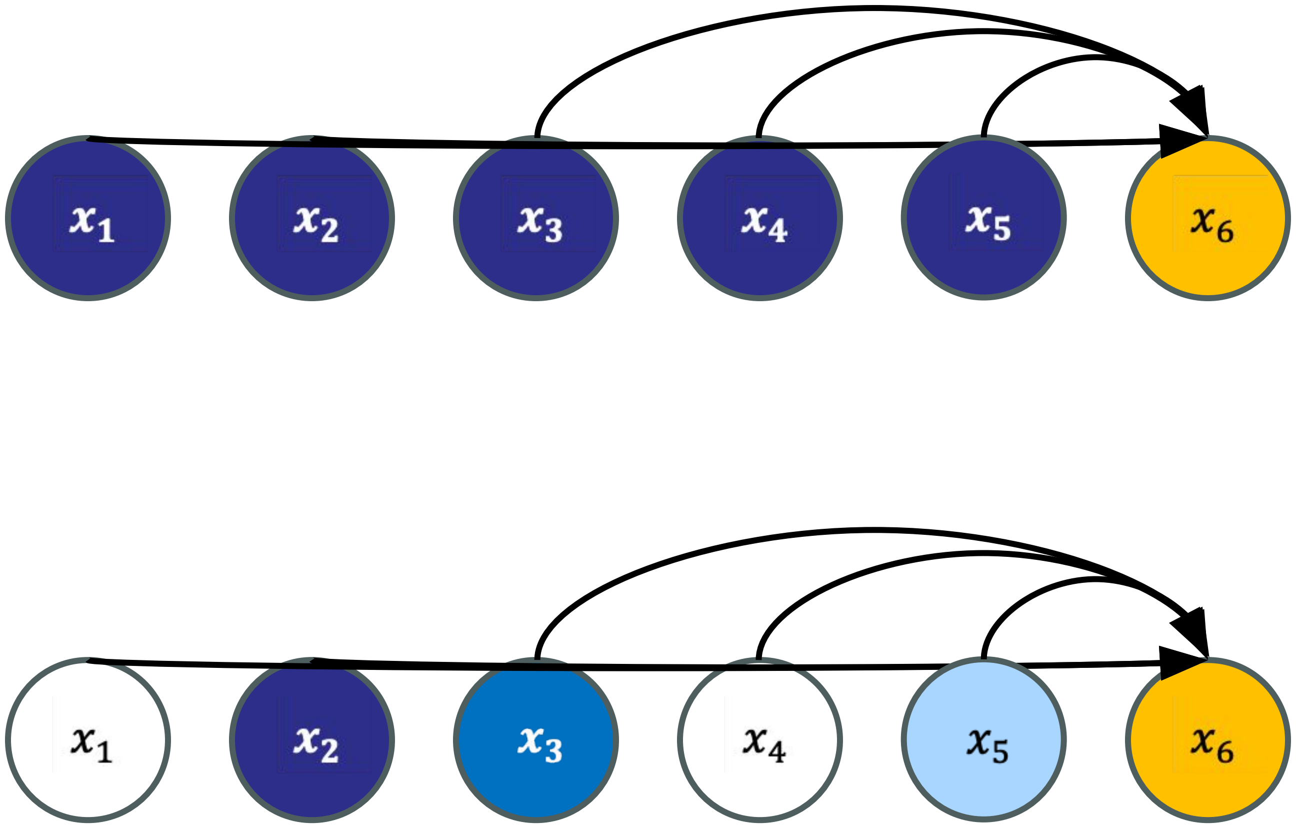
- Predict output one piece at a time (e.g. word, token, pixel, etc.)
- Concatenate: input + output
- Feed result back in as new input
- Repeat



Self-Attention Mechanisms

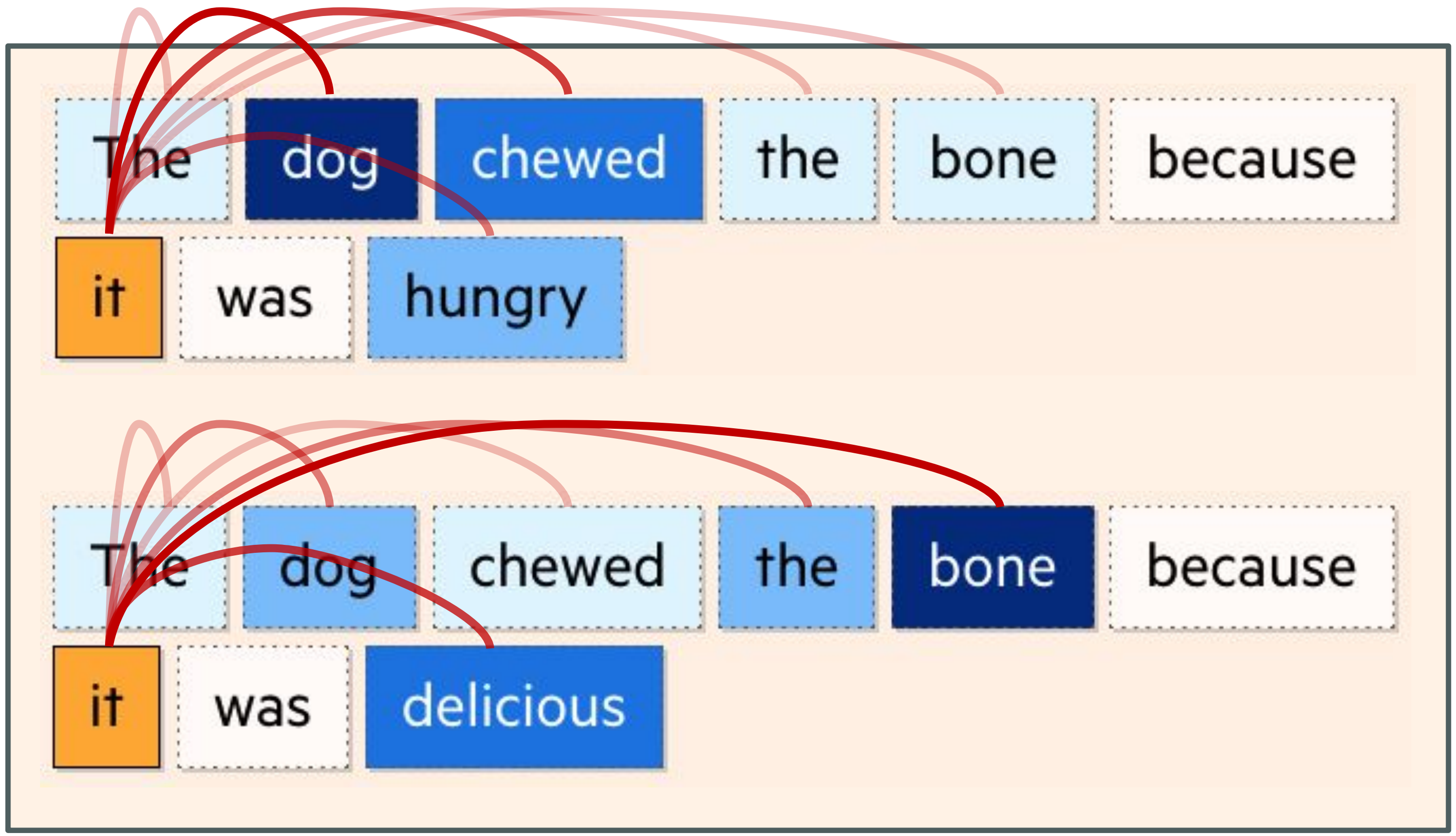


Self-Attention Mechanisms

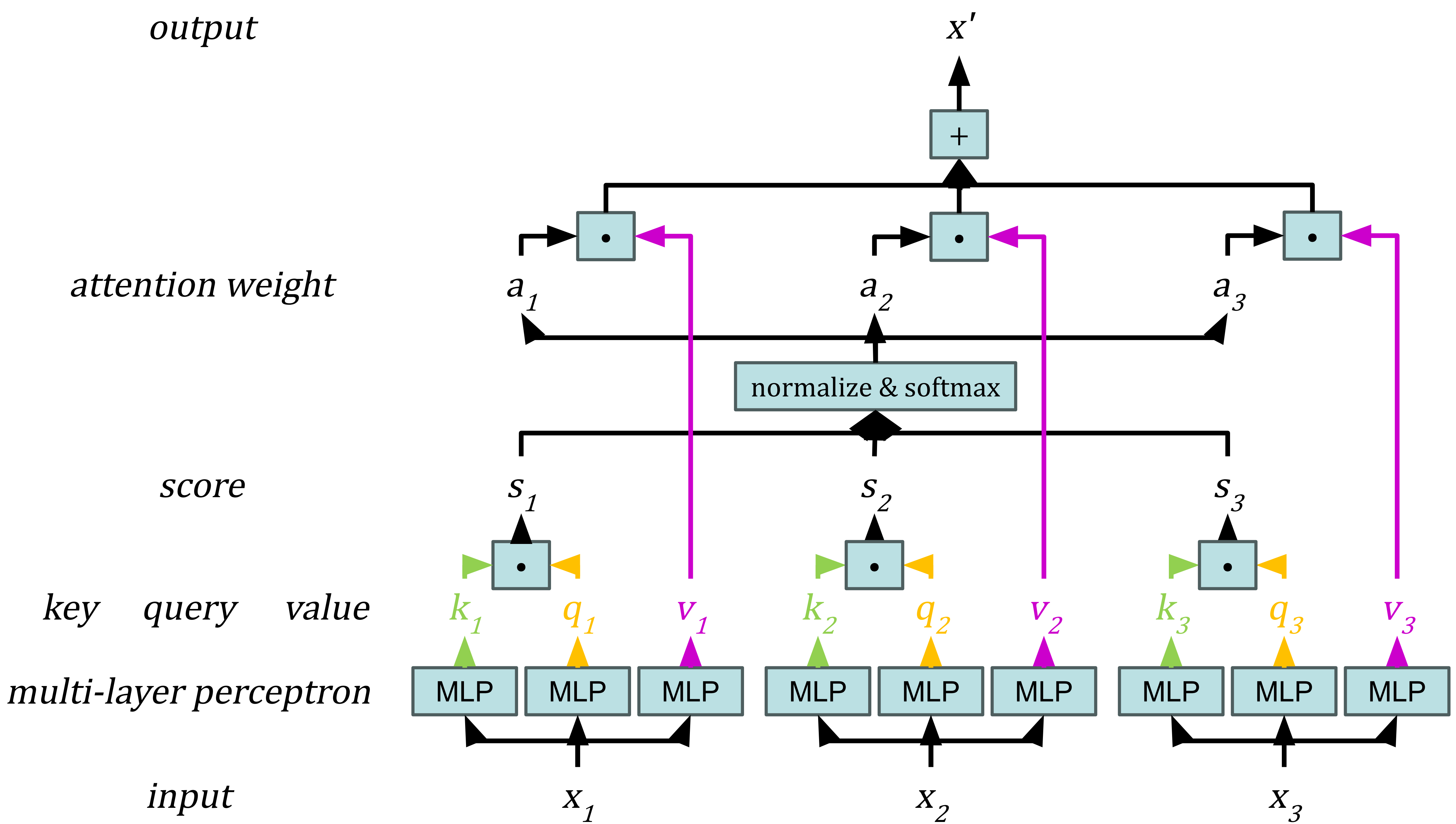


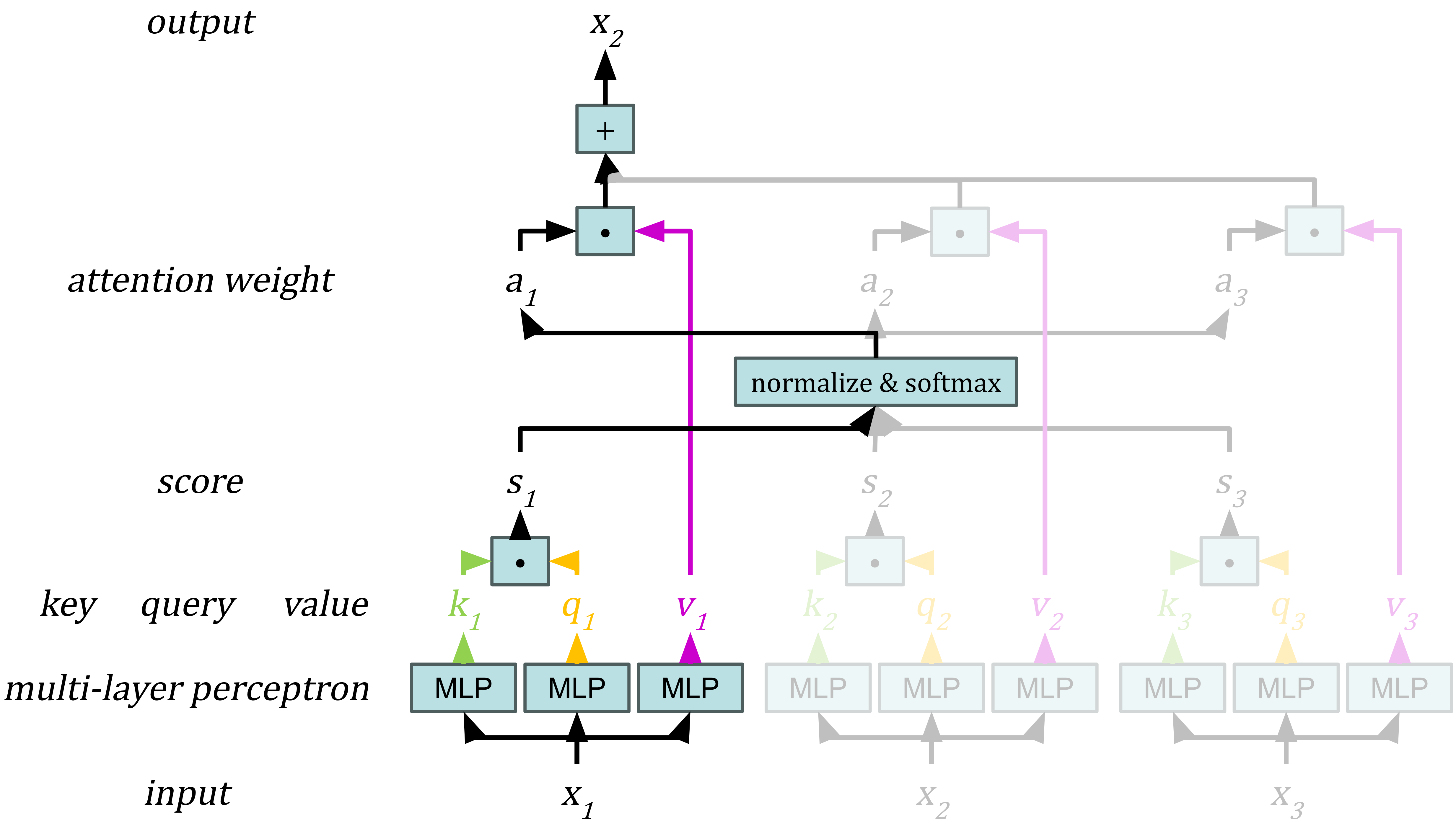
- Instead of conditioning on *all* input tokens equally…
- Pay more attention to relevant tokens!

Self-Attention Mechanisms

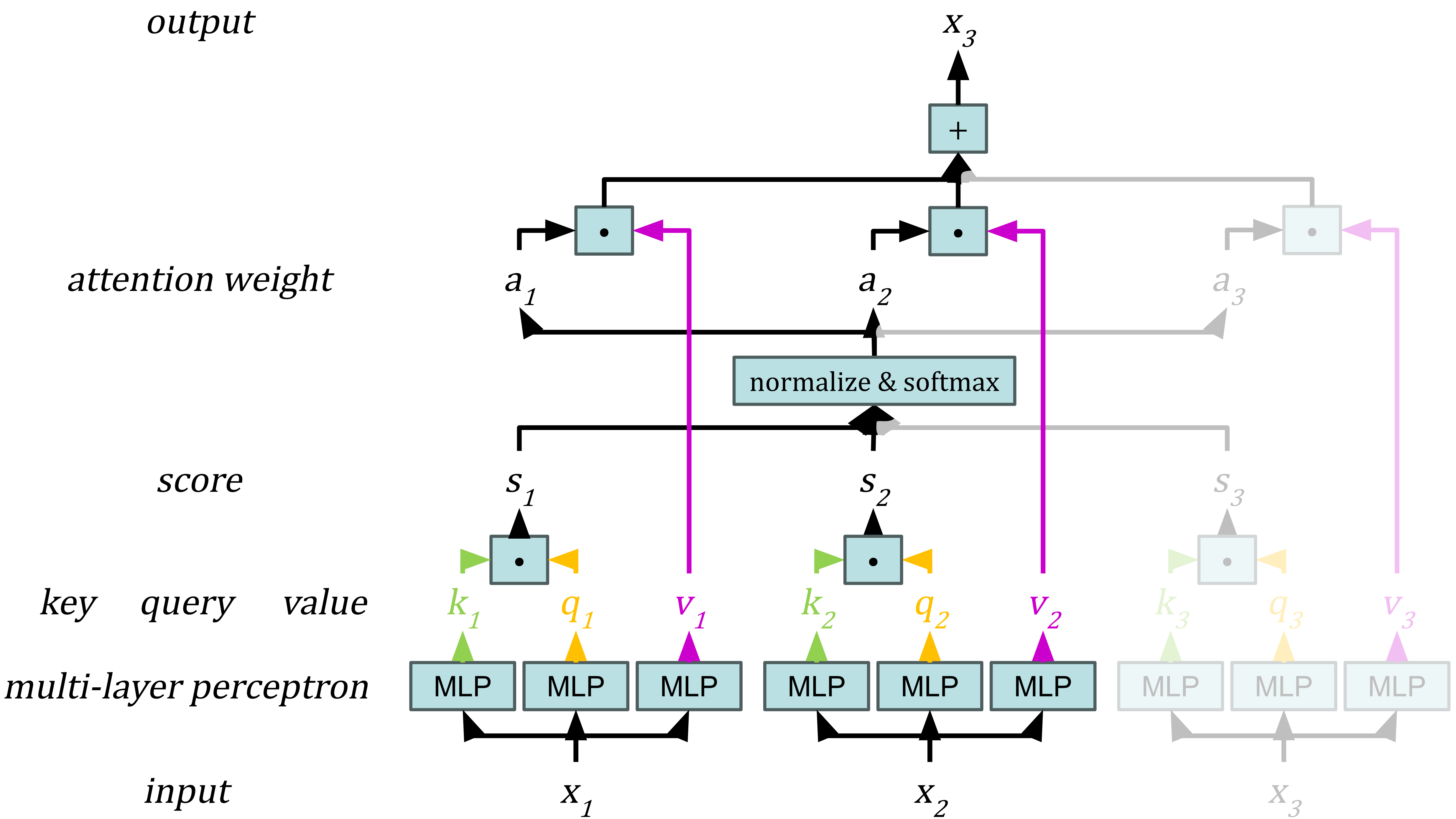


ig.ft.com/generative-ai

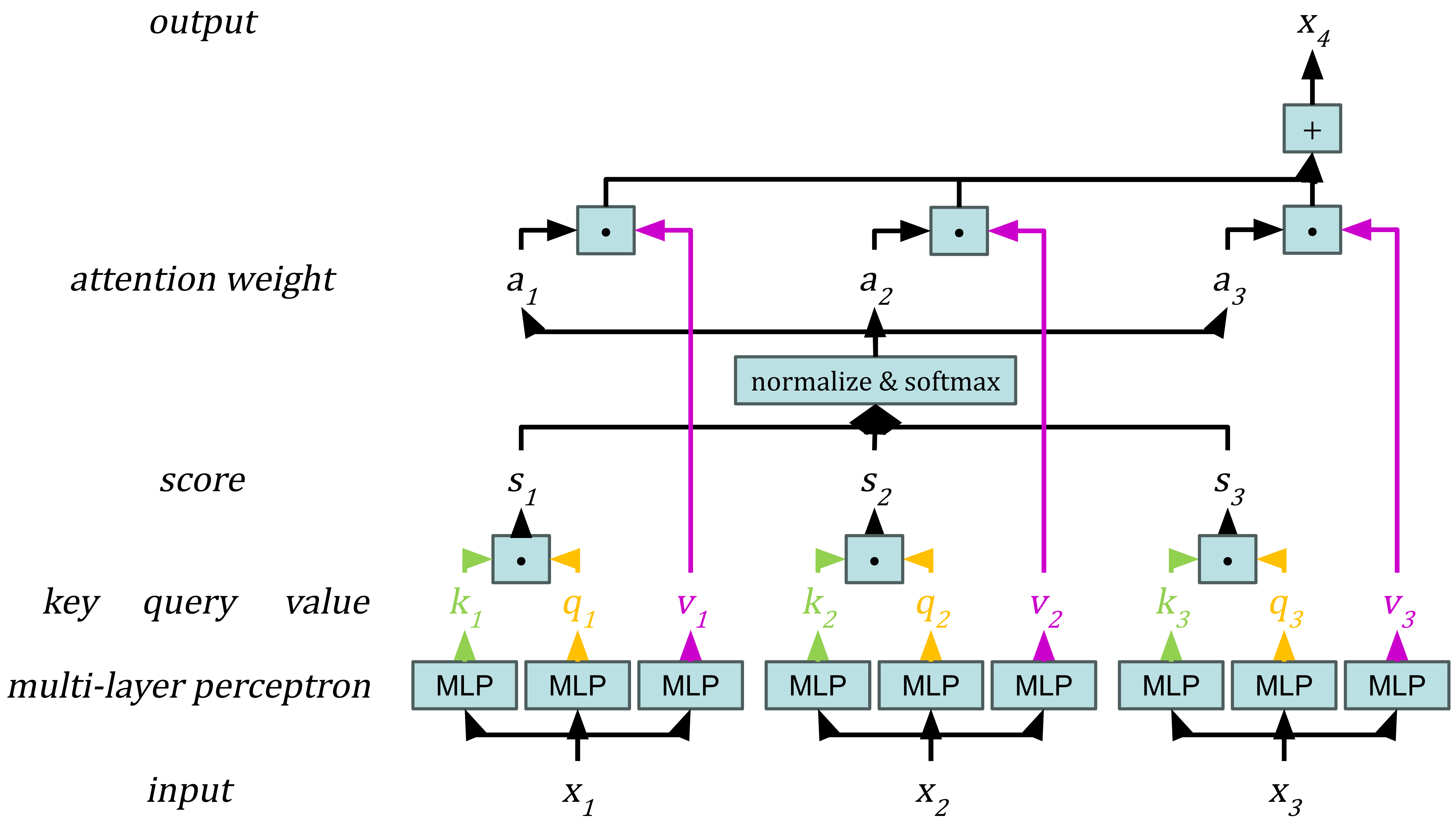

output
x'
+
attention weight
a1 a2 a3
normalize & softmax
score
s1 s2 s3
key query value
k1 q1 v1 k2 q2 v2 k3 q3 v3
multi-layer perceptron
MLP MLP MLP MLP MLP MLP MLP MLP MLP
input
x1 x2 x3



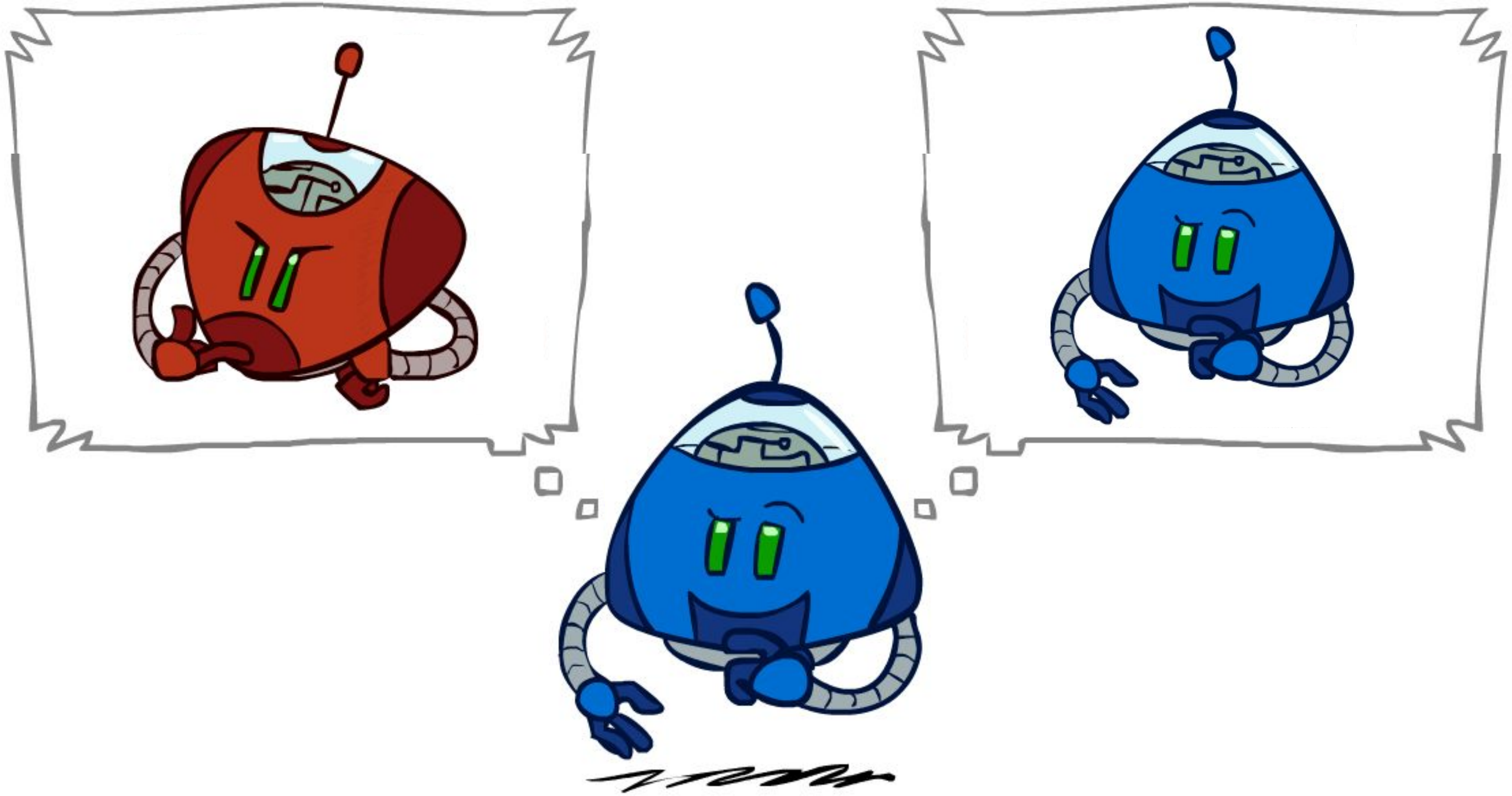
output
x_2
+
attention weight
a_1
a_2
a_3
normalize & softmax
score
s_1
s_2
s_3
key
query
value
k_1
q_1
v_1
k_2
q_2
v_2
k_3
q_3
v_3
multi-layer perceptron
MLP
MLP
MLP
MLP
MLP
MLP
MLP
MLP
MLP
input
x_1
x_2
x_3



output
x_3
+
attention weight
a_1
a_2
a_3
normalize & softmax
score
s_1
s_2
s_3
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

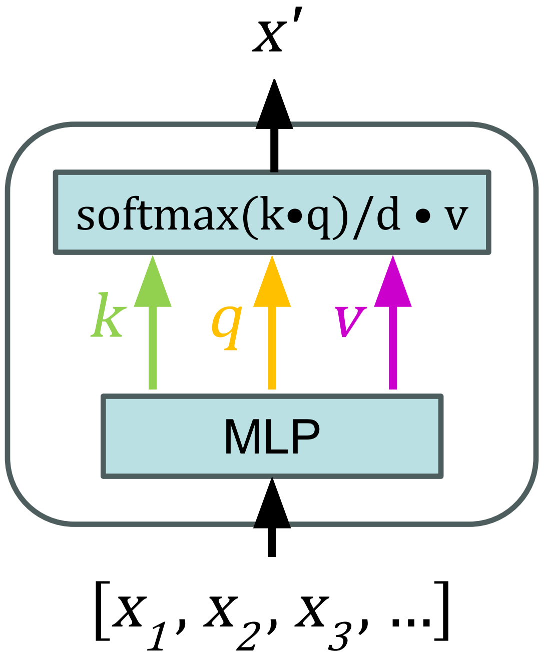

output
attention weight
score
key
query
value
multi-layer perceptron
input
x_4
+
a_1
a_2
a_3
normalize & softmax
s_1
s_2
s_3
k_1
q_1
v_1
k_2
q_2
v_2
k_3
q_3
v_3
MLP
x_1
x_2
x_3

Multi-Headed Attention

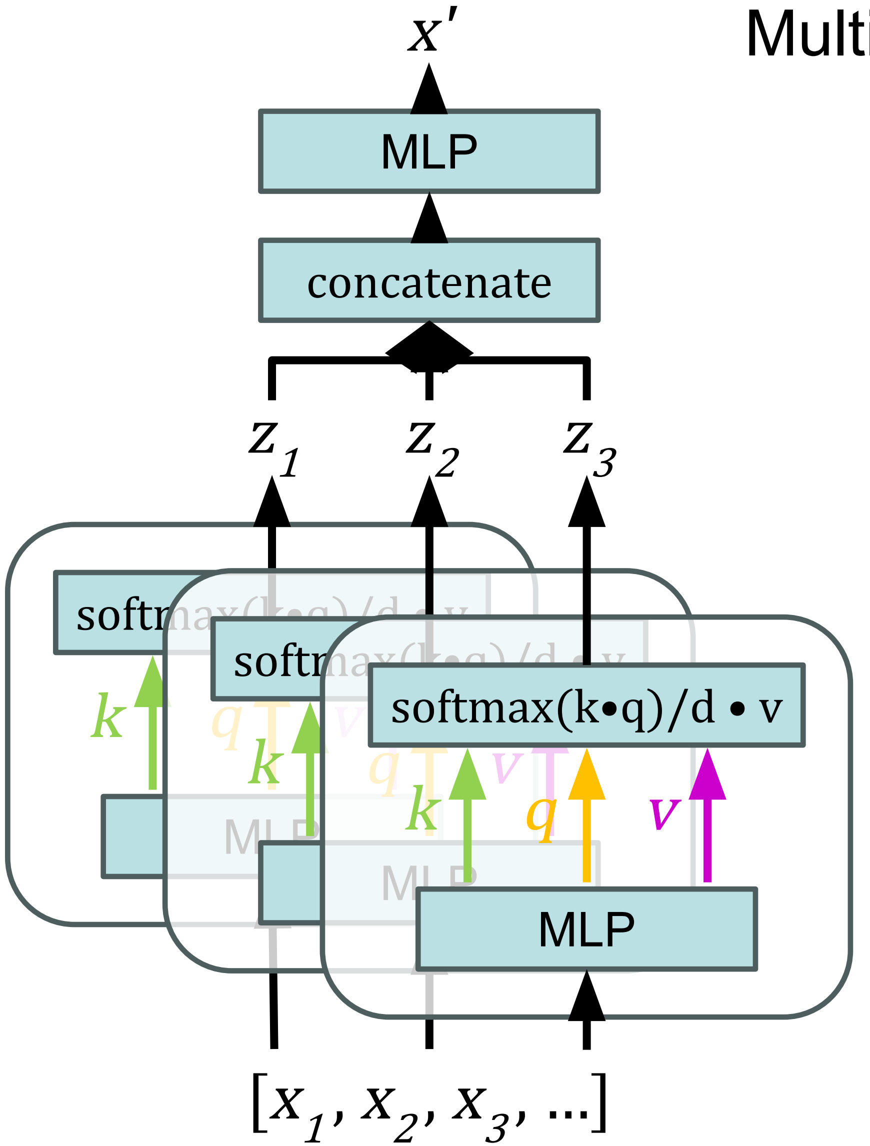


Multi-Headed Attention

Single-headed

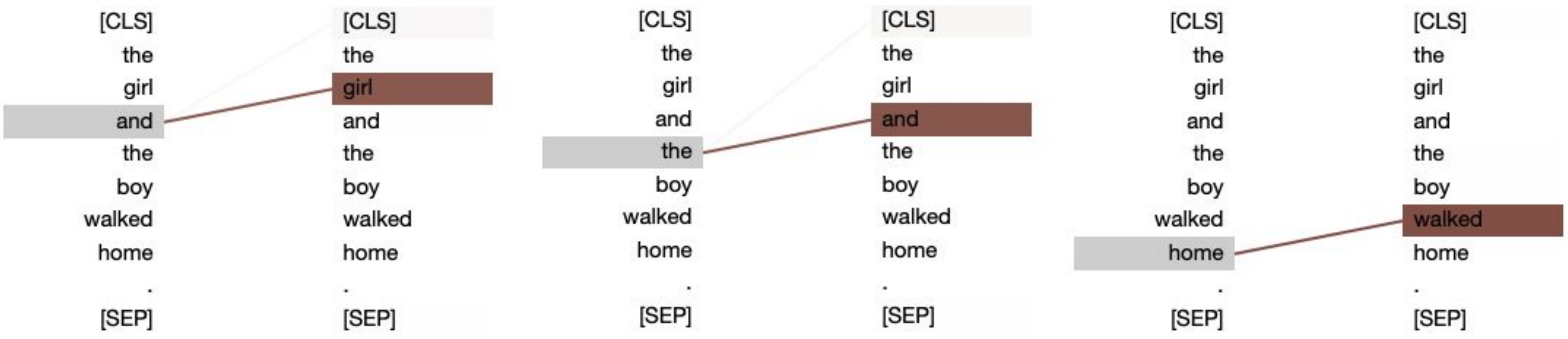


Multi-headed



Multi-Headed Attention

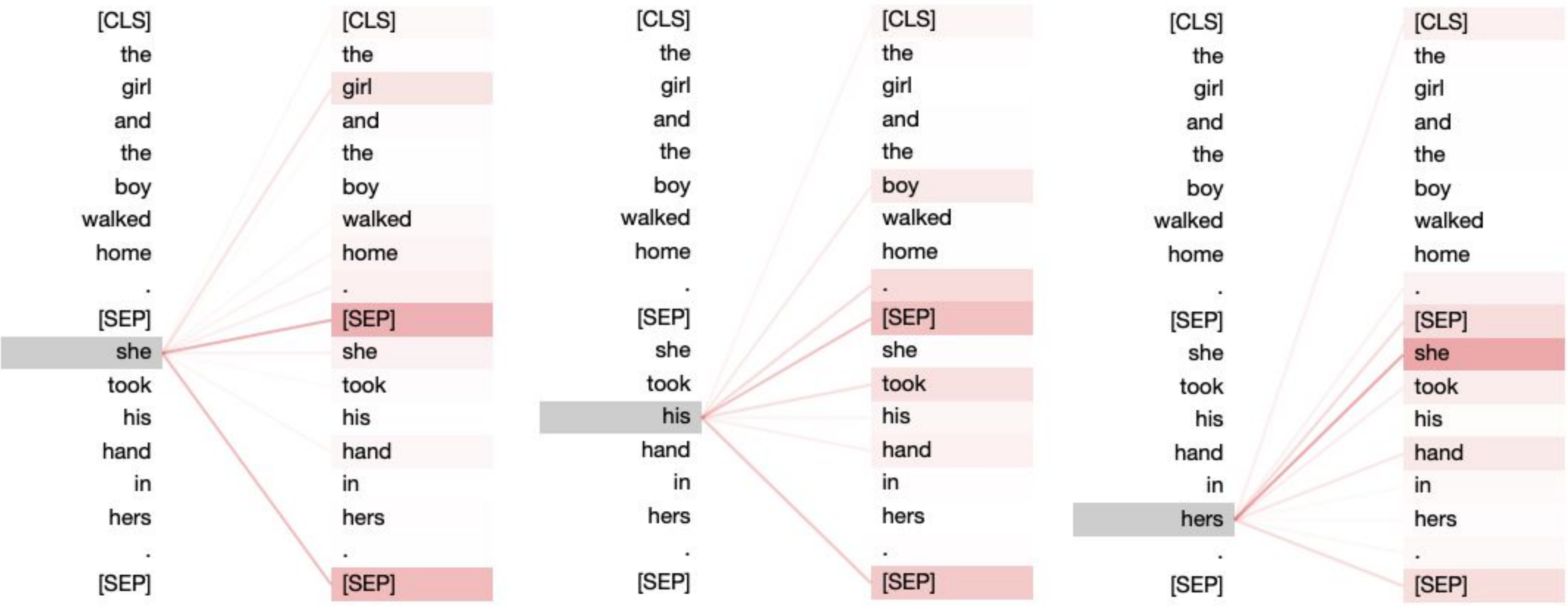
Head 6: previous word



https://github.com/jessevig/bertviz

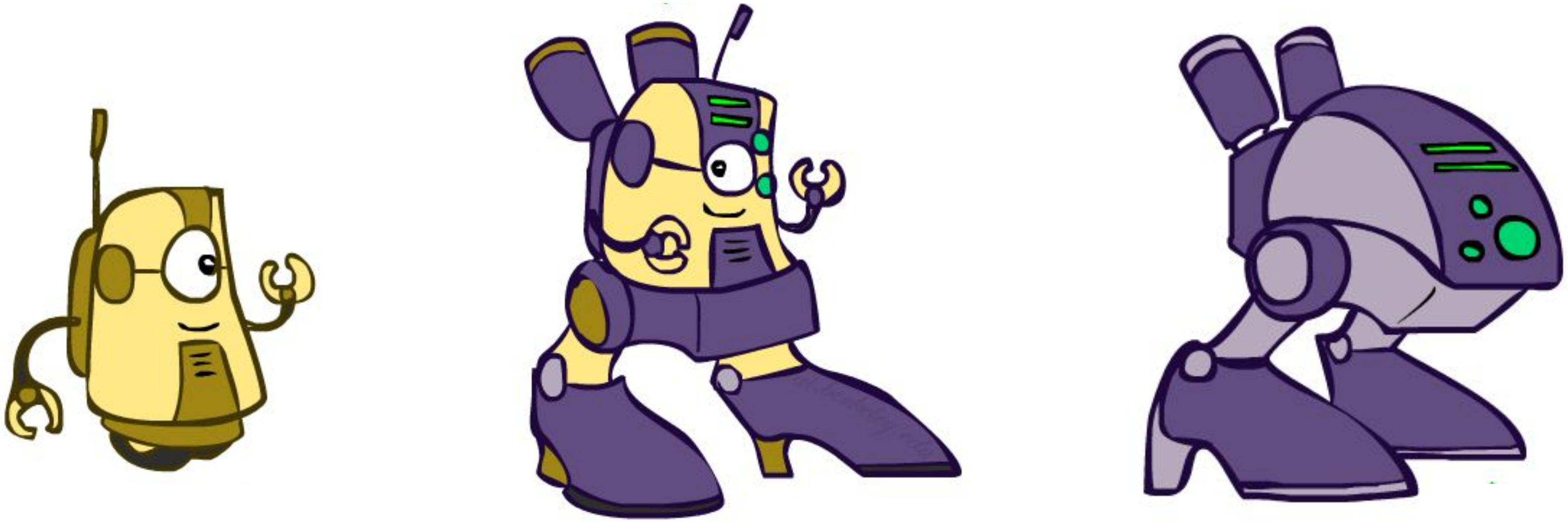
Multi-Headed Attention

Head 4: pronoun references

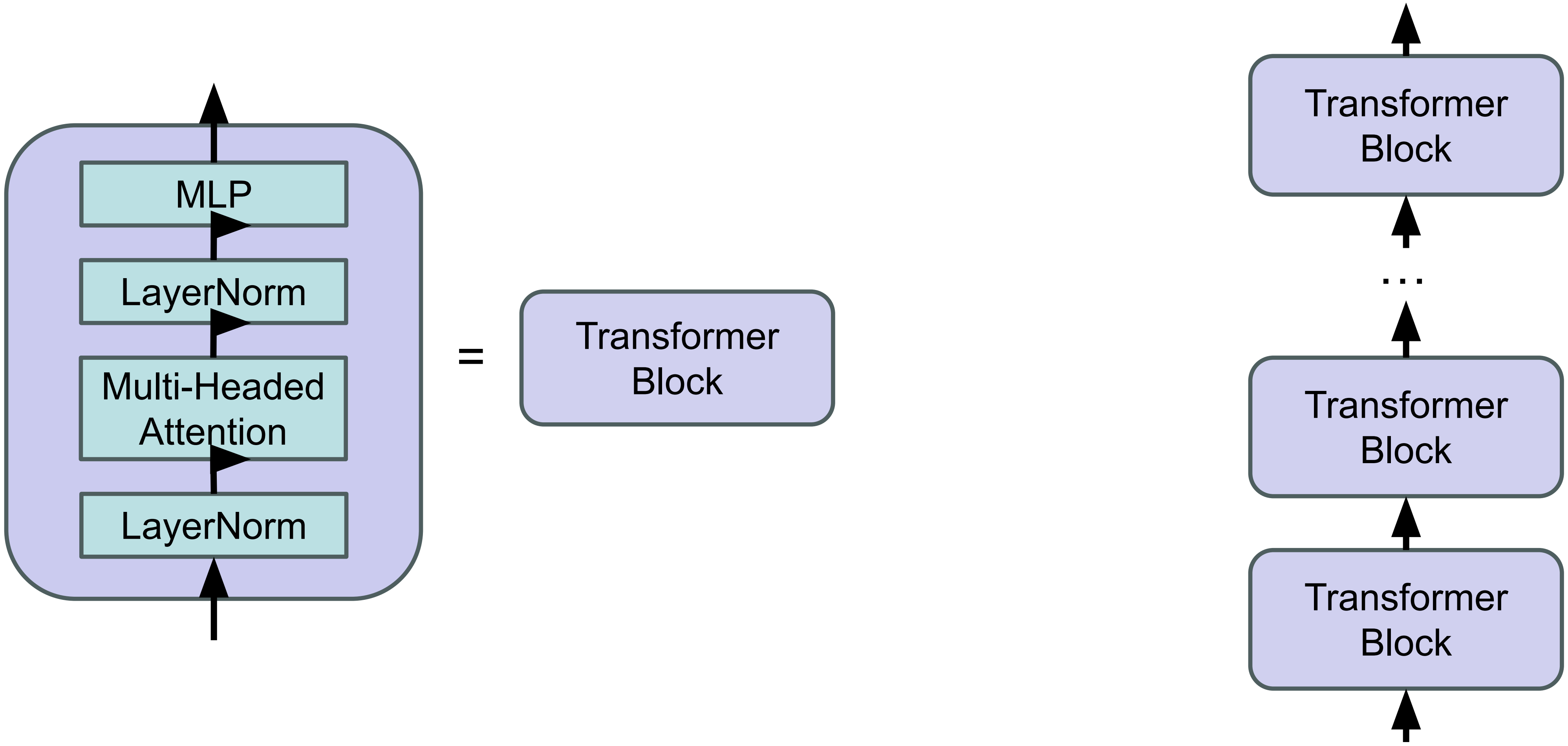


https://github.com/jessevig/bertviz

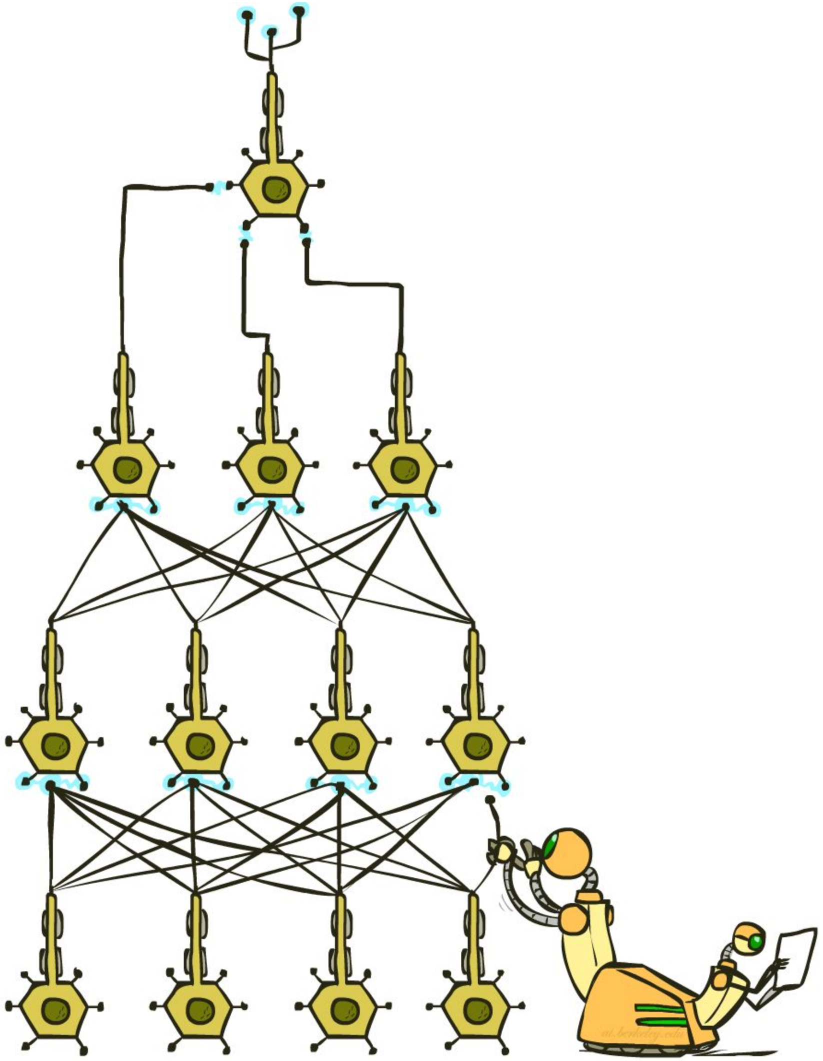
Transformer Architecture

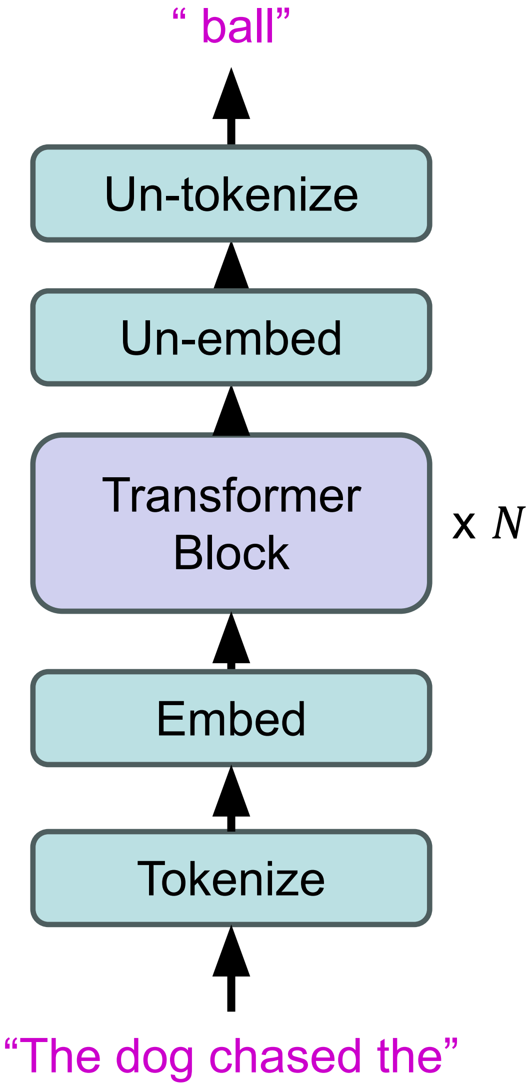


Transformer Architecture



Transformer Architecture





Large Language Models

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

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

Unsupervised / Self-Supervised Learning

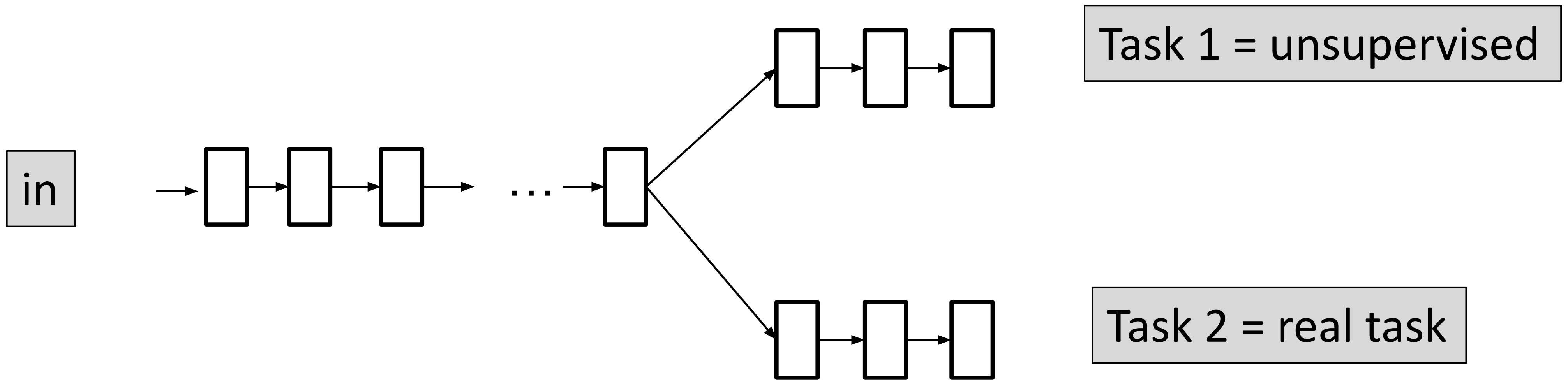
- Do we always need human supervision to learn features?
- Can't we learn general-purpose features?
- Key hypothesis:

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 very small data-set

Transfer from Unsupervised Learning



Example Setting

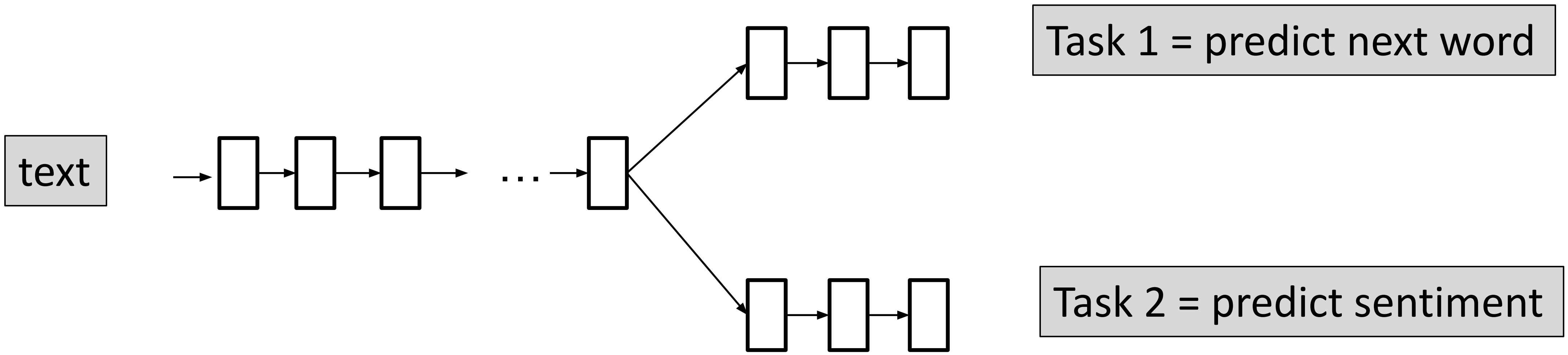


Image Pre-Training: Predict Missing Patch

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
  - …

Knowing what to optimize for is very hard

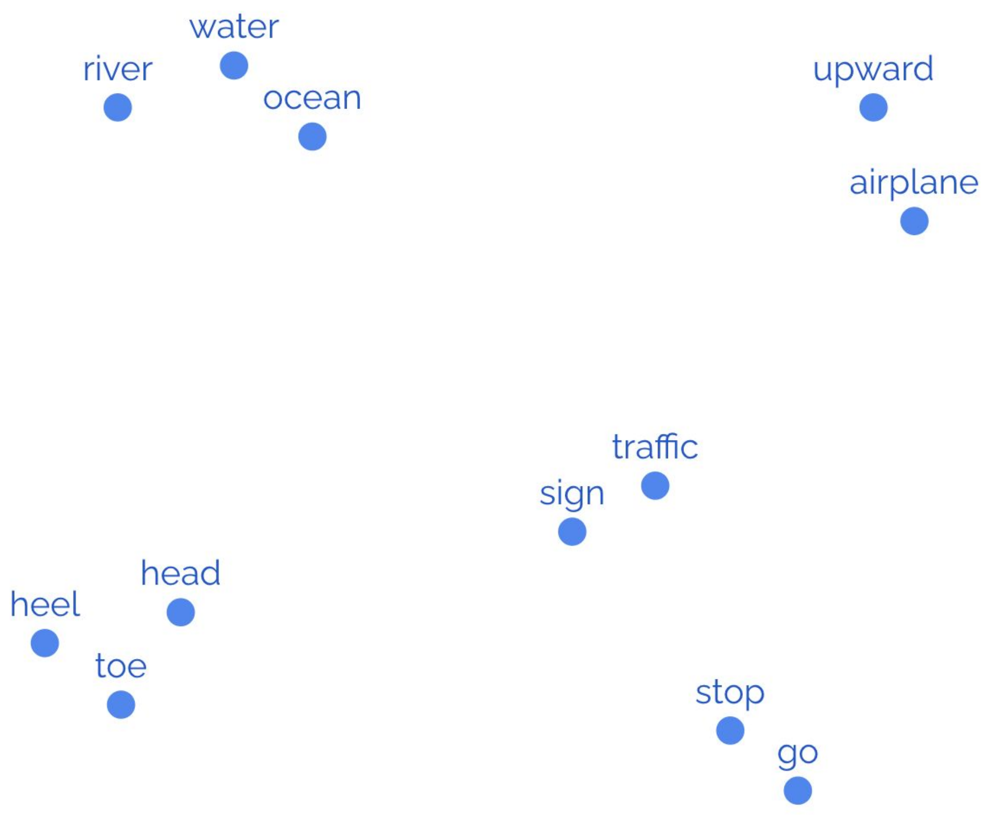
- More generally: lots of bad things happen when there is a gap between *what we really want to optimize for* and *what we train the model to optimize for*

Desired target	Actual target	Bias
Patient health needs	Patient health costs	Disparities in access to care
Severity of knee osteoarthritis	Severity as assessed by radiologist	Radiologists overlook features affecting underserved populations
Crime rates	Arrest rates	Disparities in policing
What users value	What users click on	Clickbait

Large Language Models

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

Language models build a structured concept space

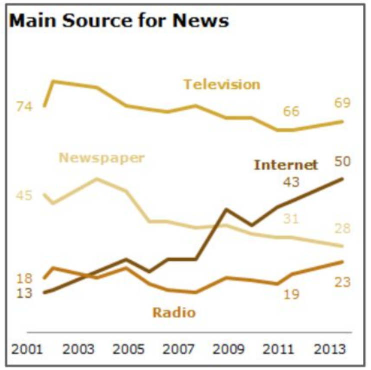


Can other data (images/audio/…) be put in this space?

Can we build a single model of all data types?

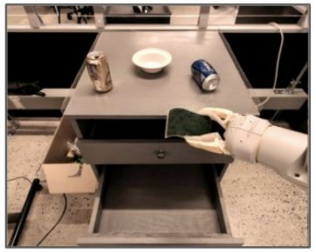
If  was invented by Wright brothers. Who invented  ?

example from [Tsimpoukelli et al, 2021]

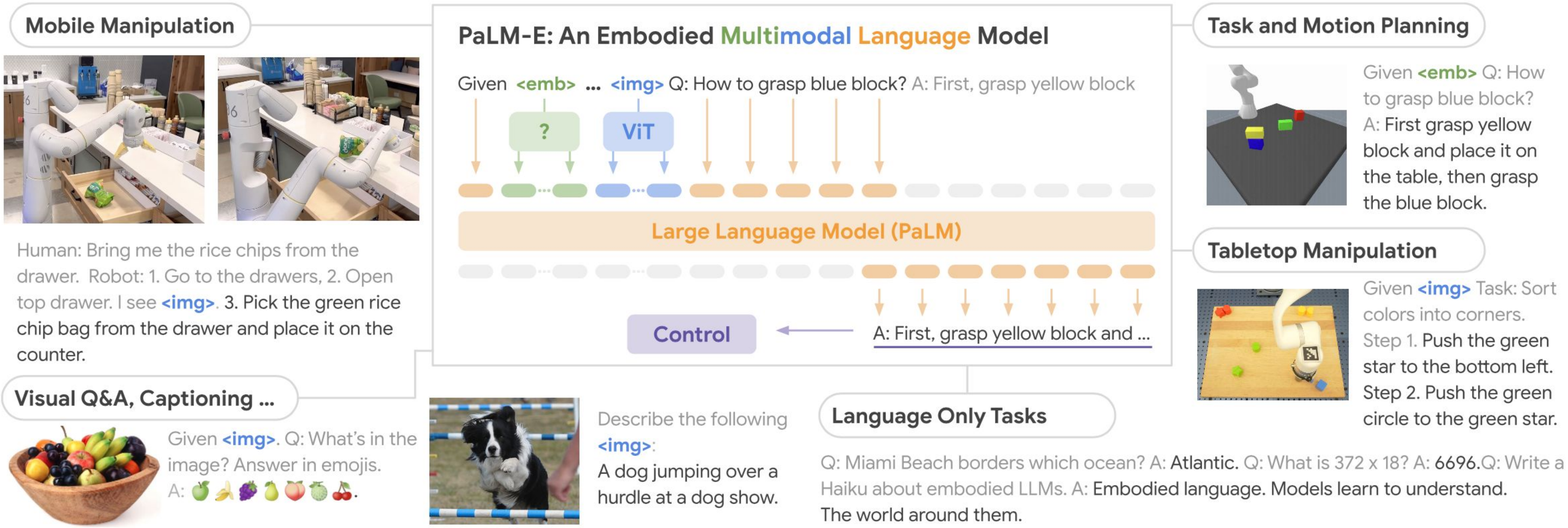
What is the fastest-growing news source according to 

?

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]