Symbolic AI. – a kind of Good old-fashioned AI (GOFAI)

↳ Represents knowledge as explicit symbols and rules
↳ System manipulate these symbols using logic and search algorithms
↳ Knowledge is (usually) hand-crafted by experts.

Ex-1

Knowledge Base (KB)

IF fever AND cough AND fatigue
THEN diagnosis = flu

IF fever AND rash
THEN diagnosis = measles

IF chest_pain AND breath_shortness
THEN diagnosis = heart_issue

⇓

Inference Engine
(match systems to rules)

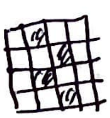
↓

Diagnosis output

Q. What about new symptoms ?
Q. What about old systems but not encoded by the KB ?

Ex-2

chess position ⇒



Hand-coded eval fun

Queen = 9 points
Rook = 5 points
...
pawn = 1 point

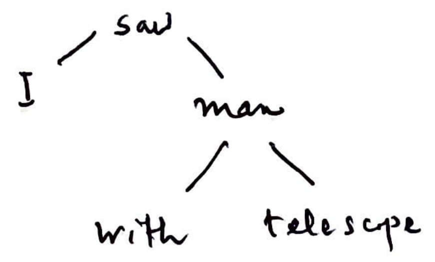
+ centre control
+ King safety
+ Pawn structure

⟹ min-max tree

Ex-3 Sentence: "I saw the man with the telescope"

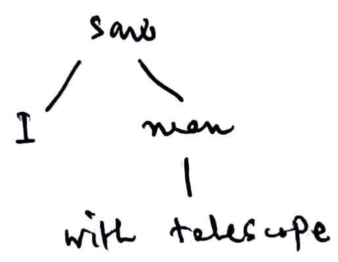
2 interpretations

①



(I used the telescope)

②



(the man used telescope)

Q. Can symbolic AI "Learn" from data ? (YES: INDUCTIVE LOGIC PROGRAMMING)

↓

Q. What is learning ?

☐ Reasoning (deductive)

Rules (KB) — exists before !!

Data / Datum → Rules (KB)

Rules (KB) → Deduce a conclusion or many conclusions

"Inductive" learning

Data

↓

Discover patterns → Rules (kind of)

↓

Hypothesis (probabilistic)

Rules (kind of) → May not always be human-understandable

Let's, for now, restrict ourselves to text data:

Q. Language patterns exist in data.
Instead of writing rules, can we learn patterns from billions of text examples ?

Foundation Models: (FMs)

FMs learn representations* and patterns* directly from massive amount of data.

A single model trained on broad data, then adapted to many tasks.

⇒ shifts from rule-based AI to learning based AI.

(Read: Brown et. al 2020, "Language models are few shot learners" (GPT3))

The man went to the bank...

which bank? for what?

GOFAI: IF bank AND deposit THEN Finbank.

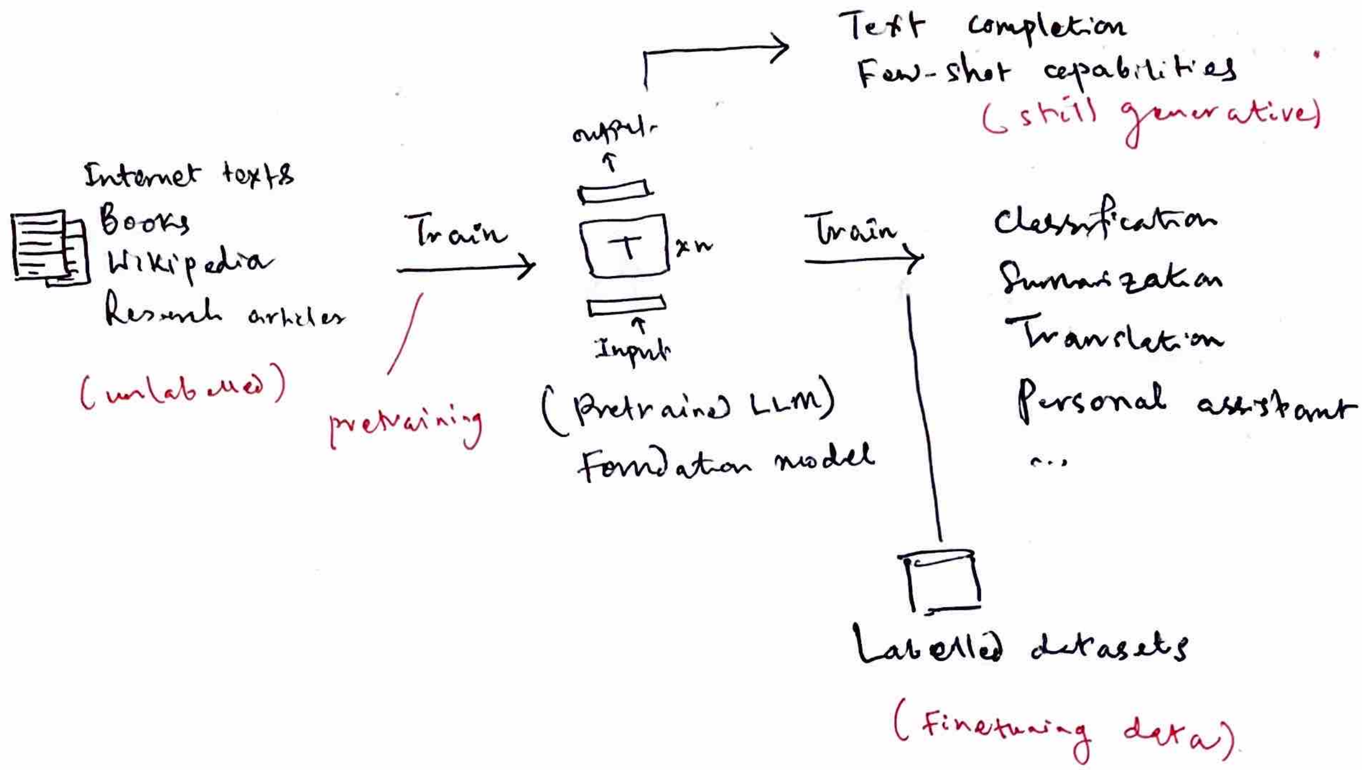
NEWFAI: Learn from millions of sentences with "bank" in different context.

Automatically learn associations.

FMs example:

Modality	models.
→ Text	GPT-4, BERT, LLaMA → Pretrained on language data (texts) also called "language model" or "Large language model" or "Pretrained LLM"
Vision	CLIP, DINOv2
Audio	Whisper, AudioLM
Code	Codex, CodeGen
Multiple modes	Gemini, GPT-4o,.
Biology	ESM, AlphaFold*
Chemistry	ChemBERTa

FMs: Serve as foundation for many downstream tasks



Q. What is being modelled by a FM?

FM is a generative model that learns a probability distribution over sequences.

→ Given context, predict next token.

Bayes' rule

$$P(x|y) = \frac{P(y|x)\,P(x)}{P(y)}$$

or

$$P(y\,|\,\underline{x}) \propto P(\underline{x}|y)\,P(y)$$

Remember this

$p($"the cat sat on the mat"$)$

$p($"the mat sat the cat on"$)$ ← very low prob.

$x_1 \quad x_2 \quad x_3 \quad \dots$

autoregressive decomposition:

$$p(\underline{x}) = p(x_1, x_2, \dots, x_n)$$

$$= p(x_1)\,p(x_2|x_1)\dots p(x_n|x_{n-1}, \dots x_1)$$

product of next token (word here) probabilities given all prev. words.

Q. How does one $\boxed{\text{estimate}}$ $p(x_i \mid x_{i-1}, \ldots x_1)$

Traditional models : n-gram counts

FMs : Deep networks $\boxed{\text{approximate}}$.

<u>Training objective</u> :

Input : The cat sat on the ← prompt

Predict : mat (high prob.)

floor (medium prob.)

water (low prob.)

Q Can we say the model must be learning :

Grammar : "cat" follows an article "the" (syntax)

Semantic : Cat is an animal (meaning)

World Knowledge : Cats do not like water.

Prompt : The cat sat on the ?

temperature scaling.

temperature = 0 (deterministic) mat

temperature = 1 (creative) flower bed

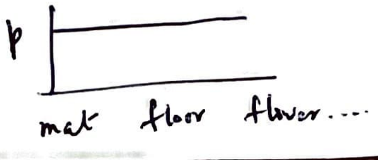
random !!

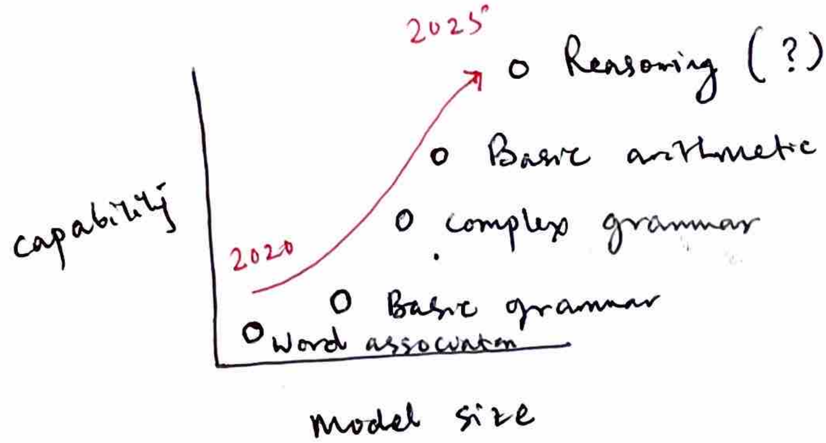
$$p(x_i) = \frac{e^{(z_i/T)}}{\sum_j e^{(z_j/T)}}$$

T > 1 : Flatter distribution over next word probs.

T < 1 : Sharper distribution over "

T = ∞





Different types of models and training objectives.

* GPT style (Causal / Autoregressive) Decoder only

The cat sat on ?

← attention flow

→ Generation task.

"the"

* BERT style (masked language model) Encoder only

The cat [MASK] on the mat

← bidirectional attention →

"sat"

→ Understanding task

* T5-style (Encoder-Decoder) → Seq to seq task

Translate to French "Hello"

↓

Bonjour

Q. chatGPT or claude?

How are these built?

GPT — FM

↓

RLHF (Future lectures)

+

Instruction Tuning

Let's walk through a quick text generation example with GPT-style model:

$\boxed{\text{USER}}$: " ~~the~~ The cat sat on the" (complete the sentence)

<u>Stage 1</u> Tokenization Text → Numerical tokens

(mostly BPE: Byte-pair encoding)

subwords

" The cat sat on the "

↓

["the" , "cat" , "sat" , "on" , "the"]

↓ ↓ ↓ ↓ ↓

[2061 , 374 , 21 , 53 , 2061] Token ID

<u>Stage 2</u> map or convert token IDs to vectors

Token ID 2061 → $\boxed{\text{Embedding Lookup table}}$ → [0.23, -0.89, ..., 0.12]

d-dimensional vector

(for GPT4 ⇒ d = 4096)

each token → d-dim.

↓

Potentially, capture semantic meaning.

additional positional info: (more on this in the DL class)

Pos 1 : [0.00, 1.00, 0.10, 1.00, ...] (d-dim.)

Pos 2 : [0.84, 0.54, ...]

⋮

Final emb for each token:

only in classical GPT imp

"The" : [0.23, -0.89...] + [0.00, 1.00, ...]

Stage 3 Transformer block.

Input: Sequence to vectors (one per token)

3.1 multi-head self-attention

Each token looks at all prev. tokens to understand content.

2. Add & Normalise

× N

Residual connection + layer norm.

3. Feedforward network

one block

Process each token independently

4. Add & Normalise

↓

Output vectors (same shape as input n×d)

N = 96 (in GPT-3)

N = ~120 (GPT4)
claude

① Self attention: " The cat sat in the "

att (sat, cat) = softmax (att_score) ≈ 0.95

0.70

att (sat, the) = Softmax (att_score) ≈ 0.01

0.1

strong relevance

③ weighted sum of token rep.

new_repr(sat) = 0.01 × repr(The) + 0.95 × repr(cat)

Multiple heads : Doing these in parallel multiple times

Note

Original token dim "d" is SPLIT across heads.

Head 1 : Learn syntactic relations

Head 2 : Learn semantic relations

Head 3 : ...

↓

Concat all heads → Final representation.

Stage 4

After N transformer layers.

Final hidden state for last token → prediction

↙

Final transformer output

d - dim → V - dim.

↓

vocab. size (50,000)

logits = [V - dim array]

↓

softmax

↓

probs.

	0	
water	0	0.001
	0	
mat	0	0.85
	:	
floor	0	0.12

0.85 → $p($ mat | "The cat sat on the"$) = 0.85$