Theory, Practice and Future of LLMs

Jon Chun
Kenyon College
2 November 2023

Jonachun.com and @jonchun2000

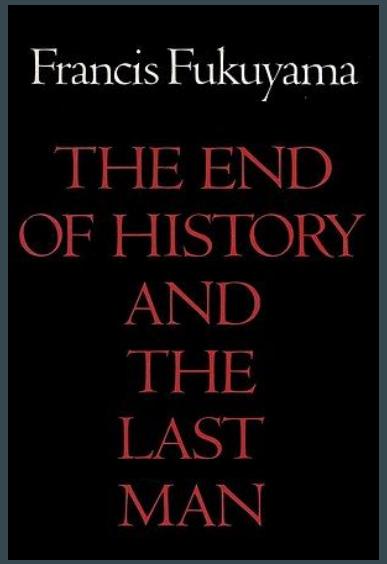
- Teaching
  - First human-centered AI curriculum (2016)
  - 300+ Mentored student research at digital.kenyon.edu/dh
- Interdisciplinary Research
  - APQ: Novel Jailbreak Ethical Reasoning Benchmarks of 12 Leading LLM
  - IJDH: Novel GreyBox Ensemble eXplainable AI
  - Spring Nature IJHAC: AI Challenges and Interdisciplinary Curriculum
  - (current): LLM Affective AI, Synthetic Data, Small 7B-13B Open LLMs
- Presentations
  - Limits of Cognition (Theoretical CS)
  - Medical Applications of AI (FDA association)
  - Manufacturing Engineering/Physics Simulation (Synthetic Time Series)
  - Higher Ed (Interdisciplinary AI)
- Consulting
  - Finance (VC, Private Equity, Industry)
  - Startup Consultant
- Industry
  - Silicon Valley Startups to Fortune 500 Director of Development (Sale Engineer to Patents)
  - Computer Security, FinTech, MedTech, InsurTech, Semiconductors: US, Asian and Latin America
- Paper/Book Reviews, Grants, Non-Profit, jonachun.com and Tweets @jonchun2000

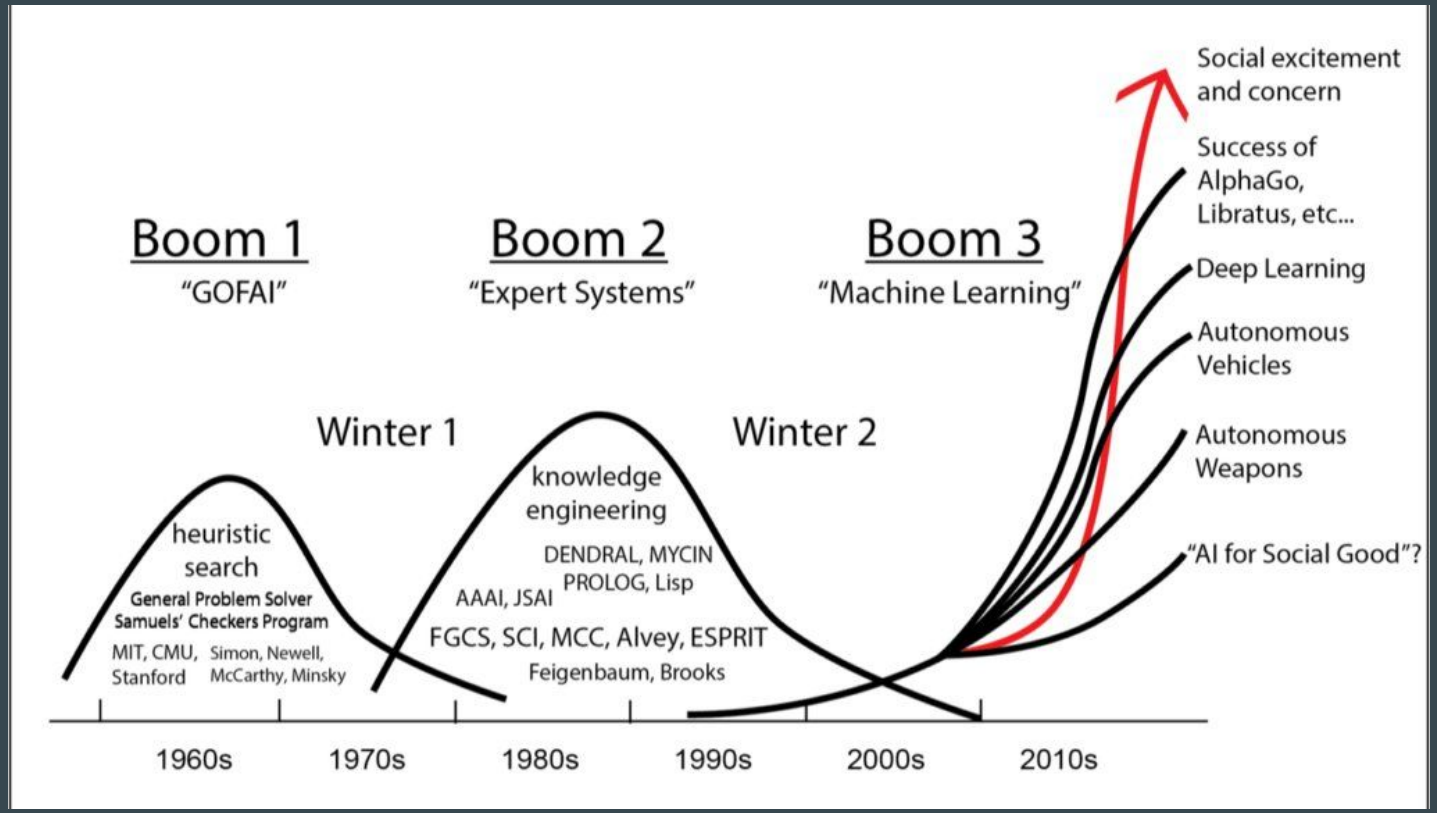
Overview

- History
- Theory
  - Information
  - Representation
  - Learning
  - Reasoning
- Practice
  - Transformers Architecture
  - Inference vs Training
- Future

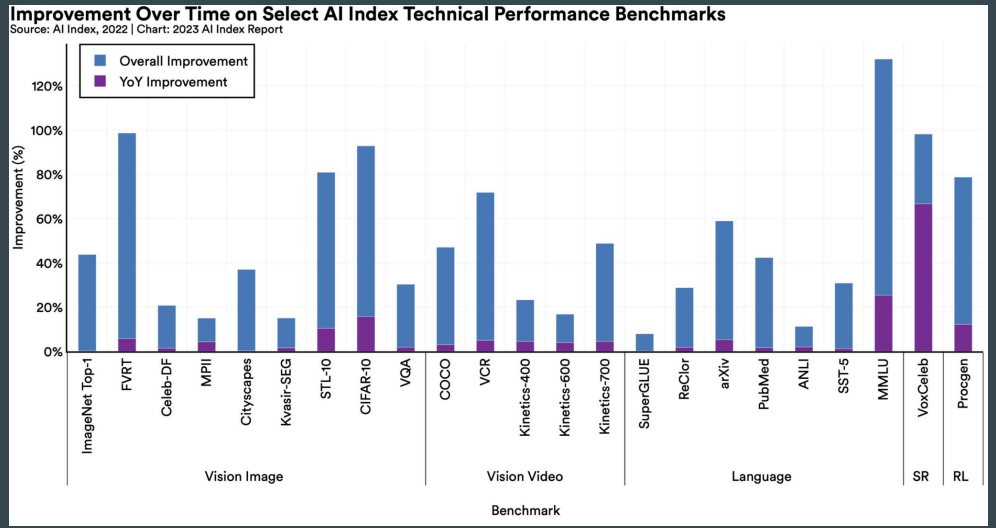
History

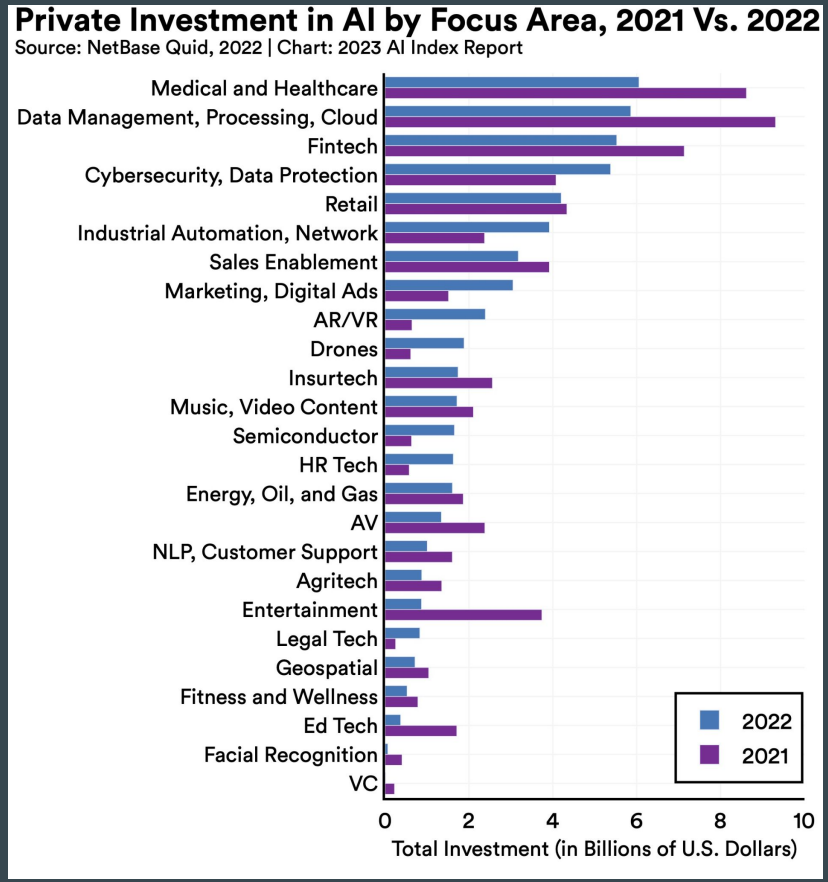
Is This Time Different?





Towards AGI


Improvement Over Time on Select AI Index Technical Performance Benchmarks
Source: AI Index, 2022 | Chart: 2023 AI Index Report
Overall Improvement
YoY Improvement
Improvement (%)
120%
100%
80%
60%
40%
20%
0%
ImageNet Top-1
FVRT
Celeb-DF
MPII
Cityscapes
Kvasir-SEG
STL-10
CIFAR-10
VQA
COCO
VCR
Kinetics-400
Kinetics-600
Kinetics-700
SuperGLUE
ReClor
arXiv
PubMed
ANLI
SST-5
MMLU
VoxCeleb
Procgen
Vision Image
Vision Video
Language
SR
RL
Benchmark


Private Investment in AI by Focus Area, 2021 Vs. 2022
Source: NetBase Quid, 2022 | Chart: 2023 AI Index Report
Medical and Healthcare
Data Management, Processing, Cloud
Fintech
Cybersecurity, Data Protection
Retail
Industrial Automation, Network
Sales Enablement
Marketing, Digital Ads
AR/VR
Drones
Insurtech
Music, Video Content
Semiconductor
HR Tech
Energy, Oil, and Gas
AV
NLP, Customer Support
Agritech
Entertainment
Legal Tech
Geospatial
Fitness and Wellness
Ed Tech
Facial Recognition
VC
2022
2021
0
2
4
6
8
10
Total Investment (in Billions of U.S. Dollars)

It was a dark and stormy night. Suddenly, a shot rang out. The sound echoed across the moors, ringing in the ears of those who were awake and alert enough t of travelers who were taking refuge i

through = 98.58%
throughout = 0.86%
across = 0.24%
off = 0.19%
in = 0.05%

Total: -6.04 logprob on 1 tokens
(99.92% probability covered in top 5 logits)

T hotgun in hand. He was a burly man, w face. He was known to be a no- n use his shotgun if necessary.

" area for any sign of danger.

The travelers, who had been huddled around the fire in the common room, jumped and looked at each other in fear. They had been on their way to the nearby city when the storm had forced them to take shelter at the inn. They were all strangers to each other, but in that moment, they were all connected by the sense of unease that hung in the air.

One of the travelers, a young woman with long, dark hair, stepped forward. "I

Model

gpt-3.5-turbo-ins...

Temperature 1

Maximum length 256

Stop sequences
Enter sequence and press Tab

Top P 1

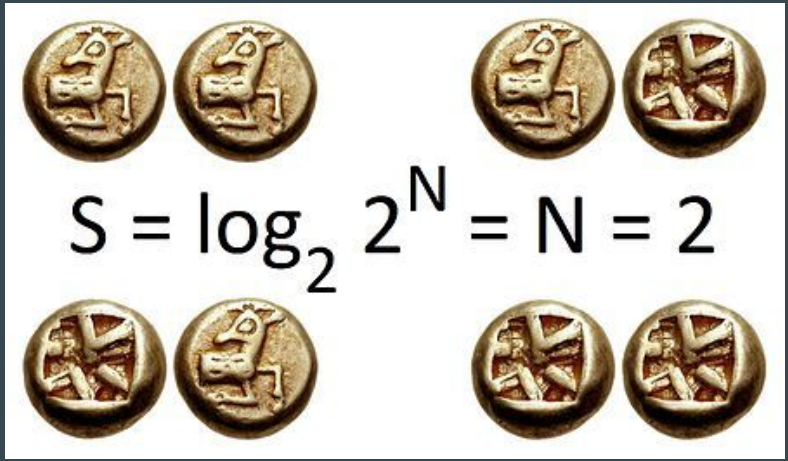
Frequency penalty 0

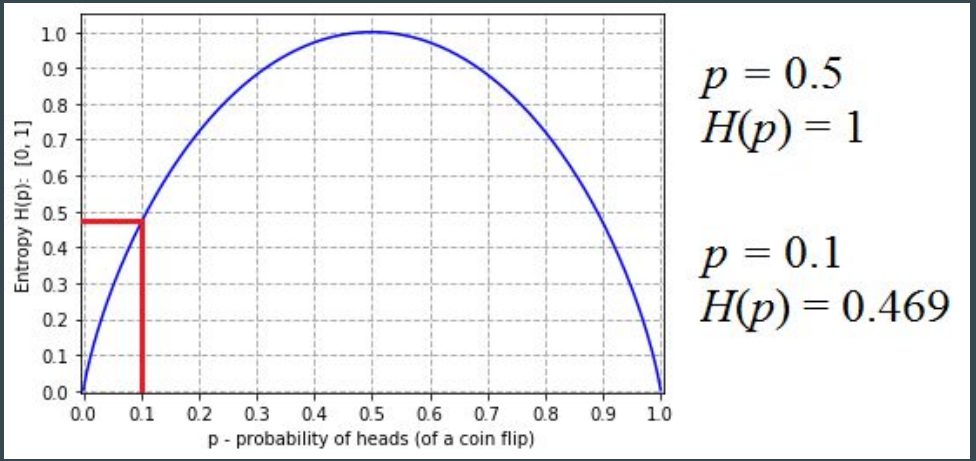
Presence penalty 0

Information

Entropy

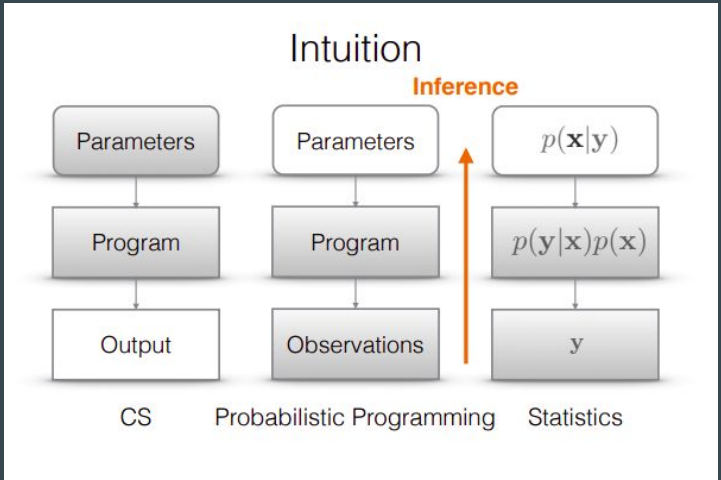
- Fundamental (mass or energy)
- Metric of uncertainty/disorder (no time, isolation, events-messages)
- Compression ~ Intelligence (id patterns, efficient representations, feature extraction, prediction, generalization)

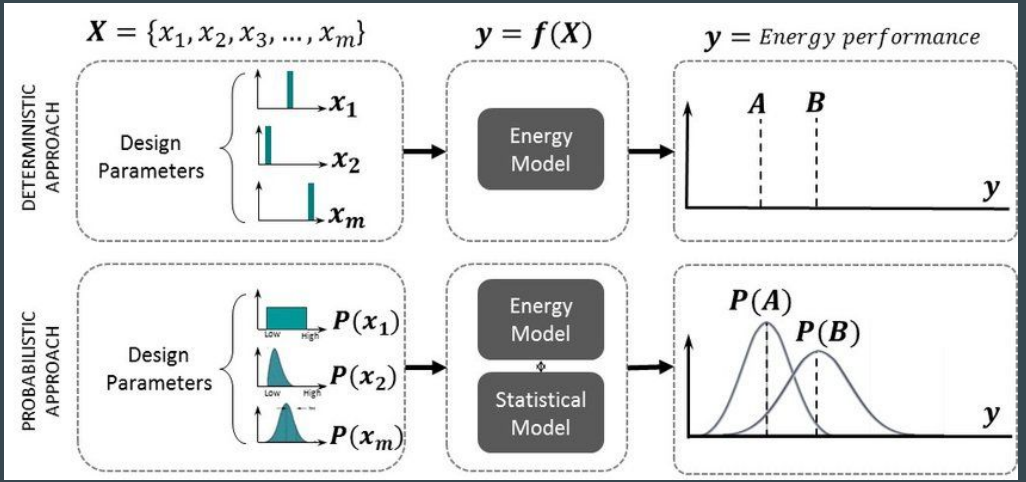




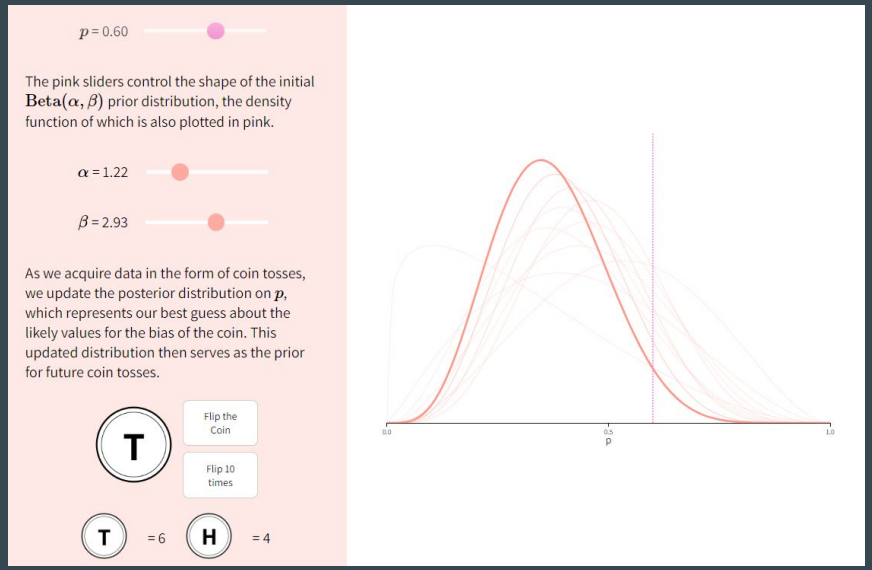
Probabilistic Thinking

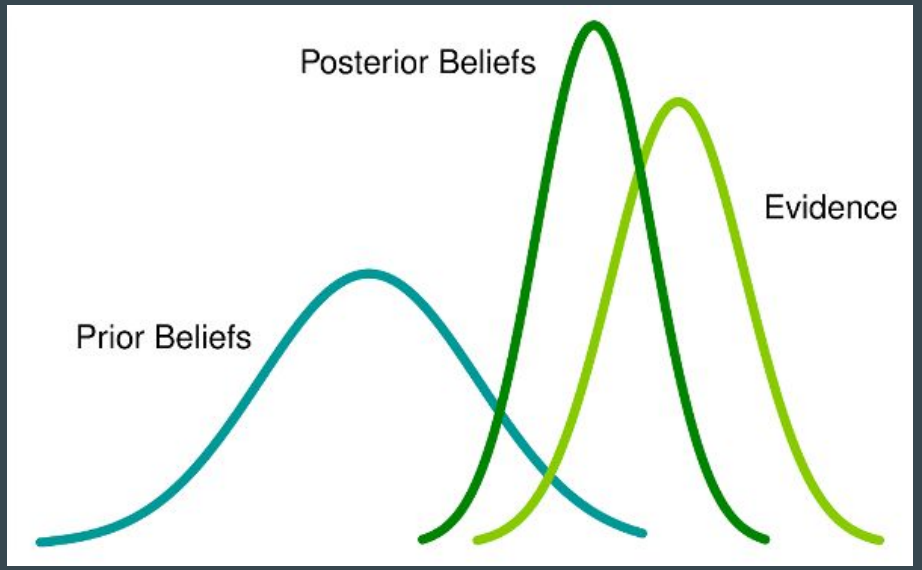
- Deterministic to Probabilistic
- Point Value to Distributions
- Implicit to Explicit Approximations



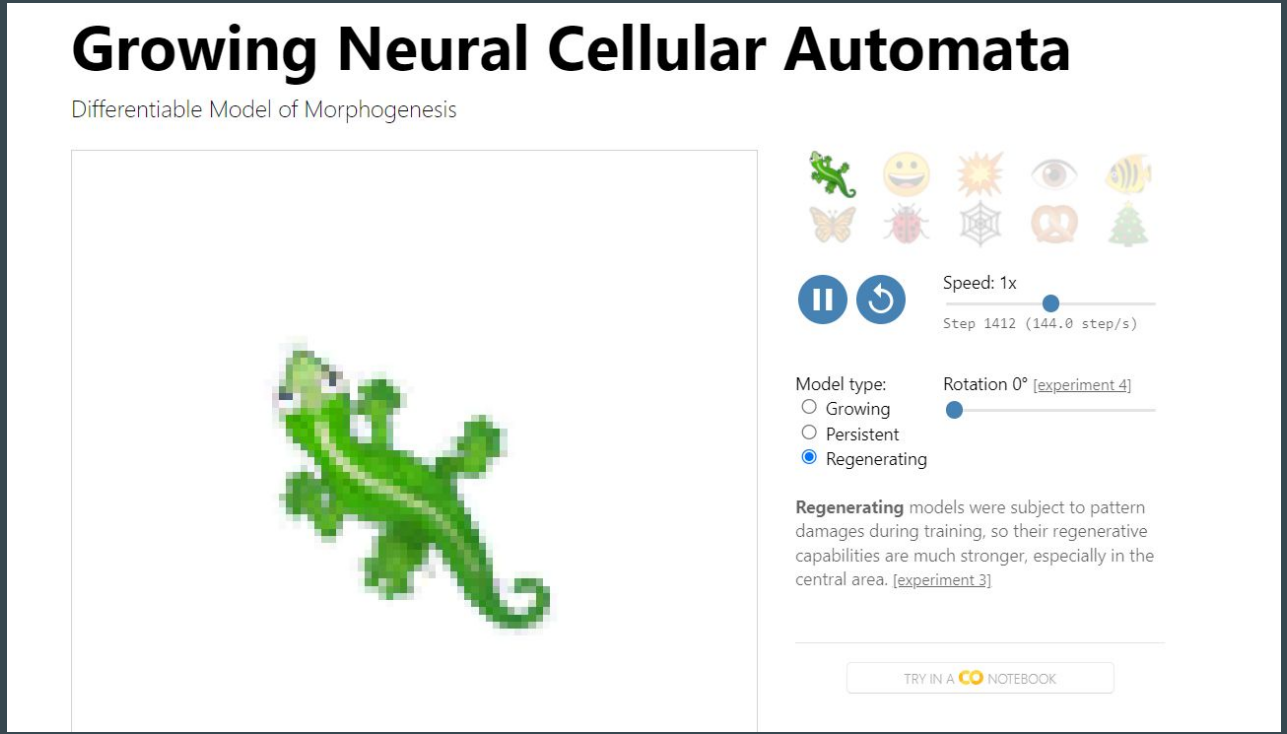


Distribution Sampling & Types

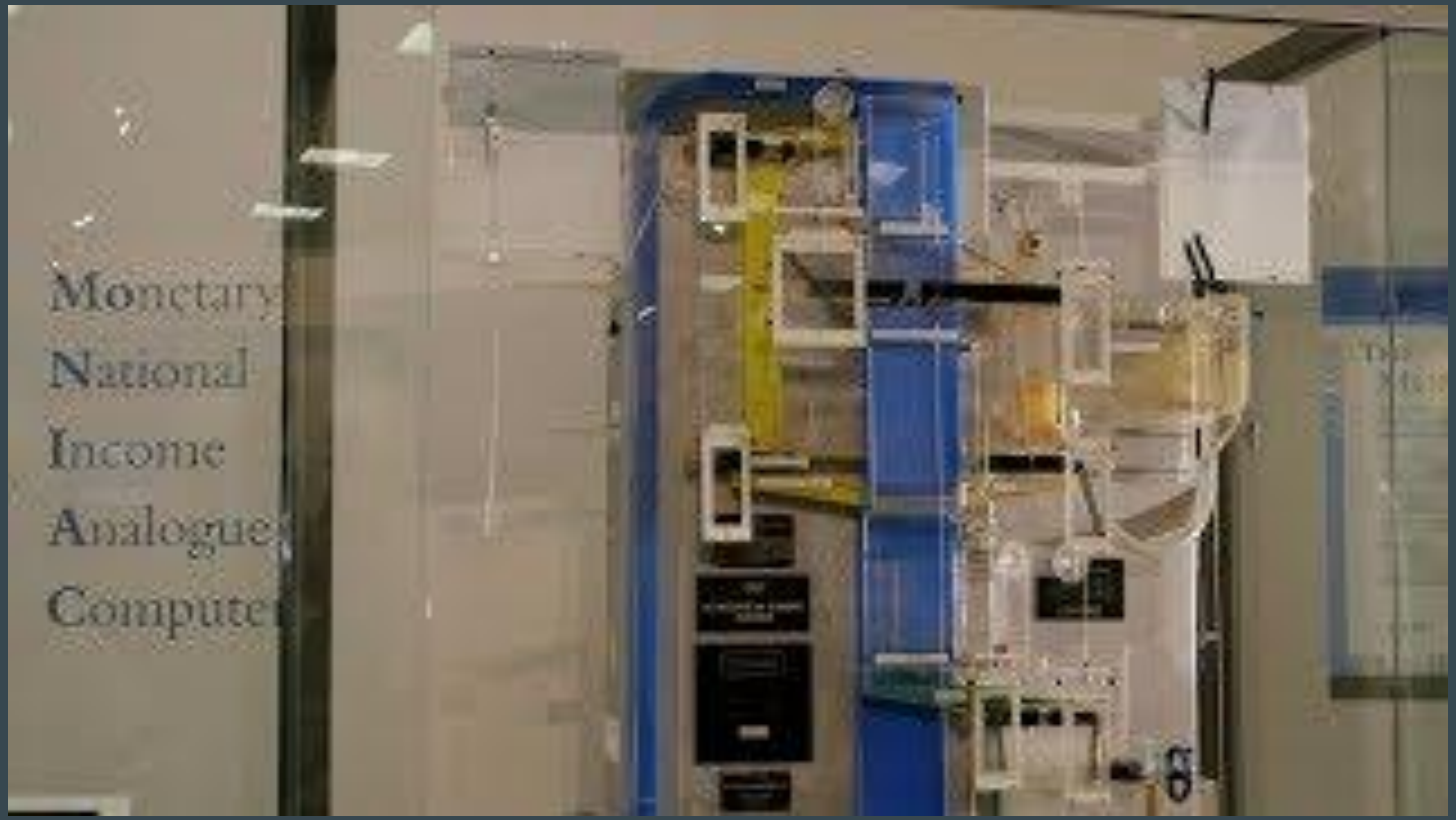




Information: Data is Code is Data

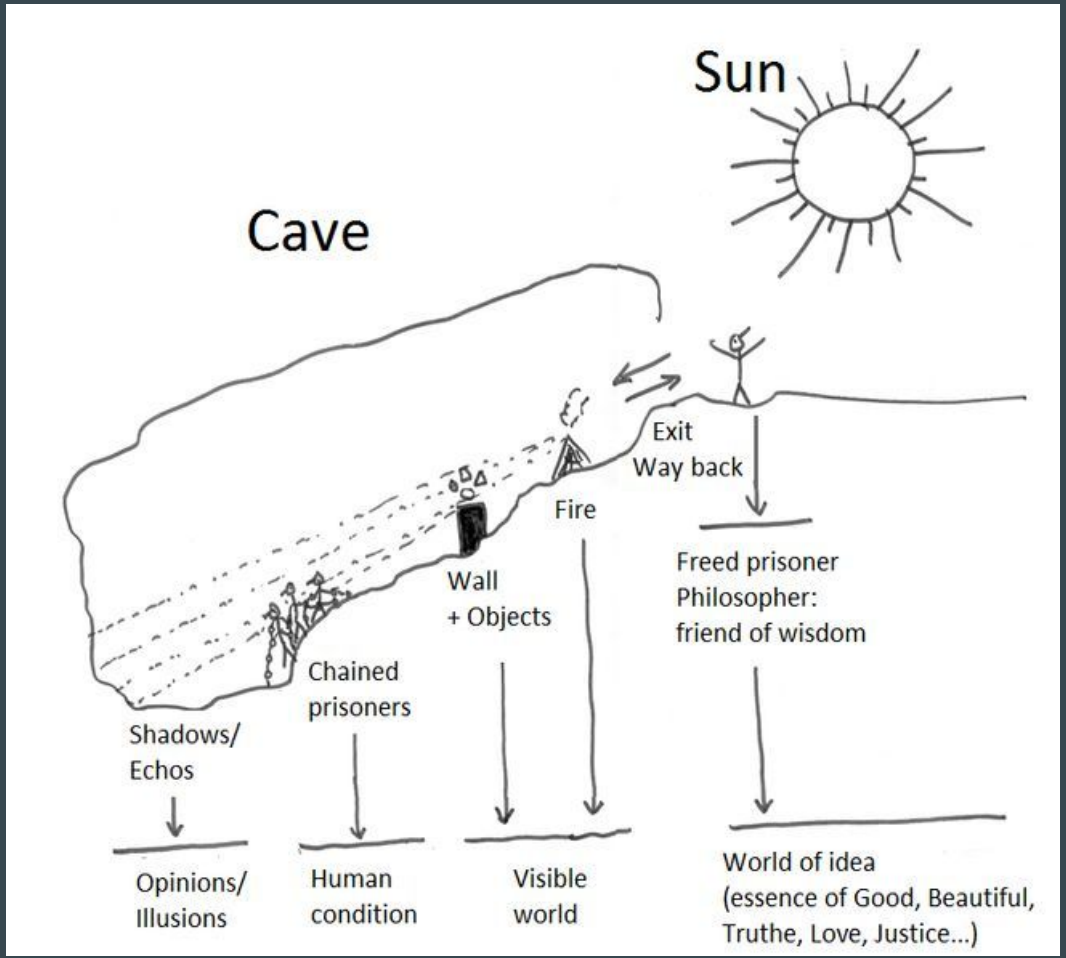


Information > Modality: MONAIC vs TensorFlow

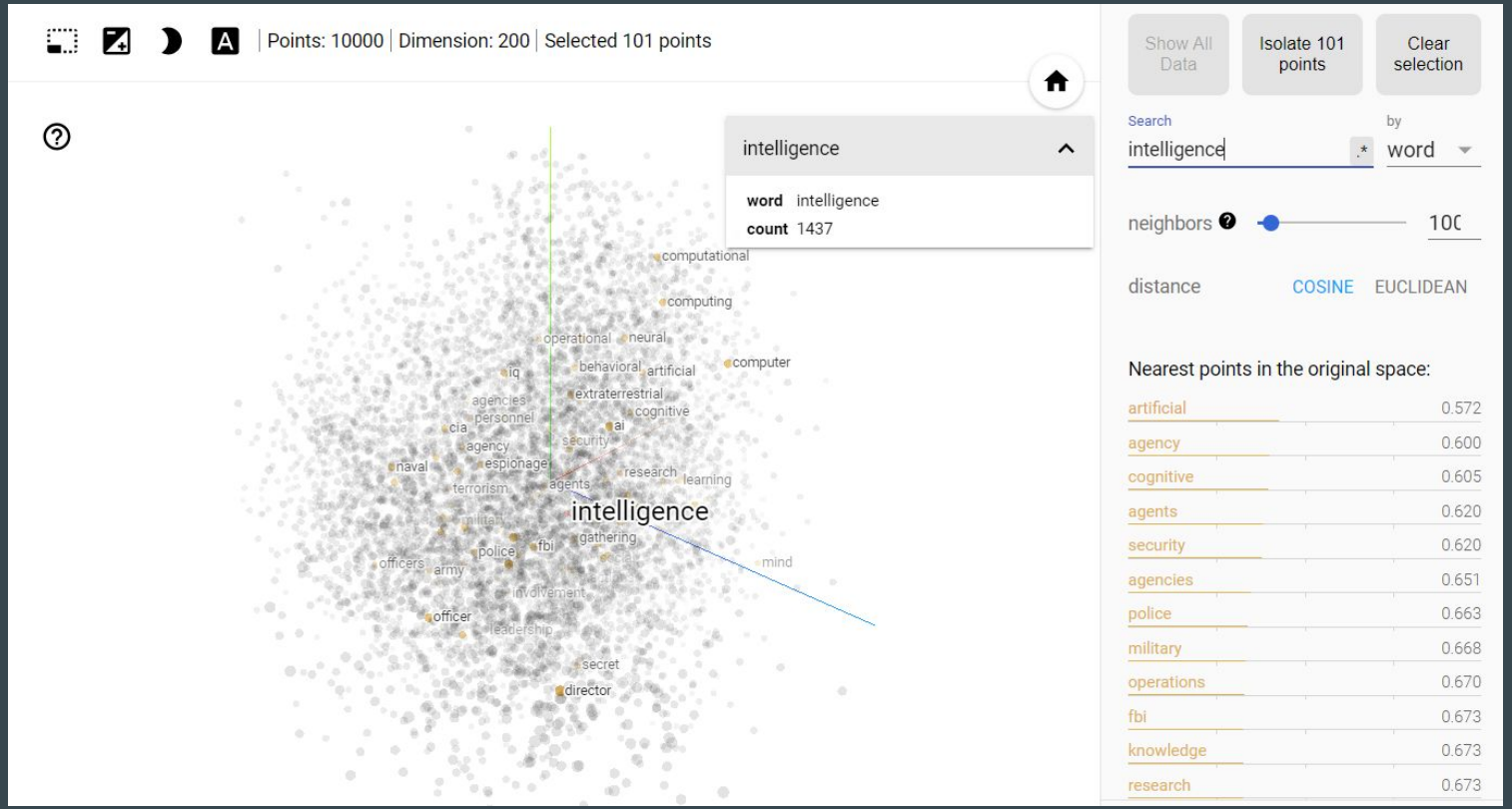


Representation

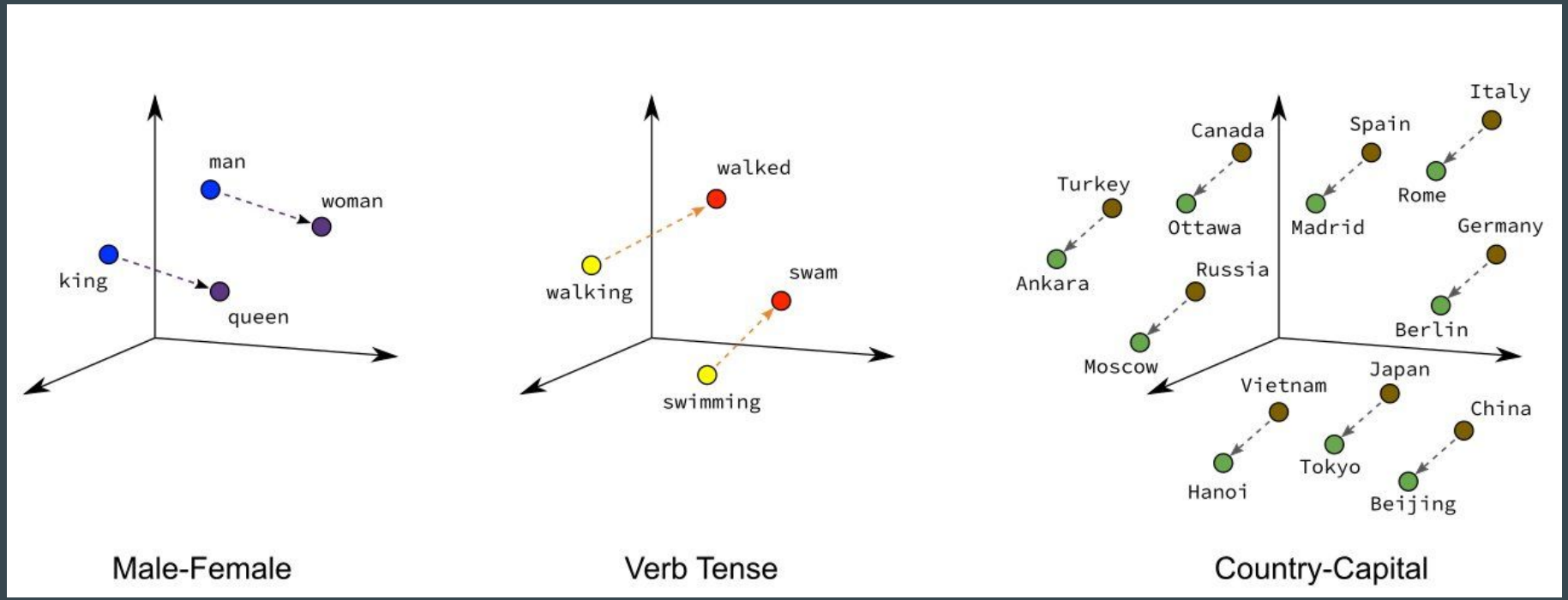
Plato's Cave


Sun
Cave
Exit
Way back
Fire
Wall
+ Objects
Freed prisoner
Philosopher:
friend of wisdom
Chained
prisoners
Shadows/
Echos
Opinions/
Illusions
Human
condition
Visible
world
World of idea
(essence of Good, Beautiful,
Truthe, Love, Justice...)

Revealing Latent Semantic Structure in Language



Semantically Meaningful Dimensions

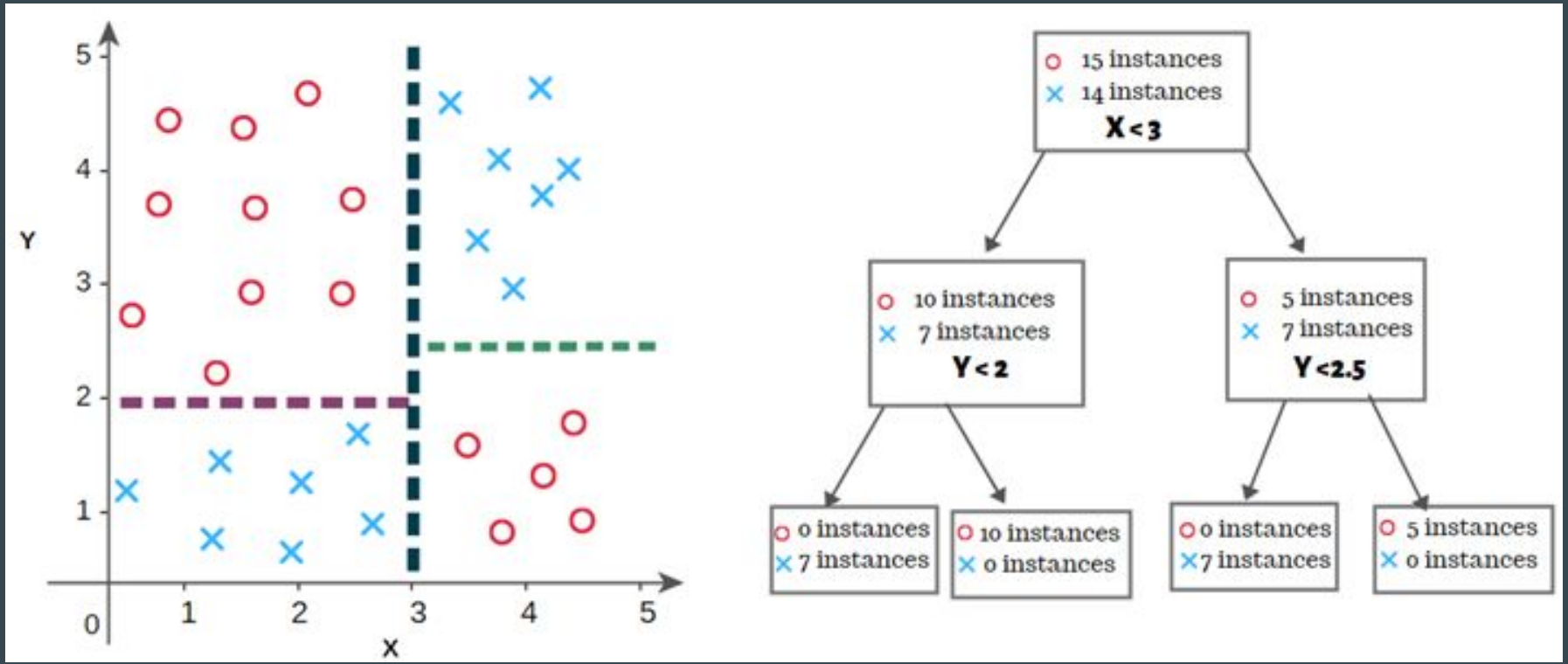


Latent < [Semantic] < Manifolds

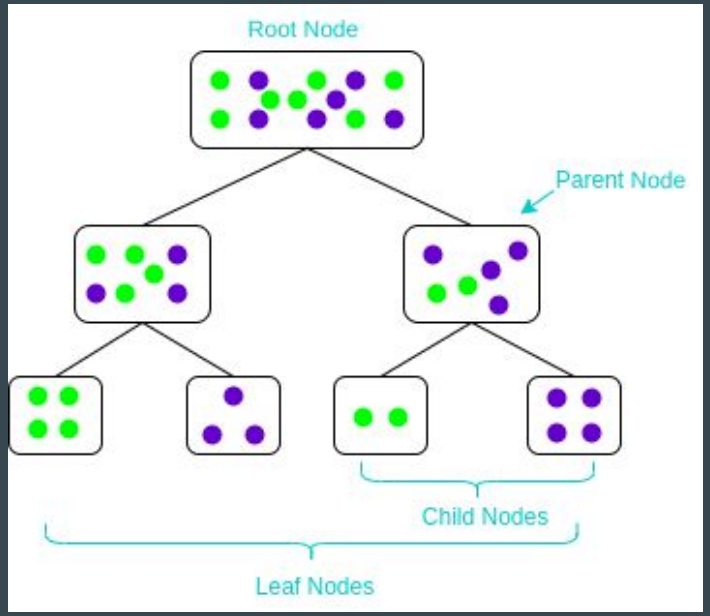
- OPERATION:
  - 1. Up-sampling Embeddings (Viz)
  - 2. Down-sampling Dimensionality Reduction/Projection (3D to 2D, pros/cons)
  - 3. Change of Basis
  - 4. Structure: None, Algebraic, and Geometric/Topological
- GOAL: Feature Engineering & Semantic Algebra (Viz)
- CHALLENGES:
  - Coverage
  - OOV
  - Sample Efficiency
  - High Dimensionality (sparsity/curse of dimensionality, overfitting, stability, interpretability, etc.)
- RESEARCH
  - Cognition as 1. goal-oriented transformations of in latent space and/or 2. directed movement within

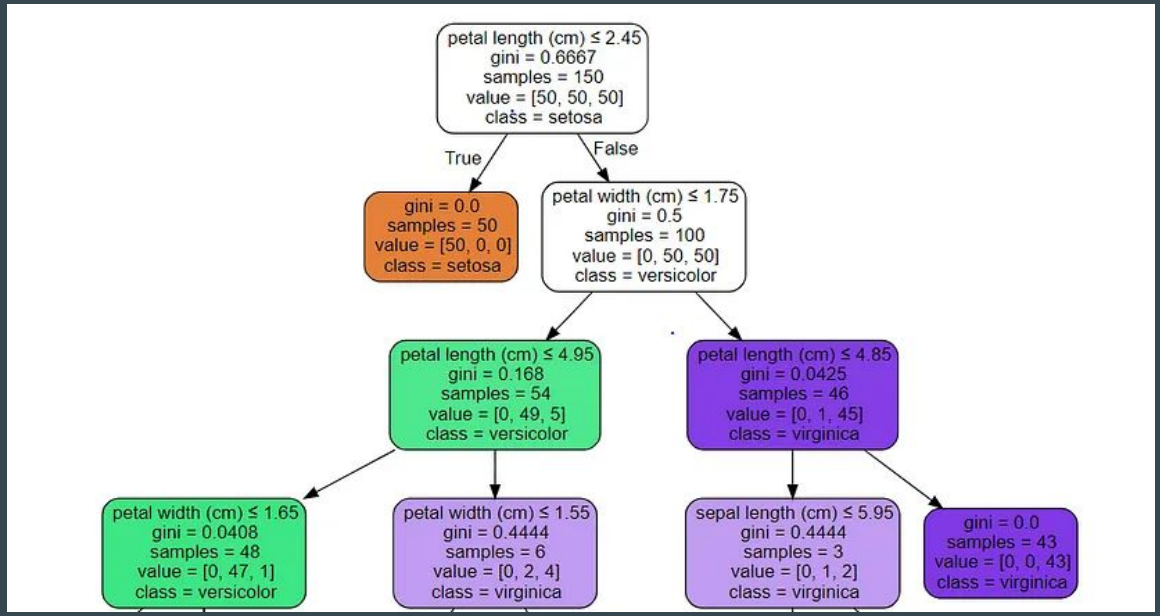
Transformations

Decision Trees (GOFAI Linear Decision Boundaries)



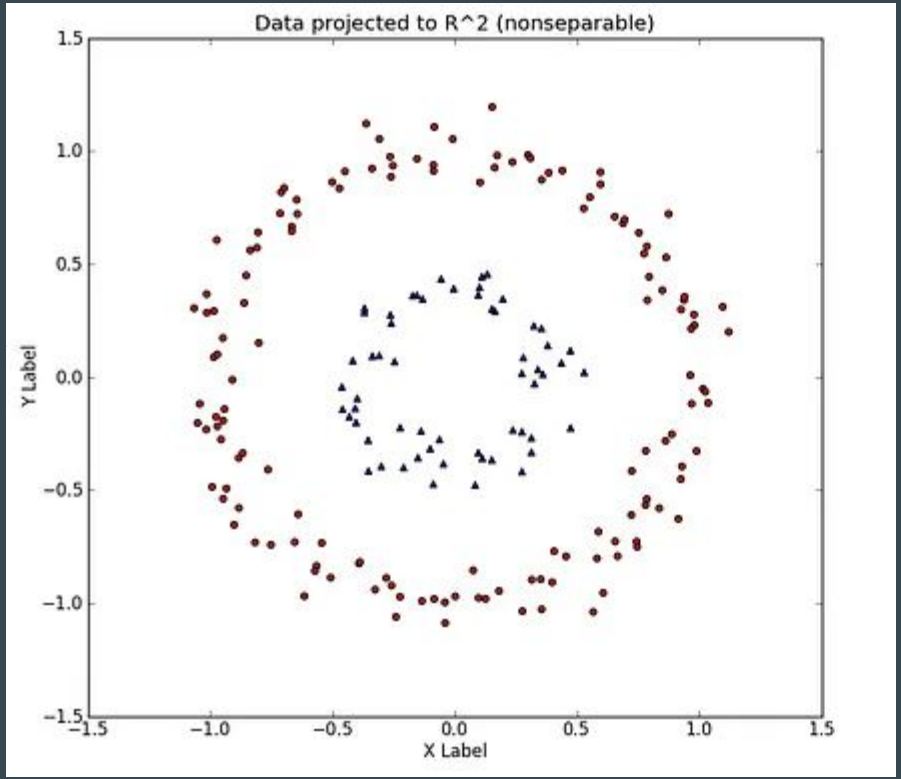
Objective Function: Min(Entropy) in Children Nodes



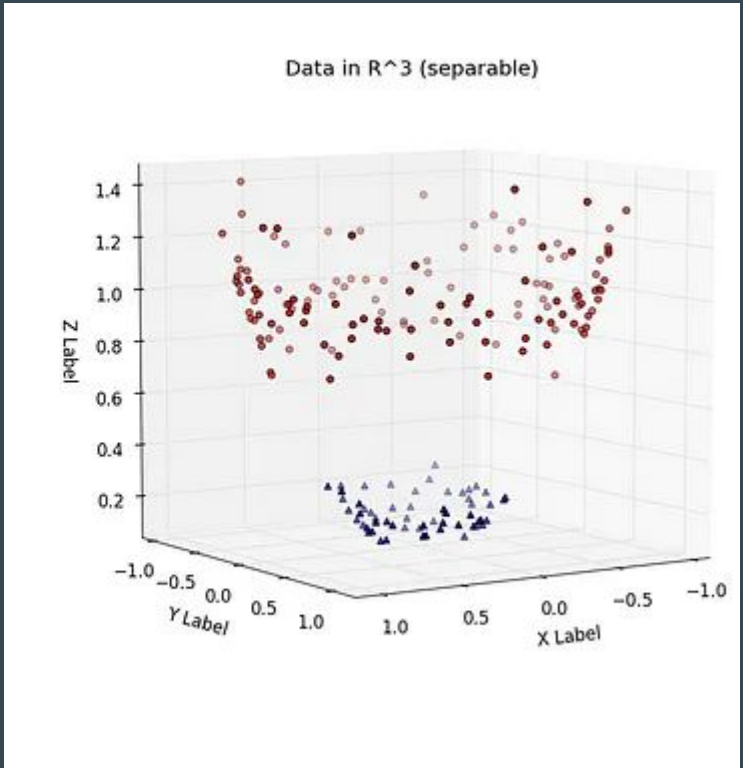


Linear Classifier

(Transform to enable Linear separation)



Transform to Higher Dimension

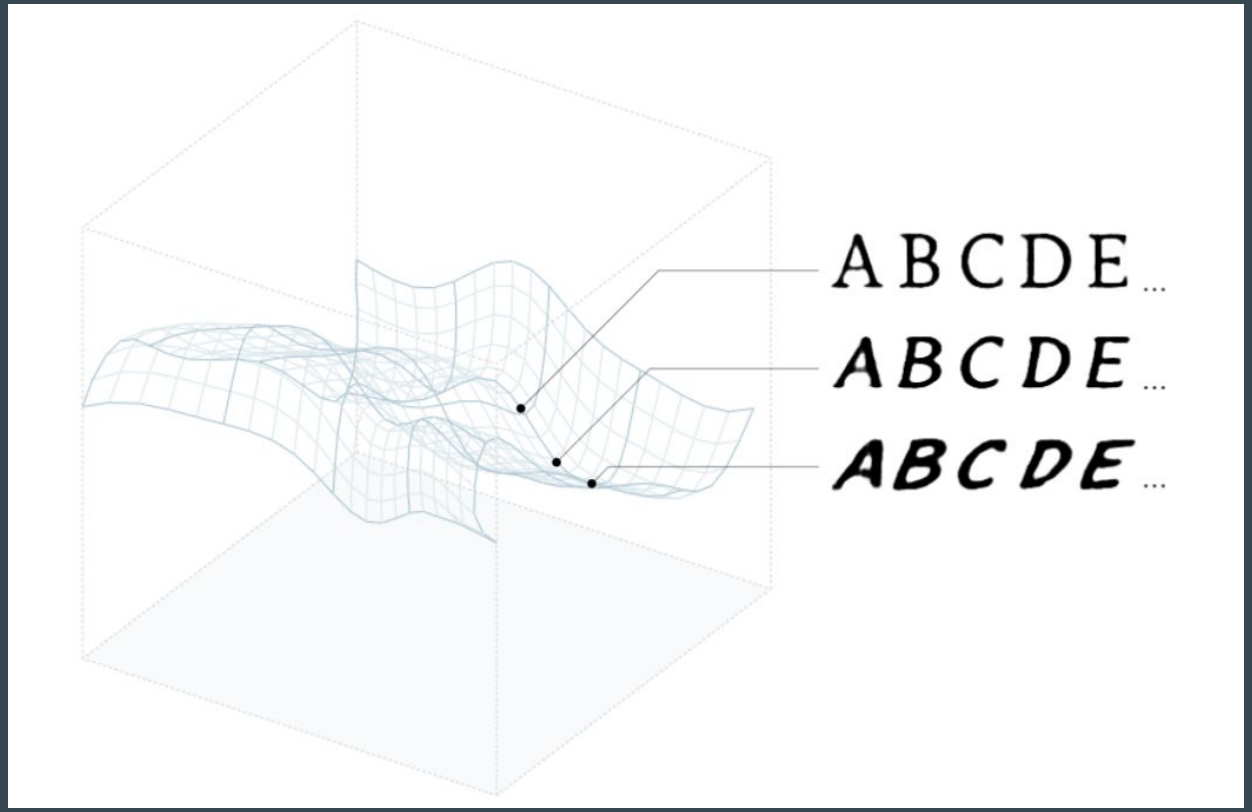


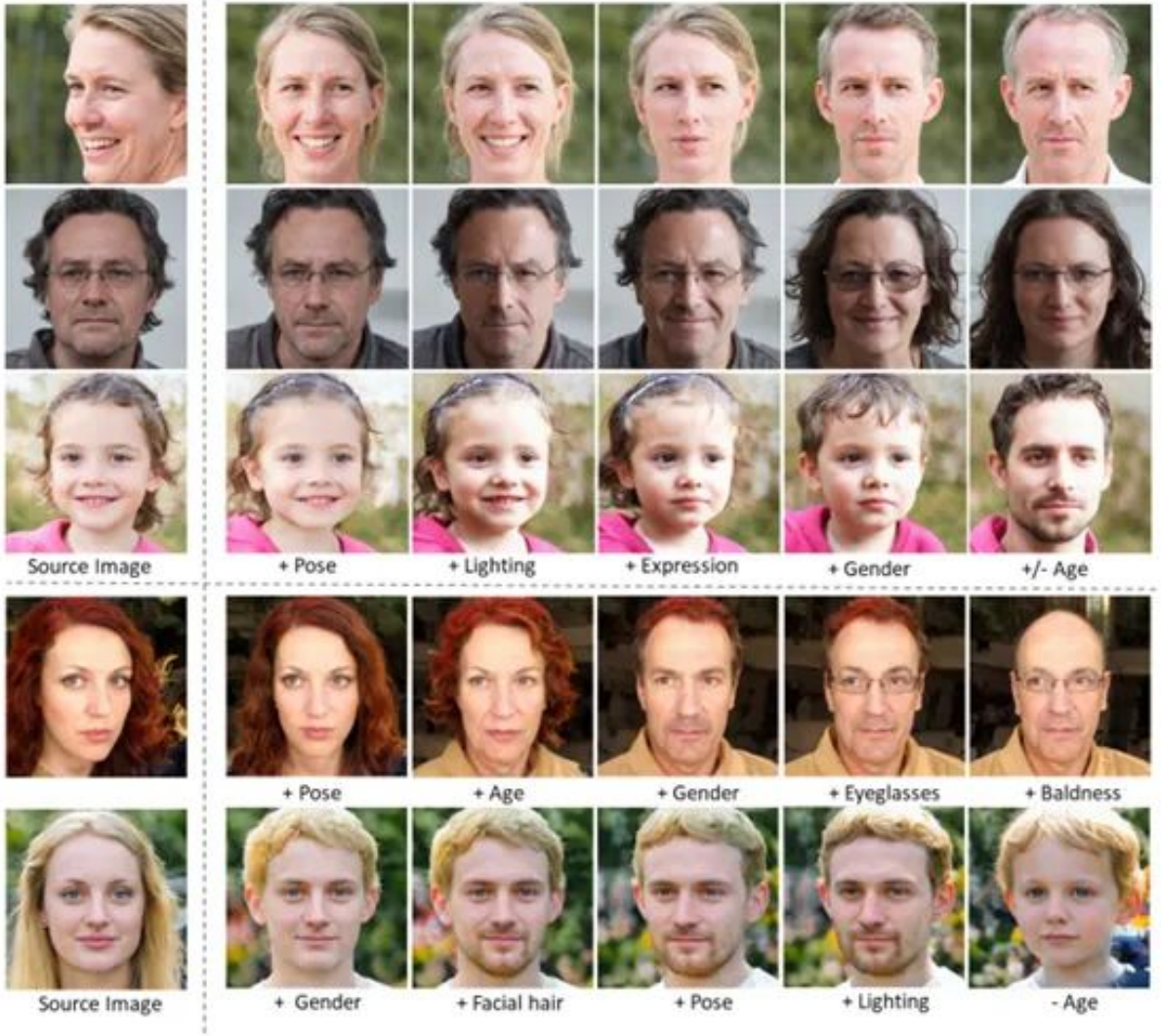
Superposition

Find 3 Transformations to Meaningful Semantic Spaces

Working in Semantic Space

Semantic Spaces with Meaningful Dimensions




Source Image
+ Pose
+ Lighting
+ Expression
+ Gender
+/- Age
+ Pose
+ Age
+ Gender
+ Eyeglasses
+ Baldness
Source Image
+ Gender
+ Facial hair
+ Pose
+ Lighting
- Age

Latent Space of Models Encoded with Language

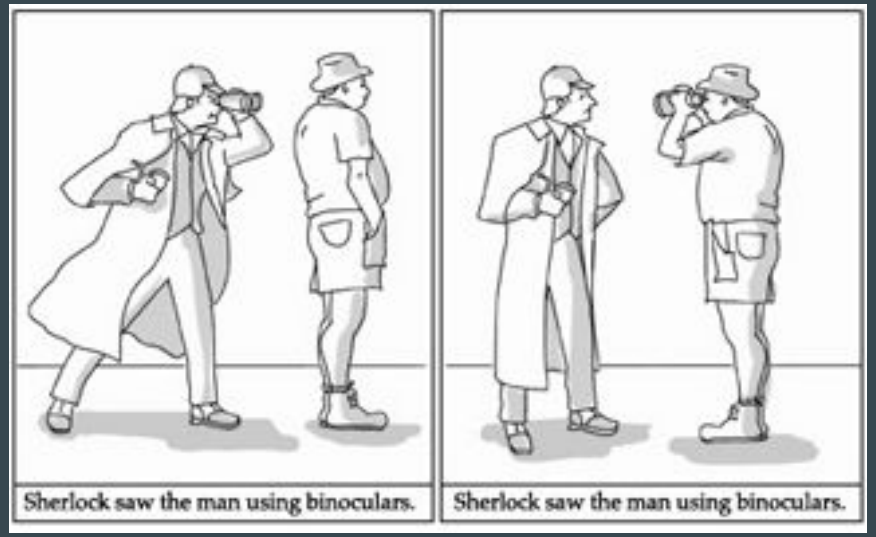
"Sherlock saw the man with binoculars."

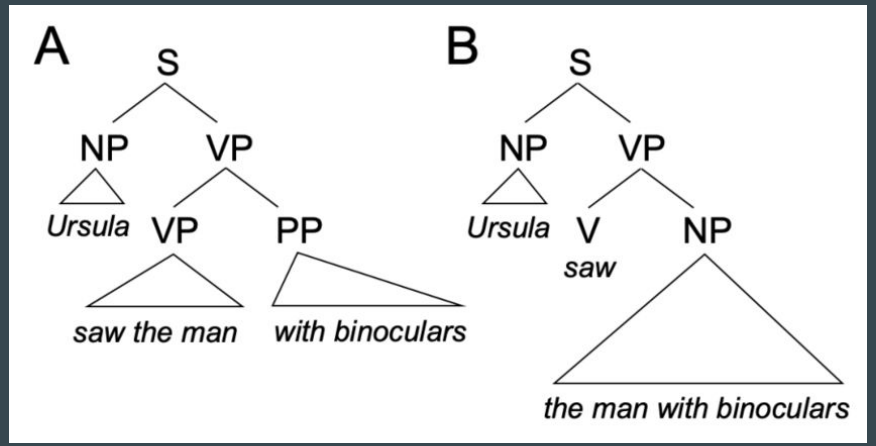
(1-5% Aphantasia)

Natural Language

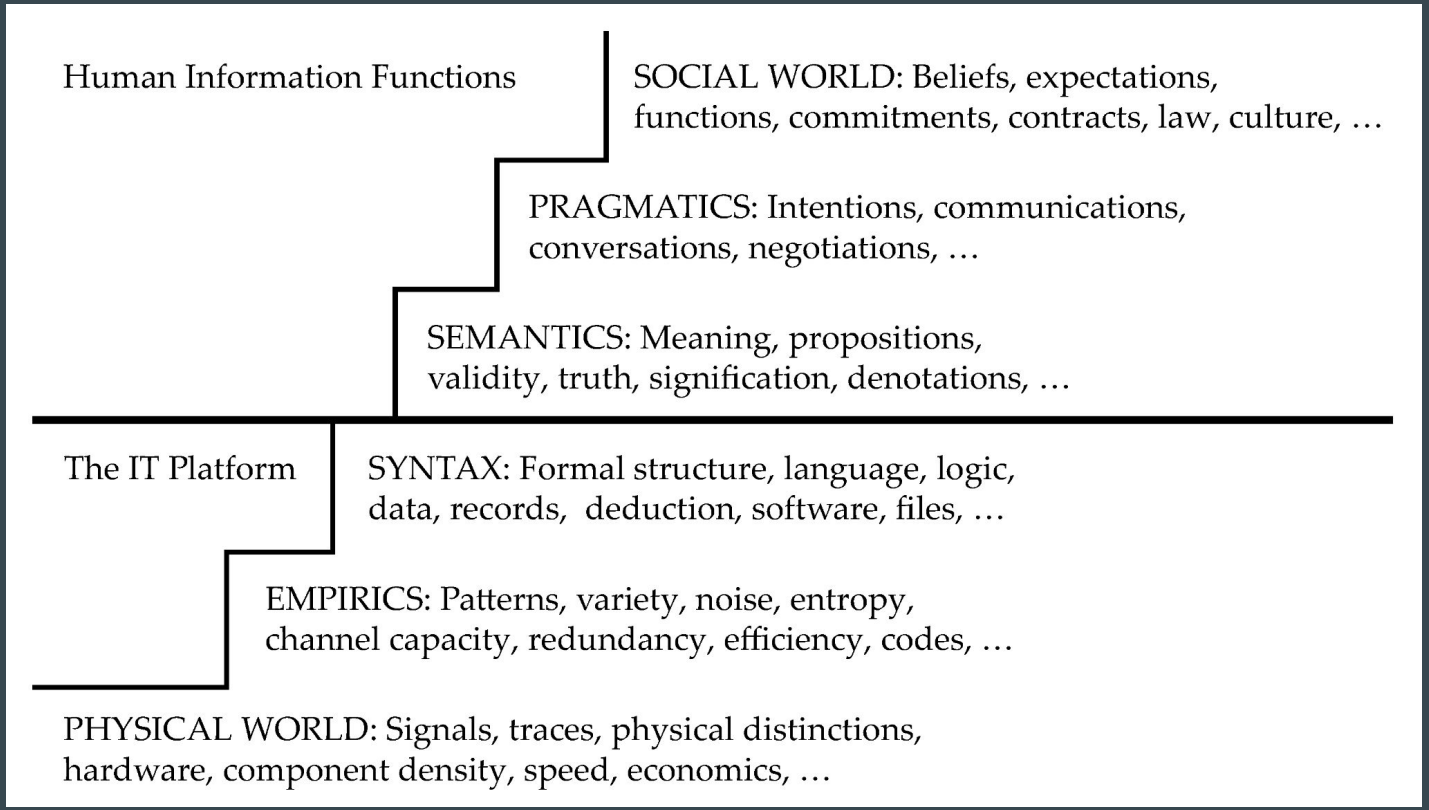
Vs

Constructed Languages (Examples?)



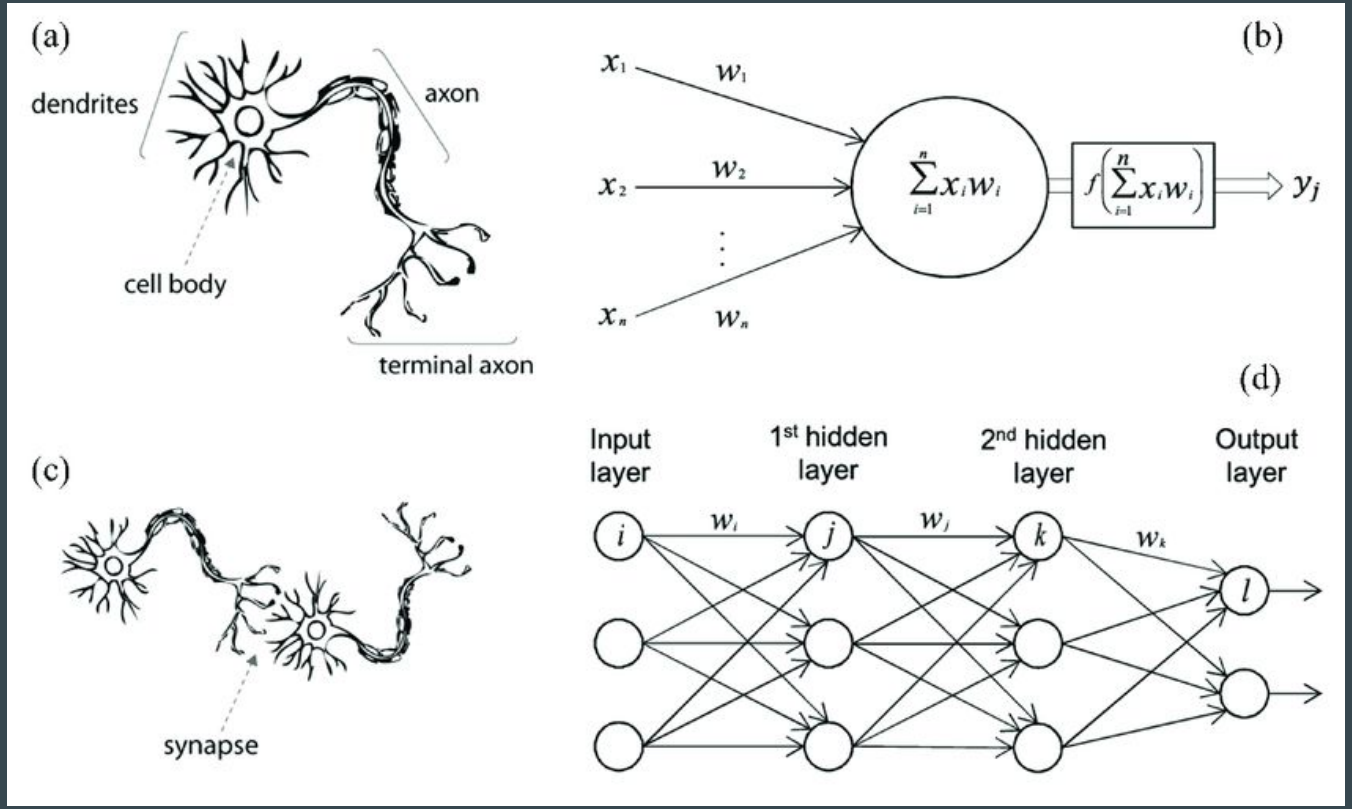


Semiotics to Linguistics to NLP

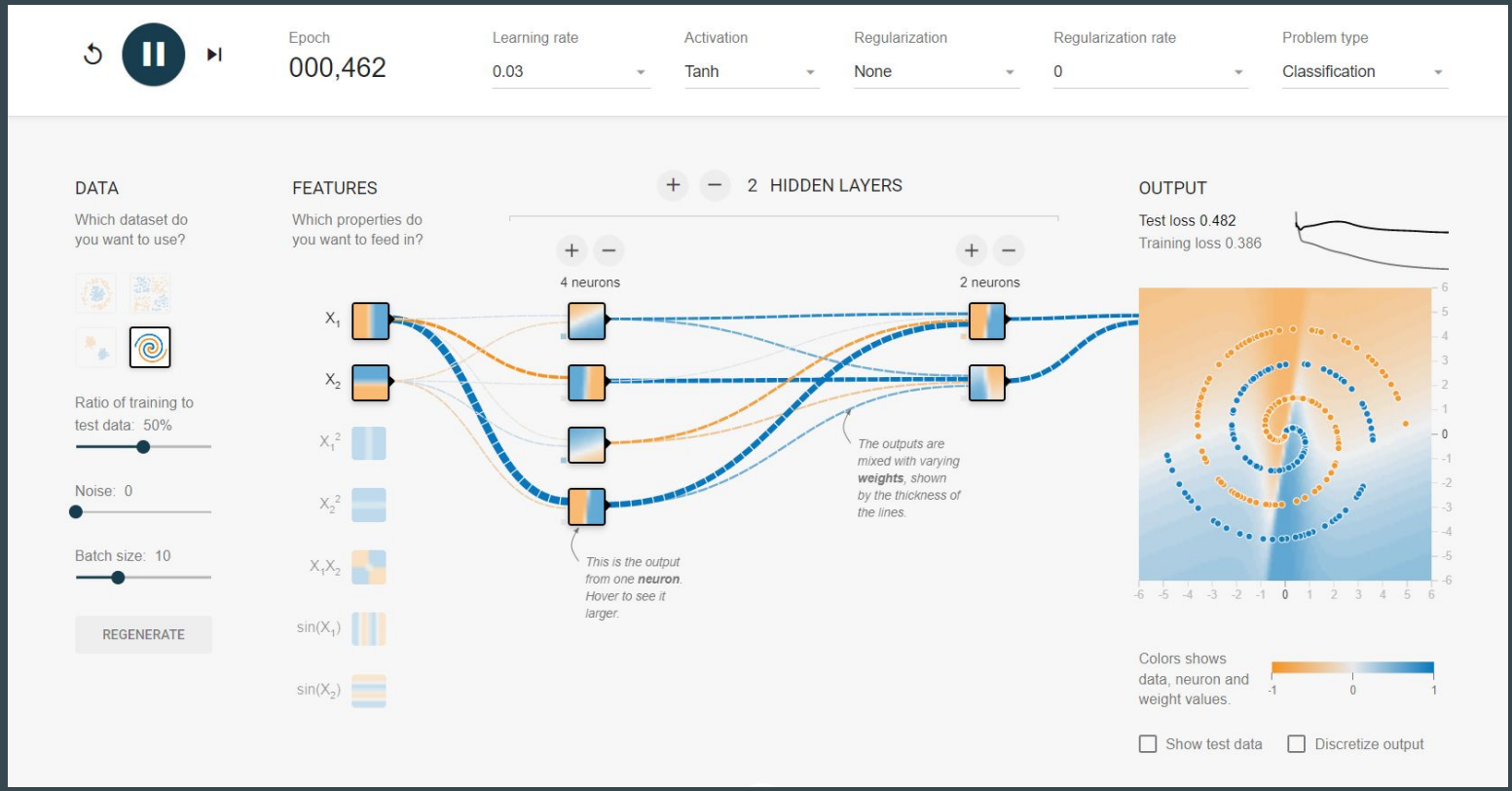


Learning

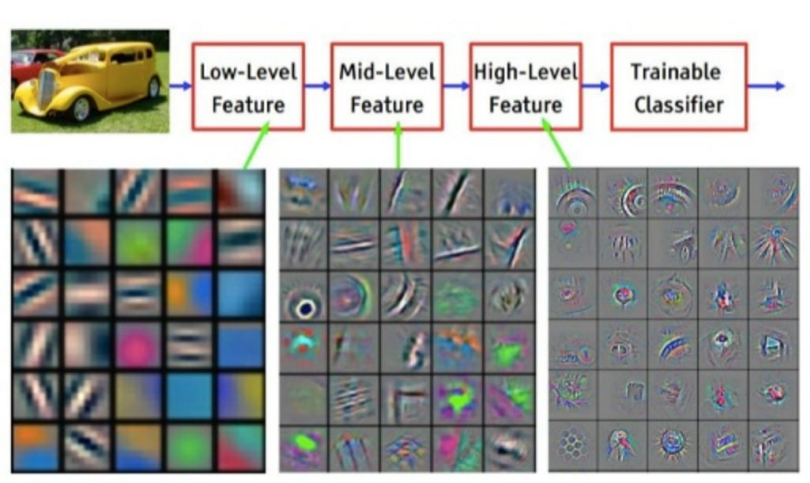
Deep Neural Networks

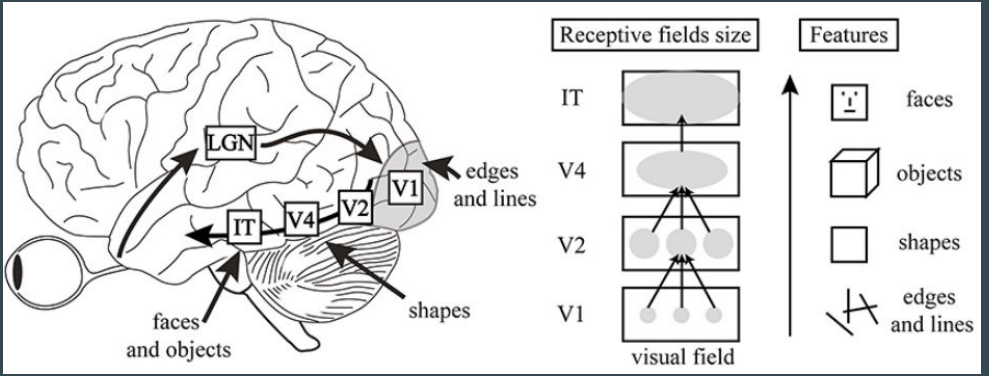


Universal Function Approximator

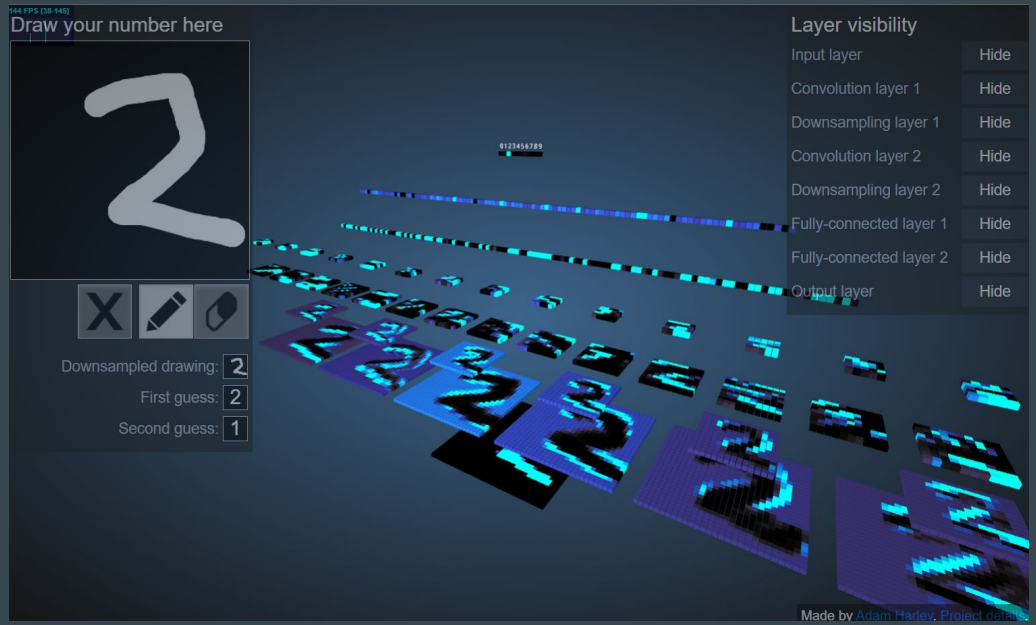


Efficient Architecture for Spatial Feature Detection (CNN)





CNN for Spatial Processing

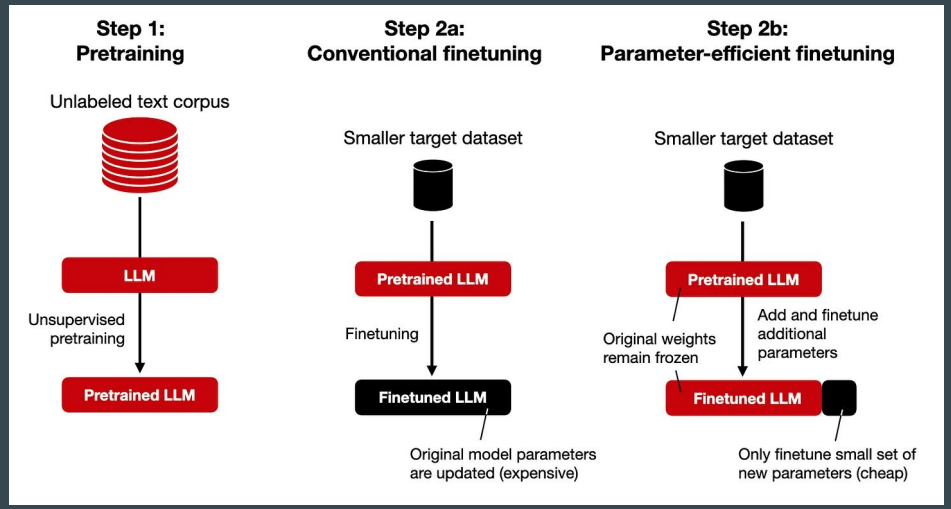


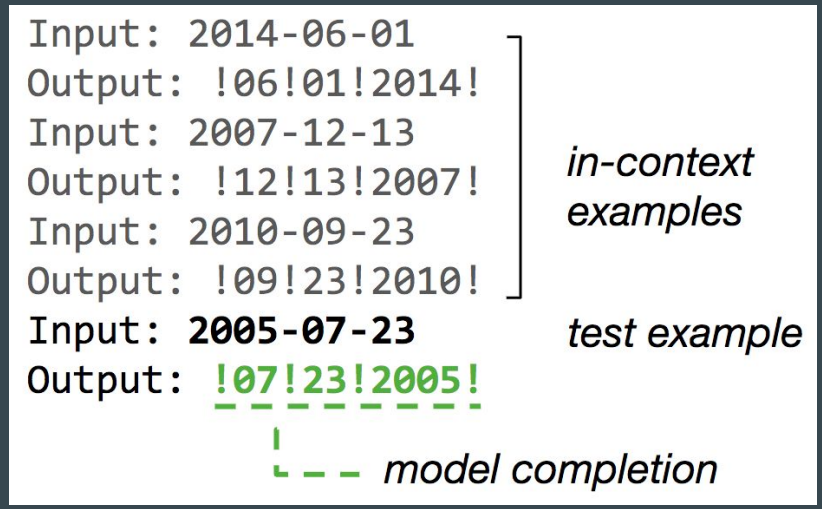

```
model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

model.summary()
```

Injecting Information into LLMs

- Pre-Training: autoregressive next word prediction (Language)
- Fine-Tuning: instruct patterns (Skills/Behaviors)
- Prompting: n-shot In-Context Learning (ICL)
- Tools (RAG, DB/KB, etc)

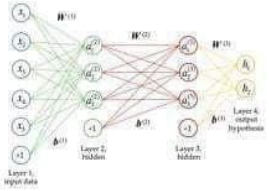




Reasoning

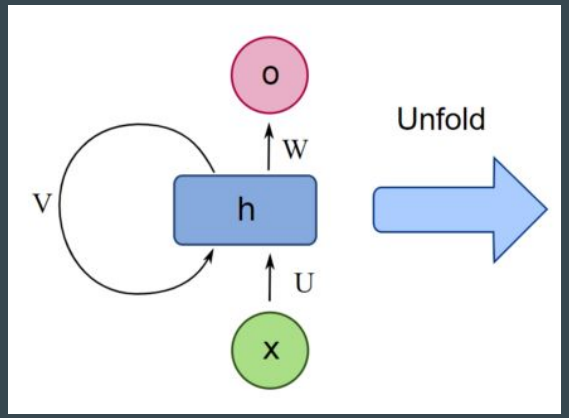
LLM: Reasoning

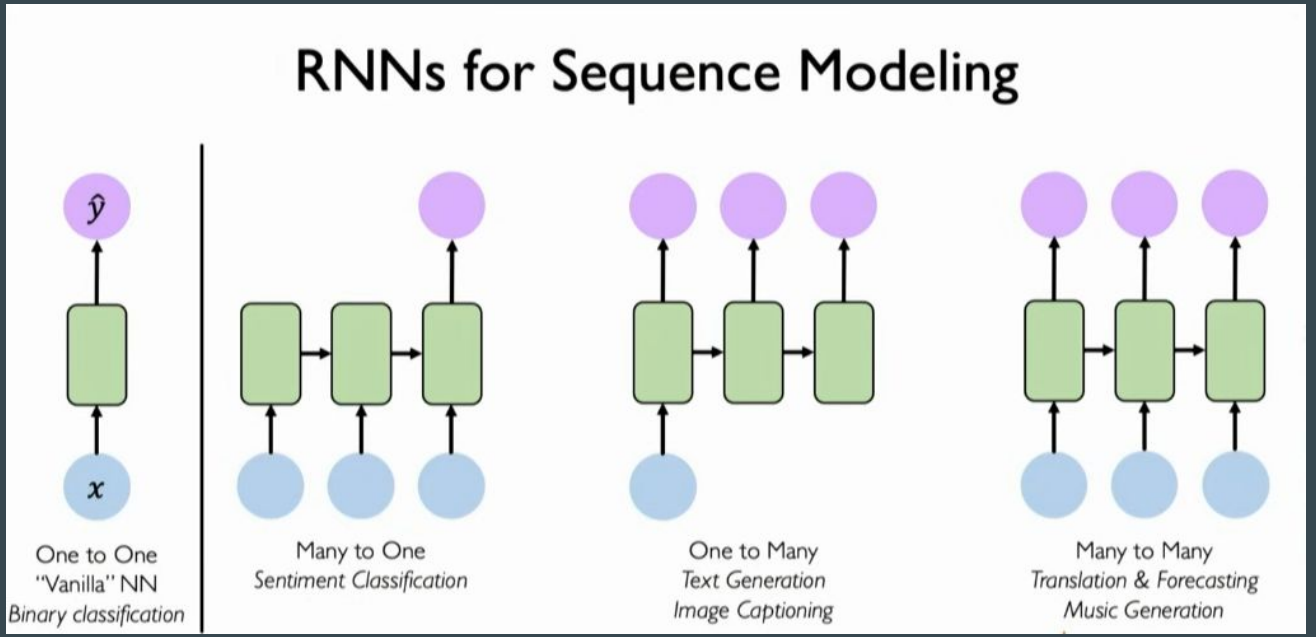
- Prompt Strategies:
  - CoT
  - Reflection
  - ReAct
- Agents
  - Rational
  - Collaborative
  - Tool Use/Making
  - Embodied
  - Autonomous

	A parrot	Machine learning algorithm
		
Learns random phrases	✅	✅
Doesn't understand [redacted] about what it learns	✅	✅
Occasionally speaks nonsense	✅	✅
Is a cute birdie parrot	✅	❌

LLM Architecture

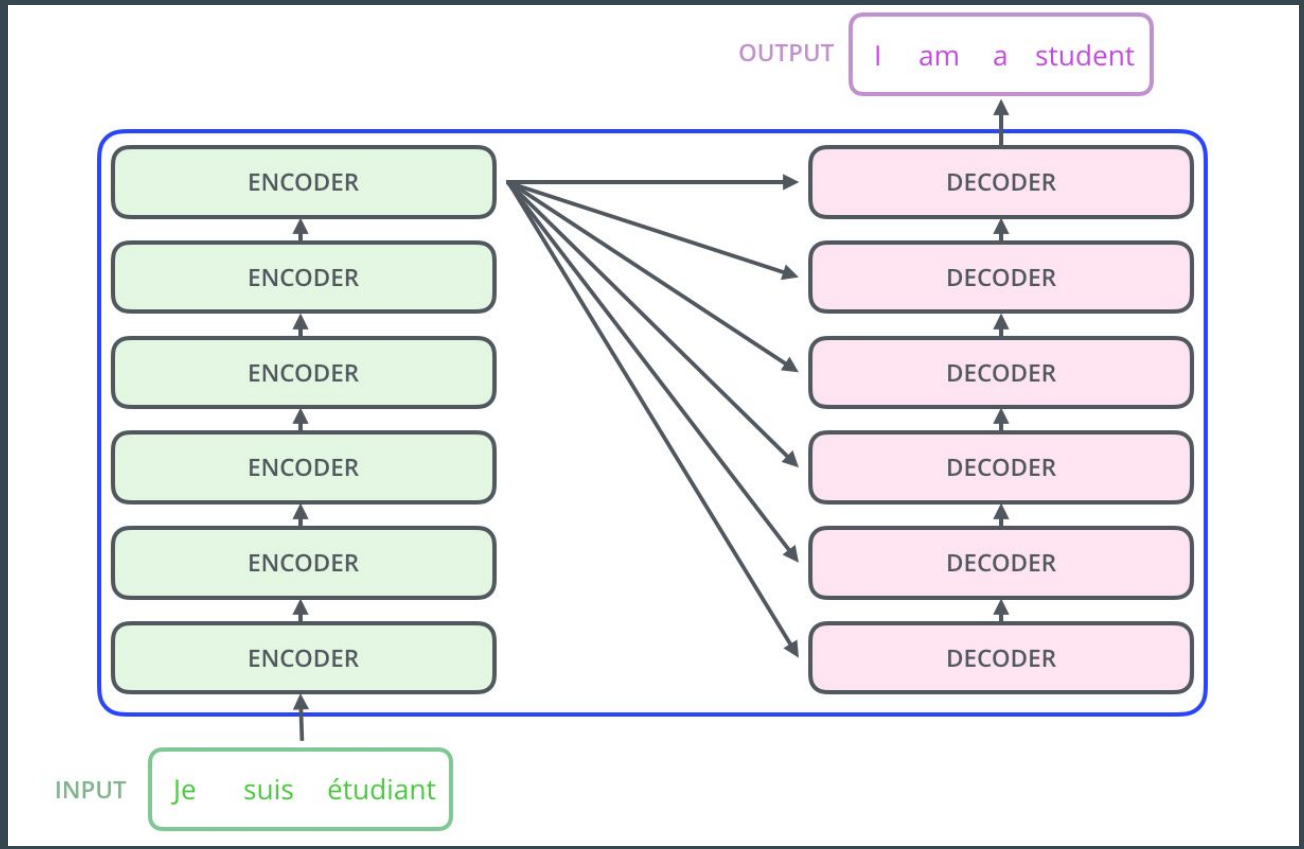
Efficient Architecture for Temporal Feature Detection (RNN)





Transformer Architecture

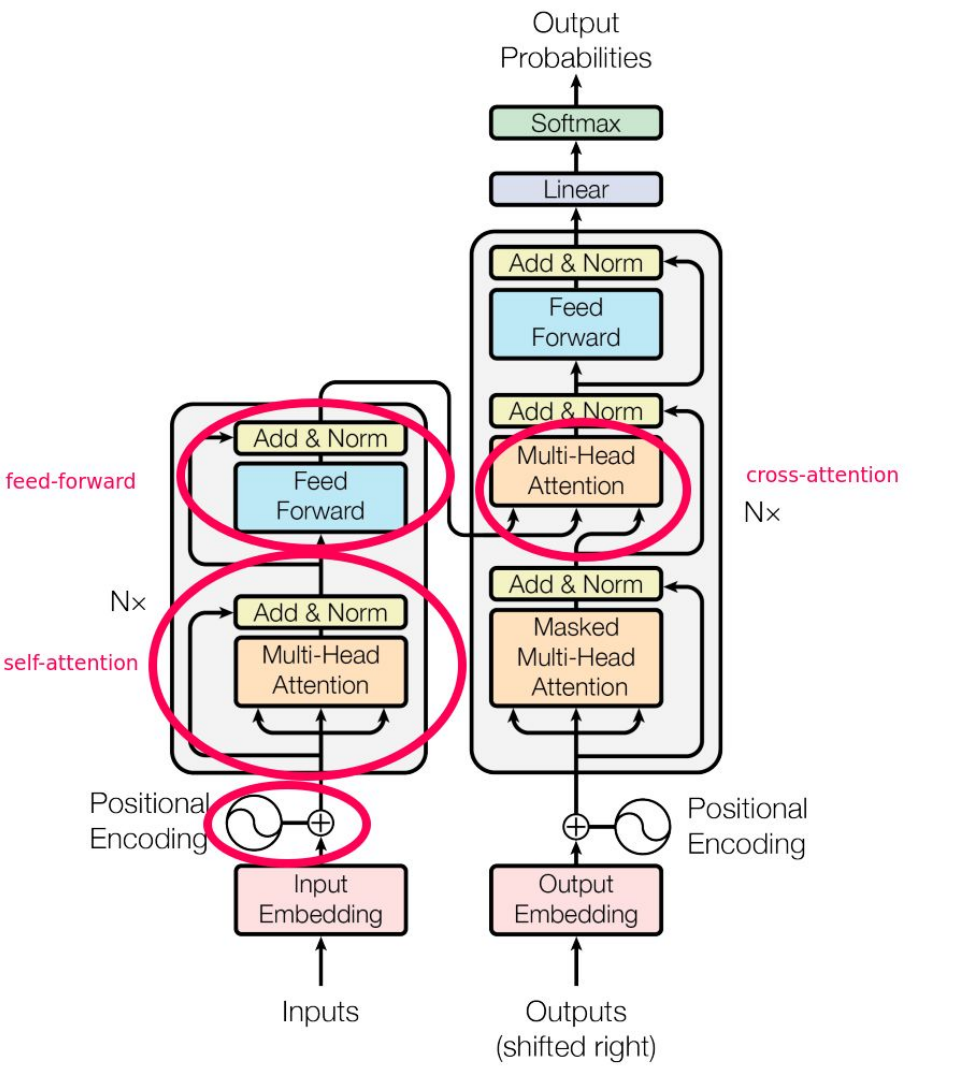
Spatial
or
Temporal
in any
Modality



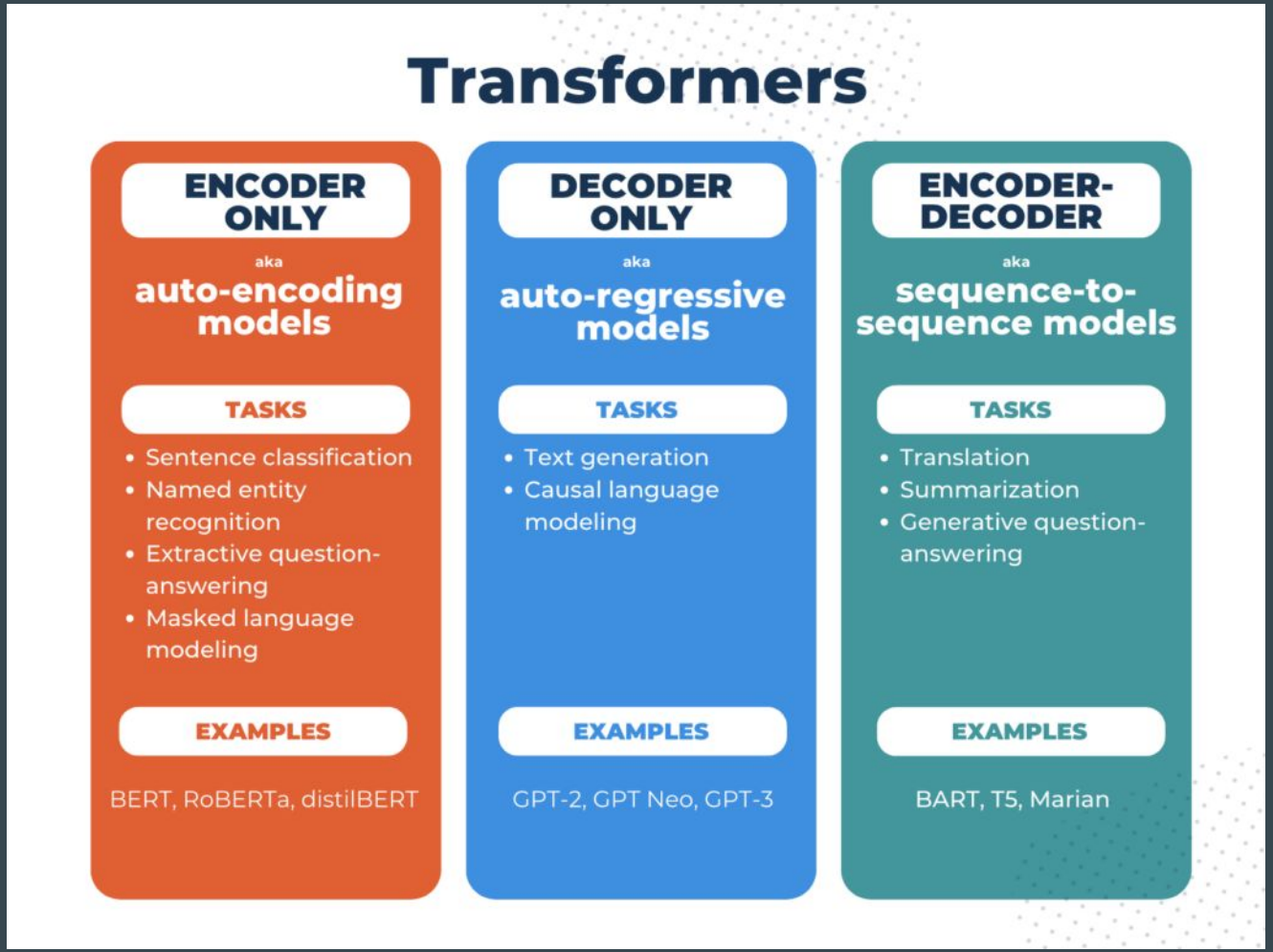
Transformers

Hyperparameters:

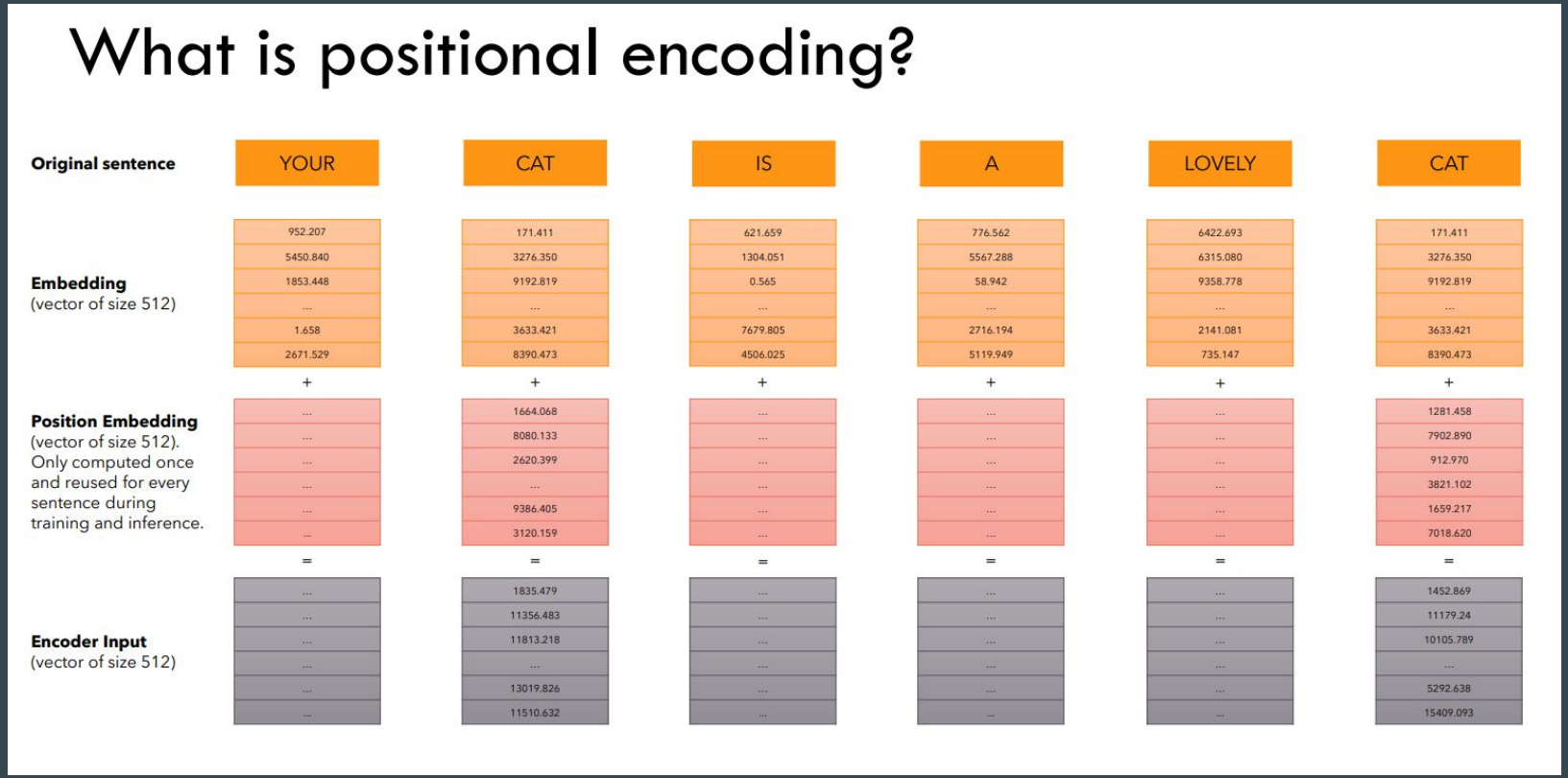
- Vocabulary (~30k-50k)
- Context Window (2k to +100k)
- hx Heads
- Nx Encoders
- Nx Decoders



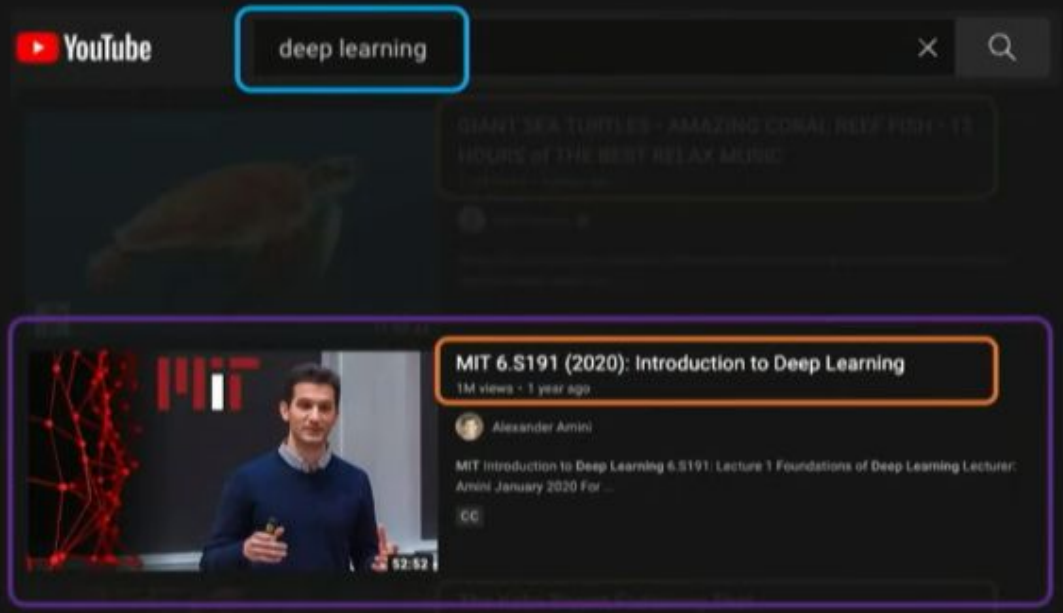
3 Variants


Transformers
ENCODER ONLY
aka
auto-encoding models
TASKS
• Sentence classification
• Named entity recognition
• Extractive question-answering
• Masked language modeling
EXAMPLES
BERT, RoBERTa, distilBERT
DECODER ONLY
aka
auto-regressive models
TASKS
• Text generation
• Causal language modeling
EXAMPLES
GPT-2, GPT Neo, GPT-3
ENCODER-DECODER
aka
sequence-to-sequence models
TASKS
• Translation
• Summarization
• Generative question-answering
EXAMPLES
BART, T5, Marian

Token Embeddings + Positional Encoding



Understanding Attention with Search



Query (Q)

Key (K_1)

Key (K_2)

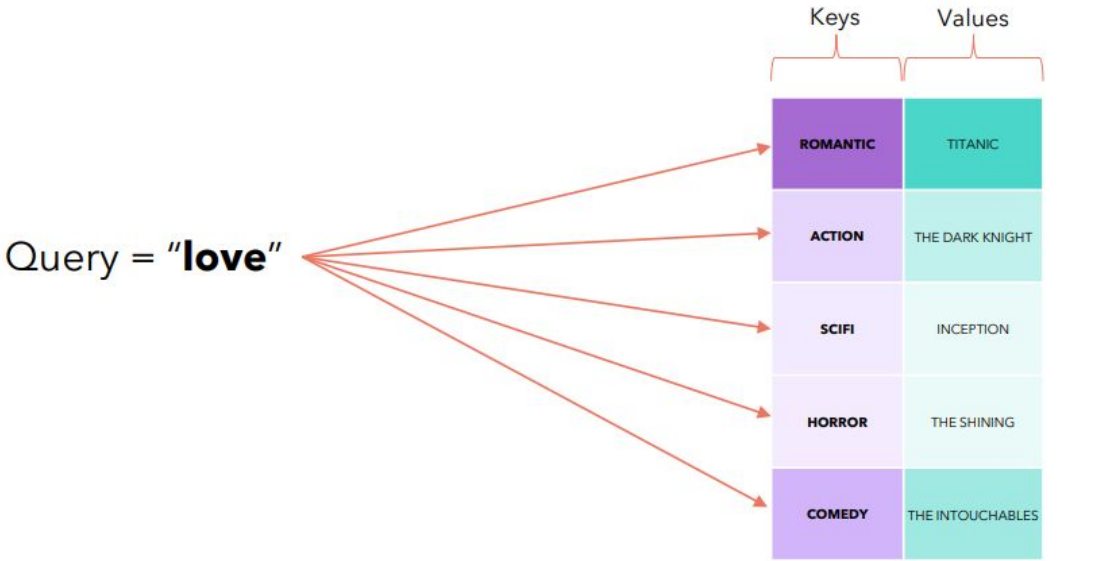
Value (V)

Key (K_3)

2. **Extract values based on attention:**
Return the values highest attention

Why query, keys and values?

The Internet says that these terms come from the database terminology or the Python-like dictionaries.



* this could be a Python dictionary or a database table.

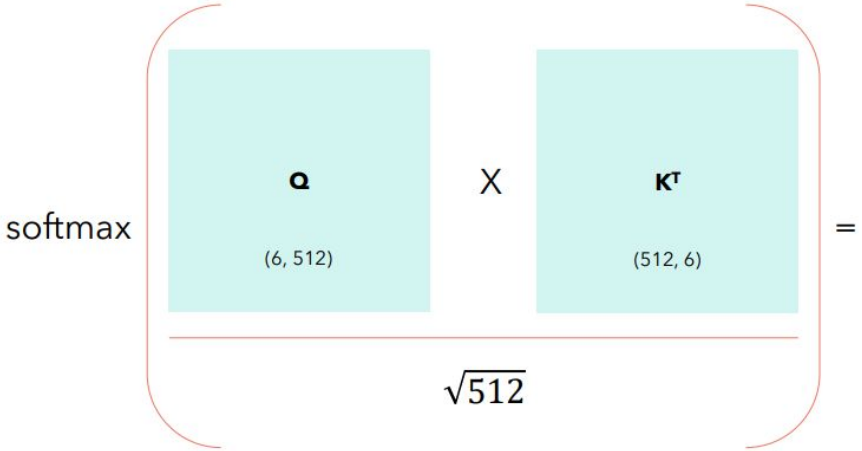
Self-Attention

Self-Attention allows the model to relate words to each other.

In this simple case we consider the sequence length **seq** = 6 and $\mathbf{d_{model}} = \mathbf{d_k} = 512$.

The matrices **Q**, **K** and **V** are just the input sentence.

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



	YOUR	CAT	IS	A	LOVELY	CAT	Σ
YOUR	0.268	0.119	0.134	0.148	0.179	0.152	1
CAT	0.124	0.278	0.201	0.128	0.154	0.115	1
IS	0.147	0.132	0.262	0.097	0.218	0.145	1
A	0.210	0.128	0.206	0.212	0.119	0.125	1
LOVELY	0.146	0.158	0.152	0.143	0.227	0.174	1
CAT	0.195	0.114	0.203	0.103	0.157	0.229	1

(6, 6)

* all values are random.

* for simplicity I considered only one head, which makes $d_{model} = d_k$.

Self-Attention

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

	YOUR	CAT	IS	A	LOVELY	CAT
YOUR	0.268	0.119	0.134	0.148	0.179	0.152
CAT	0.124	0.278	0.201	0.128	0.154	0.115
IS	0.147	0.132	0.262	0.097	0.218	0.145
A	0.210	0.128	0.206	0.212	0.119	0.125
LOVELY	0.146	0.158	0.152	0.143	0.227	0.174
CAT	0.195	0.114	0.203	0.103	0.157	0.229

(6, 6)

X

V

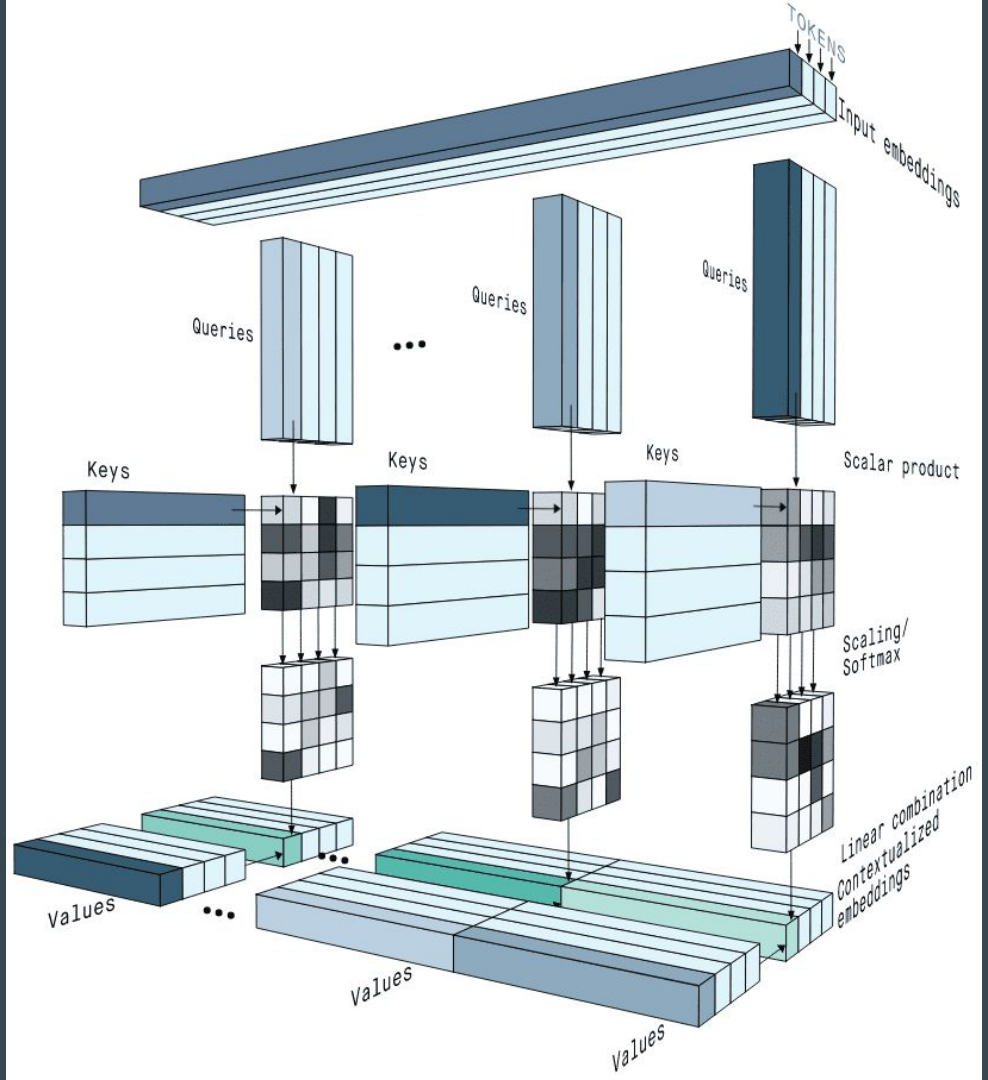
(6, 512)

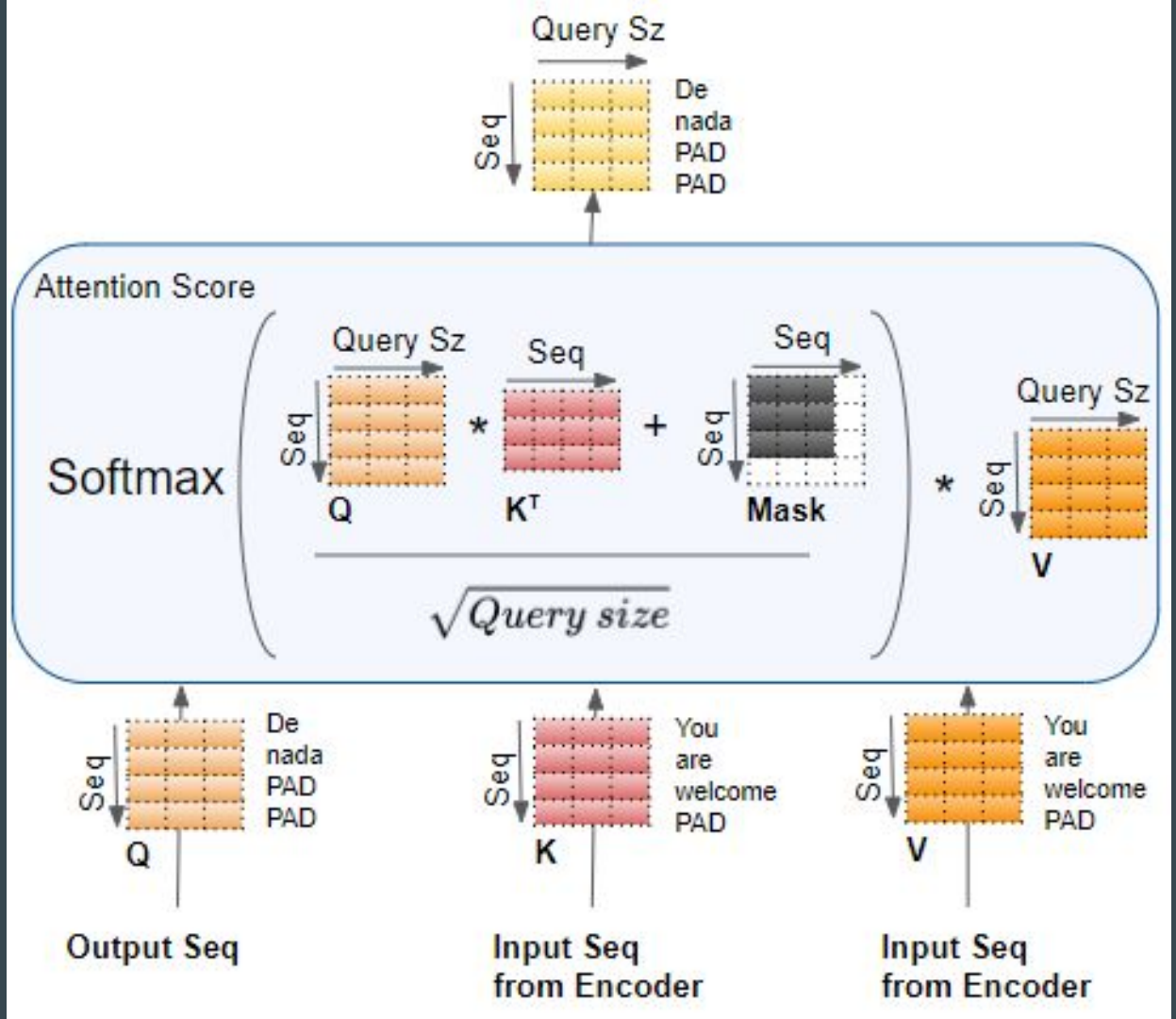
=

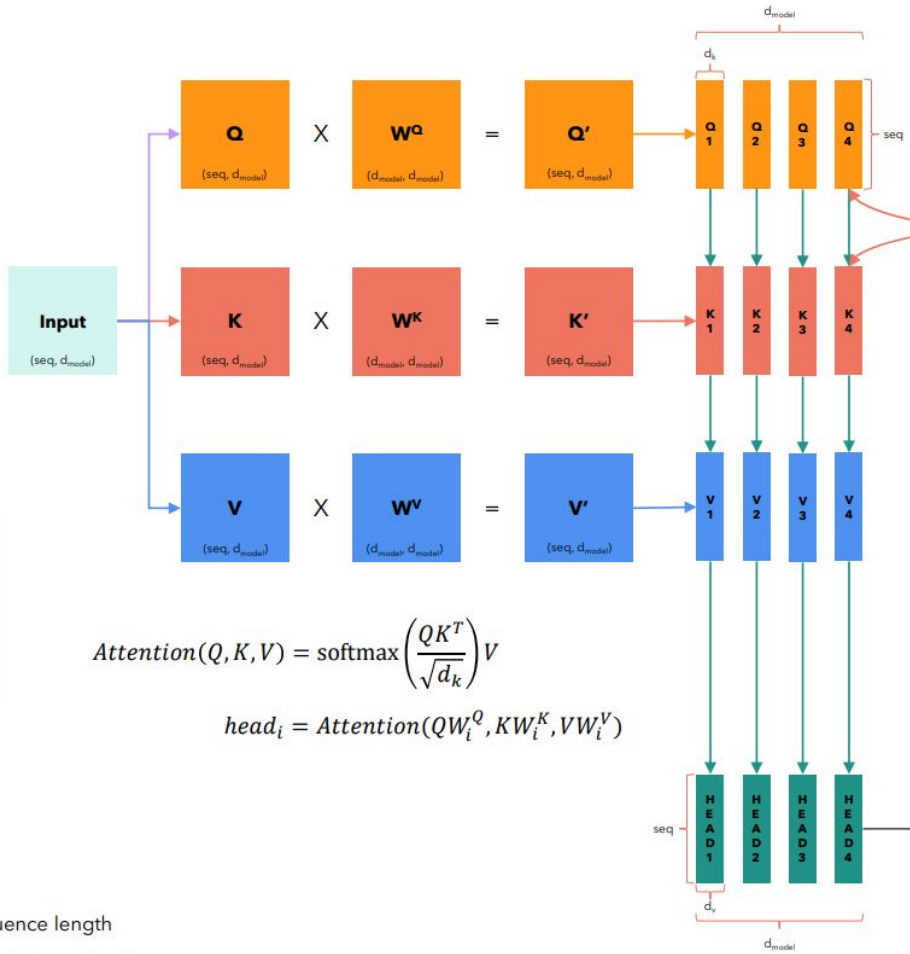
Attention

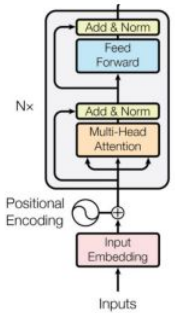
(6, 512)

Each row in this matrix captures not only the meaning (given by the embedding) or the position in the sentence (represented by the positional encodings) but also each word's interaction with other words.

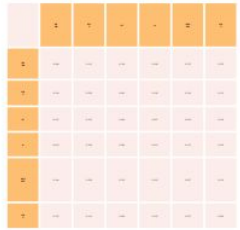

TOKENS
Input embeddings
Queries
Queries
Queries
Keys
Keys
Keys
Scalar product
Scaling/
Softmax
Linear combination
Contextualized
embeddings
Values
Values
Values


Query Sz
Seq
De
nada
PAD
PAD
Attention Score
Softmax
Query Sz
Seq
Q
Seq
K^T
Seq
Mask
Query Sz
Seq
V
$\sqrt{Query\ size}$
Seq
De
nada
PAD
PAD
Q
Output Seq
Seq
You
are
welcome
PAD
K
Input Seq
from Encoder
Seq
You
are
welcome
PAD
V
Input Seq
from Encoder

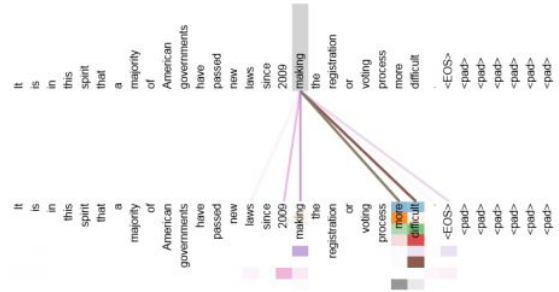




$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) =$$



Attention Visualizations



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

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

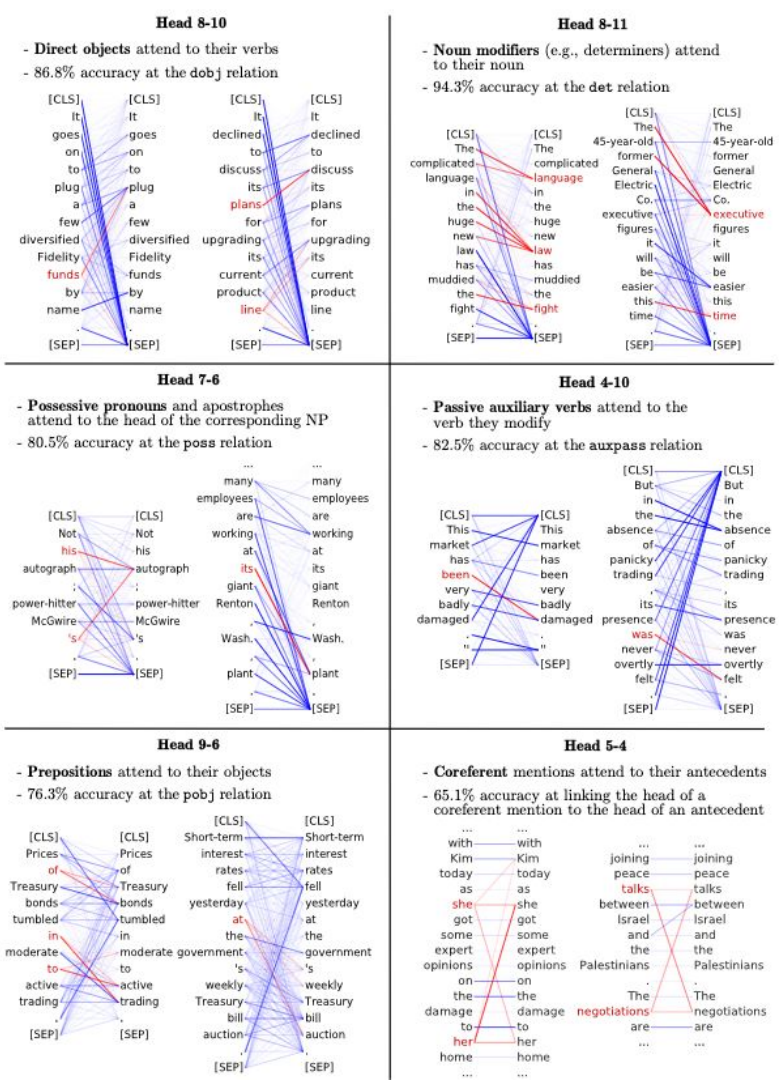
$$MultiHead(Q, K, V) = Concat(head_1 \ \ldots head_h)W^O$$

seq = sequence length

d_{model} = size of the embedding vector

h = number of heads

$d_k = d_v$ = d_{model} / h

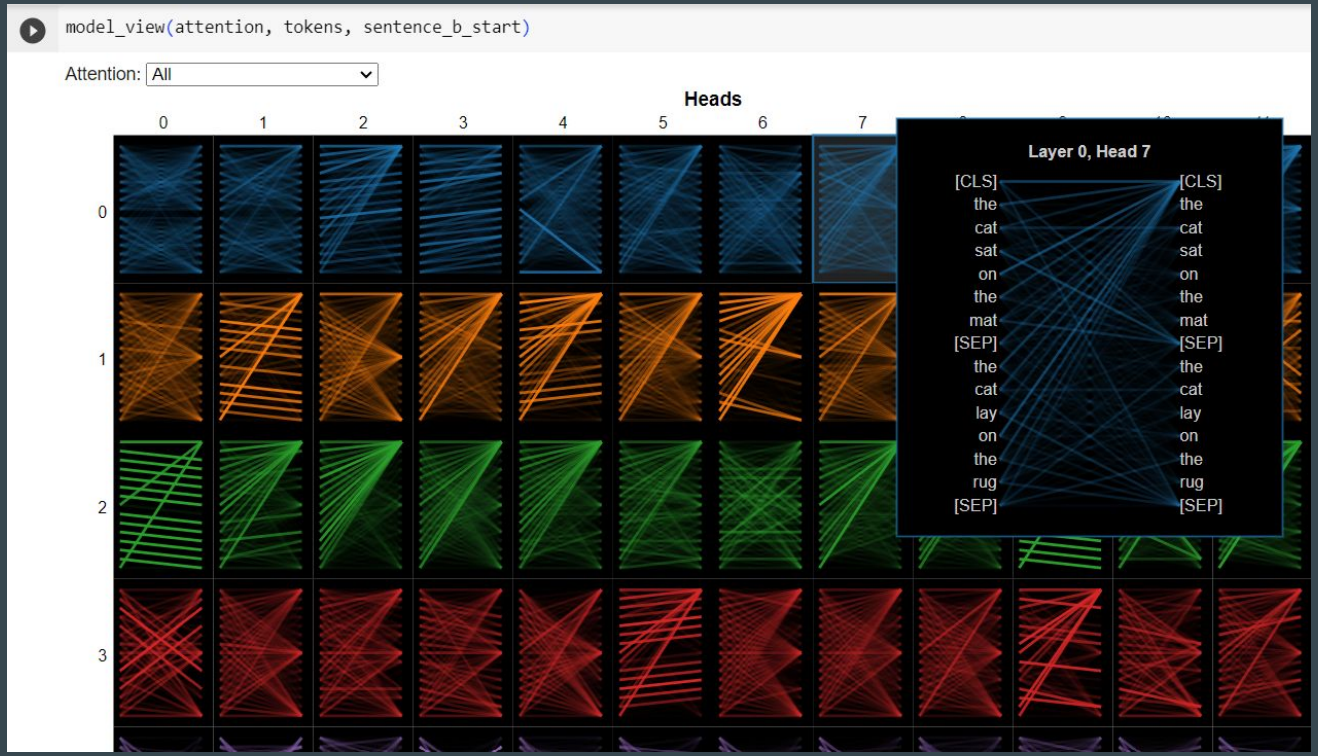


Attention weights are sparse (or close to uniform)

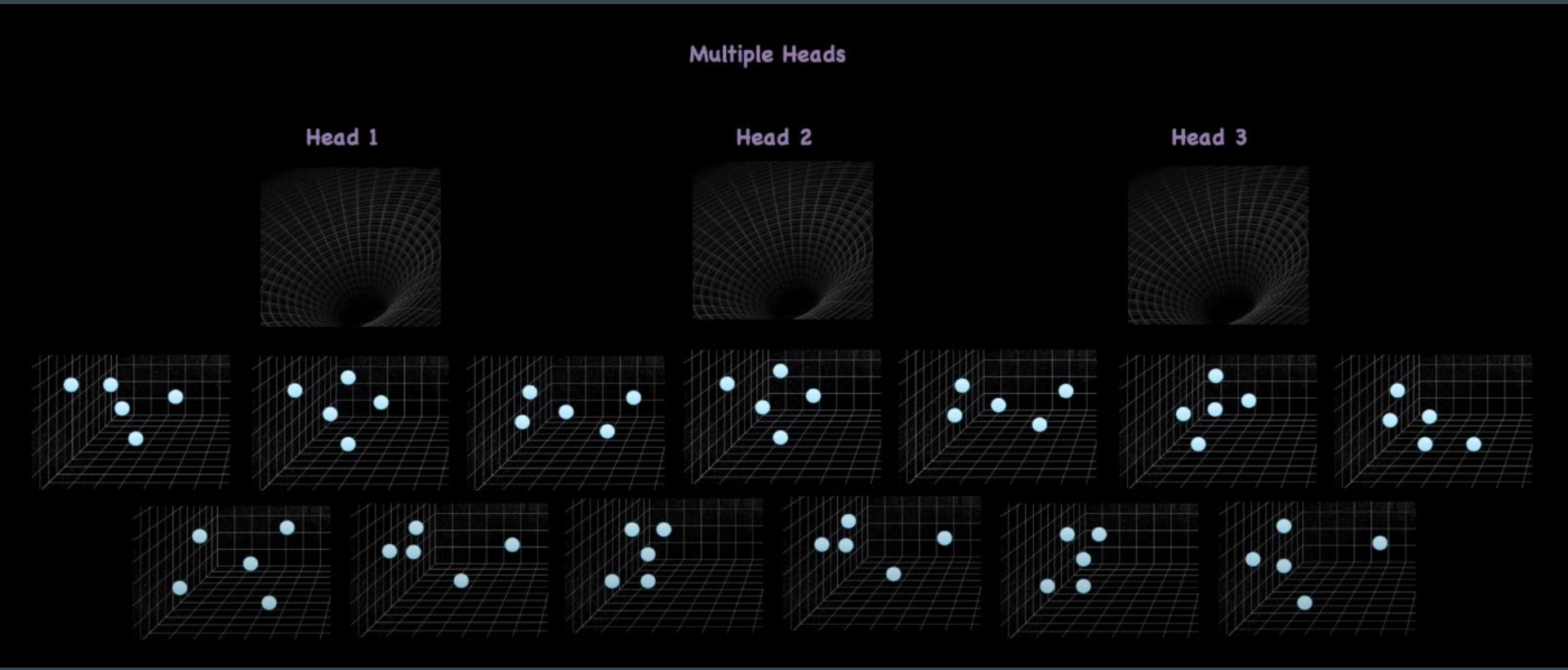
Source: "What Does BERT Look At? An Analysis of BERT's Attention"

Clark, Khandelwal, Levy, Manning

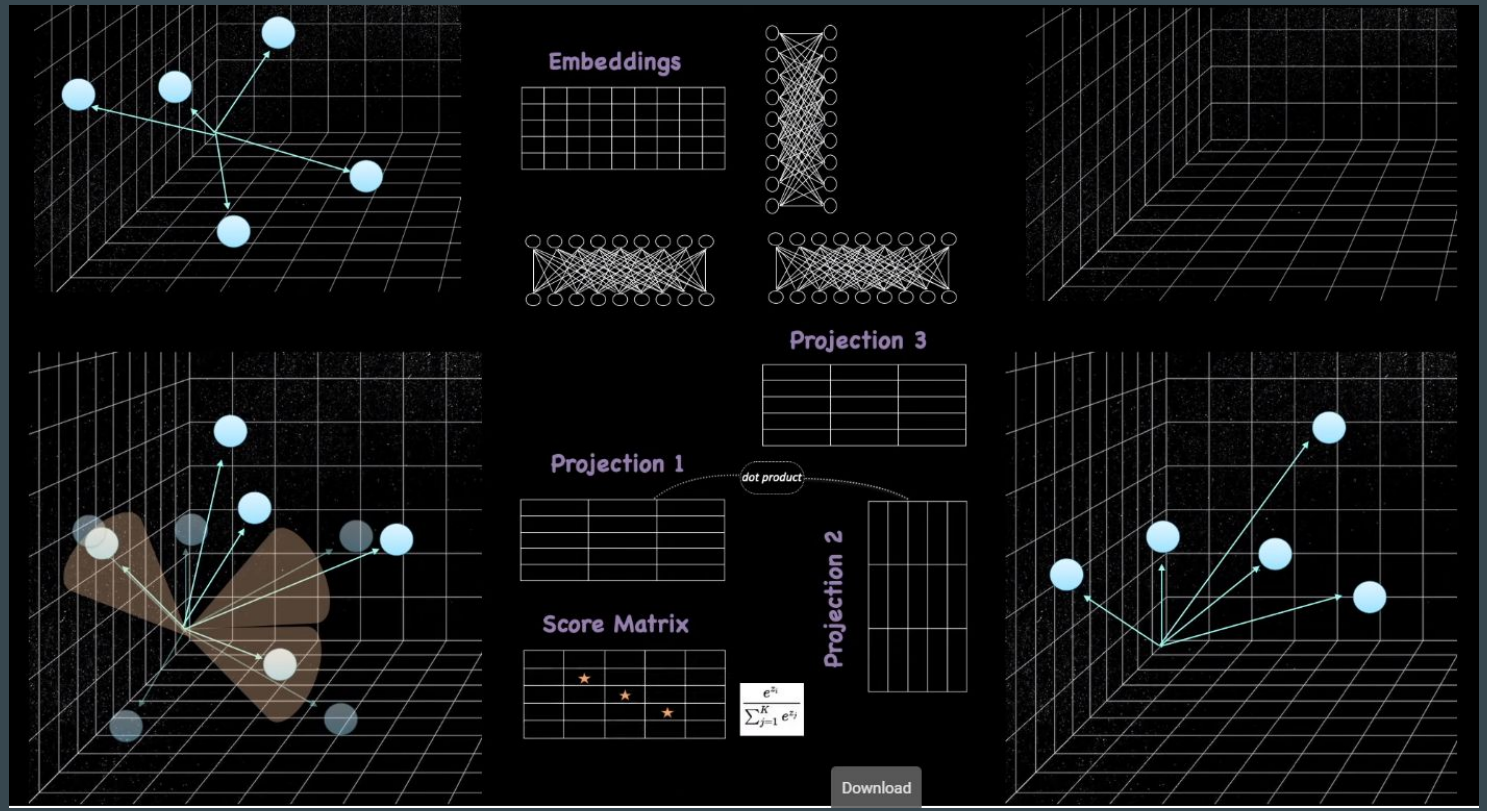
BERTViz Attention Heads



Superposition of Semantic Spaces

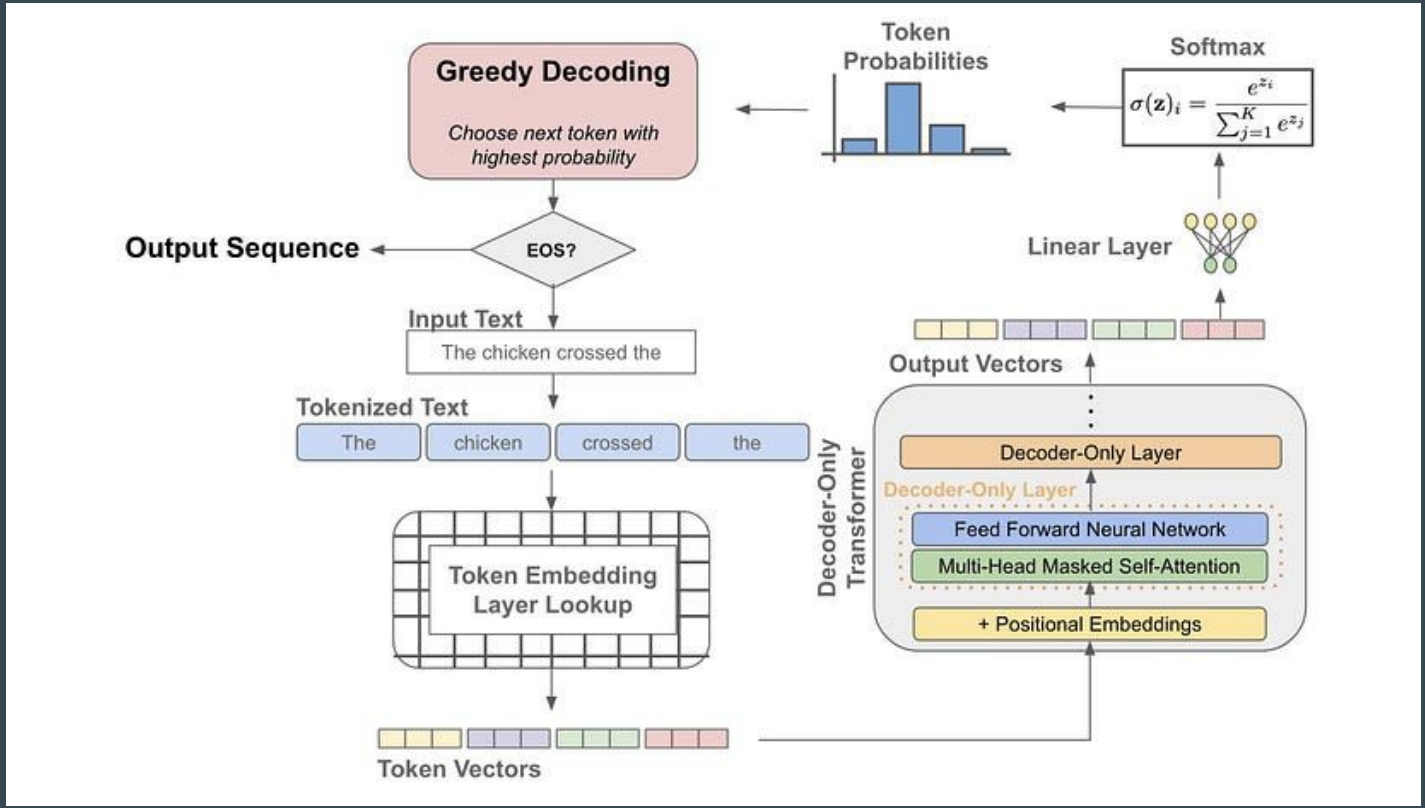


Visualize (Q x K) x V Matrix Cosine Similarities



LLM Inference

Next Token Probabilities



It was a dark and stormy night. Suddenly, a shot rang out. The sound echoed across the moors, ringing in the ears of those who were awake and alert enough t of travelers who were taking refuge ir

through = 98.58%
throughout = 0.86%
across = 0.24%
off = 0.19%
in = 0.05%

Total: -6.04 logprob on 1 tokens
(99.92% probability covered in top 5 logits)

T hotgun in hand. He was a burly man, w face. He was known to be a no- n use his shotgun if necessary.

" area for any sign of danger.

The travelers, who had been huddled around the fire in the common room, jumped and looked at each other in fear. They had been on their way to the nearby city when the storm had forced them to take shelter at the inn. They were all strangers to each other, but in that moment, they were all connected by the sense of unease that hung in the air.

One of the travelers, a young woman with long, dark hair, stepped forward. "I

Model

gpt-3.5-turbo-ins...

Temperature 1

Maximum length 256

Stop sequences
Enter sequence and press Tab

Top P 1

Frequency penalty 0

Presence penalty 0

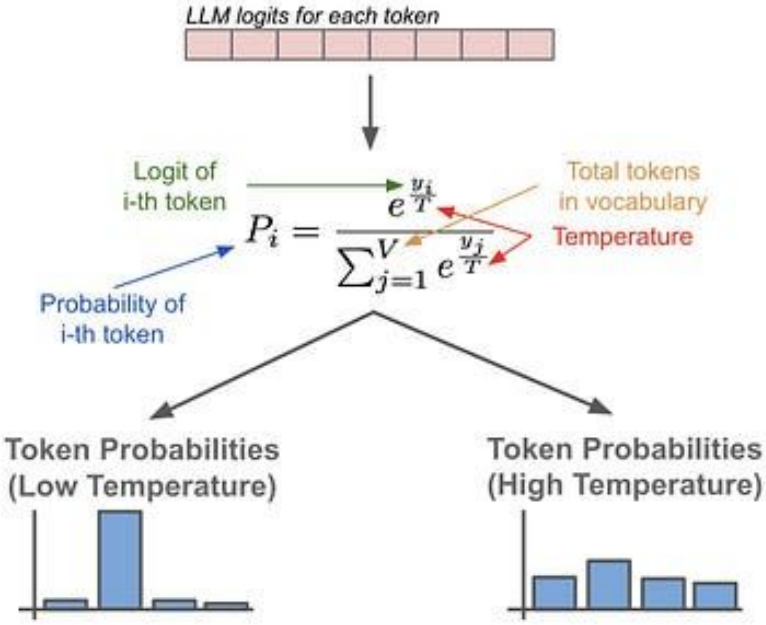
Inference Hyperparameters

Temperature in LLM APIs

`temperature` number Optional Defaults to 1

What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

Softmax with Temperature

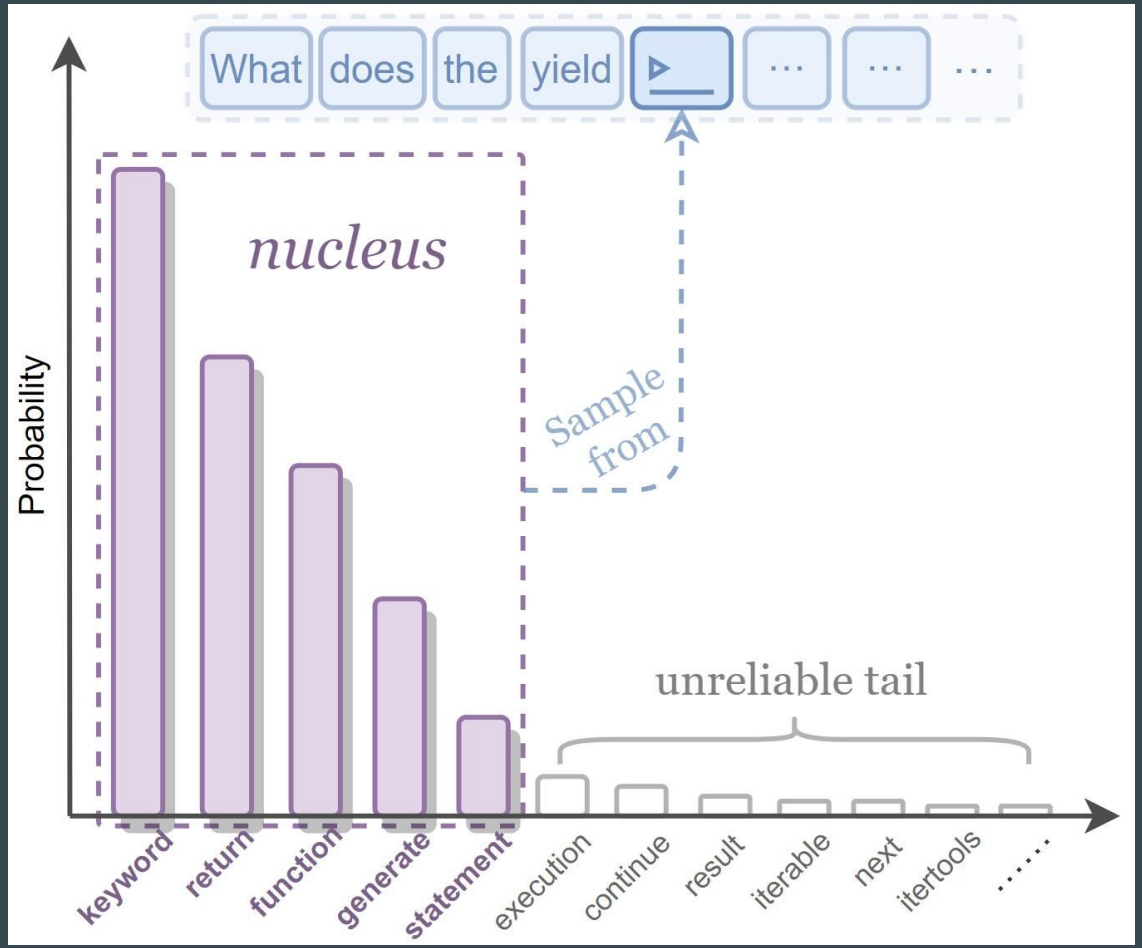


Top-P

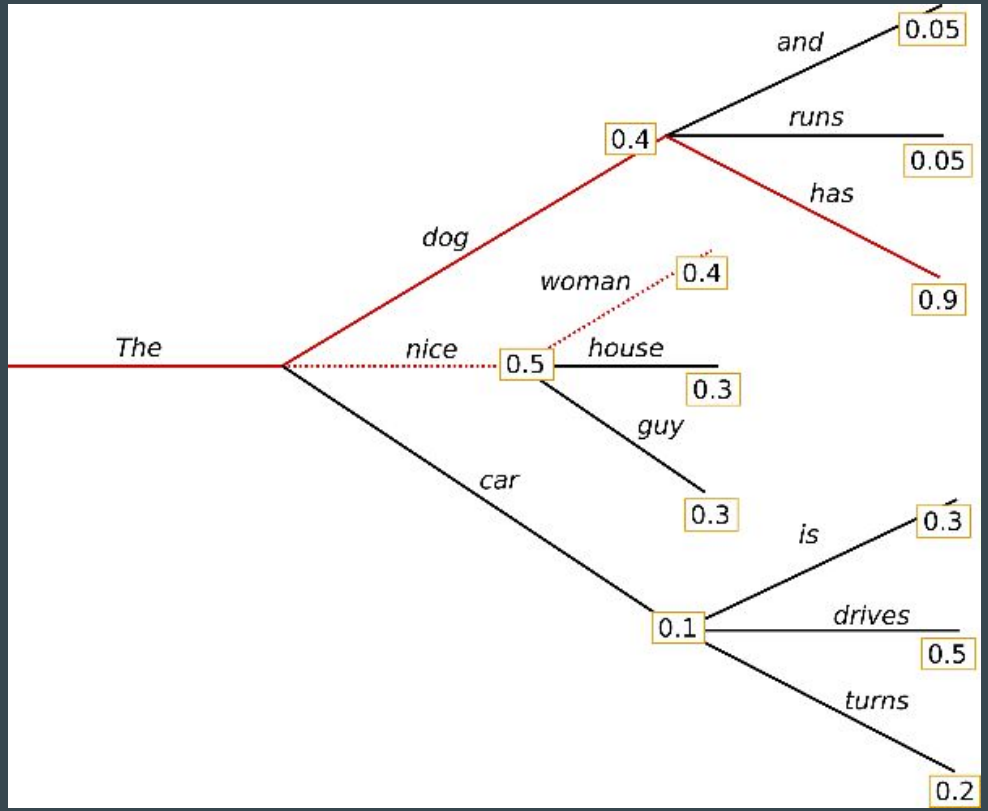
+

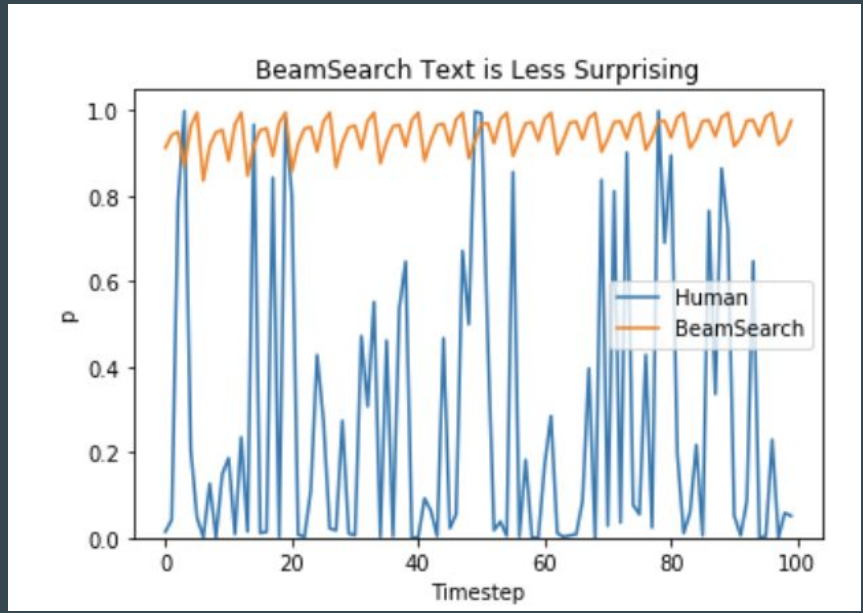
Hi Temp

~ MCMC Sampling



Beam Search


The
dog
nice
car
0.4
0.5
and
runs
has
0.05
0.05
0.9
woman
house
guy
0.4
0.3
0.3
is
drives
turns
0.1
0.3
0.5
0.2


BeamSearch Text is Less Surprising
p
Timestep
Human
BeamSearch
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100

Parameter details

Frequency and presence penalties

The frequency and presence penalties found in the Chat completions API and Legacy Completions API can be used to reduce the likelihood of sampling repetitive sequences of tokens. They work by directly modifying the logits (un-normalized log-probabilities) with an additive contribution.

```
mu[j] -> mu[j] - c[j] * alpha_frequency - float(c[j] > 0) * alpha_presence
```

Where:

- `mu[j]` is the logits of the j-th token
- `c[j]` is how often that token was sampled prior to the current position
- `float(c[j] > 0)` is 1 if `c[j] > 0` and 0 otherwise
- `alpha_frequency` is the frequency penalty coefficient
- `alpha_presence` is the presence penalty coefficient

As we can see, the presence penalty is a one-off additive contribution that applies to all tokens that have been sampled at least once and the frequency penalty is a contribution that is proportional to how often a particular token has already been sampled.

Reasonable values for the penalty coefficients are around 0.1 to 1 if the aim is to just reduce repetitive samples somewhat. If the aim is to strongly suppress repetition, then one can increase the coefficients up to 2, but this can noticeably degrade the quality of samples. Negative values can be used to increase the likelihood of repetition.

frequency_penalty number or null Optional Defaults to 0

Number between -2.0 and 2.0. Positive values penalize new tokens based on their existing frequency in the text so far, decreasing the model's likelihood to repeat the same line verbatim.

See more information about frequency and presence penalties.

logit_bias map Optional Defaults to null

Modify the likelihood of specified tokens appearing in the completion.

Accepts a json object that maps tokens (specified by their token ID in the GPT tokenizer) to an associated bias value from -100 to 100. You can use this **tokenizer tool** (which works for both GPT-2 and GPT-3) to convert text to token IDs. Mathematically, the bias is added to the logits generated by the model prior to sampling. The exact effect will vary per model, but values between -1 and 1 should decrease or increase likelihood of selection; values like -100 or 100 should result in a ban or exclusive selection of the relevant token.

As an example, you can pass `{"50256": -100}` to prevent the <|endoftext|> token from being generated.

logprobs integer or null Optional Defaults to null

Include the log probabilities on the `logprobs` most likely tokens, as well the chosen tokens. For example, if `logprobs` is 5, the API will return a list of the 5 most likely tokens. The API will always return the `logprob` of the sampled token, so there may be up to `logprobs+1` elements in the response.

The maximum value for `logprobs` is 5.

max_tokens integer or null Optional Defaults to 16

The maximum number of **tokens** to generate in the completion.

The token count of your prompt plus `max_tokens` cannot exceed the model's context length. **Example Python code** for counting tokens.

NO STREAMING | STREAMING

Example req... gpt-3.5-turbo-instruct curl Copy

```
curl https://api.openai.com/v1/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "gpt-3.5-turbo-instruct",
    "prompt": "Say this is a test",
    "max_tokens": 7,
    "temperature": 0
  }'
```

Response gpt-3.5-turbo-instruct Copy

```
{
  "id": "cmpl-uqkvlQyYK7bGYrRHQOeXlWi7",
  "object": "text_completion",
  "created": 1589478378,
  "model": "gpt-3.5-turbo-instruct",
  "choices": [
    {
      "text": "\n\nThis is indeed a test",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
```

```
"top_logprobs": [
  {
    " Hello": 0.5773516400136589,
    " Hi": 0.19021177923596874,
    "Hello": 0.067644583100199,
    "\n\n": 0.06604950779814563,
    "Hi": 0.025466052201047033
  },
  {
    "!": 0.5264271466063052,
    ".": 0.2591786724769263,
    ",": 0.09482852674076715,
    " there": 0.04379592072803984,
    "!\n": 0.024123977887792435
  }
],
"text_offset": [
  22,
  28
],
"total_logprobs": [
  0.9267235623490194,
  0.9483542444398309
```


Values for the second token you can see match the playground:

User: Say 'hello'.
AI: Hello! How can I help you?

! = 52.64%
. = 25.92%
, = 9.48%
there = 4.38%
!\n = 2.41%

Total: -0.64 logprob on 1 tokens
(94.84% probability covered in top 5 logits)

Oct 27

2 / 5
Oct 27

4d

Prompt Engineering

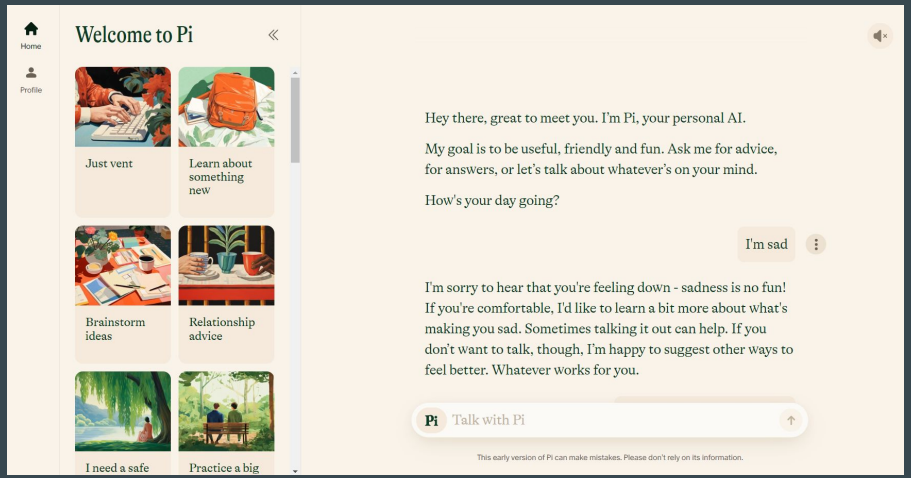
- Simple Prompts
- N-shot In Context Learning (ICL)
- Reflection
- Chain of Thought
- Tree of Thought
- ReAct

Future

Future

- LMM
- Automated Agents
- XAI & FATE
- Small Open LLM
- Affective AI
- Theoretical Basis

New Best Friend: pi.ai Chatbot



Fin