# Predictive

# ~~Big Data~~ Analytics (ISOM 5270)

by Prof. Fernández-Loría

There won’t be a quiz today.

# The Plan

~~Fundamentals~~

**Week 1**: ~~Problem formulation.~~

**Week 2**: ~~Modeling (Part 1).~~

**Week 3**: ~~Modeling (Part 2).~~

**Week 4**: ~~Model Evaluation.~~

---

(What I think are)
The Big Three

**Week 5**: ~~Causal analytics.~~

**Week 6**: ~~Big (unstructured) data.~~

**Week 7**: Generative AI.

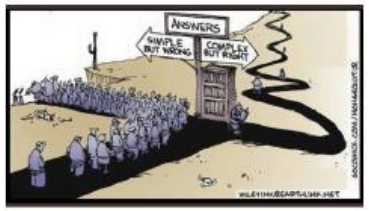
---

**Week 8**: Final Exam.

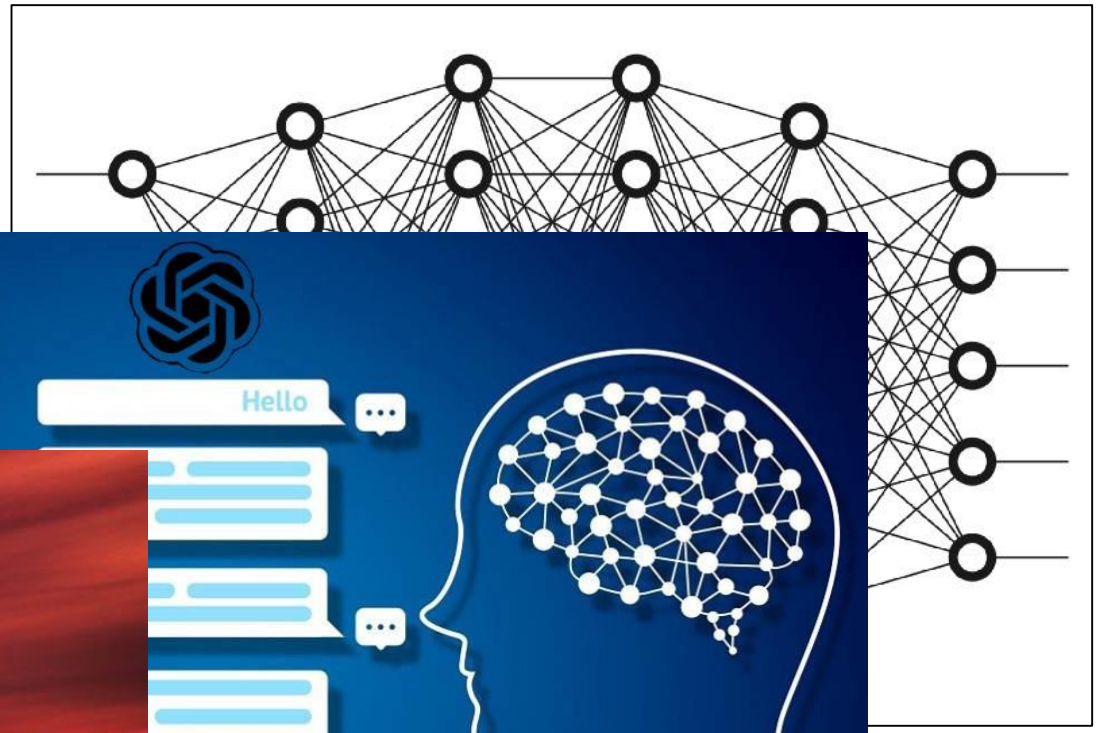
Today it will be a little more this…

…than this

# The Effectiveness (and Ineffectiveness) of Lectures

| What is a lecture NOT so good for? | What is a lecture good for? |
|---|---|
| Learn to learn on your own. | Presenting current information. |
|   Think critically about problems AND solutions. | Summarizing material. |
| Communicate effectively. | Focusing on a particular area of interest. |
| | Relaying key concepts or ideas. |
| | Building interest. |

# Focus of Today:



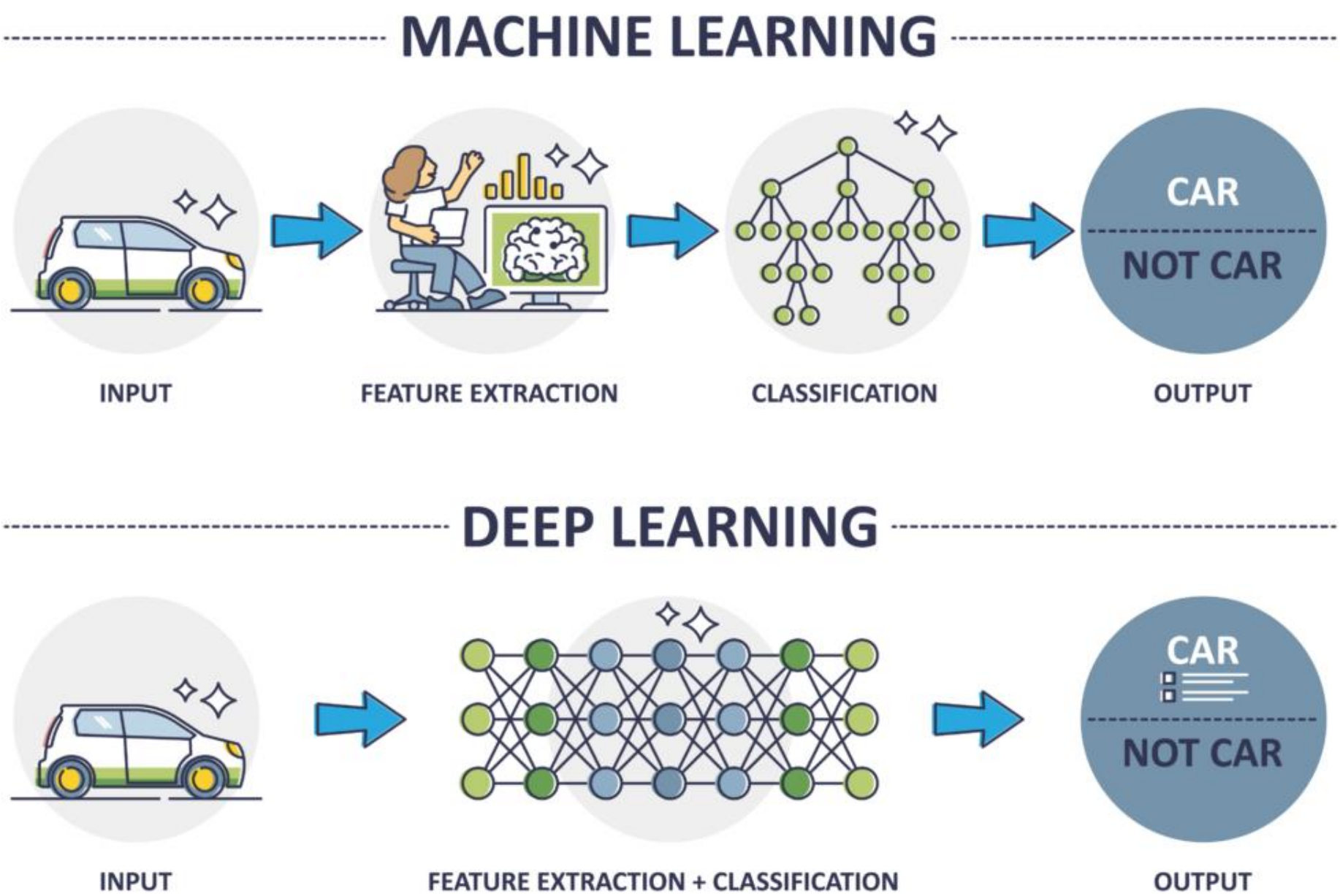
g

uage models.

The course main takeaways.

# Let’s talk about deep learning.

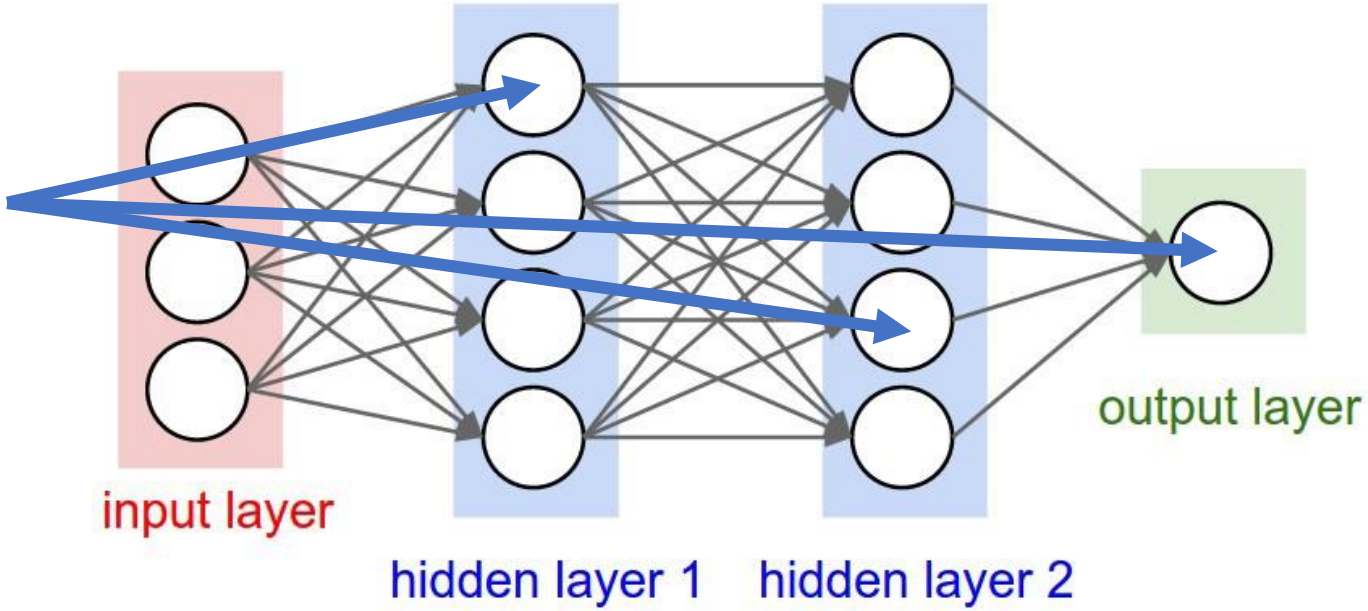
This assumes you watched the neural networks video from 3Blue1Brown.
https://www.youtube.com/watch?v=aircAruvnKk


MACHINE LEARNING
CAR
NOT CAR
INPUT
FEATURE EXTRACTION
CLASSIFICATION
OUTPUT
DEEP LEARNING
CAR
NOT CAR
INPUT
FEATURE EXTRACTION + CLASSIFICATION
OUTPUT

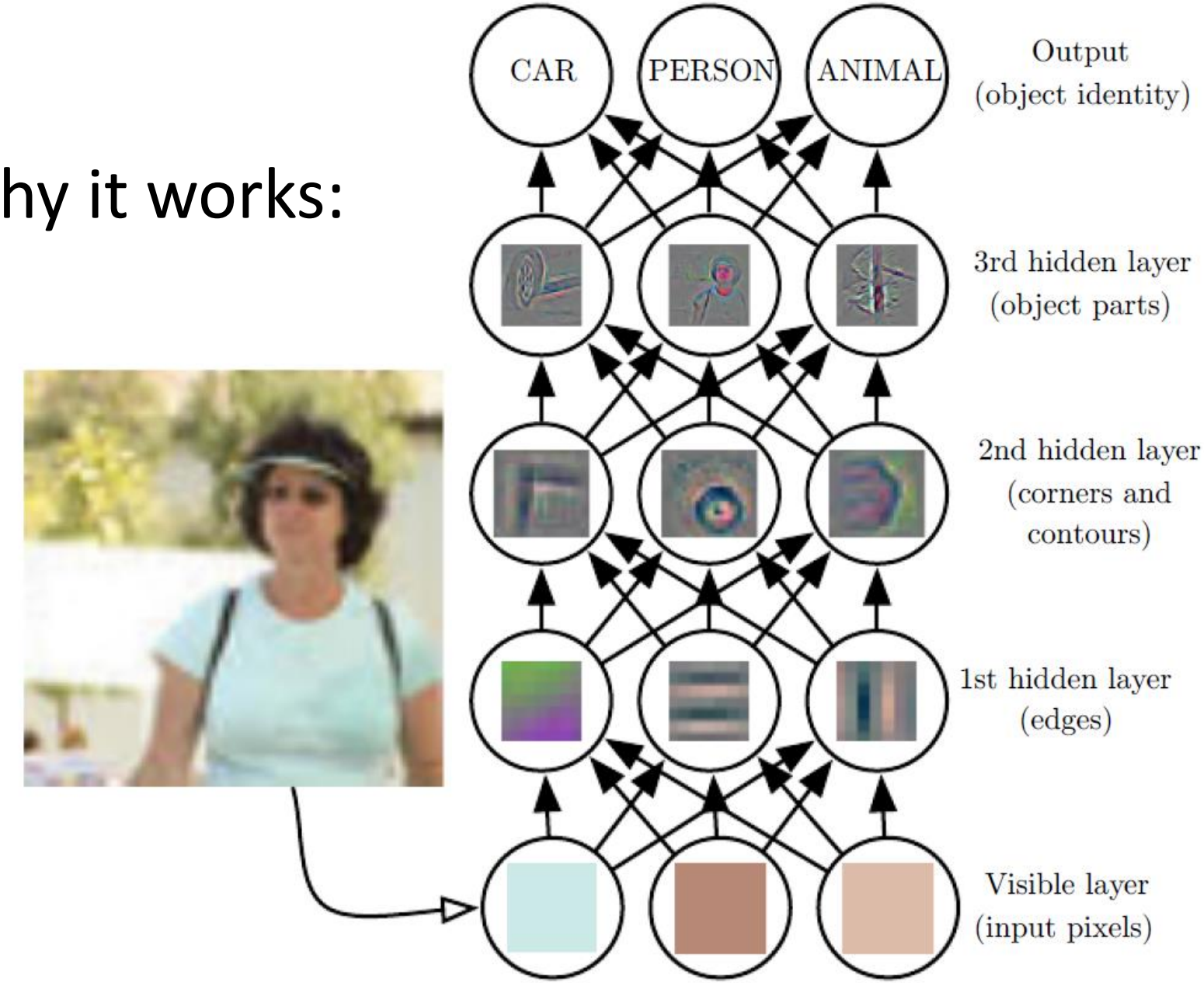
# Neural Networks (Deep Learning)

A neural network is a “stack” of models, where each layer generates increasingly complex (and non-linear) representations of the features:

**Models (e.g., logistic regression)**

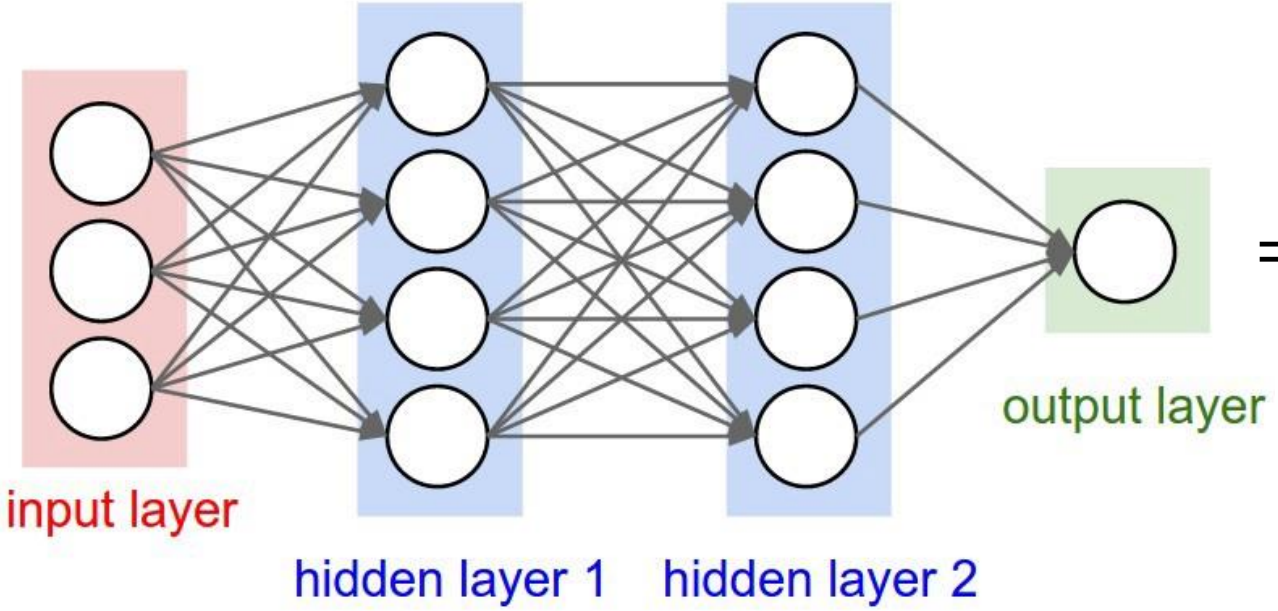


# Intuition for why it works:



# How do we learn these representations?

We only need a (differentiable) objective function:




$= f(\boldsymbol{x})$ vs $y$

The network is a group of parameters that defines a complex mathematical function. The algorithm “wiggles” parameters with large derivatives.

# Framework #5: Components of Machine Learning

Machine learning algorithms are programs that **fit** a **model** to **data** based on an **evaluation criterion**.

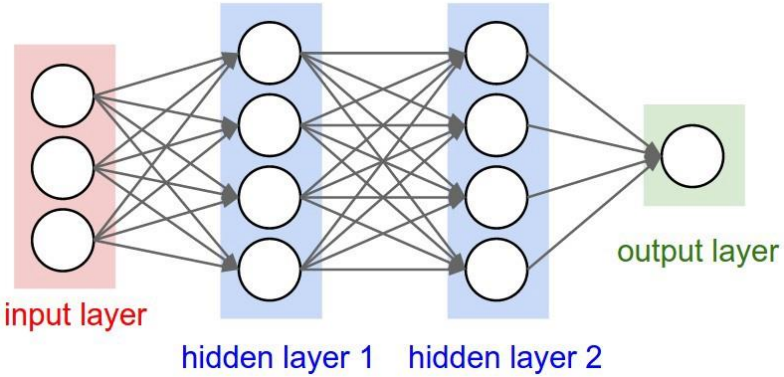
Representation + Evaluation + Optimization

$$\underset{\hat{q}}{\operatorname{argmin}} \sum_{\forall i \in \text{Data}} \text{error}\left(q_i, \hat{q}(\boldsymbol{x}_i)\right)$$

Learning = Representation + Evaluation + Optimization

# Neural Networks

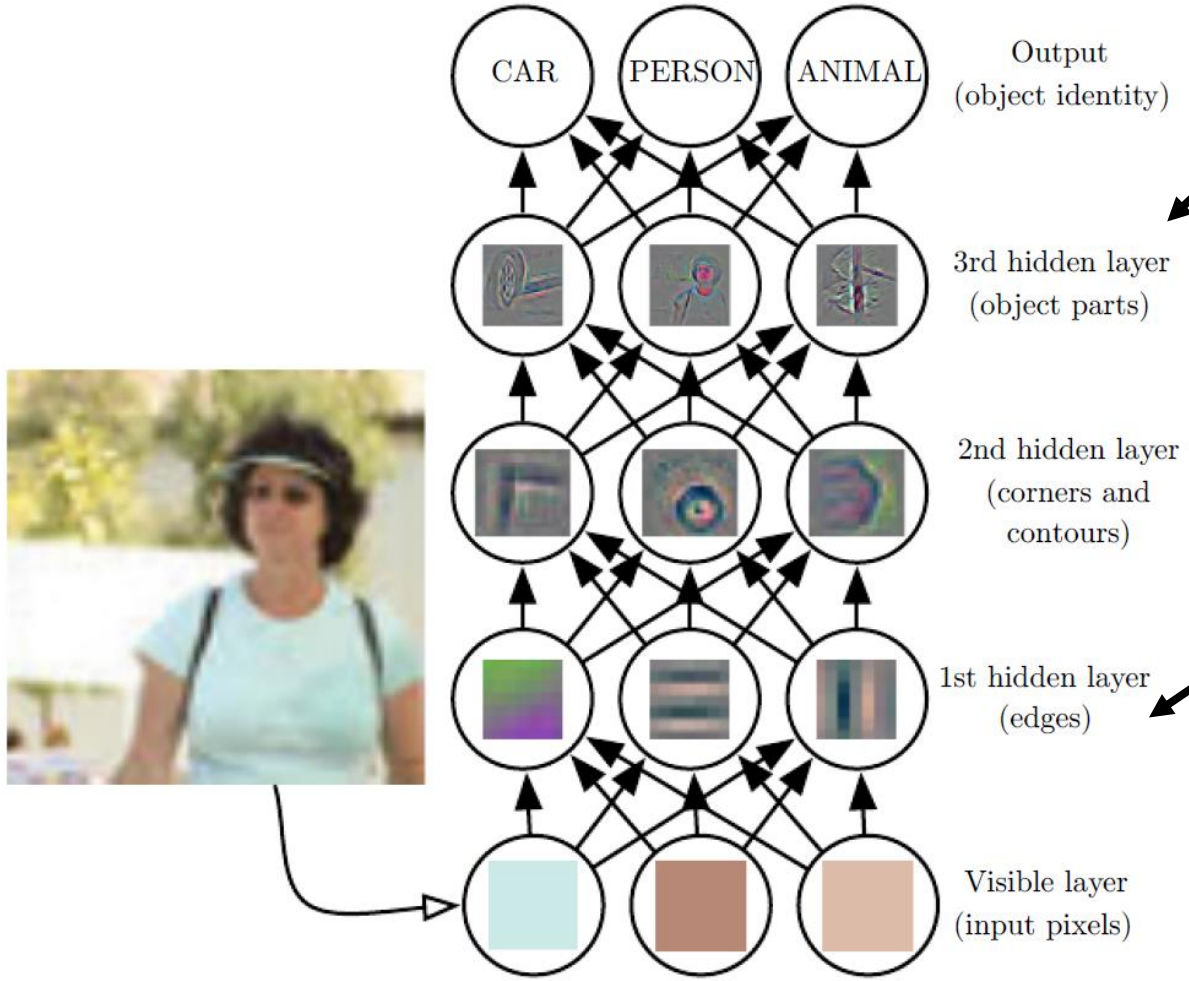
$\hat{q}(\boldsymbol{x}) =$




objective = Mean Squared Error (regression)
(Must be differentiable!) or Cross Entropy (classification)

optimization = Stochastic Gradient Descent
+ Backpropagation

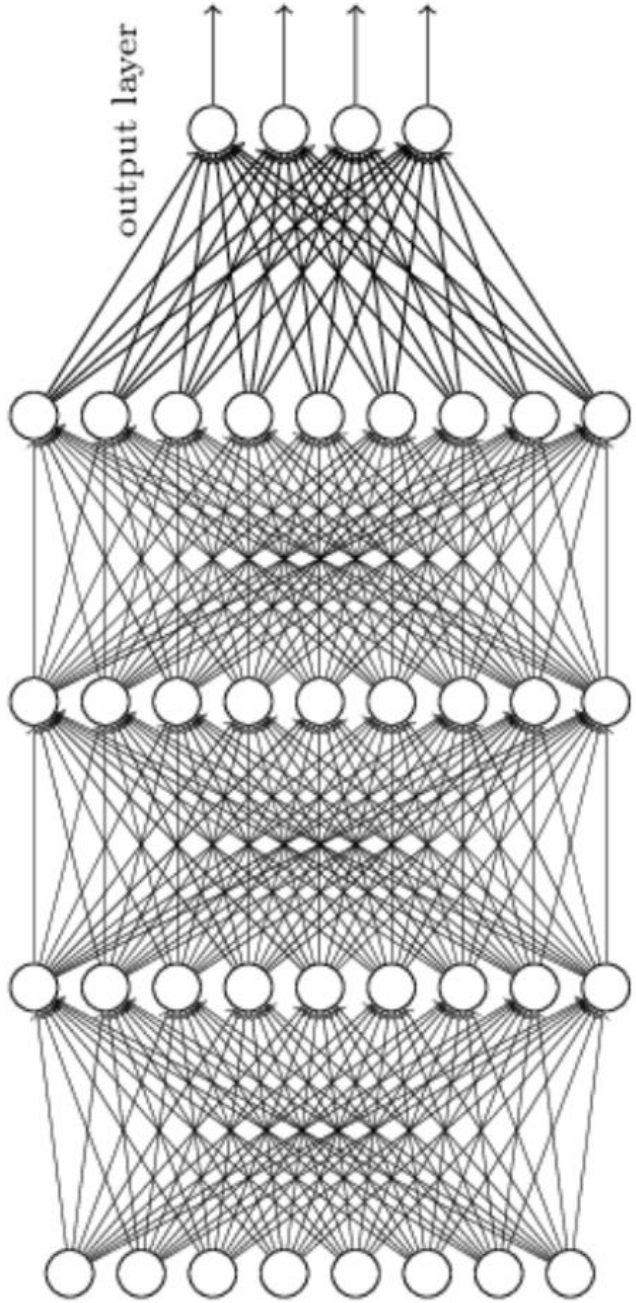
# In practice, there may be no direct match.




May not match!

May not match!

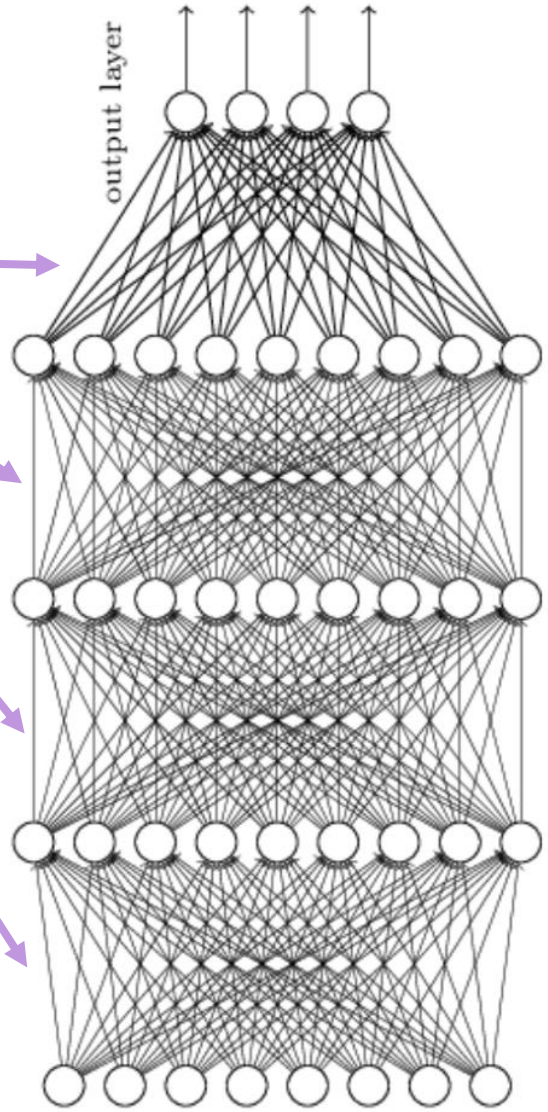
May not match!



$\hat{q}(\boldsymbol{x})$ vs. $q$

But the algorithm will learn the parameters (representations) that optimize the objective function.

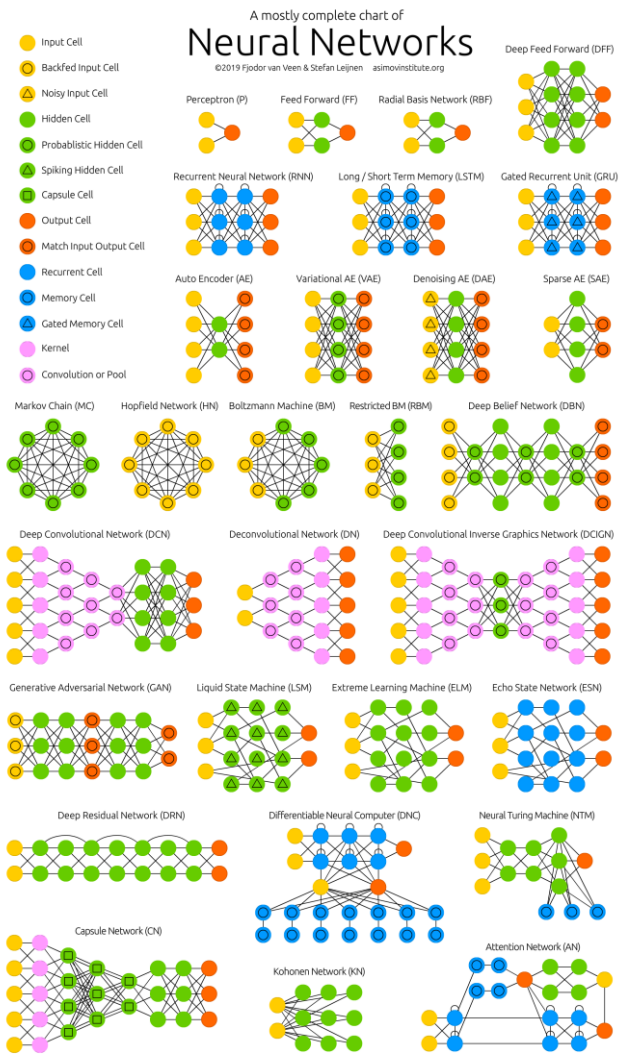
Deep learning is about learning predictive feature representations.



# Representation + Evaluation + Optimization

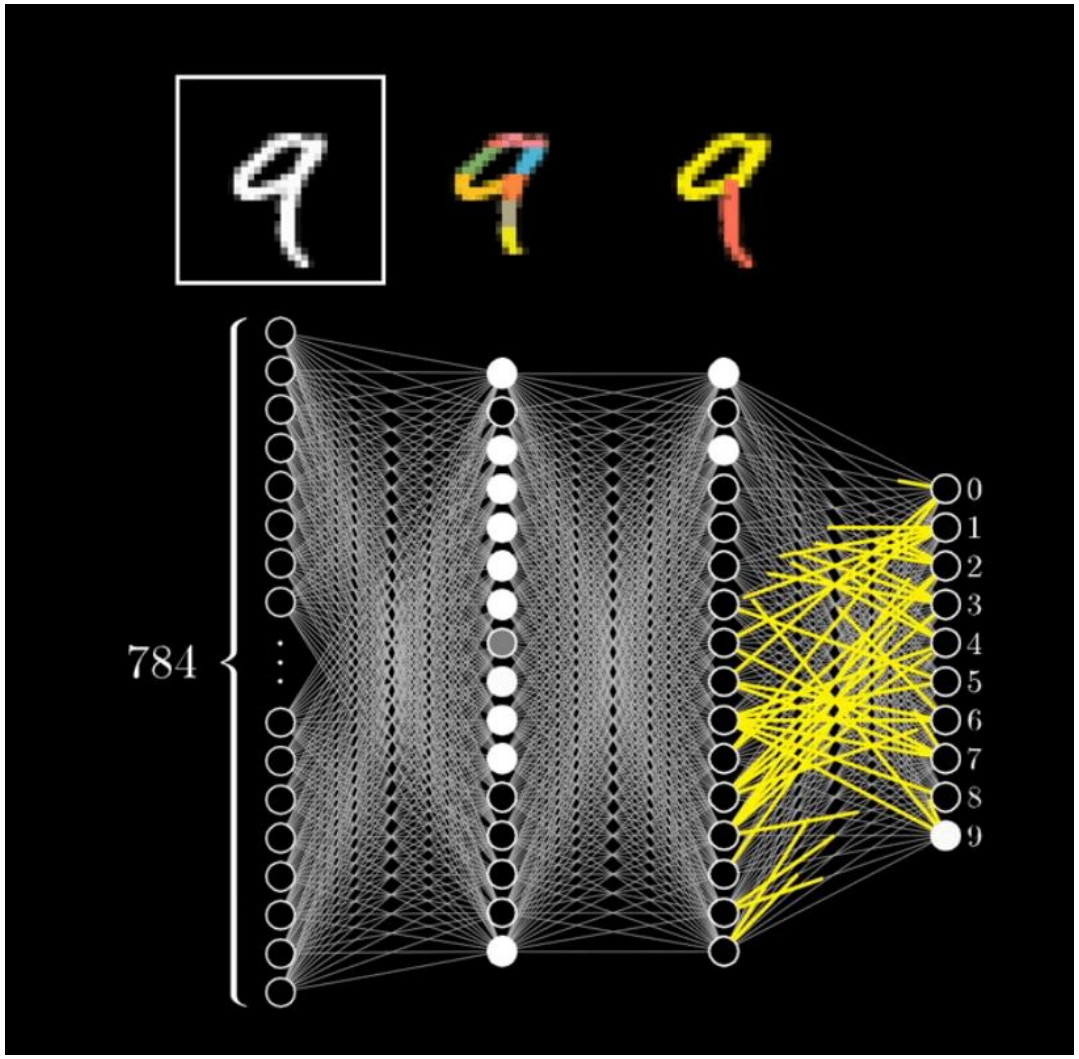
There are many “architectures” for Neural Networks. Many are for purpose-specific applications.

Image taken from:
[https://www.asimovinstitute.org/neural-network-zoo/](https://www.asimovinstitute.org/neural-network-zoo/)



# Representation + Evaluation + Optimization

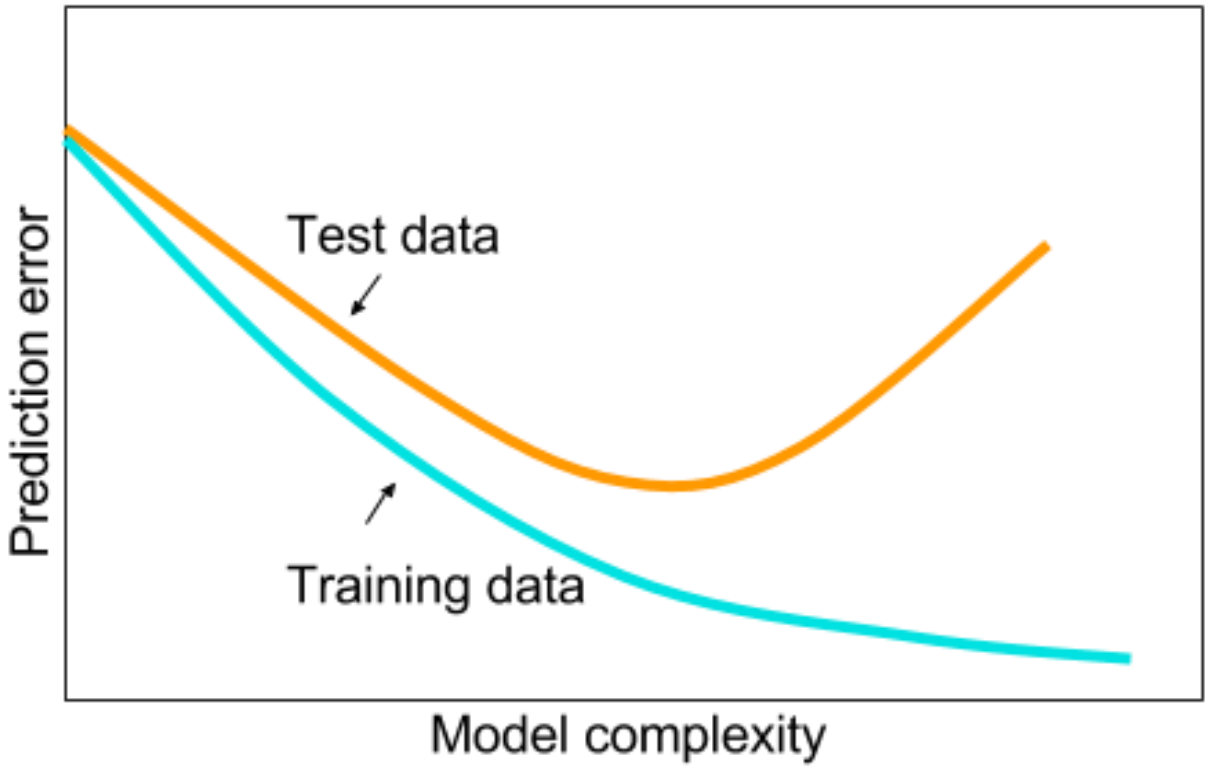
Often used for predicting **multiple** outputs.

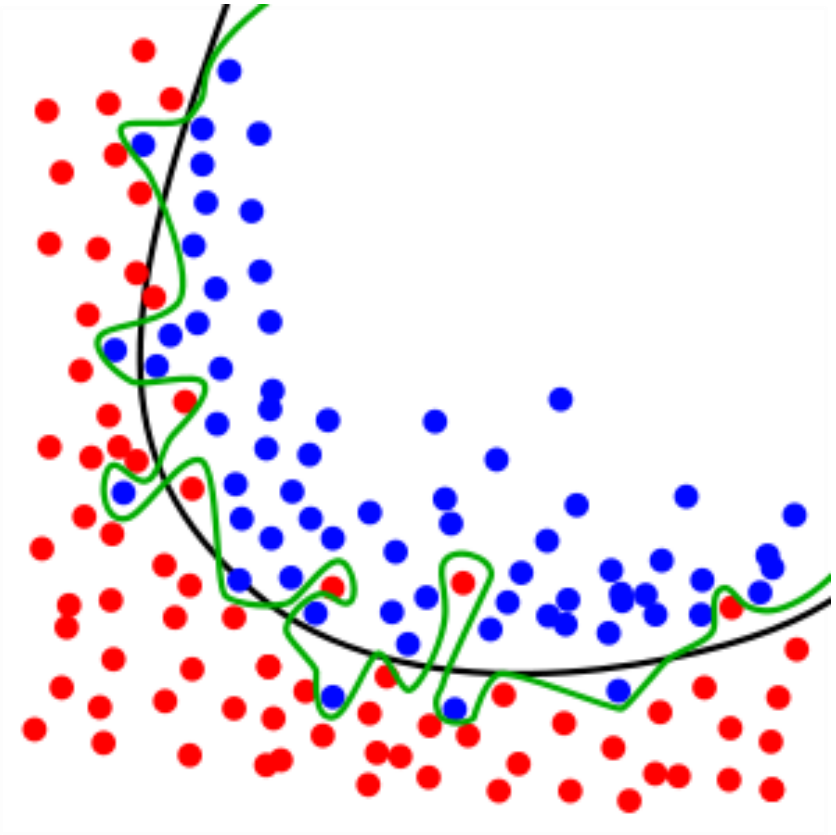


# Examples of Deep Learning Applications

In other words:
Unstructured and high-dimensional data.

# Beware of overfitting…

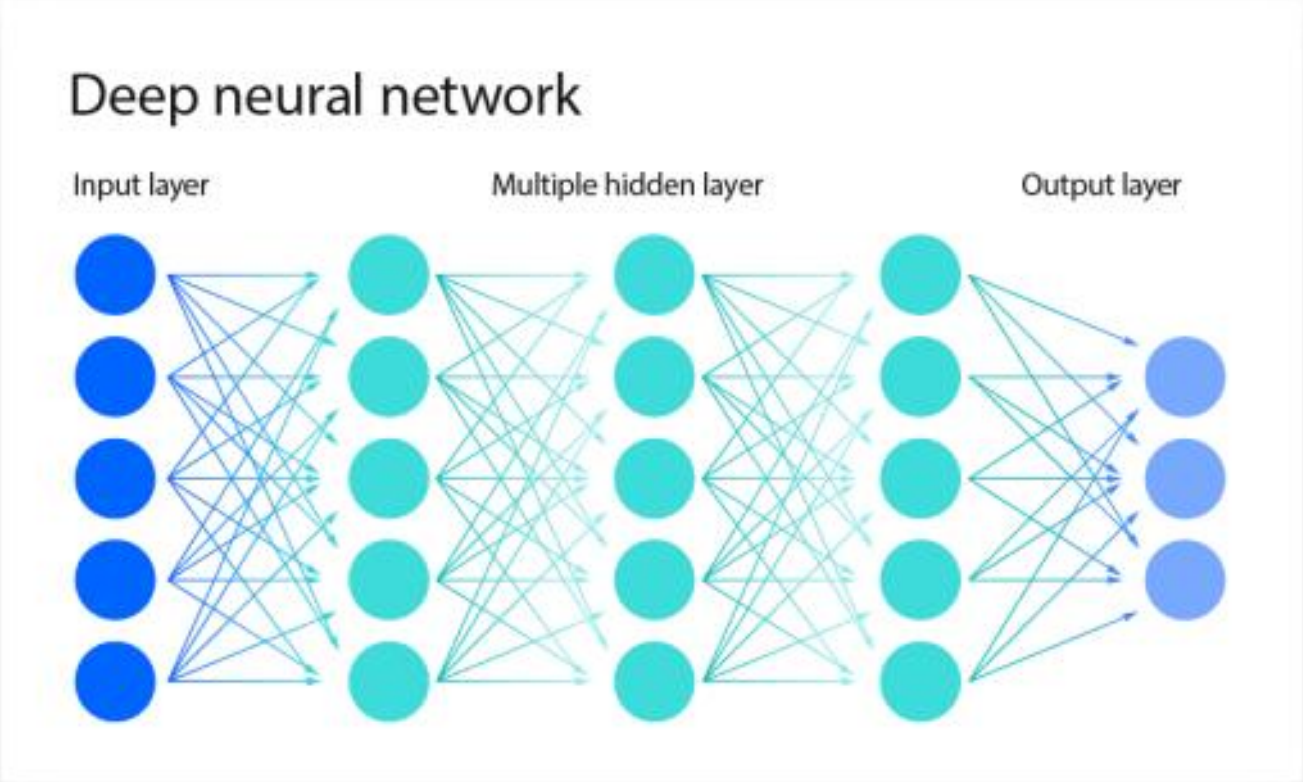


# Also…

THE #1 DATA SCIENTIST EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY MODEL'S TRAINING"
HEY! GET BACK TO WORK!
TRAINING!
OH. CARRY ON.

# Questions?


Deep neural network
Input layer
Multiple hidden layer
Output layer

# Word Embeddings:
## Deep Learning for Text Data

# What's a word embedding?

It is a vector of numbers meant to **<u>represent</u>** word **<u>meaning</u>**.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| man | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| woman | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| boy | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| girl | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| prince | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| princess | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| queen | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| king | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| monarch | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

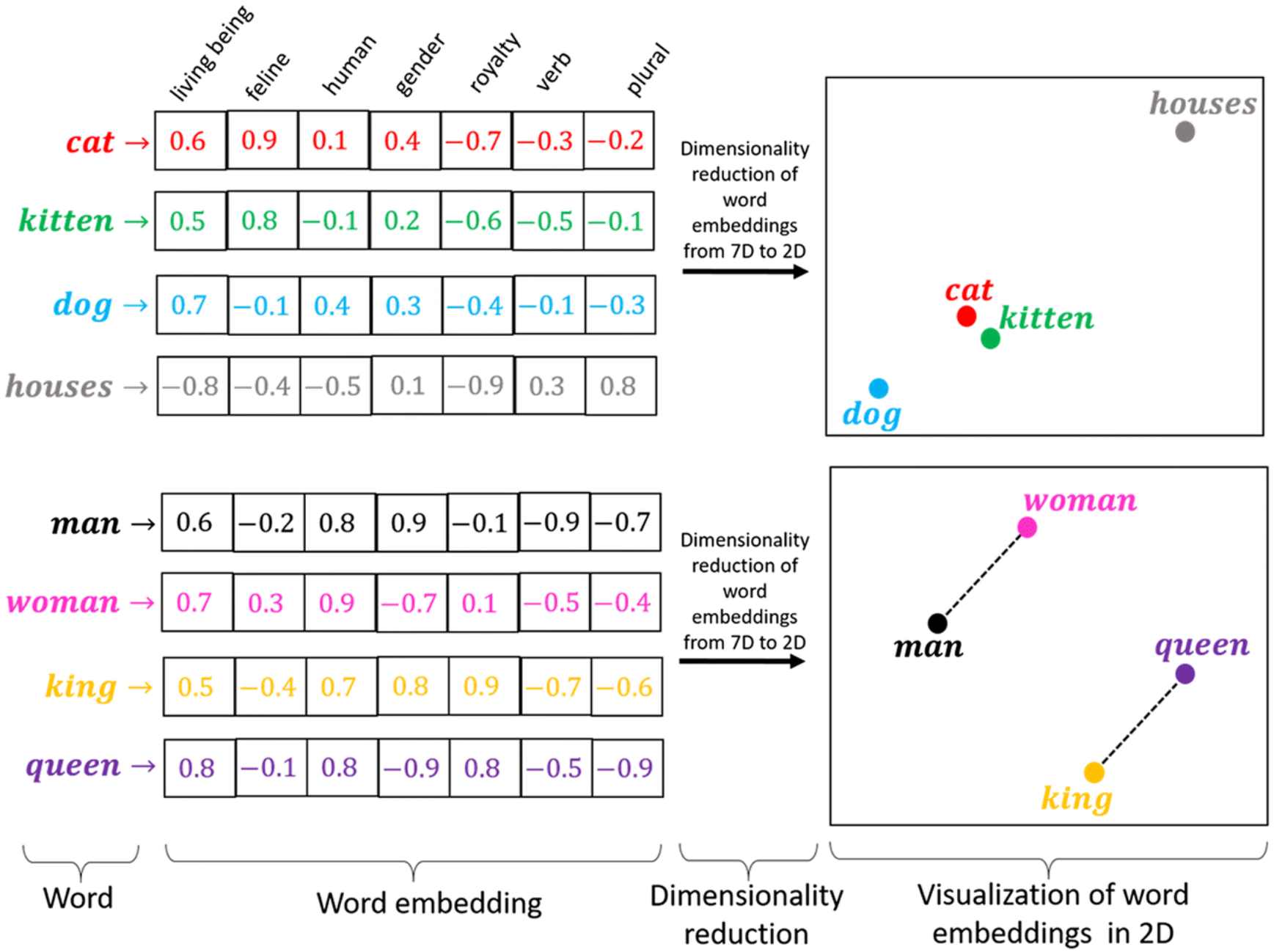
Let's move from **bag of words**…

To **word Embeddings**!

| | Femininity | Youth | Royalty |
|---|---|---|---|
| Man | 0 | 0 | 0 |
| Woman | 1 | 0 | 0 |
| Boy | 0 | 1 | 0 |
| Girl | 1 | 1 | 0 |
| Prince | 0 | 1 | 1 |
| Princess | 1 | 1 | 1 |
| Queen | 1 | 0 | 1 |
| King | 0 | 0 | 1 |
| Monarch | 0.5 | 0.5 | 1 |

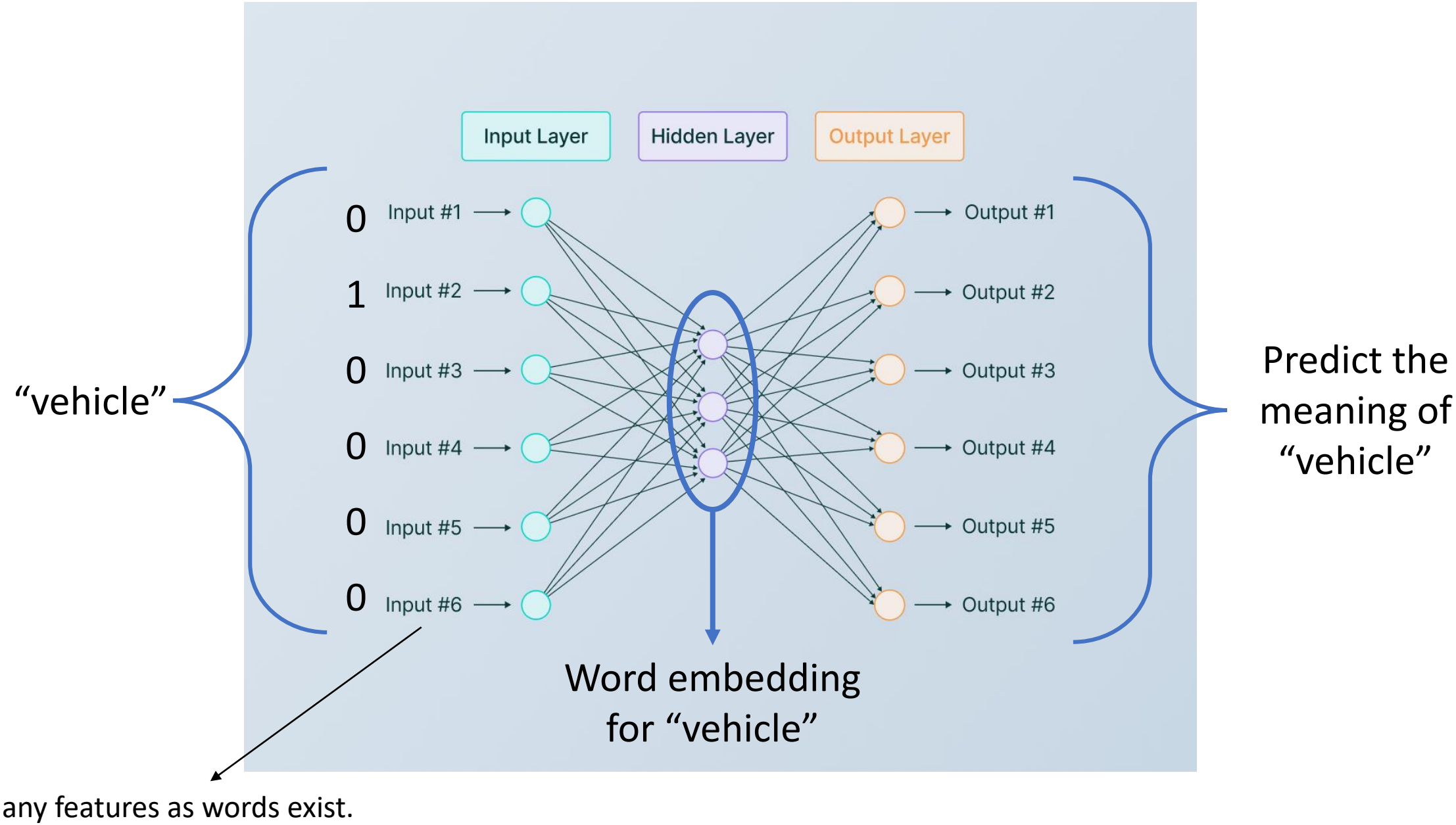
# Why are they useful?

To capture the **meaning** of words and their **relationships** with one another.



Can we learn **representations** of word meaning from data?

The hidden layers in a neural network are representations!


Input Layer
Hidden Layer
Output Layer
0
1
0
0
0
0
Input #1
Input #2
Input #3
Input #4
Input #5
Input #6
Output #1
Output #2
Output #3
Output #4
Output #5
Output #6
"vehicle"
Predict the meaning of "vehicle"
Word embedding for "vehicle"
As many features as words exist.

# Context ≈ Meaning

"I painted the bench ___red___."

"I painted the bench ___blue___."

"I painted the bench ___green___."

Based on their context, these words have a "similar" meaning.

# Context ≈ Meaning

"I painted the bench ___red___."

"I painted the bench ___blue___."

"I painted the bench ___green___."

Based on their context, these words have a "similar" meaning.

"I painted the bench ___**today**___."

With some caveats...

# Learning embeddings from data

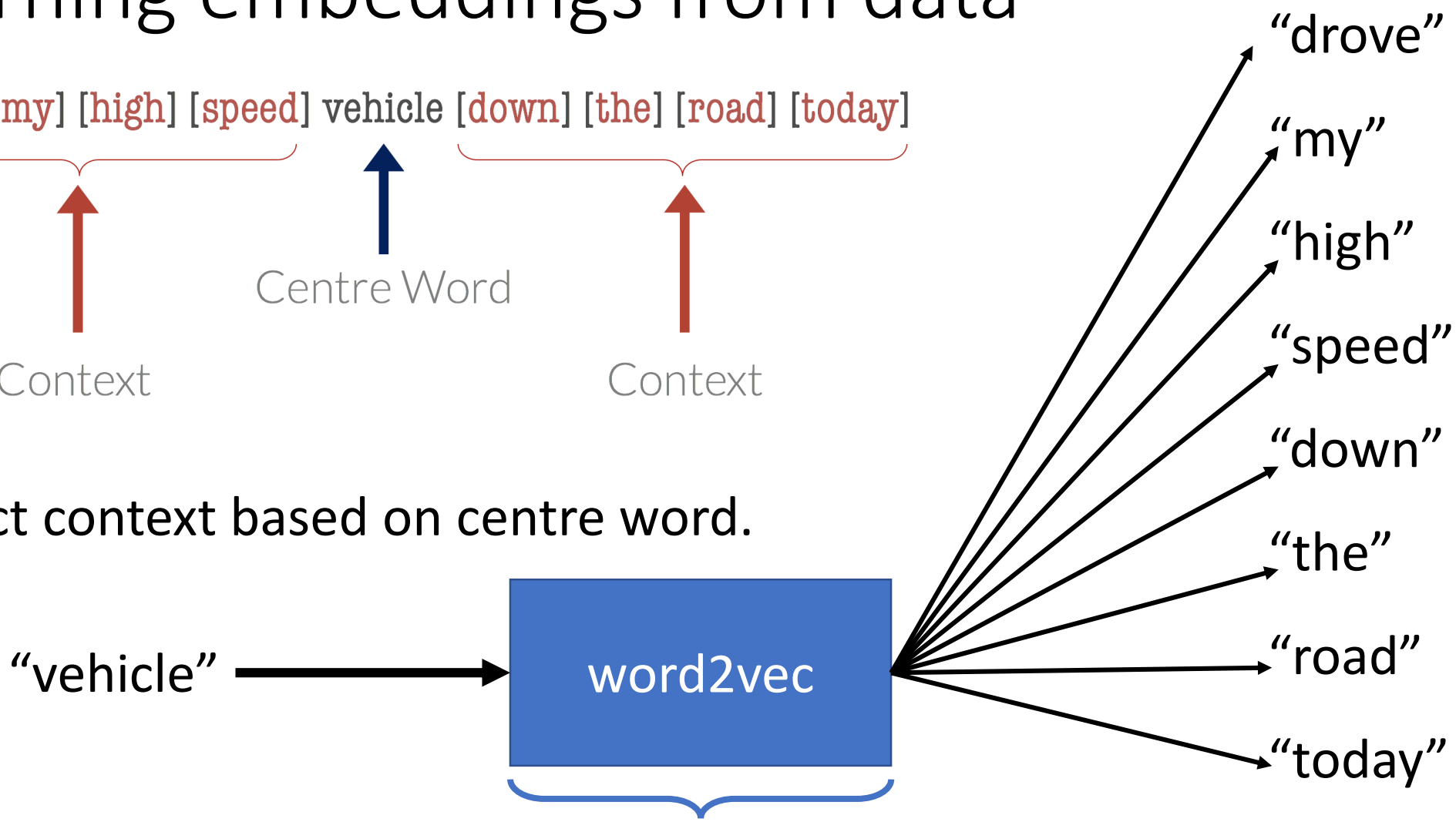
[drove] [my] [high] [speed] vehicle [down] [the] [road] [today]

Context

Centre Word

Context

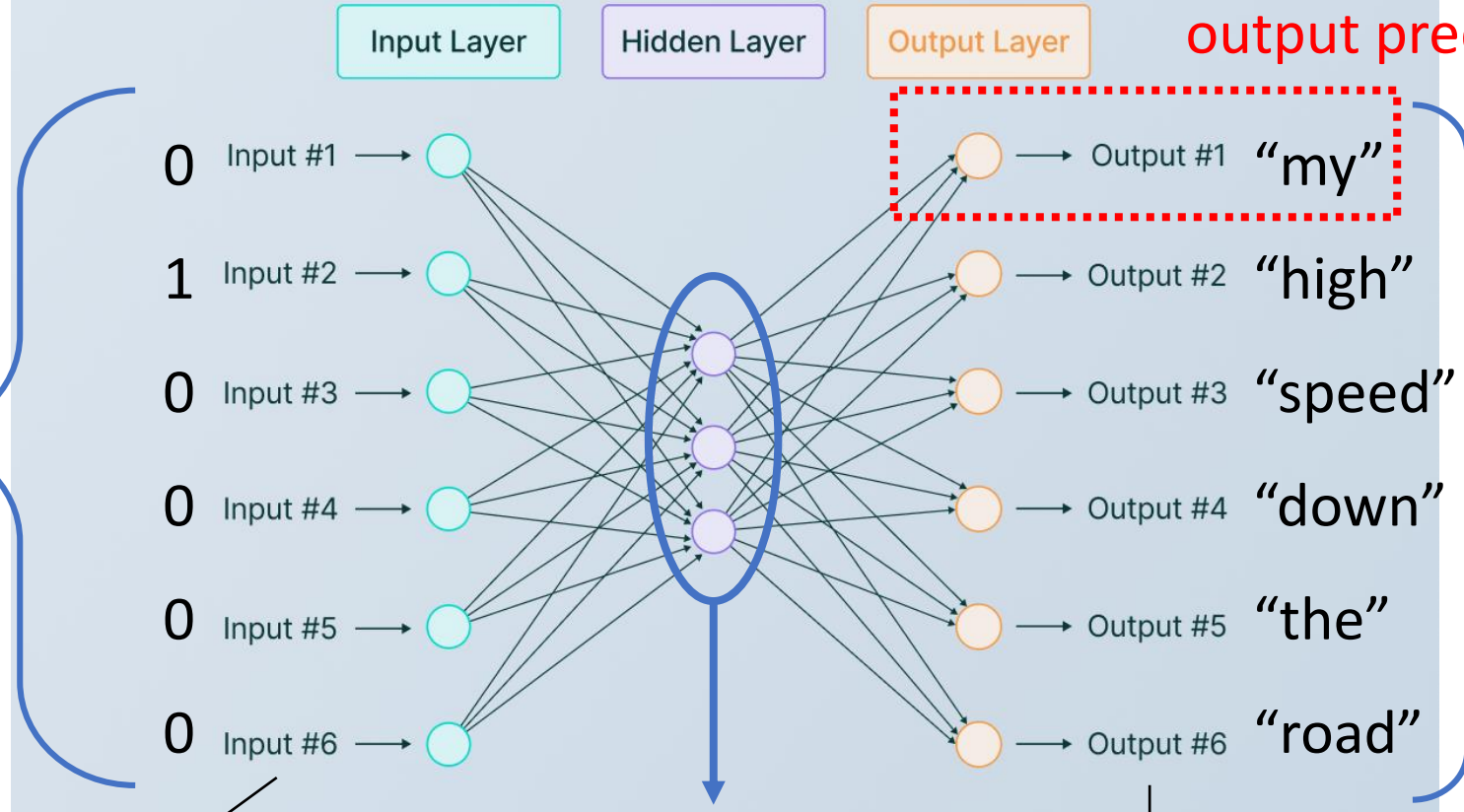
Predict context based on centre word.



Questions?

# word2vec

Let's zoom into a single output prediction




"vehicle"

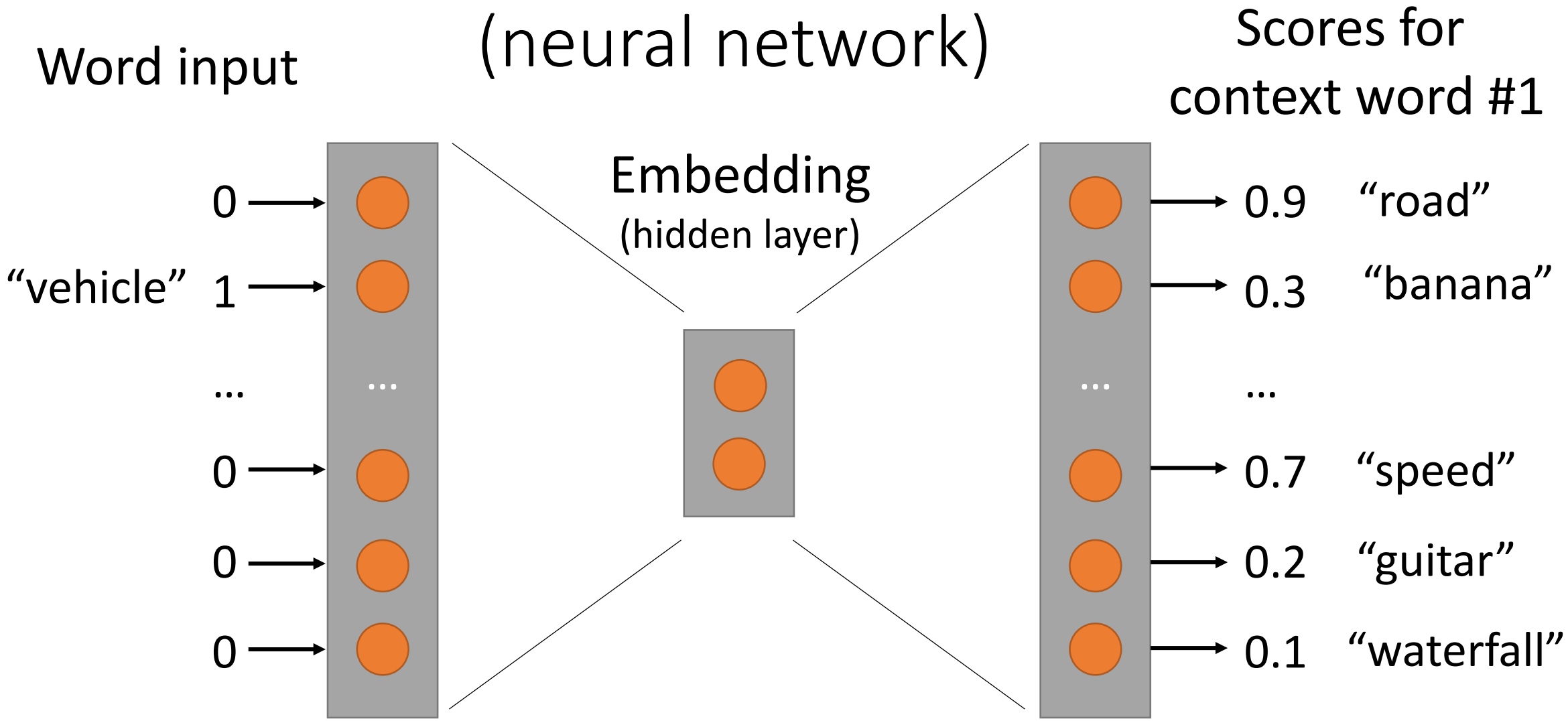
Predict the ~~meaning~~ context of "vehicle"

Word embedding for "vehicle"

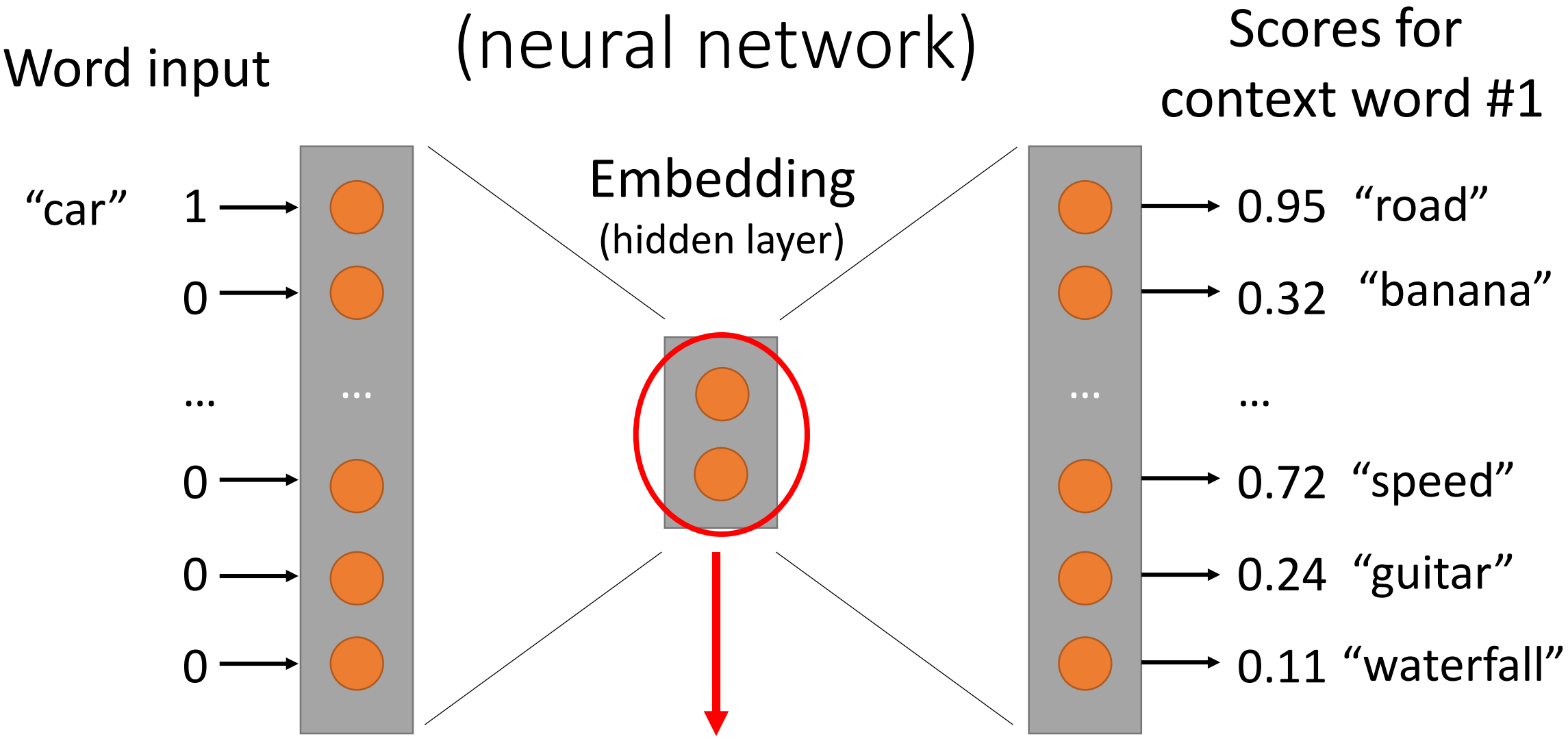
As many features as words exist.

One output for each word in the context window.

# word2vec (neural network)

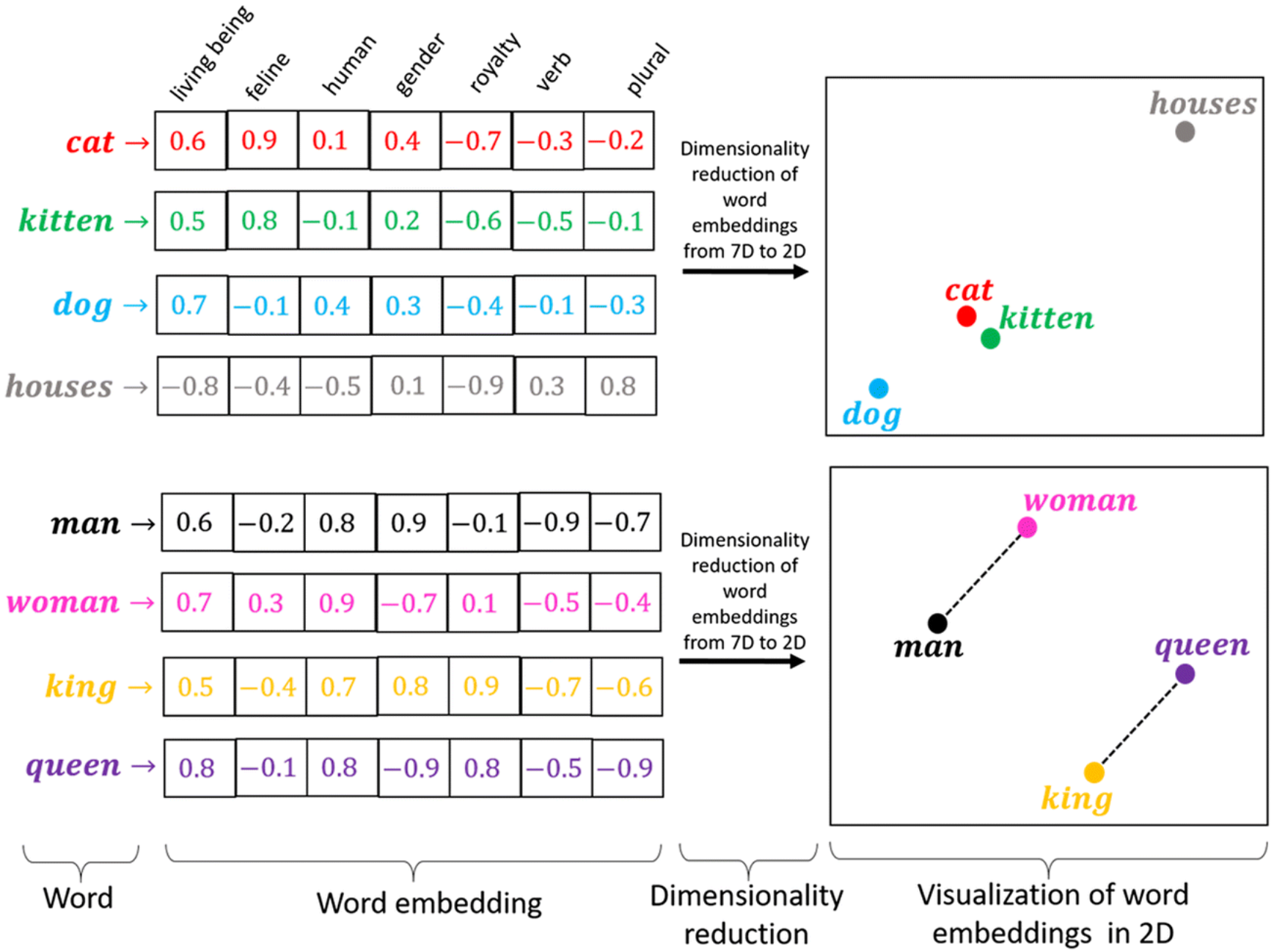


# word2vec (neural network)




Learn similar **representations** for words that appear in similar contexts!

# Questions?

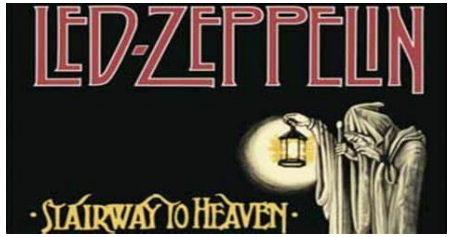

living being
feline
human
gender
royalty
verb
plural
cat → 0.6 0.9 0.1 0.4 −0.7 −0.3 −0.2
kitten → 0.5 0.8 −0.1 0.2 −0.6 −0.5 −0.1
dog → 0.7 −0.1 0.4 0.3 −0.4 −0.1 −0.3
houses → −0.8 −0.4 −0.5 0.1 −0.9 0.3 0.8
Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
man → 0.6 −0.2 0.8 0.9 −0.1 −0.9 −0.7
woman → 0.7 0.3 0.9 −0.7 0.1 −0.5 −0.4
king → 0.5 −0.4 0.7 0.8 0.9 −0.7 −0.6
queen → 0.8 −0.1 0.8 −0.9 0.8 −0.5 −0.9
Dimensionality reduction of word embeddings from 7D to 2D
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

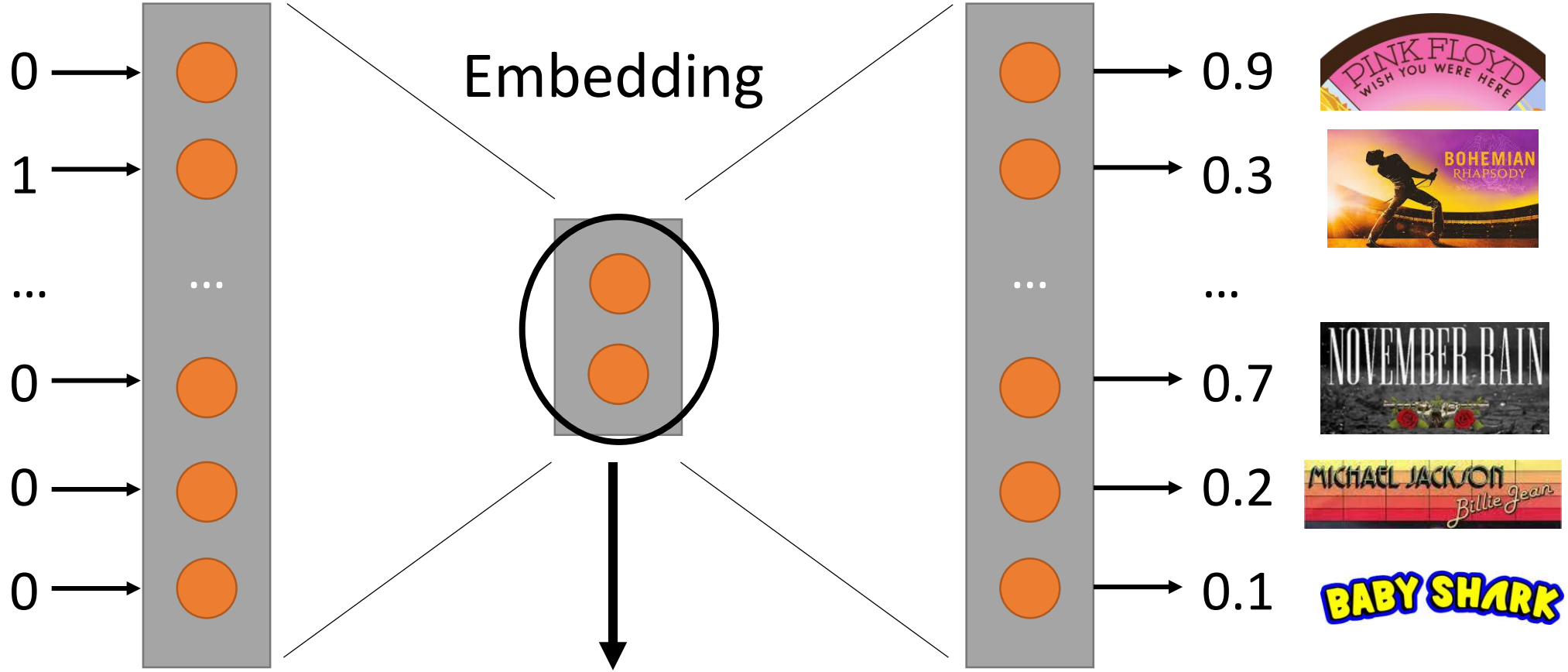
# Other interesting use cases

# Spotify song2vec

Song input

~~Word input~~




Playlist output

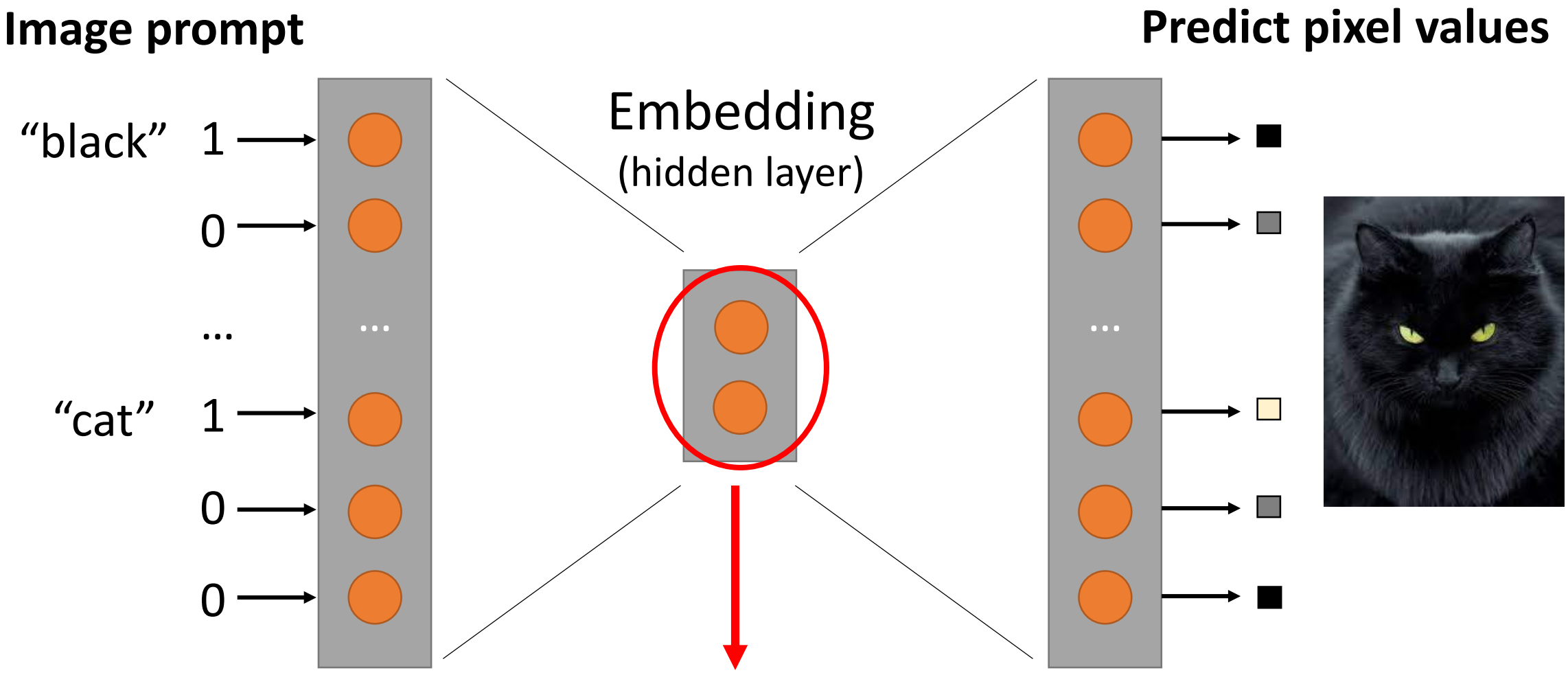
~~Context output~~

songs

playlists

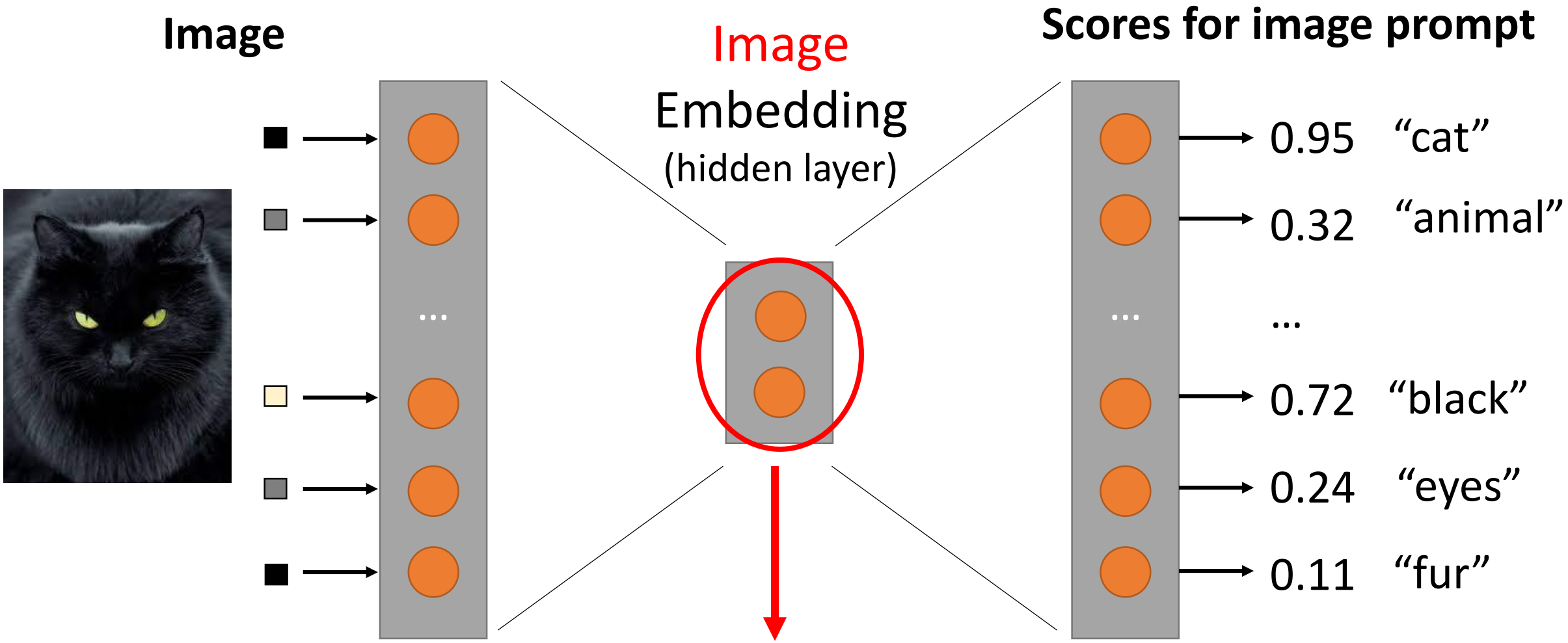
Learn similar **representations** for ~~words~~ that appear in similar ~~contexts~~!

# What about learning embeddings with images?

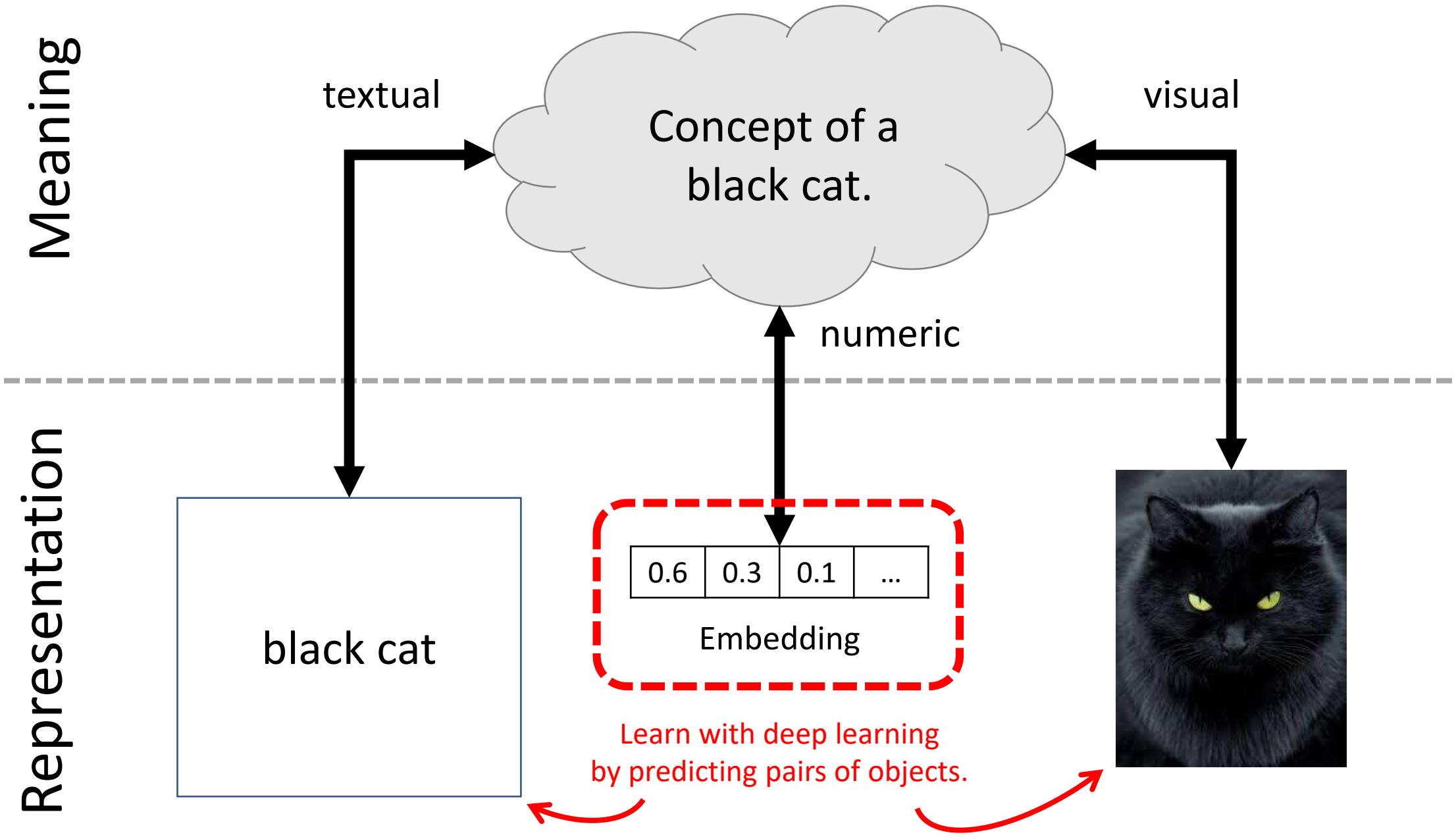



Learn word **representations** that capture the "meaning" of images.

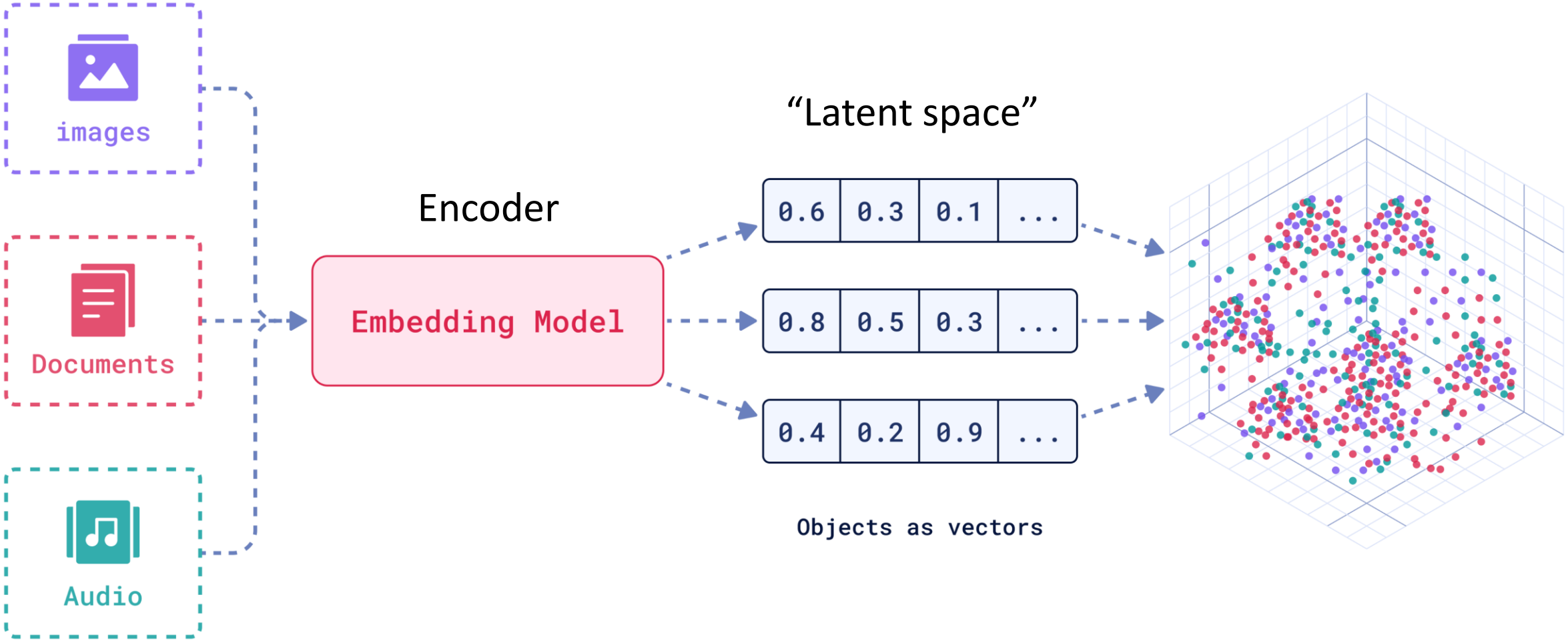
# Or embeddings for other unstructured data?

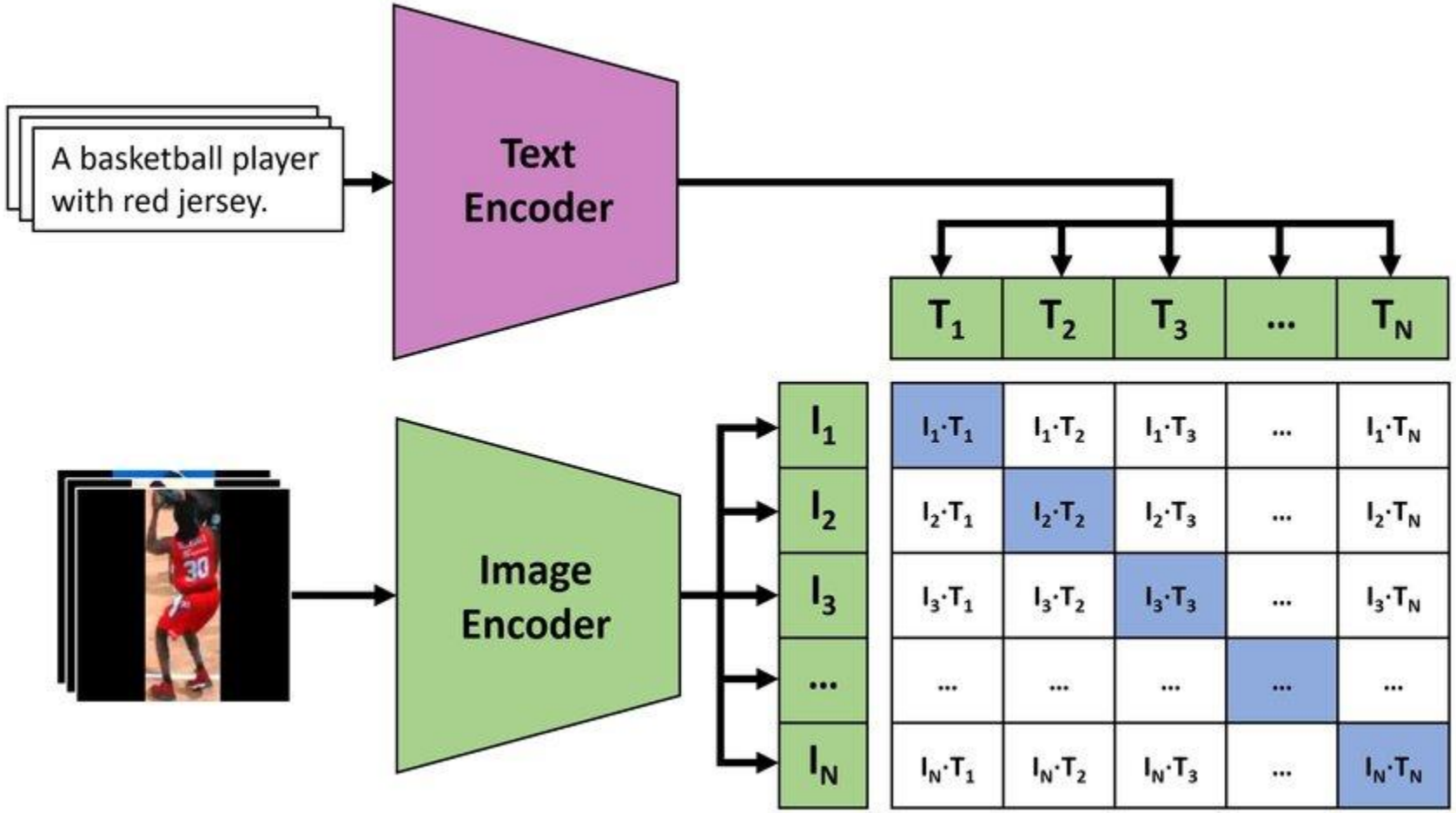



Learn IMAGE **representations** that match the image prompts.


Meaning
Representation
textual
visual
numeric
Concept of a black cat.
black cat
0.6
0.3
0.1
...
Embedding
Learn with deep learning by predicting pairs of objects.

# Unstructured data

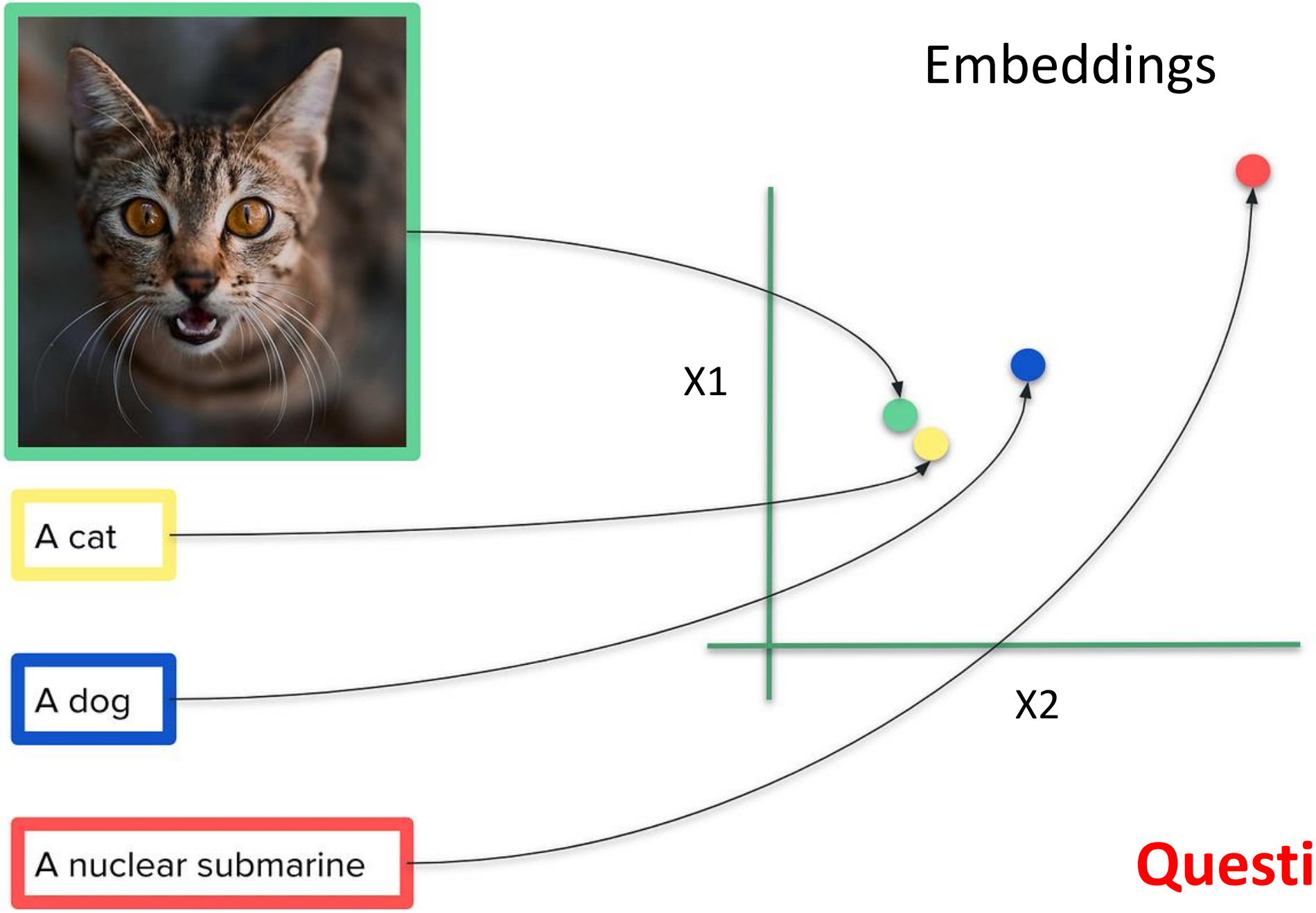





Learn embeddings for images and text that have:

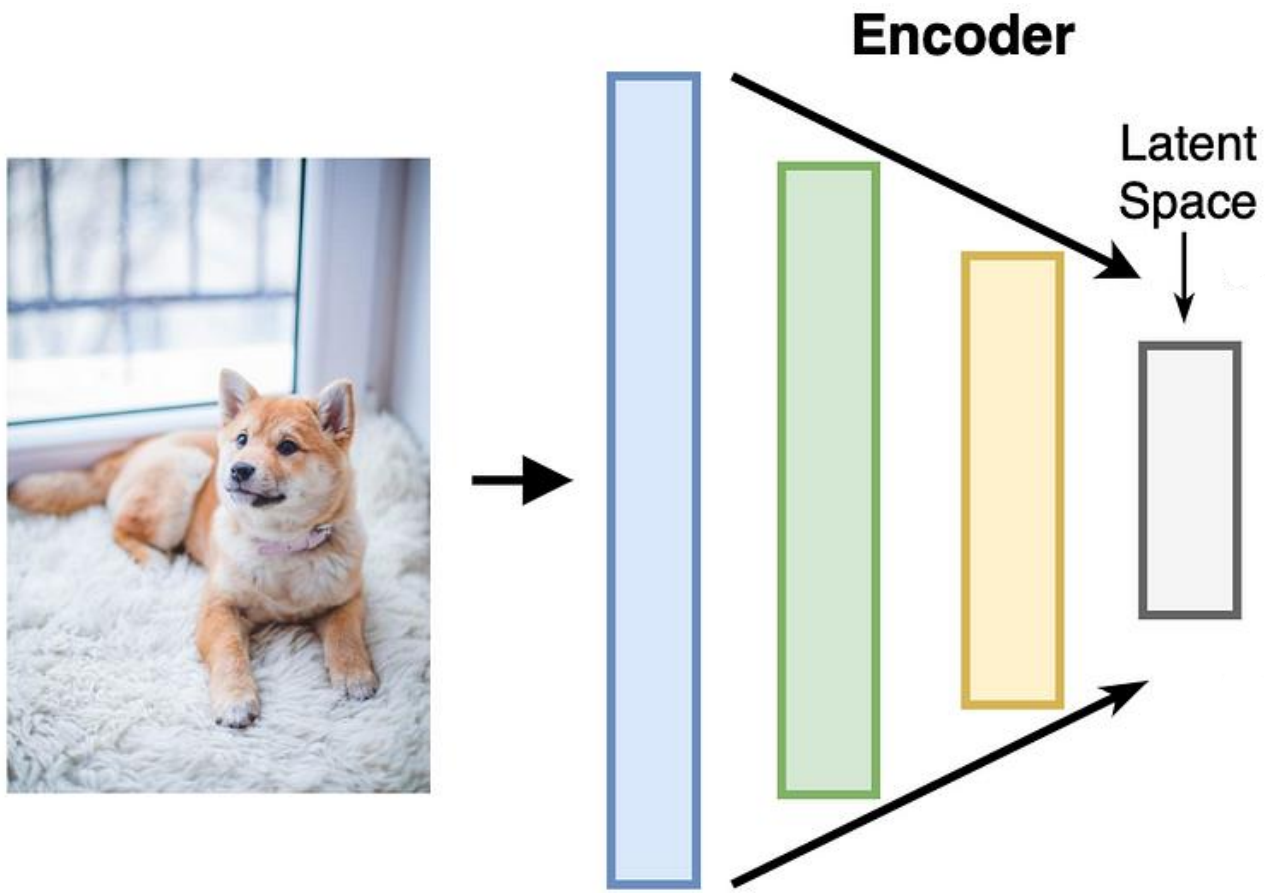
- High cosine similarity for matching pairs.
- Low cosine similarity for non-matching pairs.

# Embeddings




**Questions?**

We discussed transforming unstructured data into latent space (embeddings).



We discussed transforming unstructured data into latent space (embeddings).

We could also learn how to go from latent space back to the transformed data.

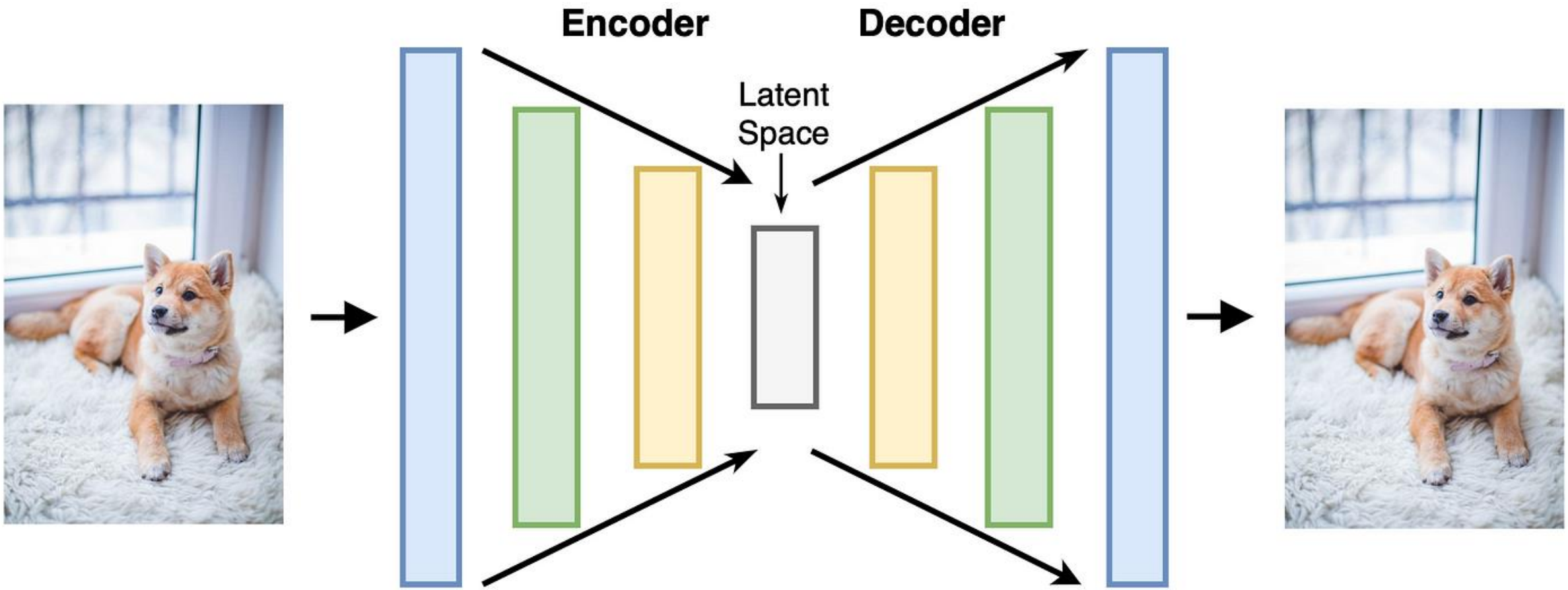


Image Embedding

Image Encoder

Image Decoder

"A sleek **black cat** with velvety fur and piercing yellow-green eyes stares intently, exuding an air of mystery and quiet confidence. Its dark, elegant coat contrasts sharply with its vivid gaze, creating a captivating and powerful presence."

Text Embedding

Text Encoder

High Cosine Similarity

Image Embedding

Image Encoder

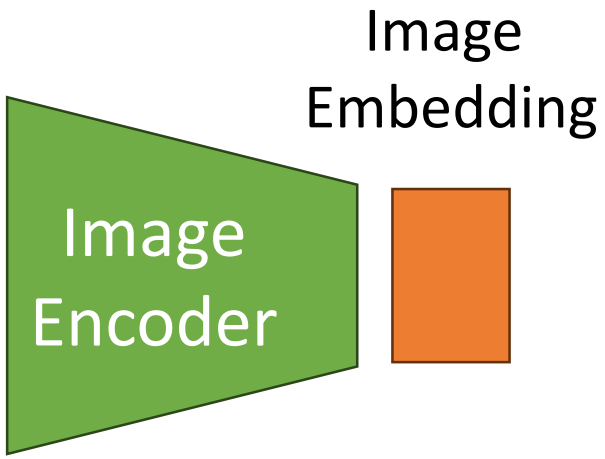
Image Decoder

Text Embedding

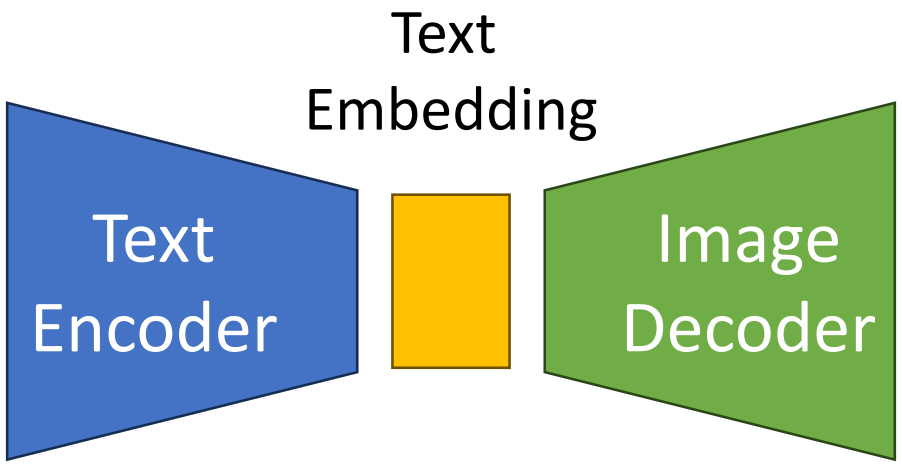
Similar values!

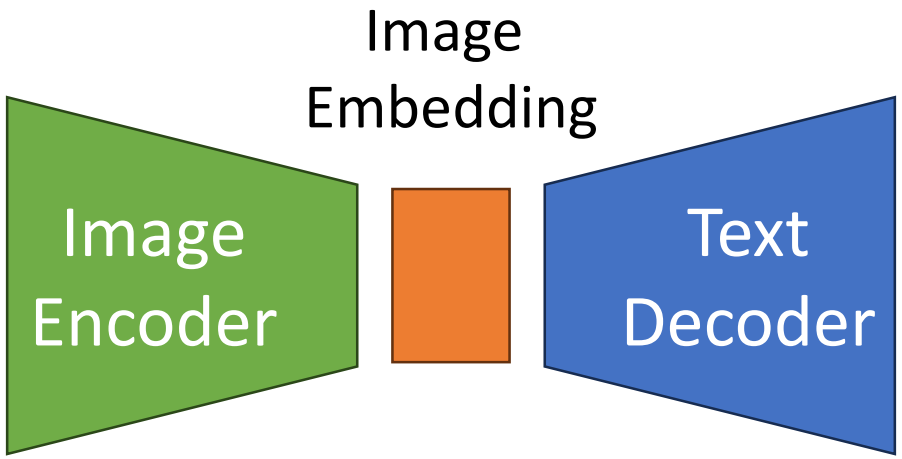
"A sleek **black cat** with velvety fur and piercing yellow-green eyes stares intently, exuding an air of mystery and quiet confidence. Its dark, elegant coat contrasts sharply with its vivid gaze, creating a captivating and powerful presence."

Text Encoder



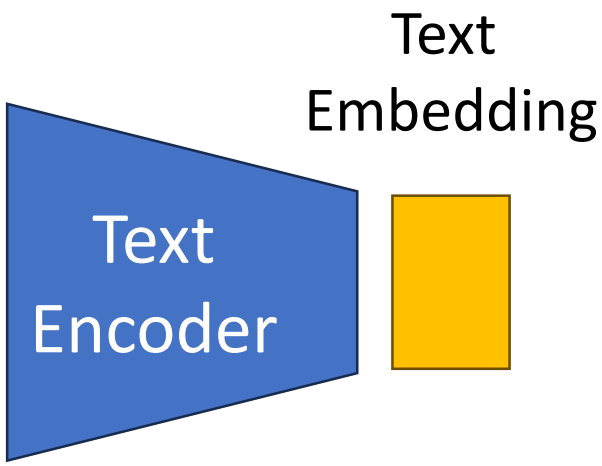

"A sleek **black cat** with velvety fur and piercing yellow-green eyes stares intently, exuding an air of mystery and quiet confidence. Its dark, elegant coat contrasts sharply with its vivid gaze, creating a captivating and powerful presence."



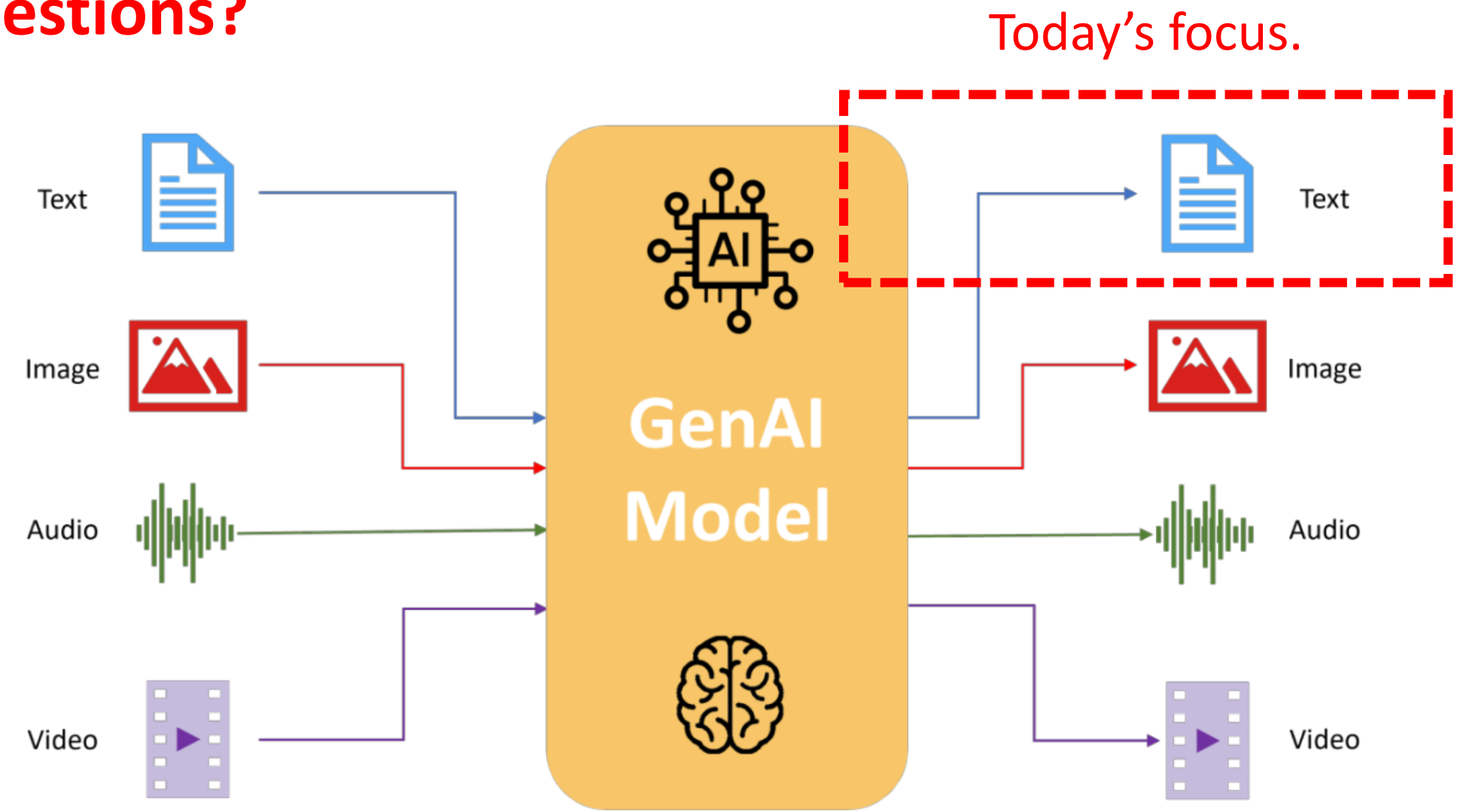



"A sleek **black cat** with velvety fur and bright, luminous green eyes gazes intently, exuding a sense of calm and quiet confidence. Its elegant posture and soft, shadowy coat create a serene yet powerful presence, while its vivid eyes draw attention with an almost mystical glow."

"A sleek **black cat** with velvety fur and piercing yellow-green eyes stares intently, exuding an air of mystery and quiet confidence. Its dark, elegant coat contrasts sharply with its vivid gaze, creating a captivating and powerful presence."



# Questions?


Today's focus.
Text
Image
Audio
Video
AI
GenAI
Model
Text
Image
Audio
Video

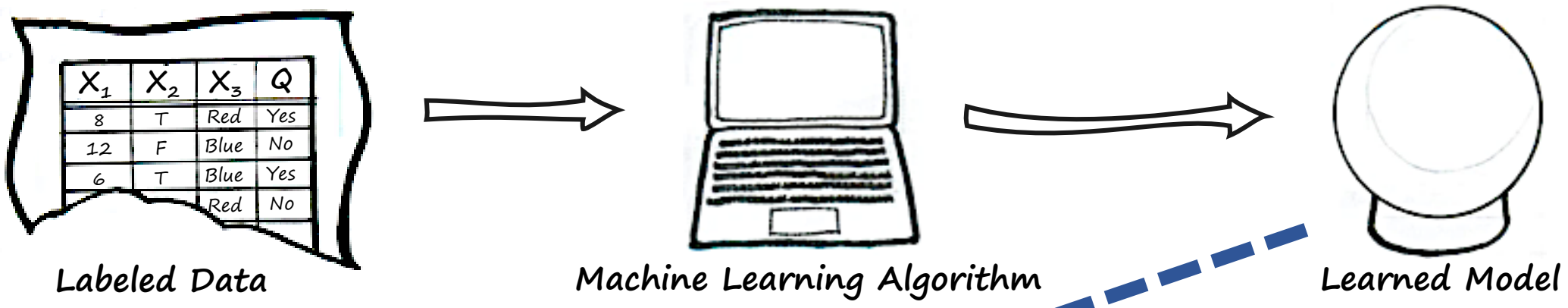
Is ChatGPT a Large Language Model (LLM)?

It is actually not!

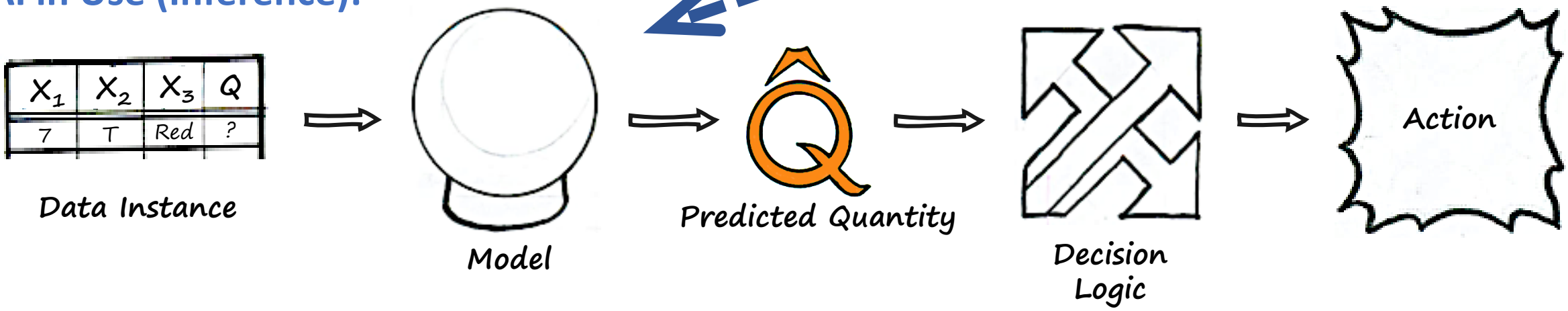
# Framework #3: AI in Use vs. Machine Learning

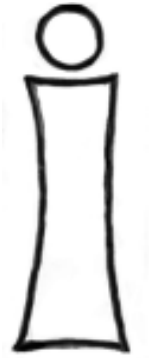
Let's use our framework to understand ChatGPT.

## Machine Learning:

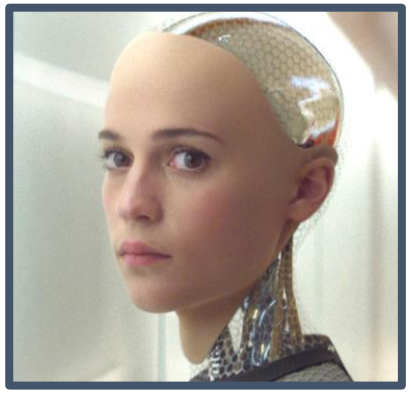



## AI in Use (Inference):

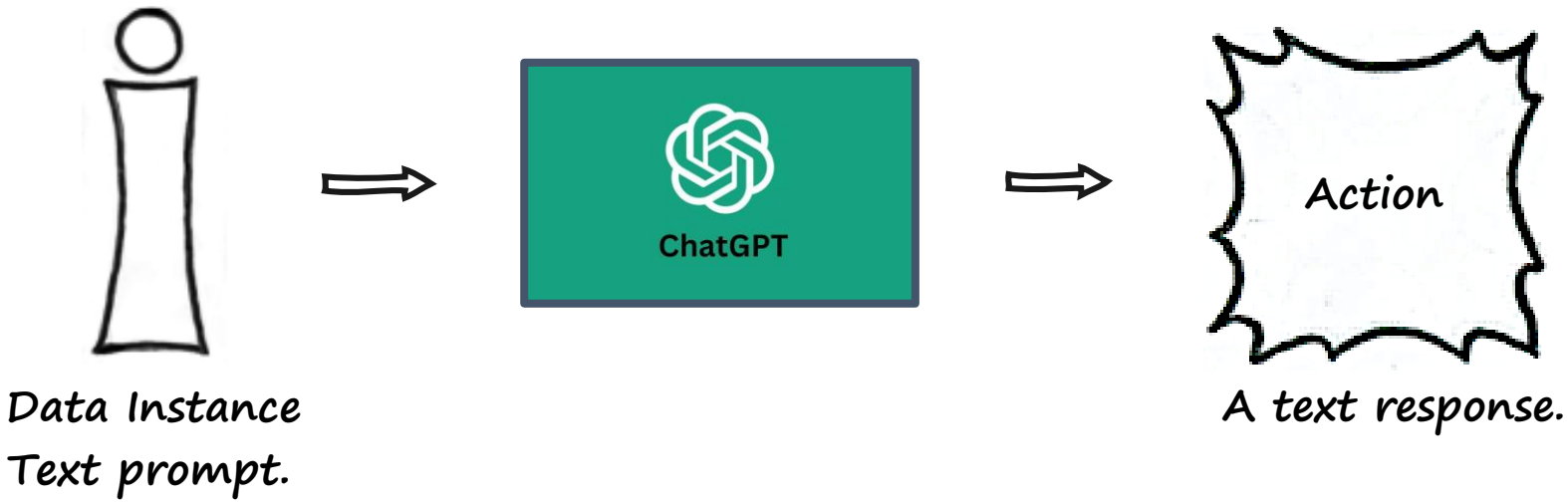




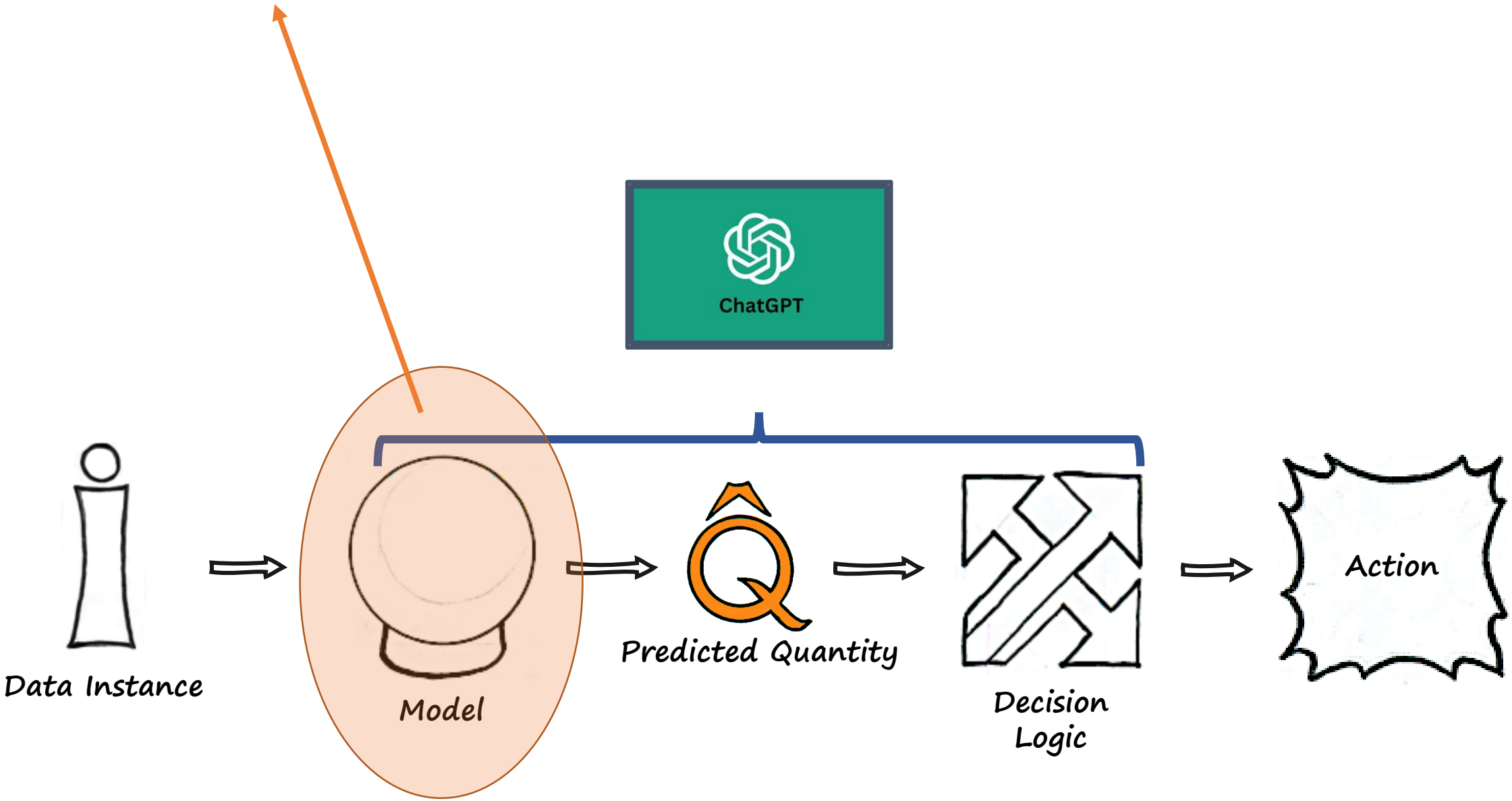
Data Instance
Example: a customer



Example: give retention offer
(or not)


ChatGPT
Action
Data Instance
Text prompt.
A text response.

The large language model (LLM), such as GPT 3



# GPT stands for:

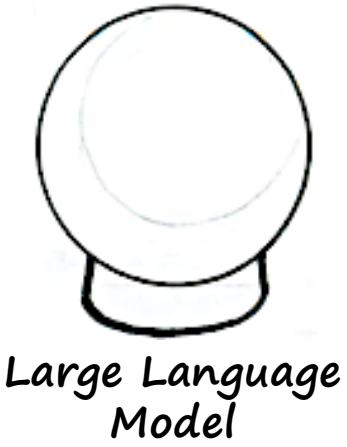
Generative Pre-trained Transformer

# GPT stands for:

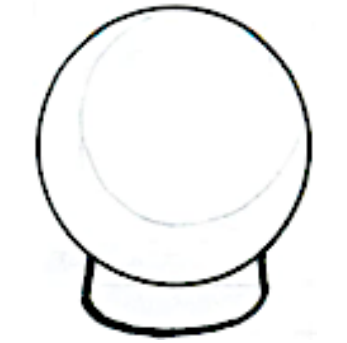
Generative Pre-trained Transformer

Let’s discuss what each of these words mean.

What’s the instance?

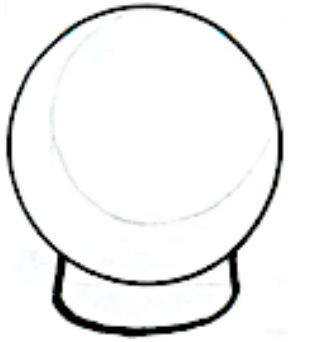


A sequence of text:
“The quick brown”




What do we want to predict?

A sequence of text:
“The quick brown”




The next word:
“fox”

What are the features?

# Generative Pre-trained Transformer

GPT-3 Codex

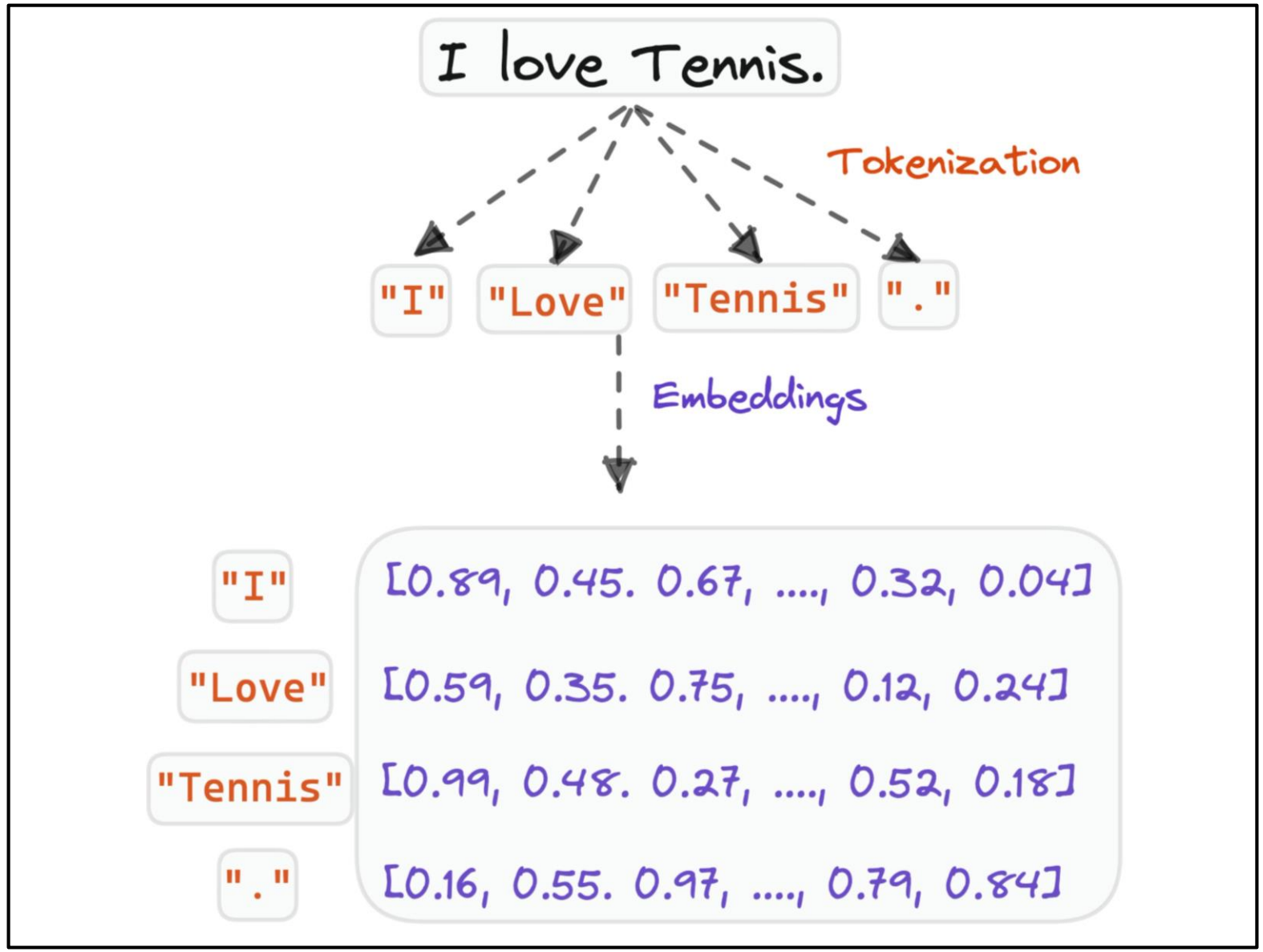
Finding the right balance is crucial to ensure that the tokenizer effectively represents the text while keeping computational efficiency in check. GPT uses a custom tokenizer, and you can even try out how it works.

Clear Show example

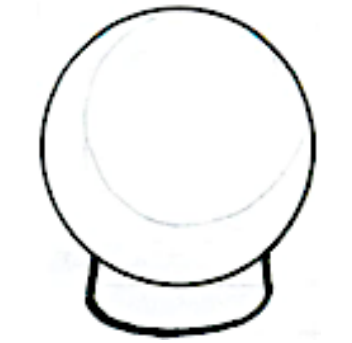
| Tokens | Characters |
| --- | --- |
| 41 | 214 |

Finding the right balance is crucial to ensure that the tokenizer effectively represents the text while keeping computational efficiency in check. GPT uses a custom tokenizer, and you can even try out how it works.

TEXT TOKEN IDS


I love Tennis.
Tokenization
"I"
"Love"
"Tennis"
"."
Embeddings
"I"
[0.89, 0.45. 0.67, ...., 0.32, 0.04]
"Love"
[0.59, 0.35. 0.75, ...., 0.12, 0.24]
"Tennis"
[0.99, 0.48. 0.27, ...., 0.52, 0.18]
"."
[0.16, 0.55. 0.97, ...., 0.79, 0.84]

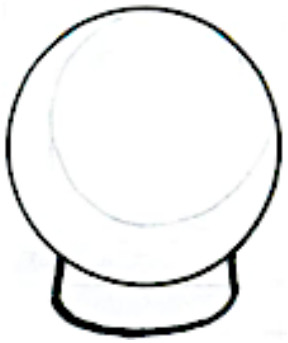
| TK | … | T4 | T3 | T2 | T1 |
|---|---|---|---|---|---|
| Empty | … | Empty | “The” | “quick” | “brown” |




The next token: “fox”

What is the model’s prediction?

| TK | … | T4 | T3 | T2 | T1 |
|---|---|---|---|---|---|
| Empty | … | Empty | “The” | “quick” | “brown” |

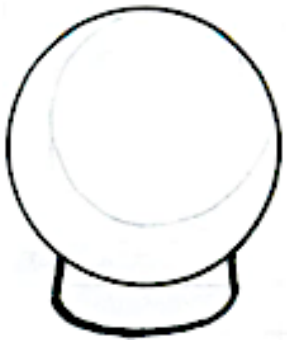



| Token | Prob. |
|---|---|
| “fox” | 70% |
| “dog” | 10% |
| “jumps” | 3% |
| … | |
| “rainbow” | 0.00001% |

One probability for each possible token.

And we use some decision logic to pick a token.

| TK | … | T4 | T3 | T2 | T1 |
|---|---|---|---|---|---|
| Empty | … | Empty | “The” | “quick” | “brown” |



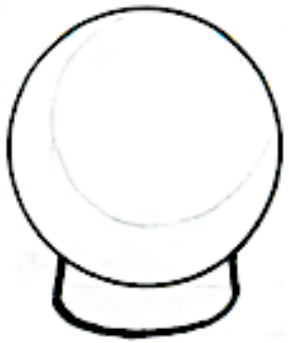

| Token | Prob. |
|---|---|
| “fox” | 70% |
| “dog” | 10% |
| “jumps” | 3% |
| … | |
| “rainbow” | 0.00001% |

One probability for each possible token.

# What do you think happens next?

| TK | … | T3 | T2 | T1 |
|---|---|---|---|---|
| … | … | “The” | “quick” | “brown” |

Data Instance



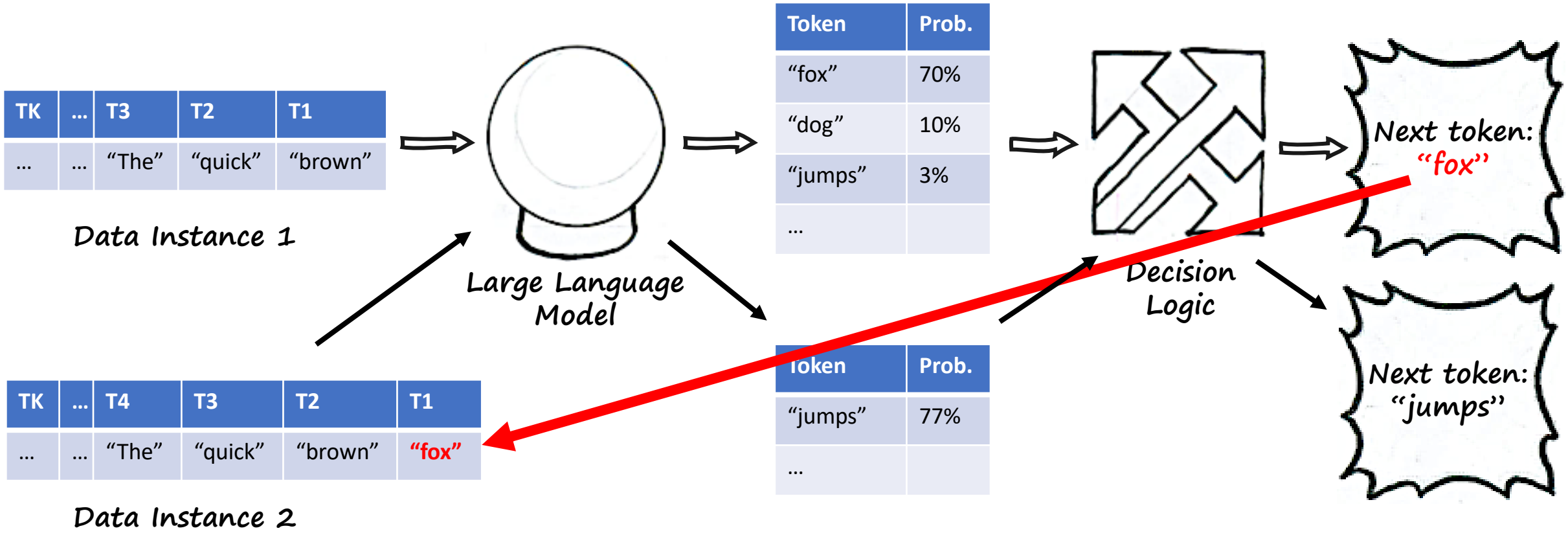
Large Language Model

| Token | Prob. |
|---|---|
| “fox” | 70% |
| “dog” | 10% |
| “jumps” | 3% |
| … | |

Predicted Quantity

Decision Logic

Next token: “fox”


TK
...
T3
T2
T1
...
...
"The"
"quick"
"brown"
Data Instance 1
Large Language Model
Token
Prob.
"fox"
70%
"dog"
10%
"jumps"
3%
...
Decision Logic
Next token: "fox"
Next token: "jumps"
Token
Prob.
"jumps"
77%
...
TK
...
T4
T3
T2
T1
...
...
"The"
"quick"
"brown"
"fox"
Data Instance 2

# Questions?

| TK | … | T3 | T2 | T1 |
|---|---|---|---|---|
| … | … | “The” | “quick” | “brown” |

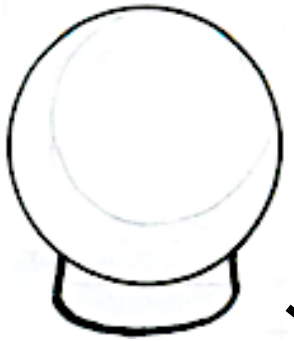
Data Instance 1

| TK | … | T4 | T3 | T2 | T1 |
|---|---|---|---|---|---|
| … | … | “The” | “quick” | “brown” | “fox” |

Data Instance 2

| TK | … | T5 | T4 | T3 | T2 | T1 |
|---|---|---|---|---|---|---|
| … | … | “The” | “quick” | “brown” | “fox” | “jumps” |

Data Instance 3



Large Language Model

| Token | Prob. |
|---|---|
| “fox” | 70% |
| “dog” | 10% |
| “jumps” | 3% |
| … | |

| Token | Prob. |
|---|---|
| “jumps” | 77% |
| … | |

| Token | Prob. |
|---|---|
| “over” | 82% |
| … | |

Decision Logic

Next token: “fox”

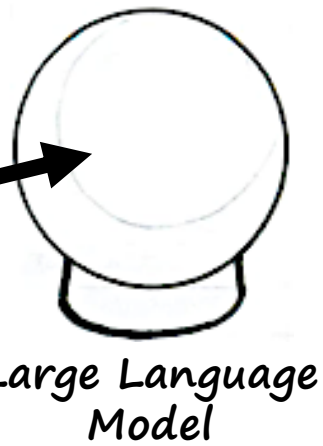
Next token: “jumps”

Next token: “over”

And this continues until we get the End-Of-Transmission token.

# Let’s talk more about the implantation details.

What’s inside this thing?



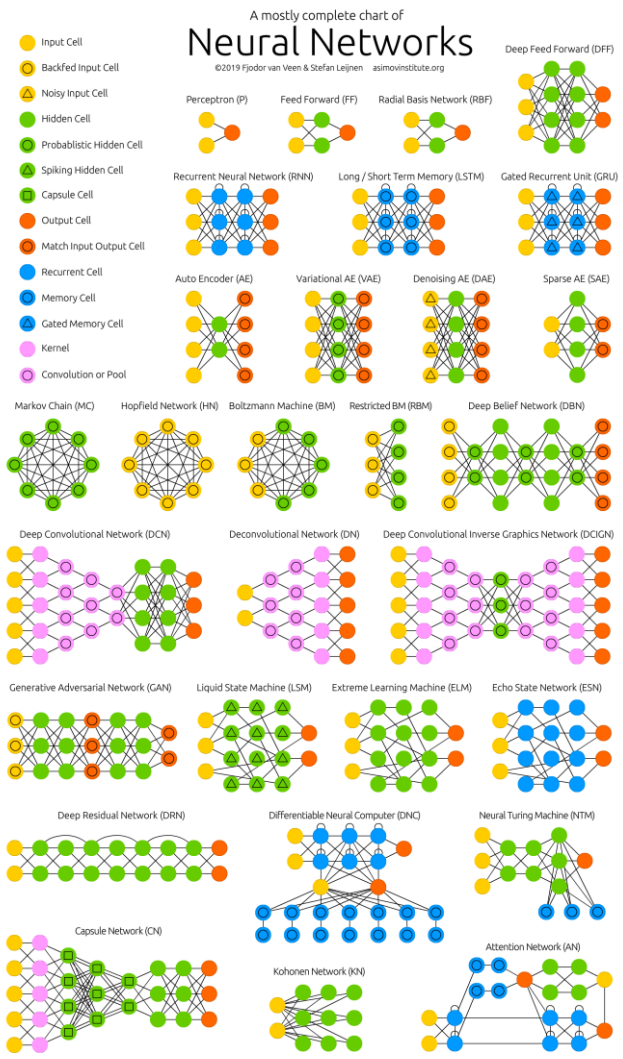
Representation + Evaluation + Optimization

Generative Pre-trained Transformer

# Representation + Evaluation + Optimization

There are many “architectures” for neural networks, each designed for purpose-specific applications.

Image taken from:
[https://www.asimovinstitute.org/neural-network-zoo/](https://www.asimovinstitute.org/neural-network-zoo/)



Published in NeurIPS conference in 2017.

**Ashish Vaswani***
Google Brain
avaswani@google.com

**Llion Jones***
Google Research
llion@google.com

The dominant sequ
convolutional neur
performing models
mechanism. We p
based solely on atte
entirely. Experime
be superior in qual
less time to train.
to-German translat
ensembles, by over
our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

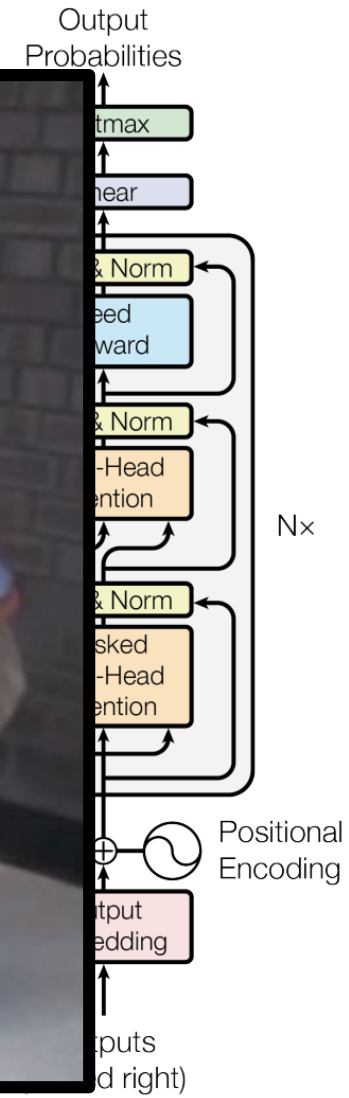
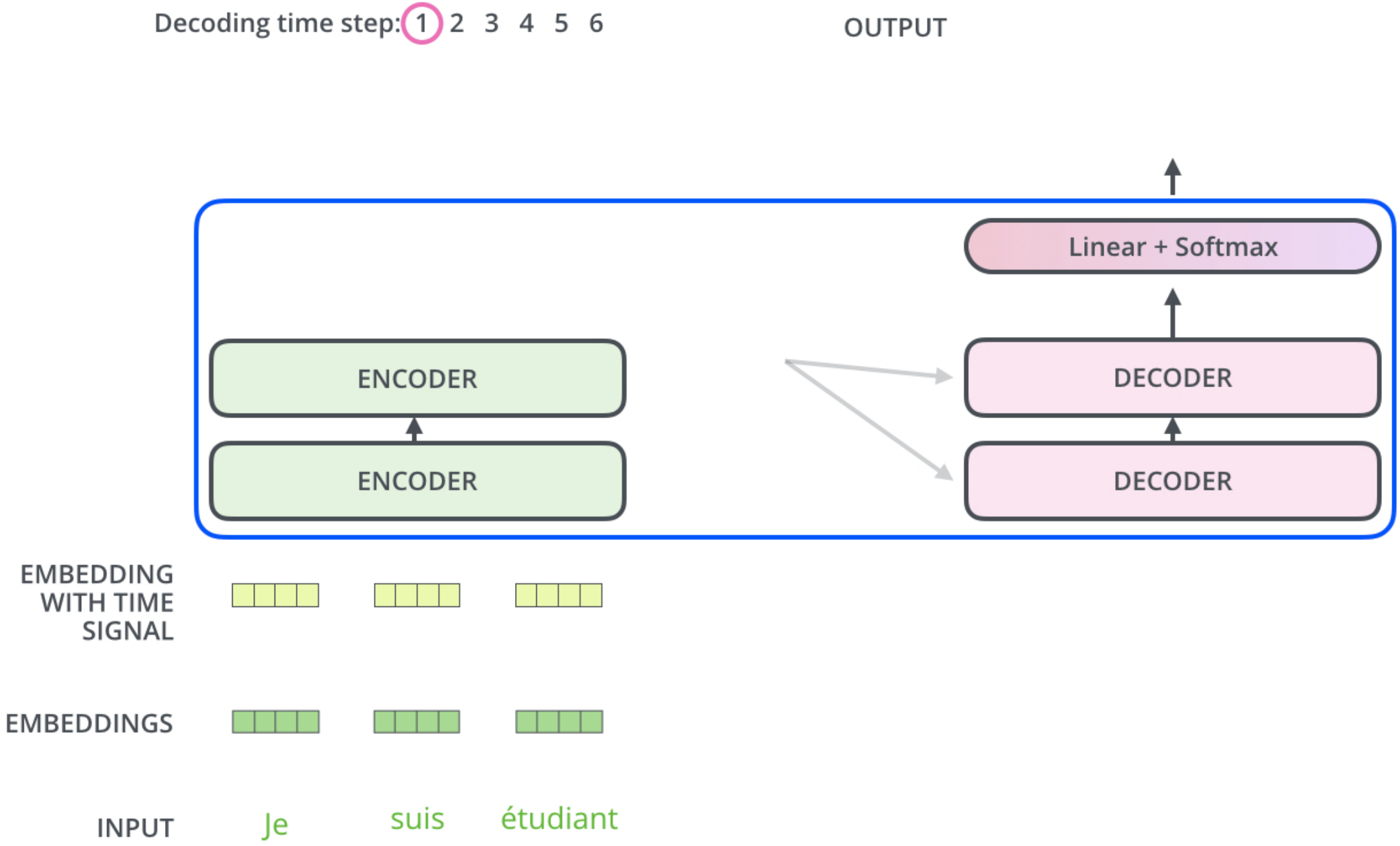
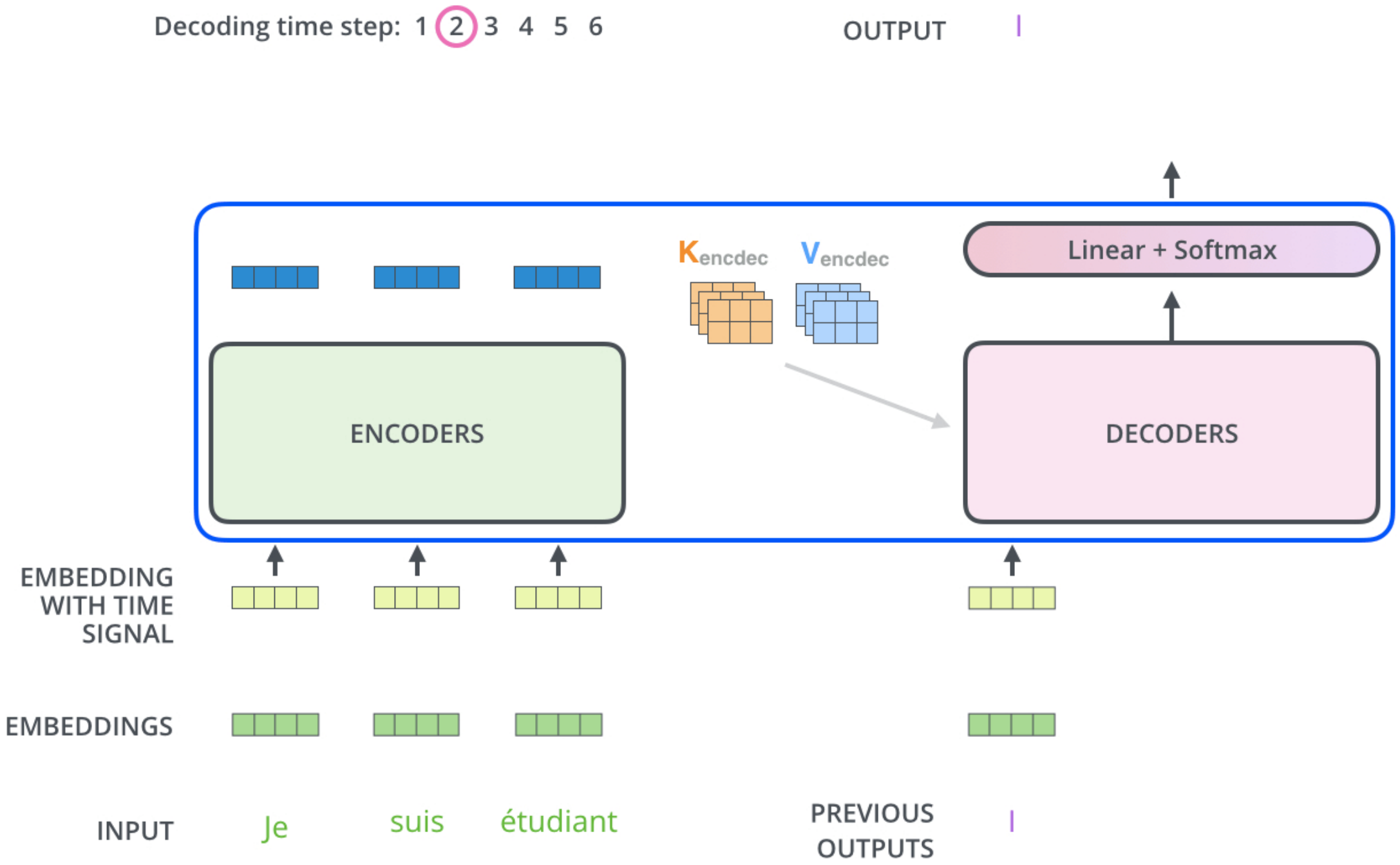
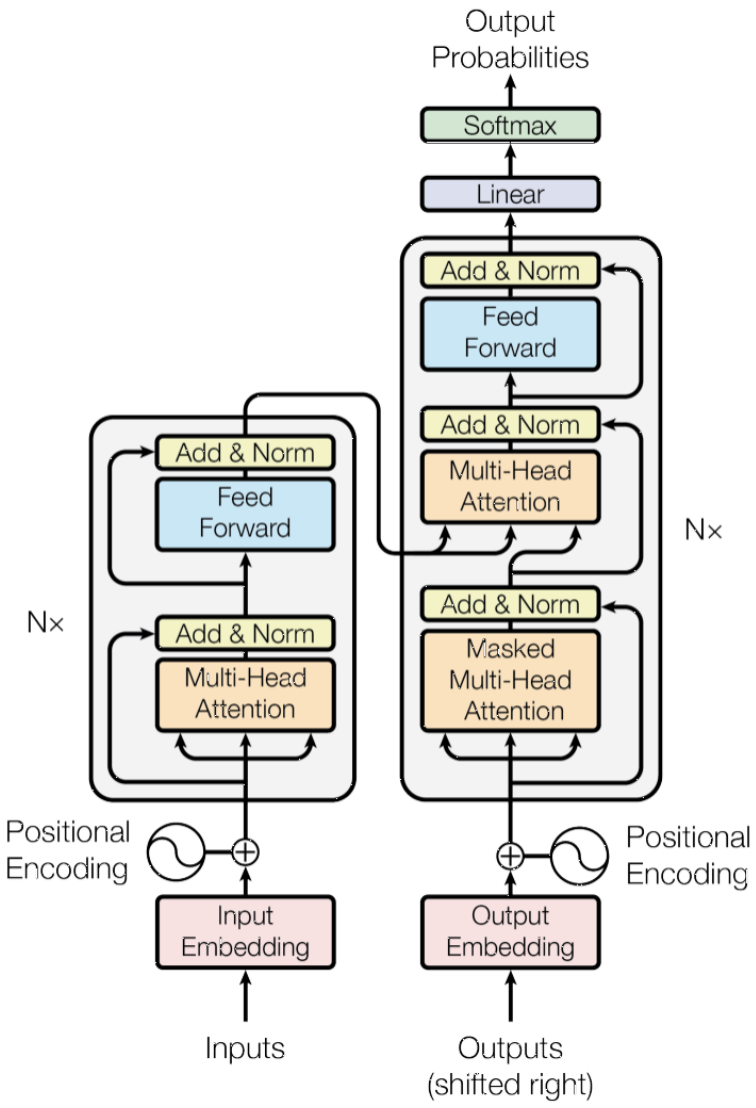



Figure 1: The Transformer - model architecture.

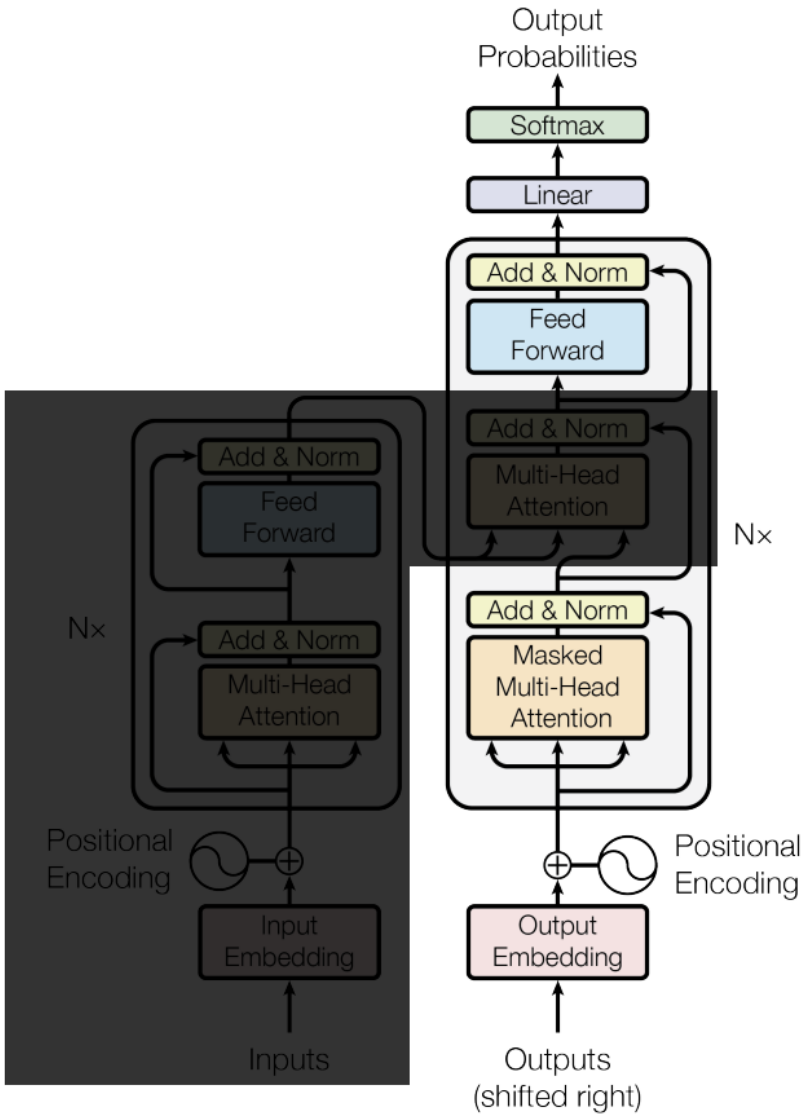

Decoding time step: 1 2 3 4 5 6
OUTPUT
Linear + Softmax
ENCODER
ENCODER
DECODER
DECODER
EMBEDDING WITH TIME SIGNAL
EMBEDDINGS
INPUT
Je
suis
étudiant

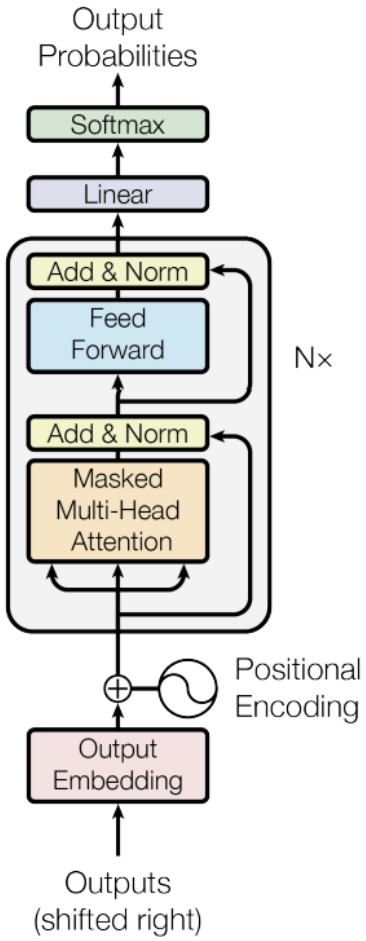

Decoding time step: 1 2 3 4 5 6
OUTPUT
I
Linear + Softmax
Kencdec
Vencdec
ENCODERS
DECODERS
EMBEDDING WITH TIME SIGNAL
EMBEDDINGS
INPUT
Je
suis
étudiant
PREVIOUS OUTPUTS
I



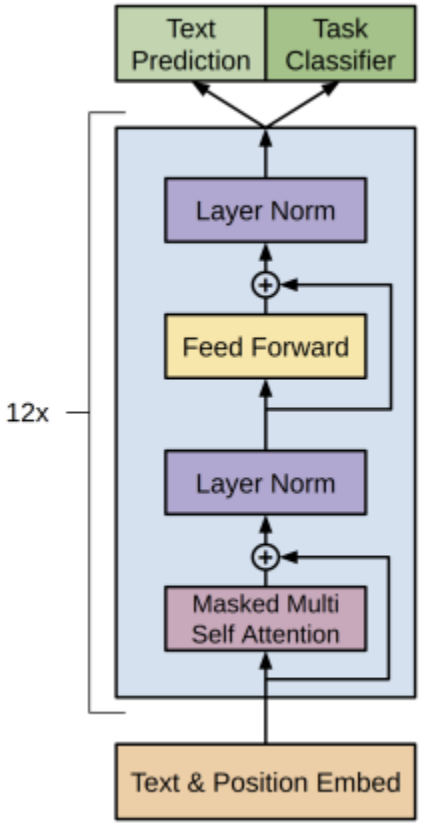

# Original Transformer Architecture

# GPT Architecture

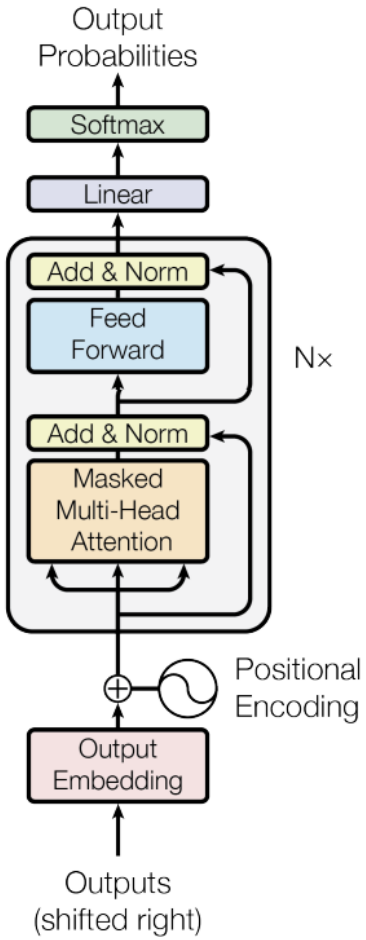


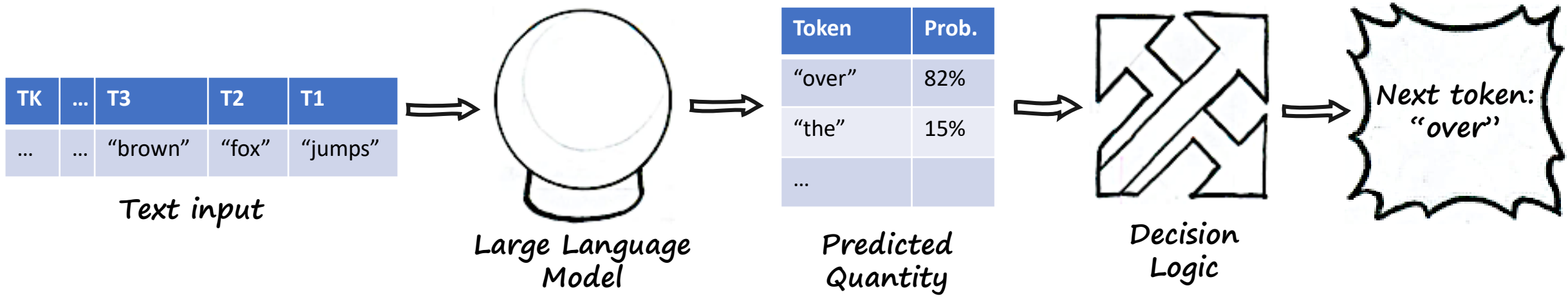



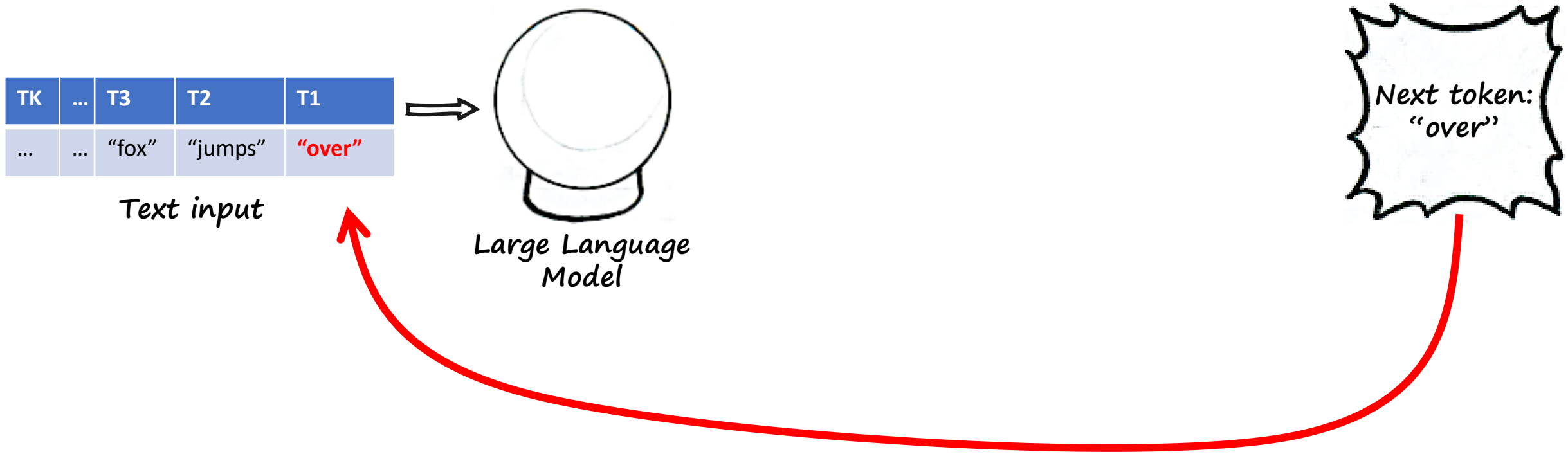
# Official diagram of GPT

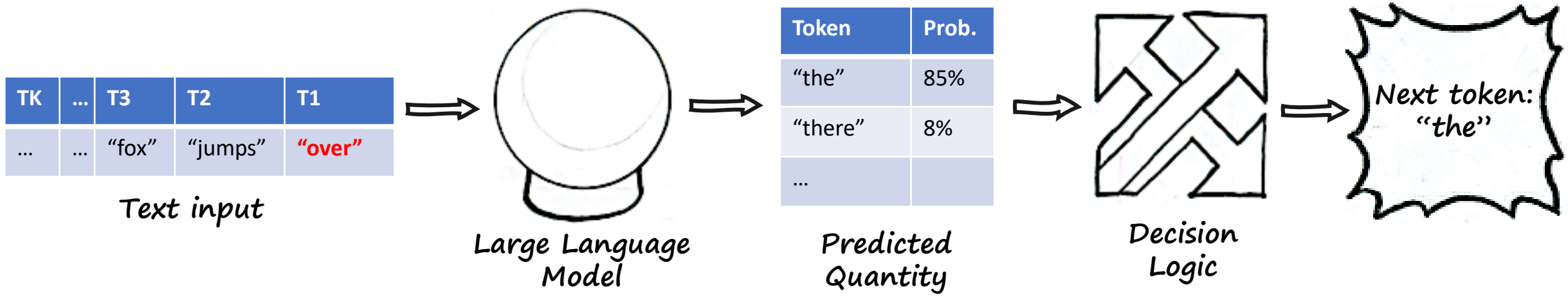



# GPT Architecture

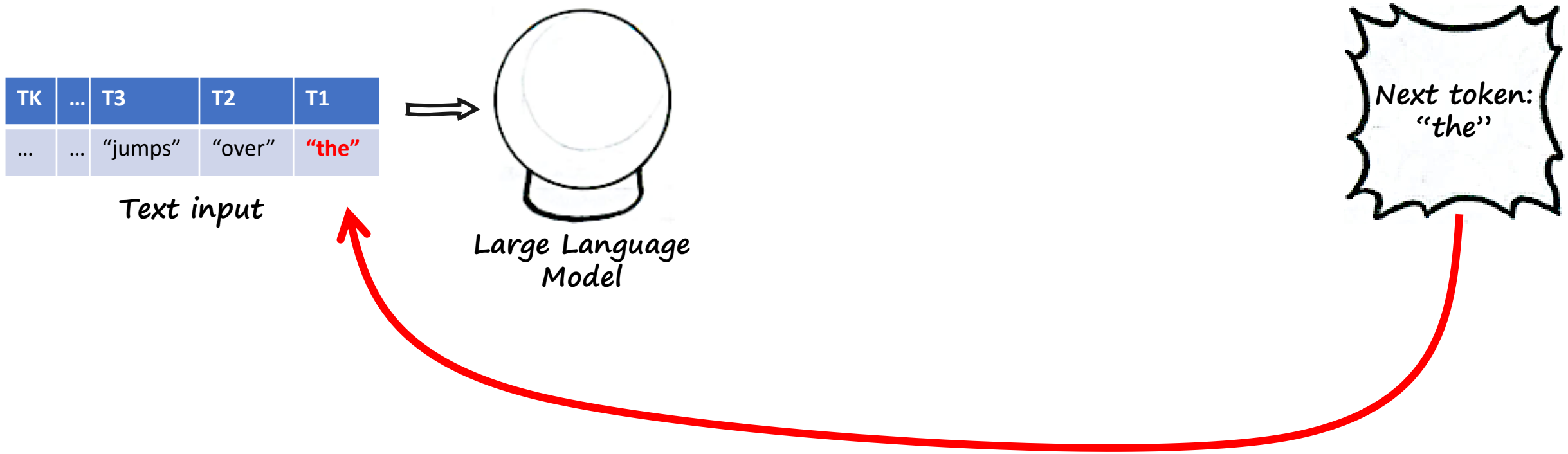



TK
...
T3
T2
T1
...
...
“brown”
“fox”
“jumps”
Text input
Large Language Model
Token
Prob.
“over”
82%
“the”
15%
...
Predicted Quantity
Decision Logic
Next token: “over”

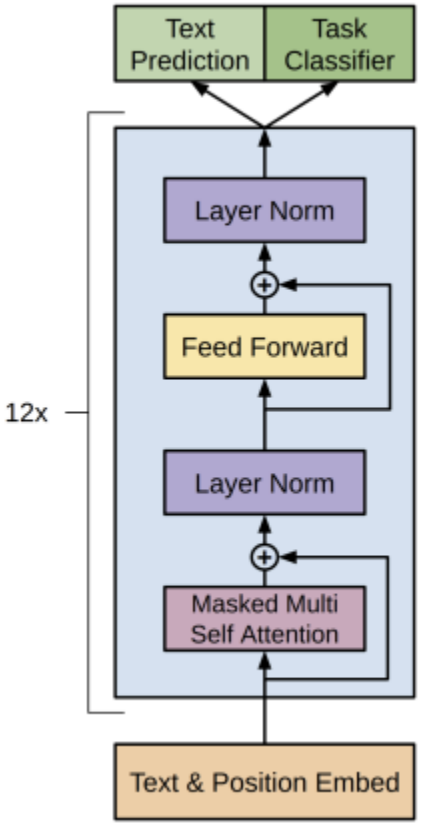

TK
...
T3
T2
T1
...
...
"fox"
"jumps"
"over"
Text input
Large Language Model
Next token: "over"


TK
...
T3
T2
T1
...
...
"fox"
"jumps"
"over"
Text input
Large Language Model
Token
Prob.
"the"
85%
"there"
8%
...
Predicted Quantity
Decision Logic
Next token: "the"

# And this continues until we get the End-Of-Transmission token.



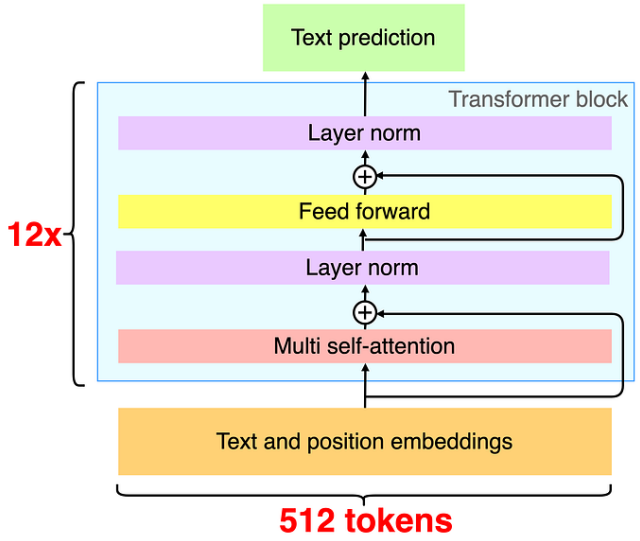
# GPT Architecture




It is straightforward to build a GPT model…
~100 lines of code would do it.

But training it is another story!

## GPT-1

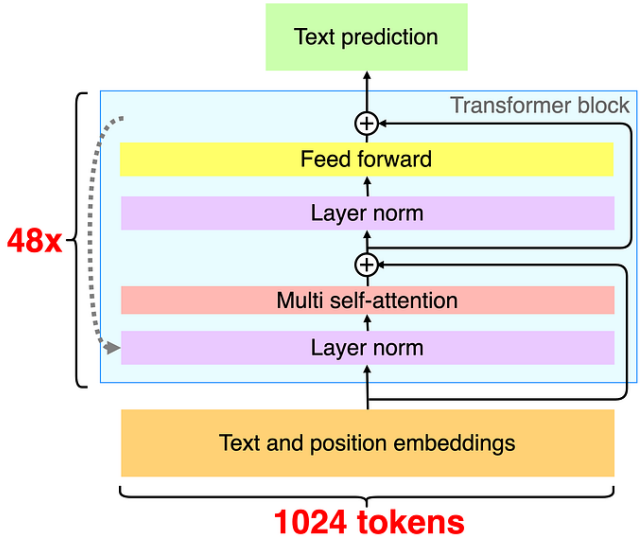



GPT-1, GPT-2, and GPT-3 are very similar.

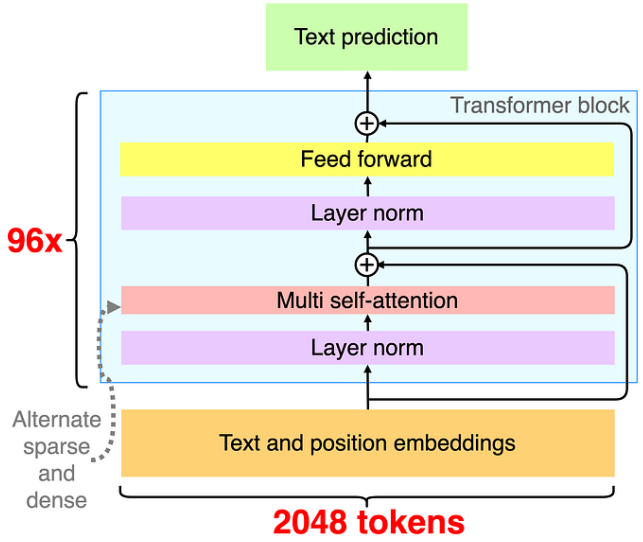
They differ mostly on:

- Number of transformer blocks.
- Number of incoming tokens.
- **<u>The data used and its size.</u>**

## GPT-2




## GPT-3

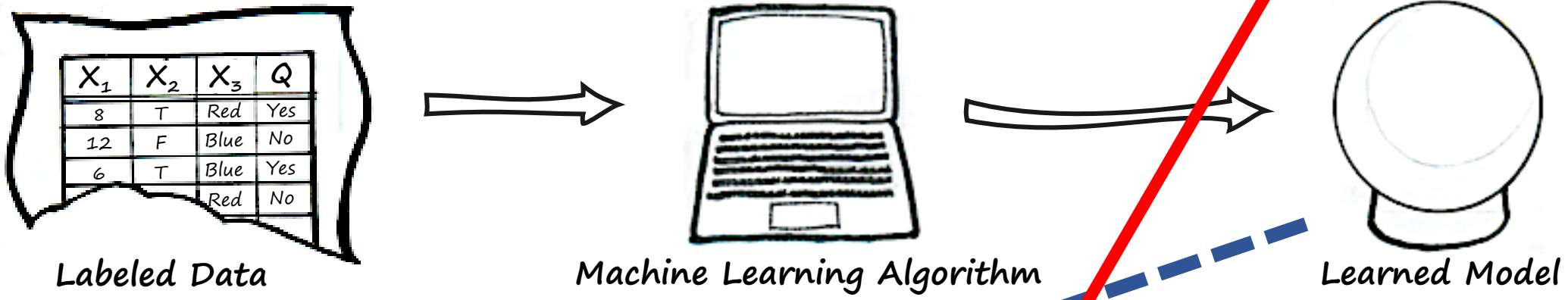



**Questions?**

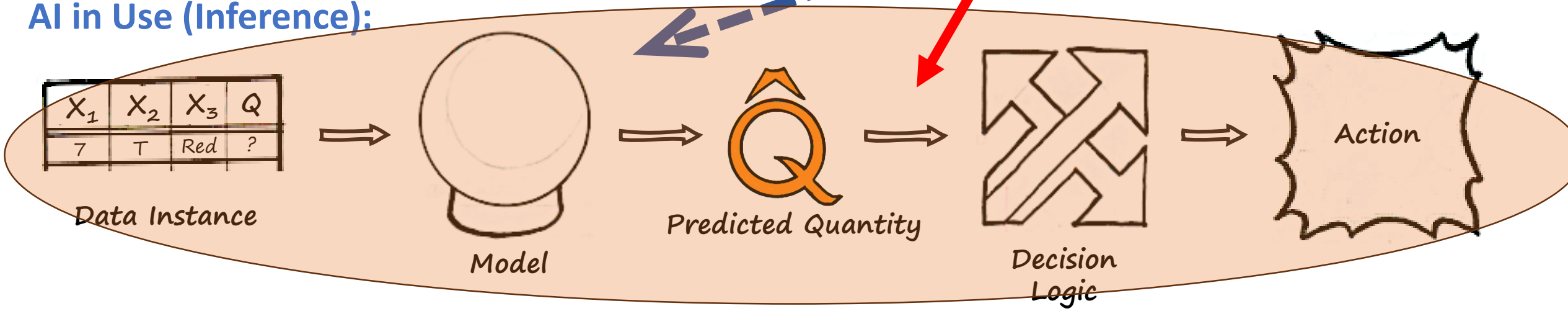
# Framework #3: AI in Use vs. Machine Learning

We just discussed this.

## Machine Learning:



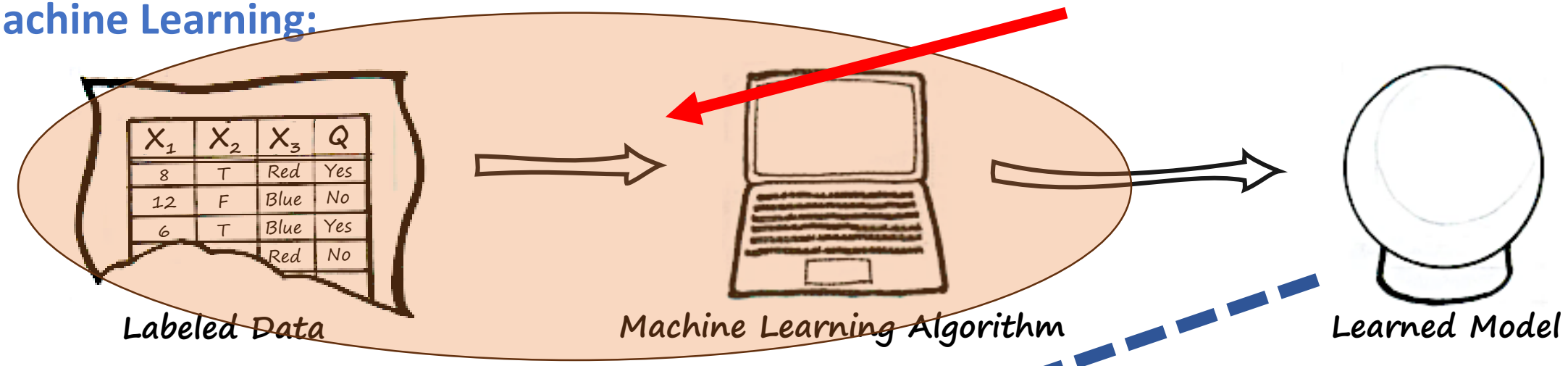

## AI in Use (Inference):



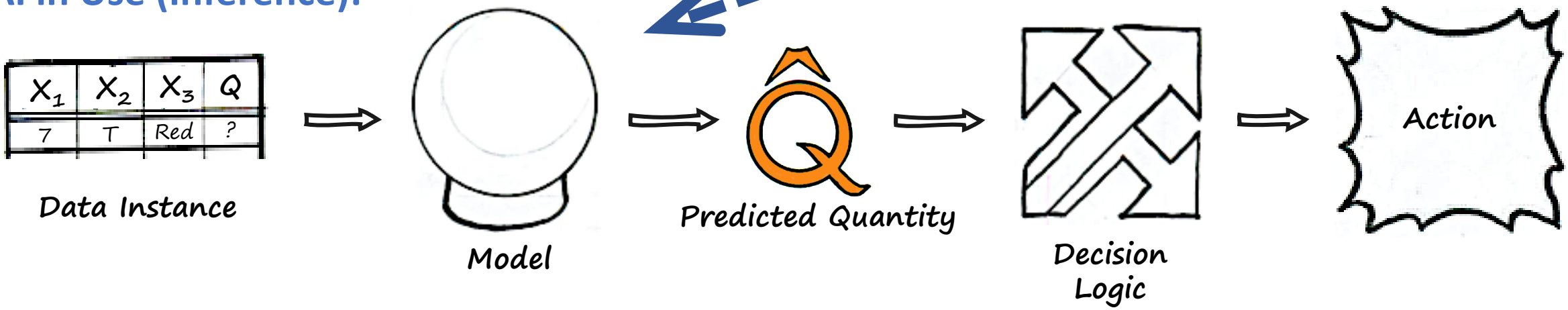
# Framework #3: AI in Use vs. Machine Learning

Let's discuss this now.

## Machine Learning:




## AI in Use (Inference):



Generative Pre-trained Transformer

# What sort of data do you think they used to train the models?

**Pre-trained** on a lot of text data from the Internet.

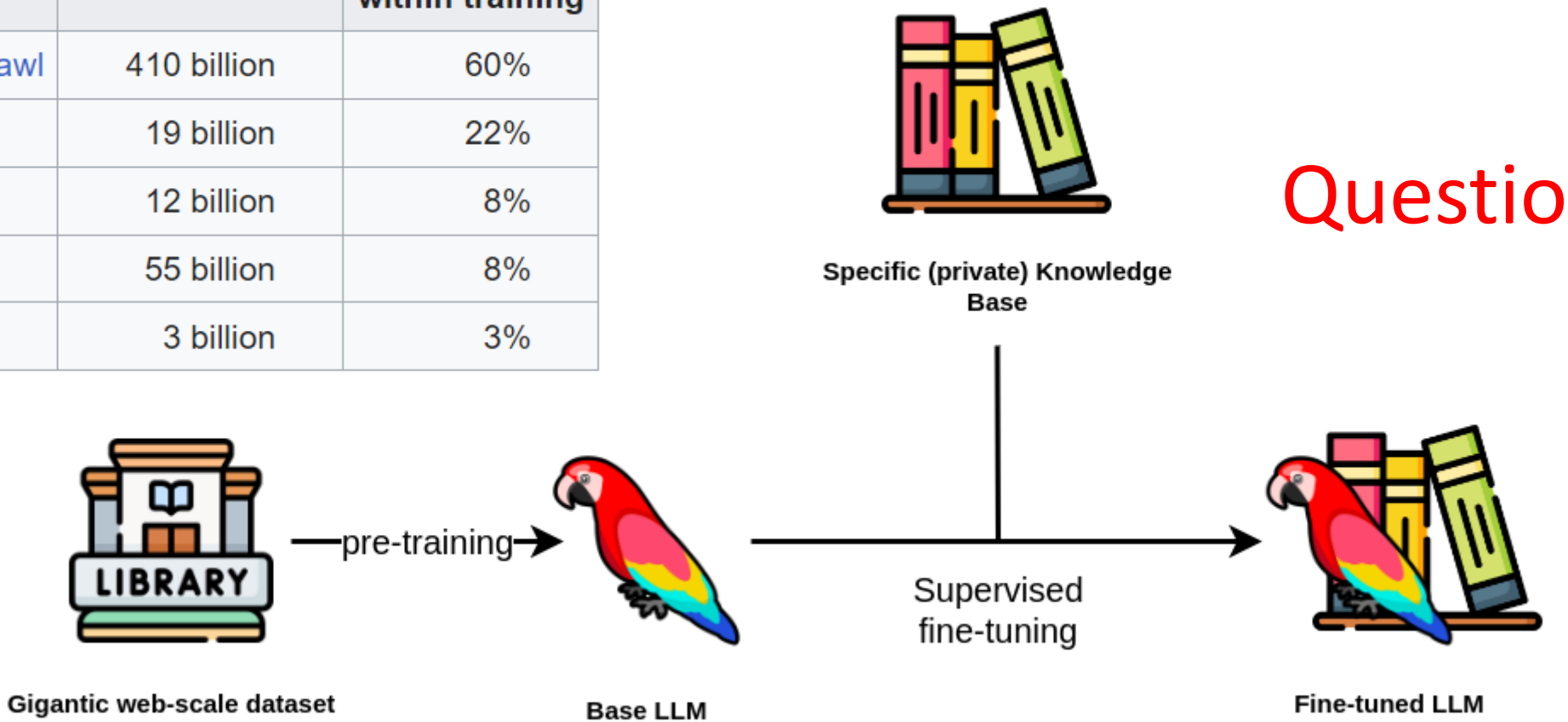
Then **fine-tuned** for chatting.

# Fine Tuning, Part 1 (curated data)

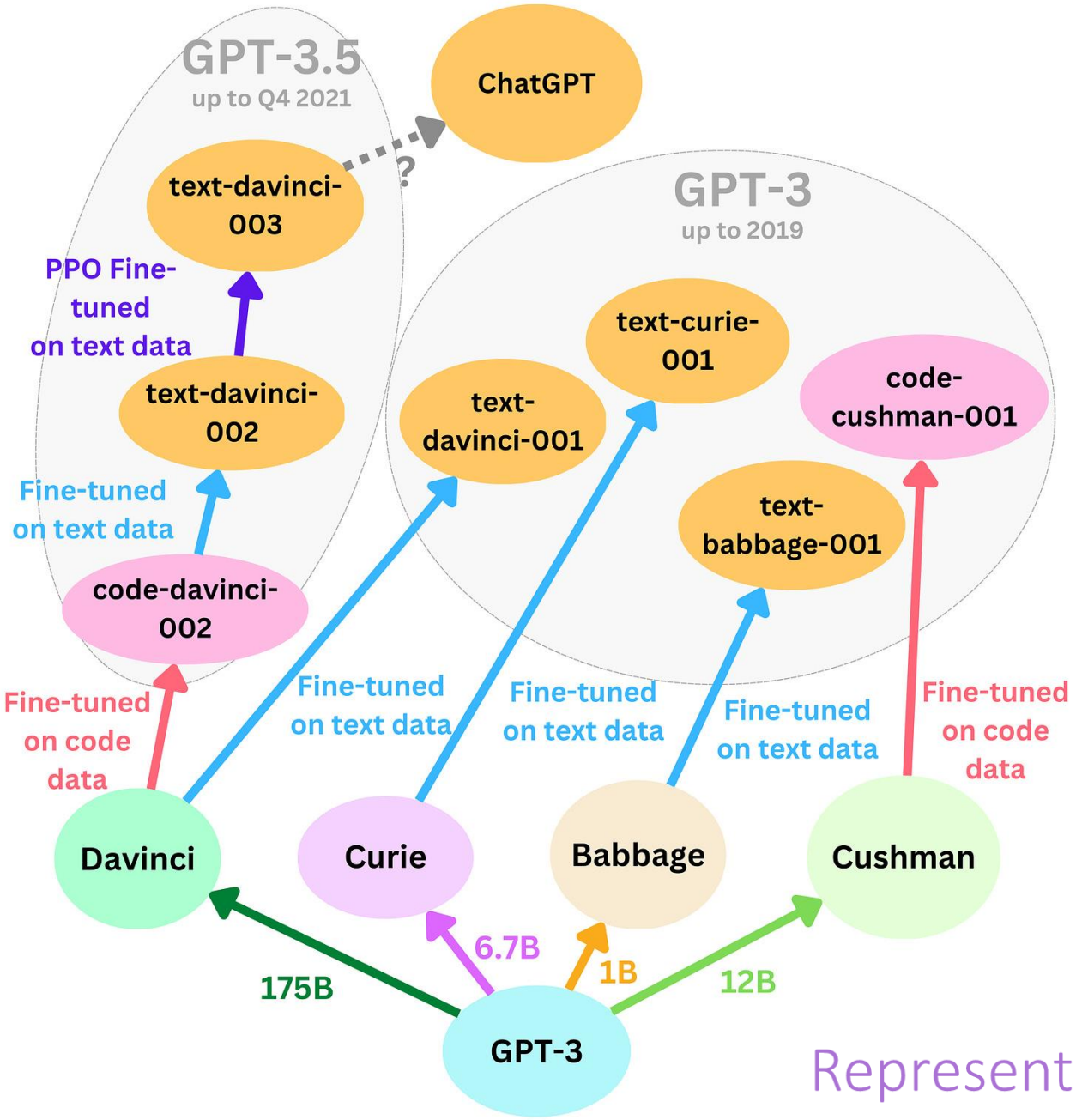
**GPT-3 training data**[1]:9

| Dataset | # tokens | Proportion within training |
|---|---|---|
| Common Crawl | 410 billion | 60% |
| WebText2 | 19 billion | 22% |
| Books1 | 12 billion | 8% |
| Books2 | 55 billion | 8% |
| Wikipedia | 3 billion | 3% |

For example, data from chats or Q&A websites




Questions?




GPT-3 represents a **family** of models.

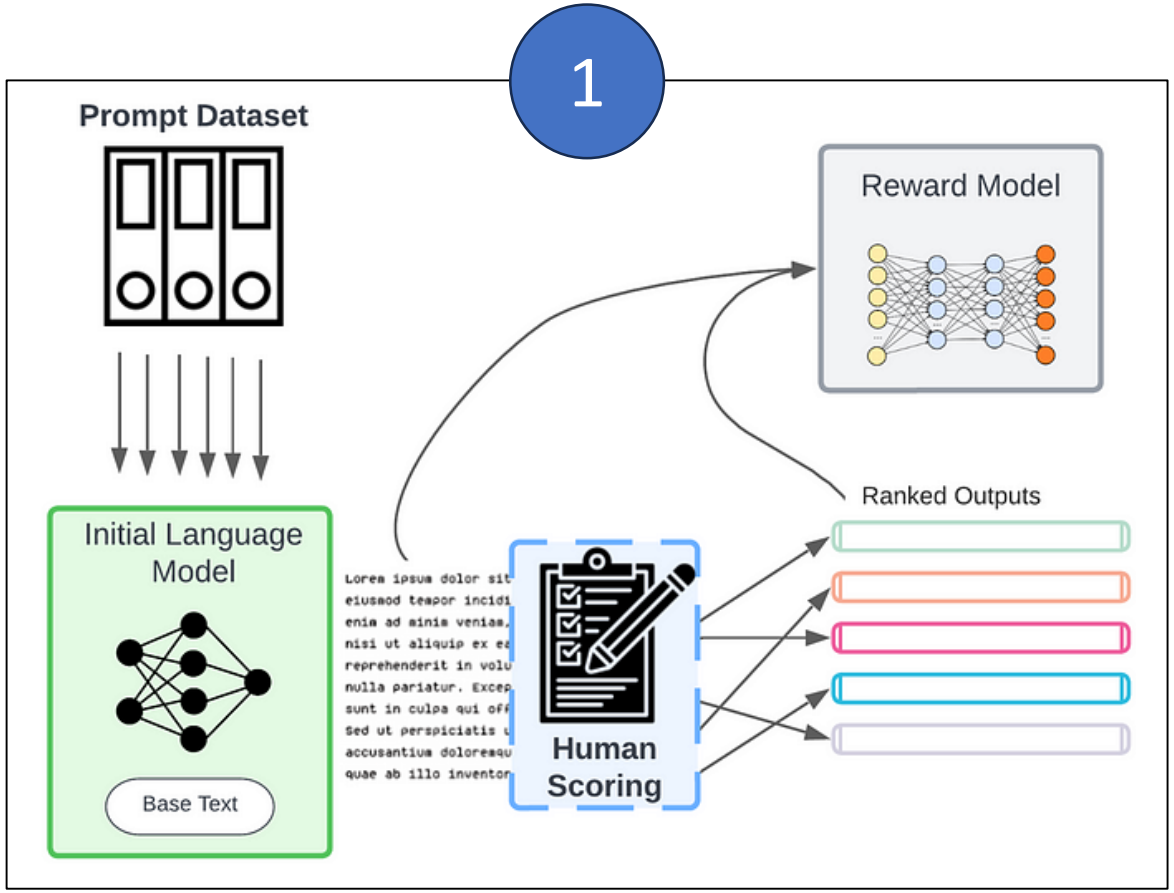
The models differ on:

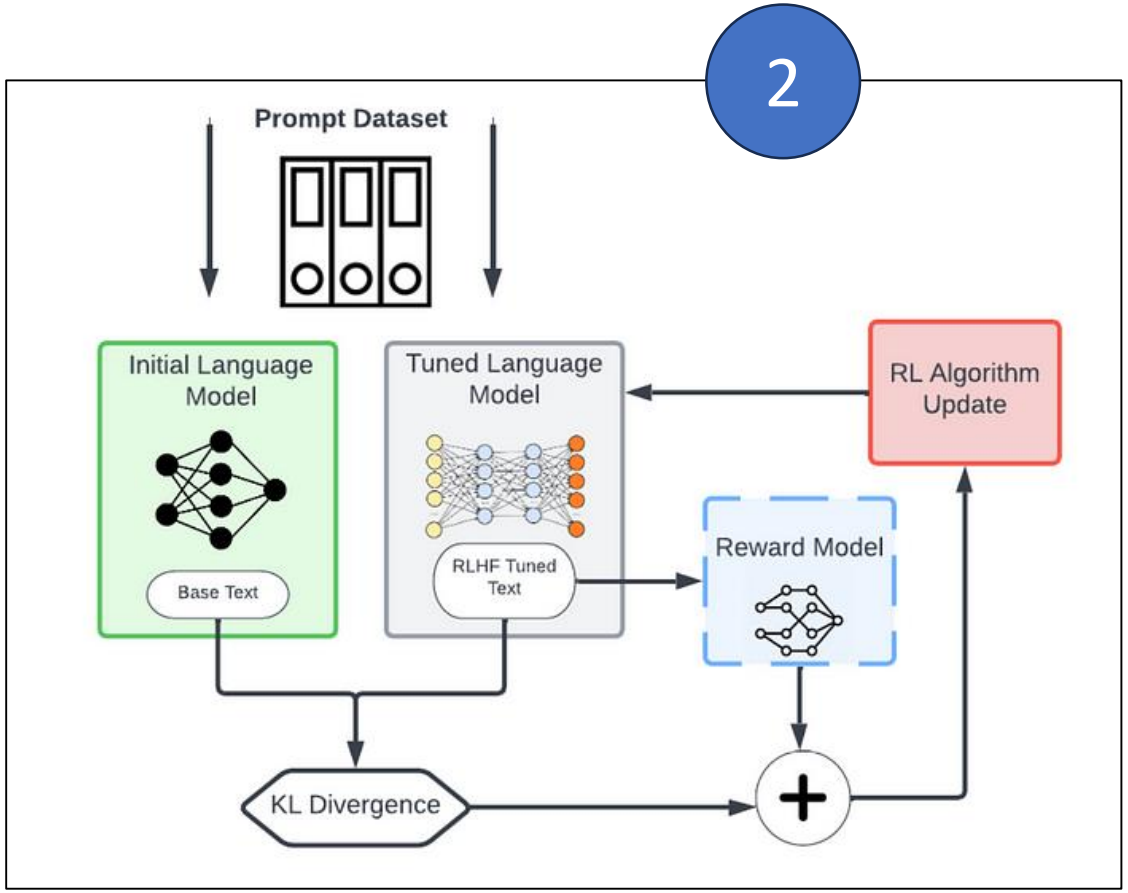
- # of parameters
- Data used
- Training strategy

Representation + Evaluation + Optimization

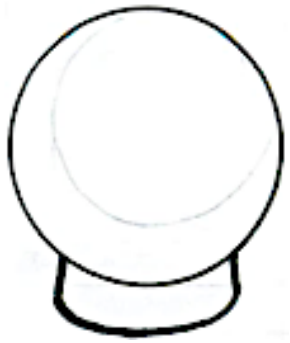
# Fine Tuning, Part 2 (human feedback)

Questions?



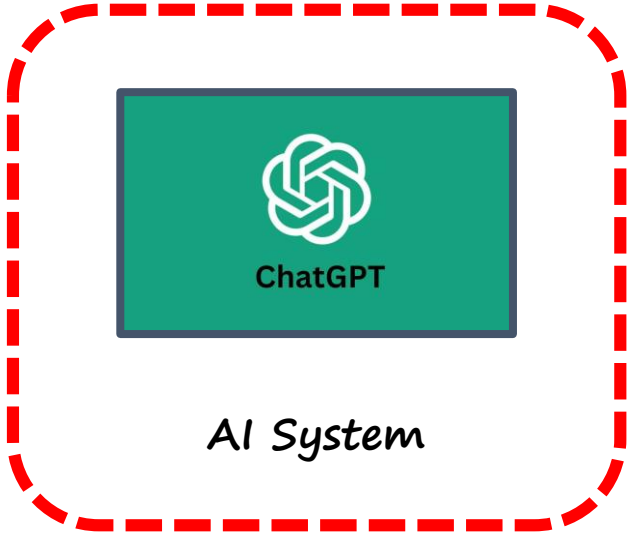



# Recall that:



Large Language Model

≠




Let’s talk about this now.

## Official bots

**Assistant**
General-purpose assistant bot with...

**Web-Search**
General-purpose assistant bot capable...

**GPT-4**
OpenAI's most powerful model....

**DALL-E-3**
OpenAI's most powerful image...

## Bots for you

**DataExpert**
📊 DataExpert: Your Data & Code...
2.9K monthly users

**HumanWrite**
Writing Like a Real Human, the solution...
20K monthly users

**Mrstherapist**
Your very own therapist with...
2.9K monthly users

**SummarizeURLs**
Our AI agent generates summarie...
68 monthly users

## Popular bots

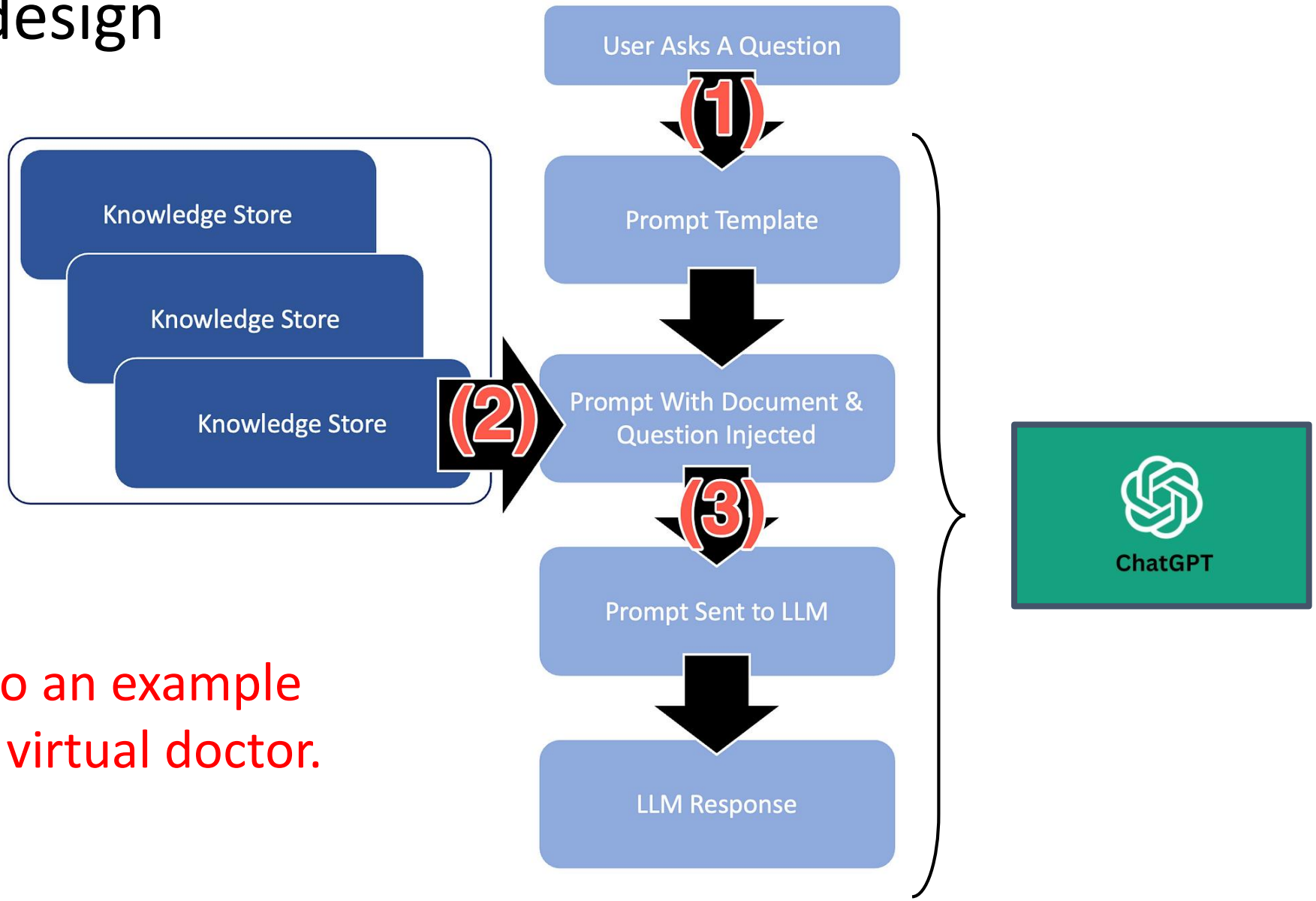
**LOGO-CREATOR...**
⭐⭐⭐⭐⭐ 🎨
Your Creative...
64K monthly users

**UniversityGPT**
A bot to help you learn about anything.
23K monthly users

**Photo_CreateE**
This bot generates realistic, stock photo...
183K monthly users

**EnglishTeacher_J...**
You can chat in English (free choice ...
7.2K monthly users

# Prompt design




Let’s do an example with a virtual doctor.

**User asks a question**:

What should I do about my frequent headaches?

**Prompt template**:

**User asked**: "[USER_QUESTION]"

**Patient's Health History**:
- Age: [AGE]
- Existing Conditions: [CONDITIONS]
- Medications: [MEDICATIONS]
- Lifestyle Factors: [LIFESTYLE]

**Relevant Medical Information**:
- Common Causes: [CAUSES]
- Treatment Options: [TREATMENTS]
- When to Seek Help: [SEEK_HELP]

Provide a detailed response to the user incorporating this information.

**User asks a question**:

What should I do about my frequent headaches?

**Prompt template**:

**User asked**: What should I do about my frequent headaches?

**Patient's Health History**:
- Age: [AGE]
- Existing Conditions: [CONDITIONS]
- Medications: [MEDICATIONS]
- Lifestyle Factors: [LIFESTYLE]

**Relevant Medical Information**:
- Common Causes: [CAUSES]
- Treatment Options: [TREATMENTS]
- When to Seek Help: [SEEK_HELP]

Provide a detailed response to the user incorporating this information.

**User asks a question**:

What should I do about my frequent headaches?

**Prompt template**:

**User asked**: What should I do about my frequent headaches?

**Patient's Health History**:
- Age: 35
- Existing Conditions: Hypertension
- Medications: Lisinopril
- Lifestyle Factors: Sedentary, high-stress job

**Patient data**

from medical records

**Relevant Medical Information**:
- Common Causes: [CAUSES]
- Treatment Options: [TREATMENTS]
- When to Seek Help: [SEEK_HELP]

Provide a detailed response to the user incorporating this information.

**User asks a question**:

What should I do about my frequent headaches?

## Prompt template:

**User asked**: What should I do about my frequent headaches?

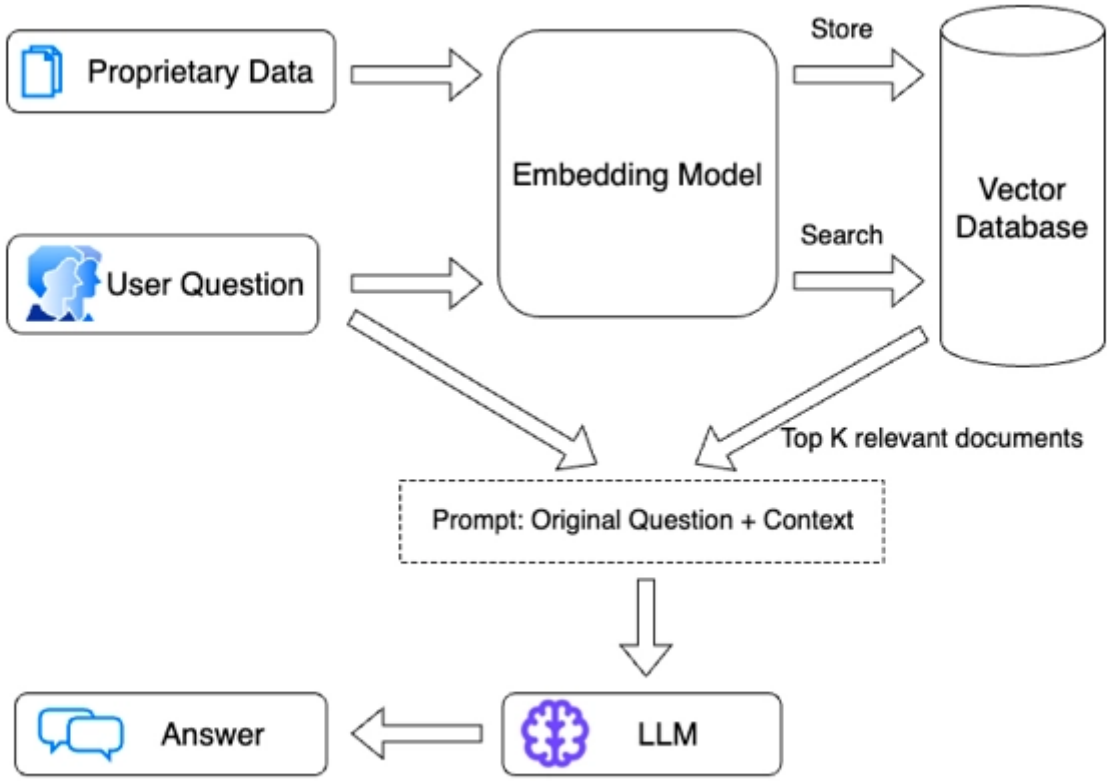
**Patient's Health History**:
- Age: 35
- Existing Conditions: Hypertension
- Medications: Lisinopril
- Lifestyle Factors: Sedentary, high-stress job

How can we fill this?

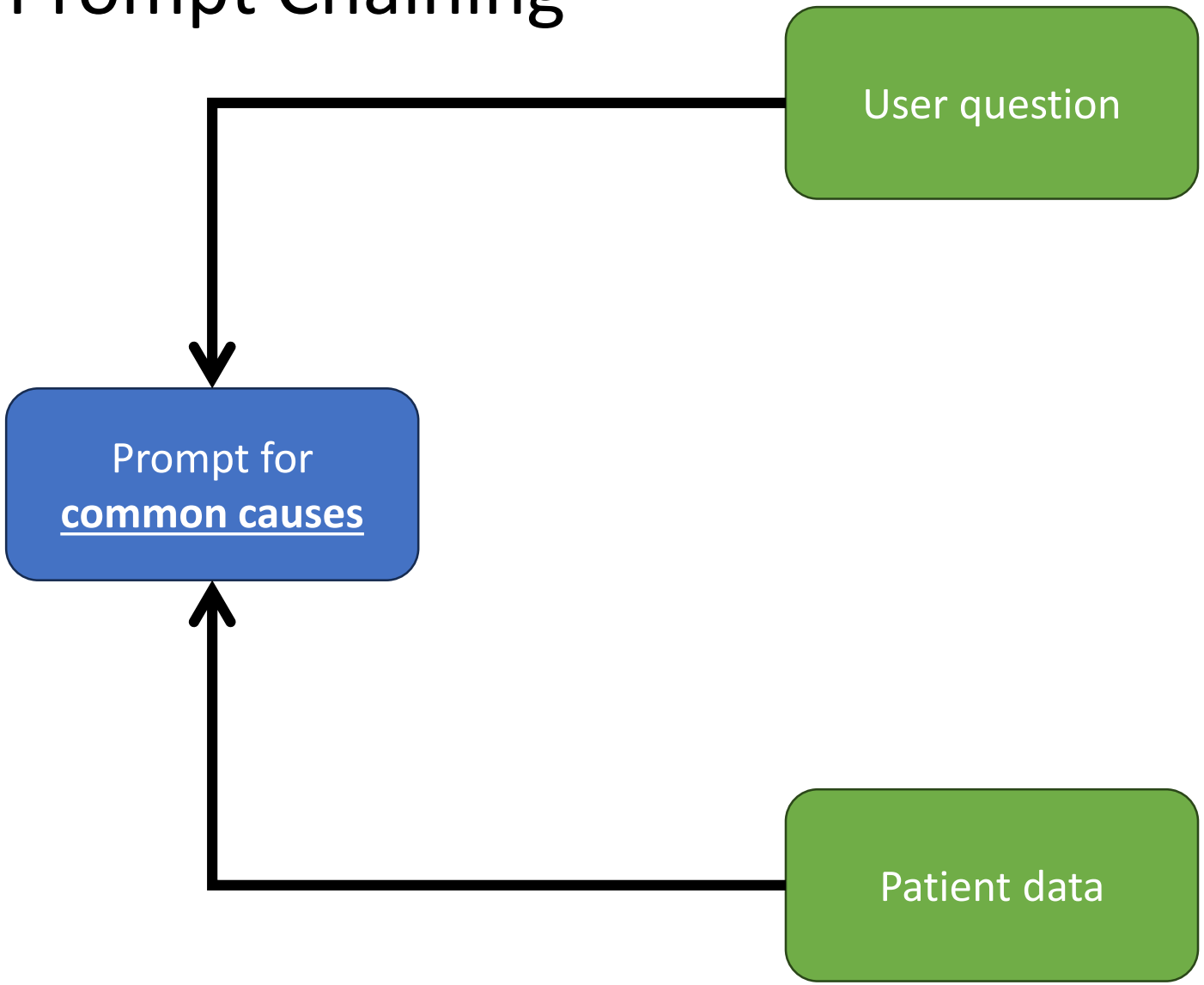
**Relevant Medical Information**:
- Common Causes: Stress, poor posture, dehydration
- Treatment Options: Drink more water, take more breaks, relaxation techniques
- When to Seek Help: If headaches persist or worsen, consult a healthcare provider

Provide a detailed response to the user incorporating this information.

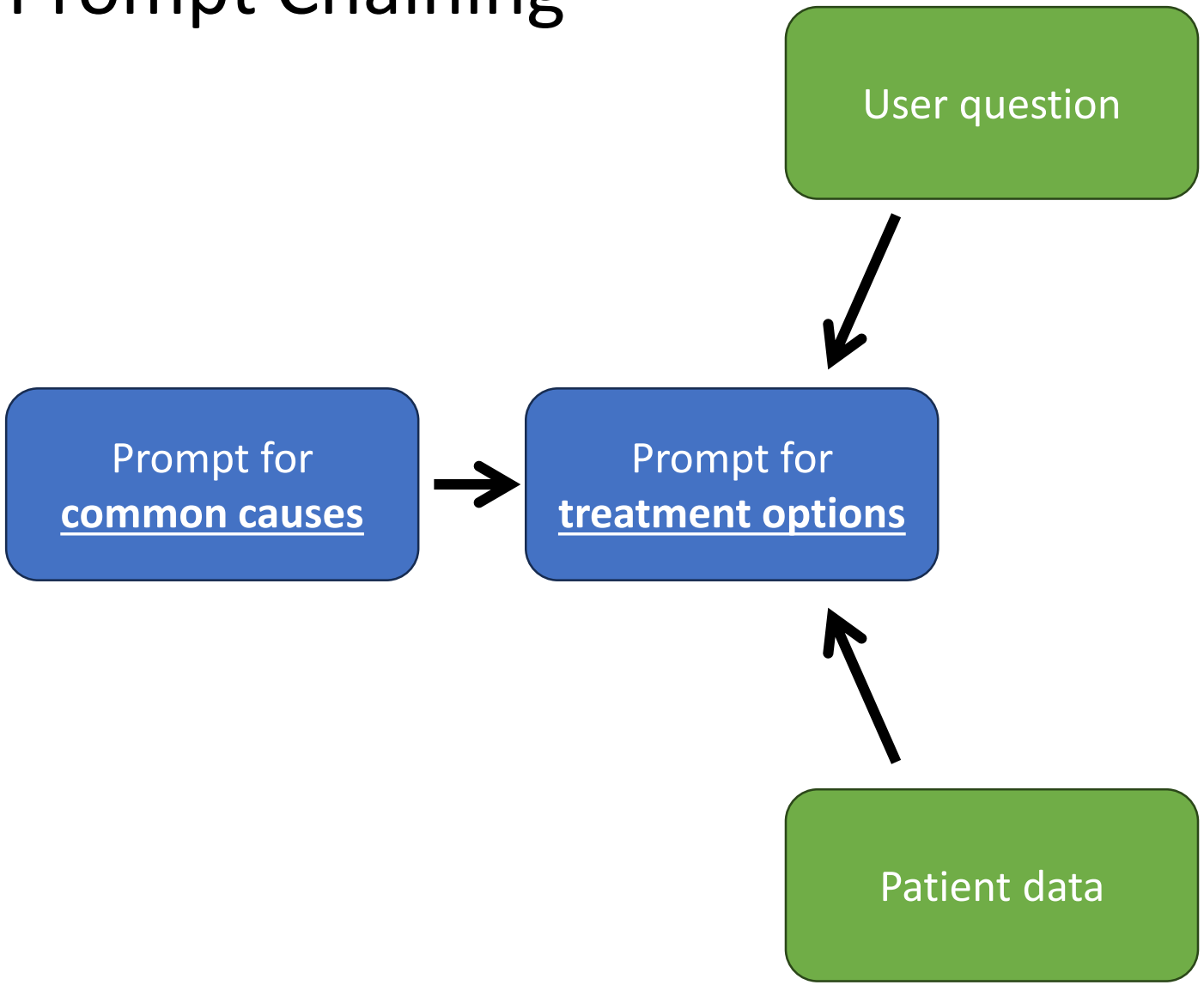
# Retrieval Augmented Generation (RAG)



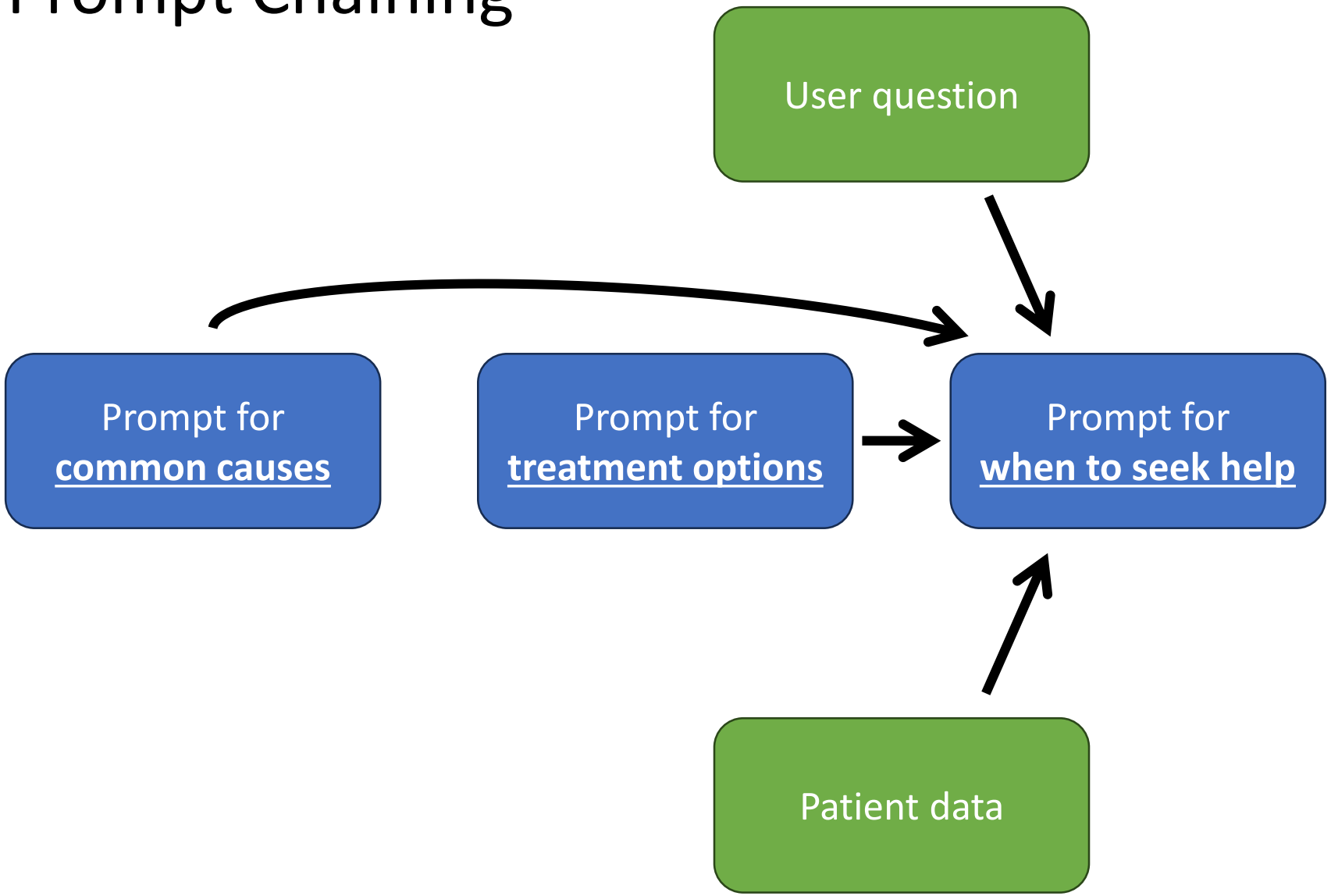
# Prompt Chaining



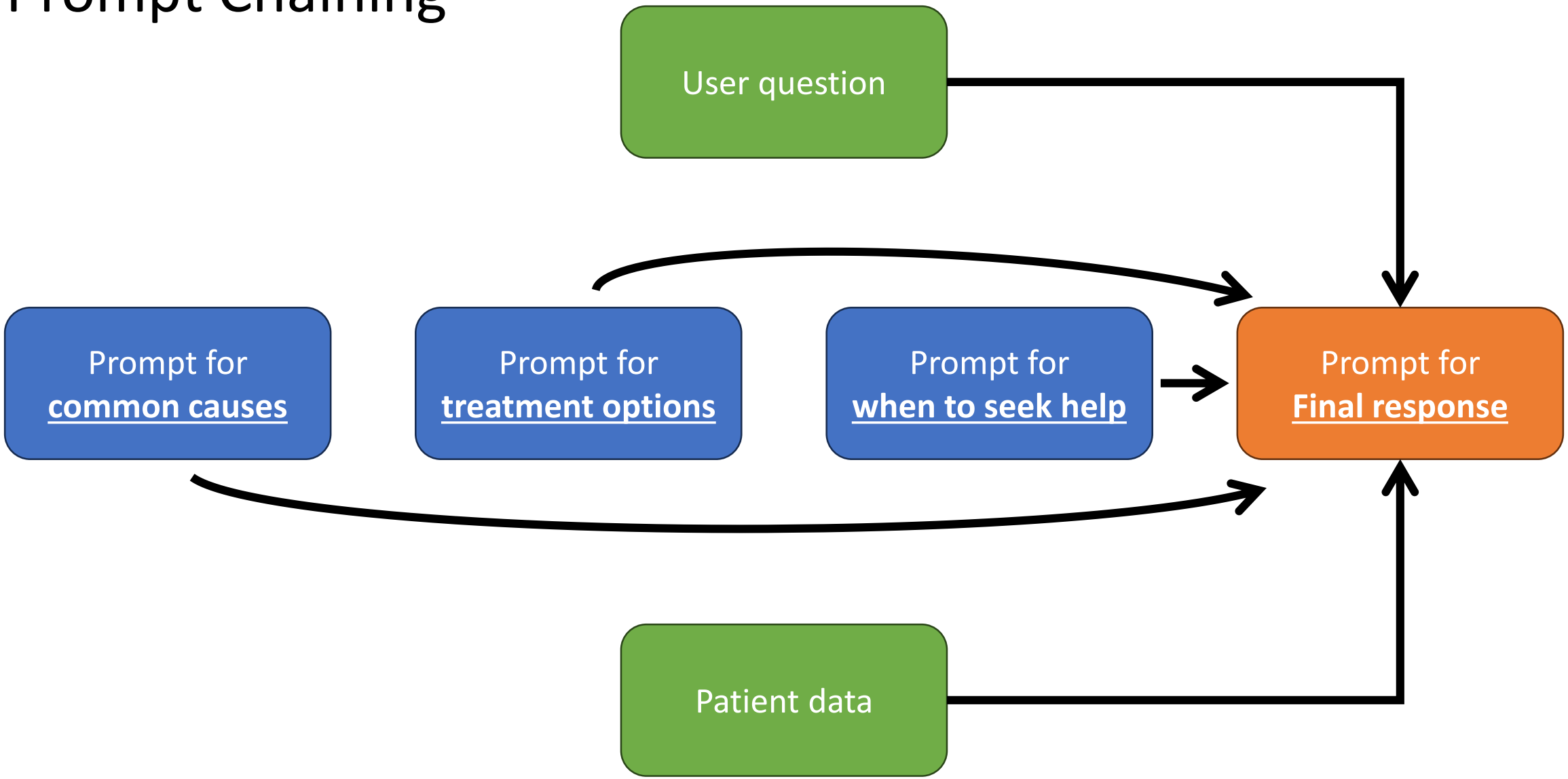
# Prompt Chaining



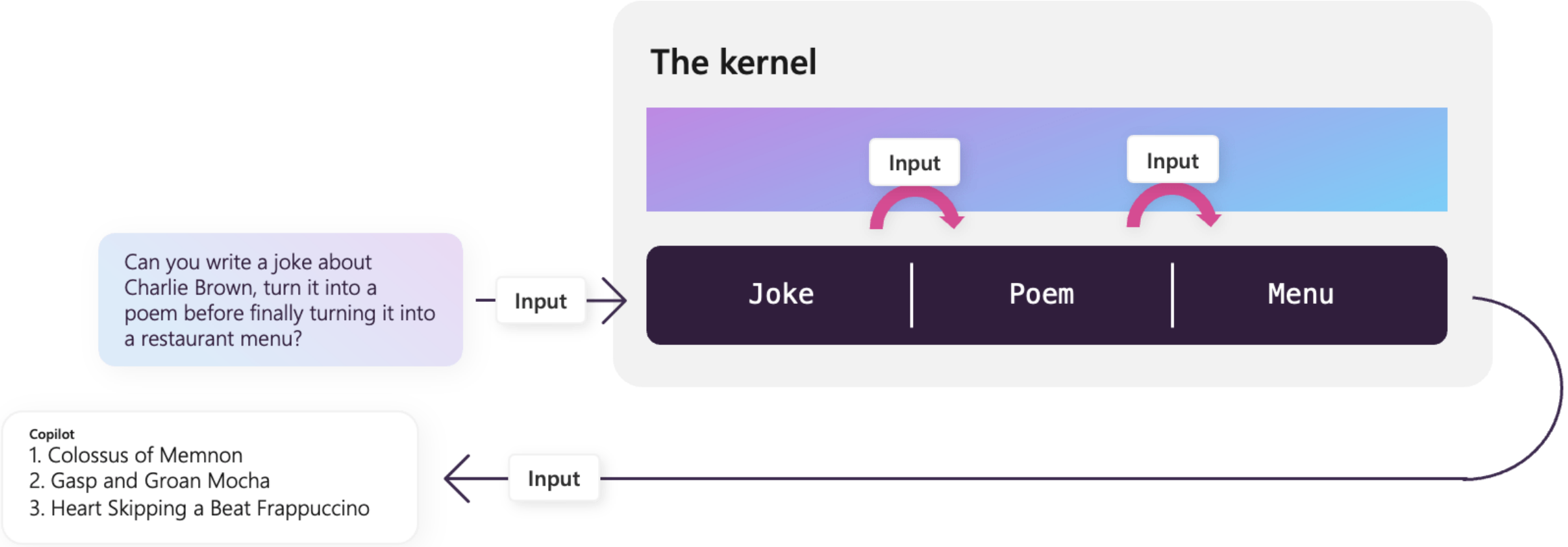
# Prompt Chaining



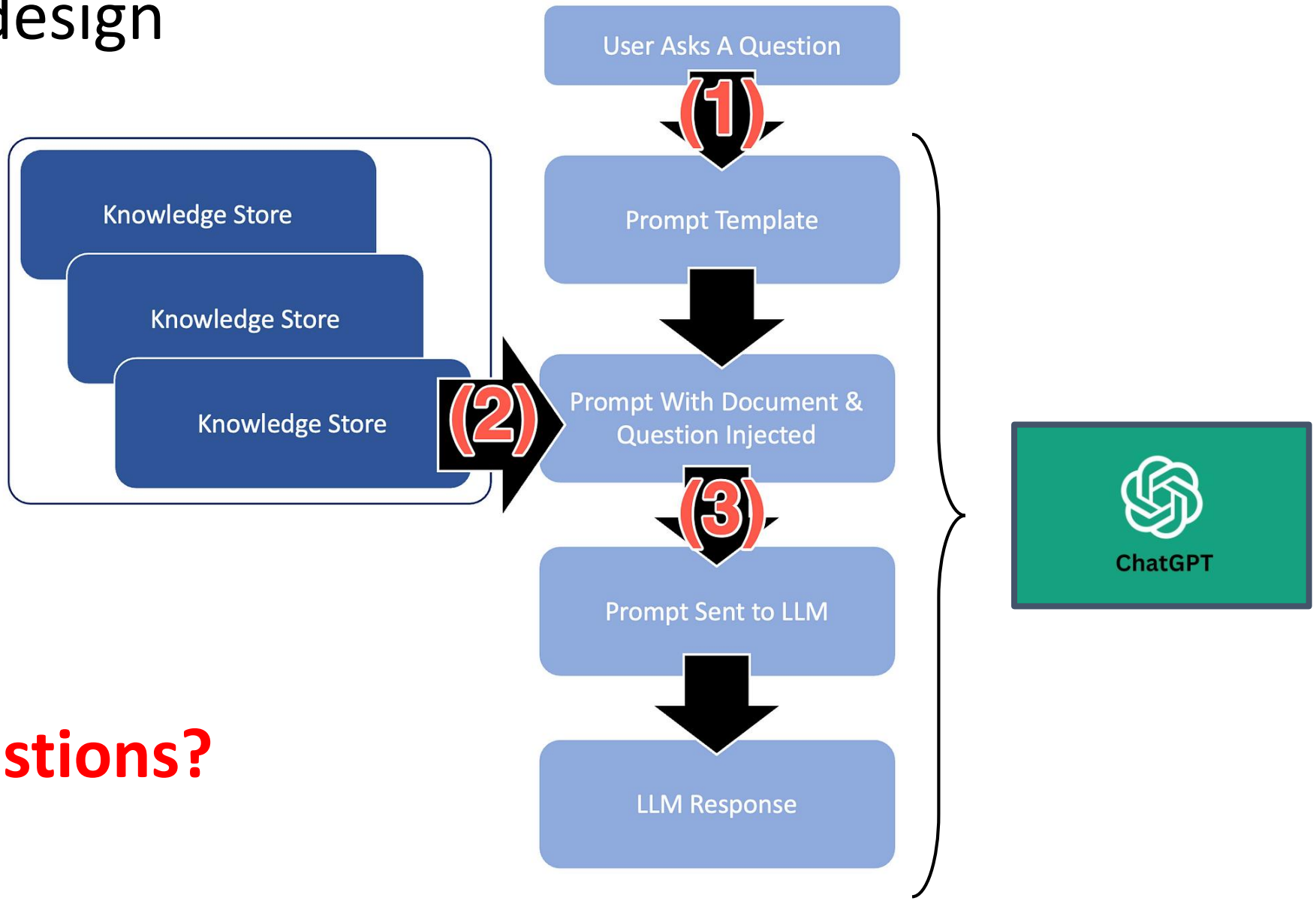
# Prompt Chaining



# Semantic Kernel

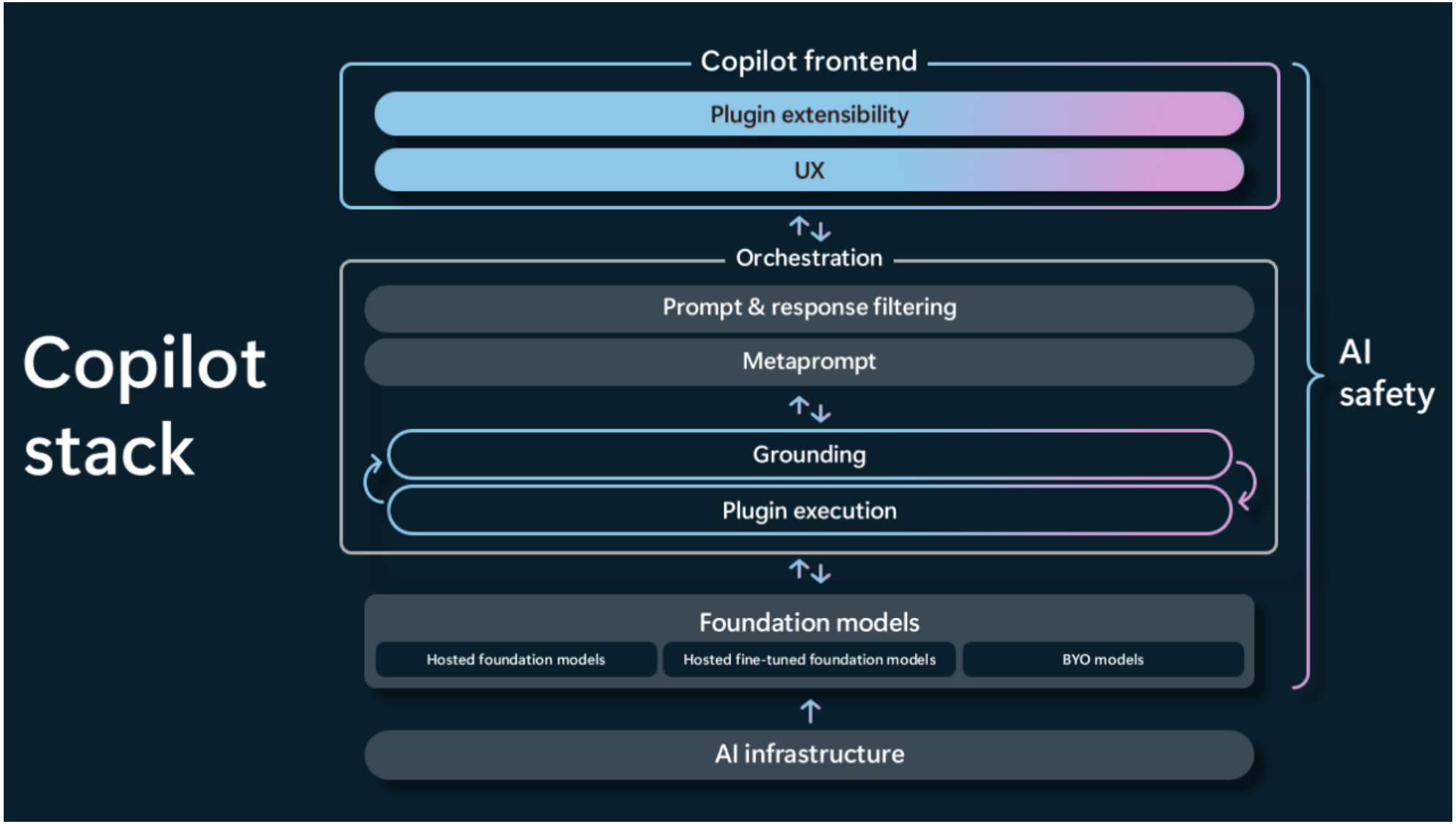


# Prompt design



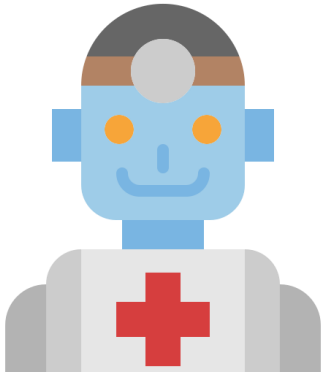

**Questions?**

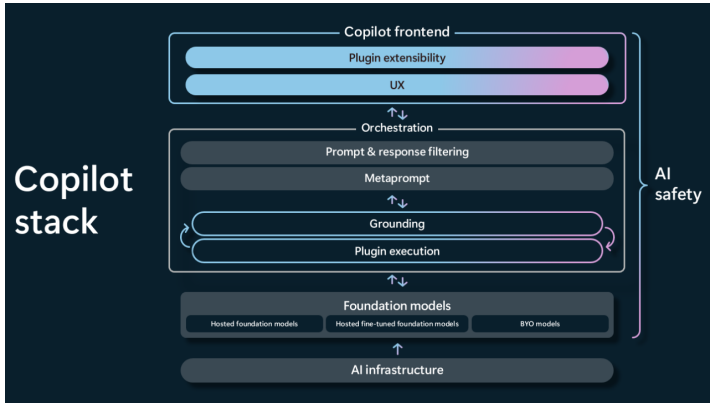
# Microsoft Build 2023: AI Redefines Software Development and Future of Work.



**User asks a question**:

What should I do about my frequent headaches?






*"Frequent headaches require attention to lifestyle and medical factors. At **age 35**, and with a condition like **hypertension**, you should…"*

# How do we evaluate LLM-based systems?

Ep 6. Conquer LLM Hallucinations with an Evaluation Framework

2K views • 2 months ago

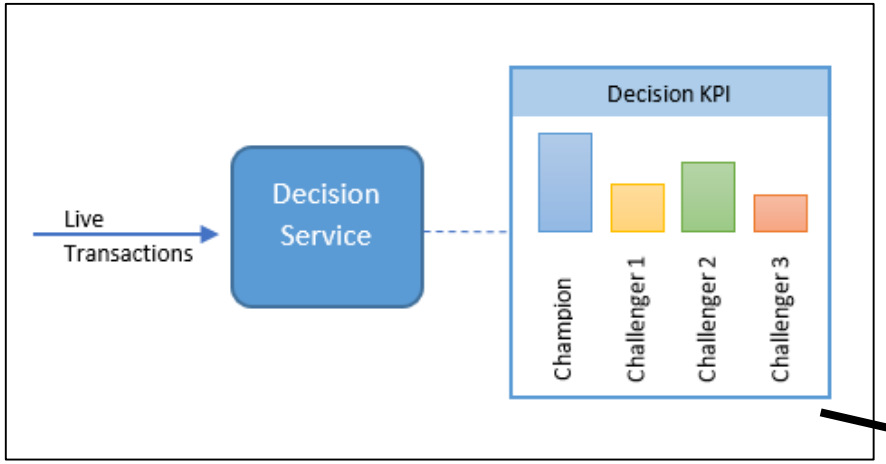
Prolego

ABOUT THIS VIDEO Large language models (LLMs), such as GPT-4, are intelligent tools that allow for rapid, cost-effective ...

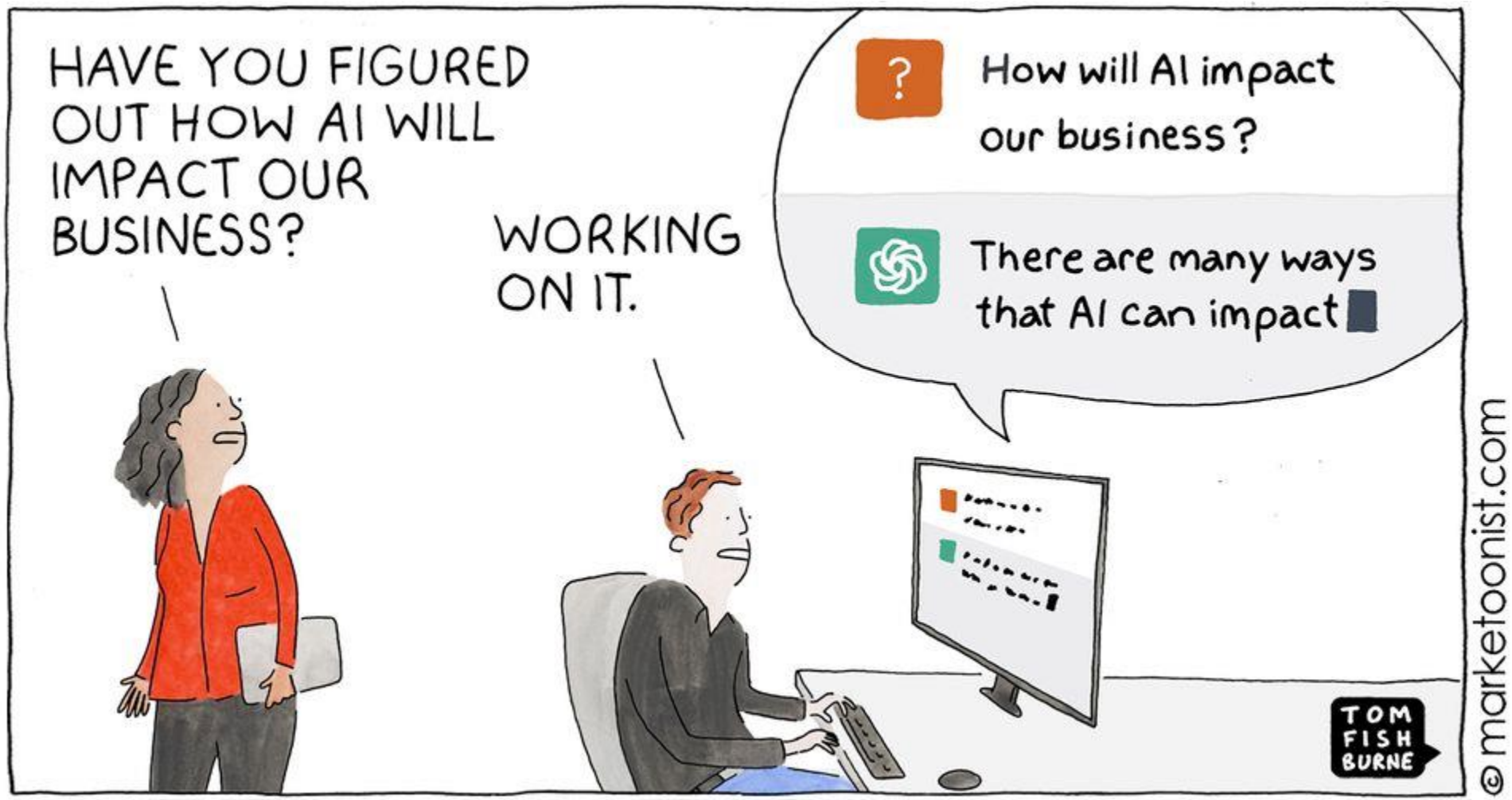
4K

In this video, Prolego CEO, Kevin Dewalt illustrates the creation of a basic evaluation framework. He designed five scenarios, merging various models and LLM agent instructions, and assessed them using four metrics:

1. Cost
2. Speed
3. Reliability
4. Accuracy




Champion Challenger Approach

**Questions?**

# Be careful who you trust.