Platonic Research

August Update

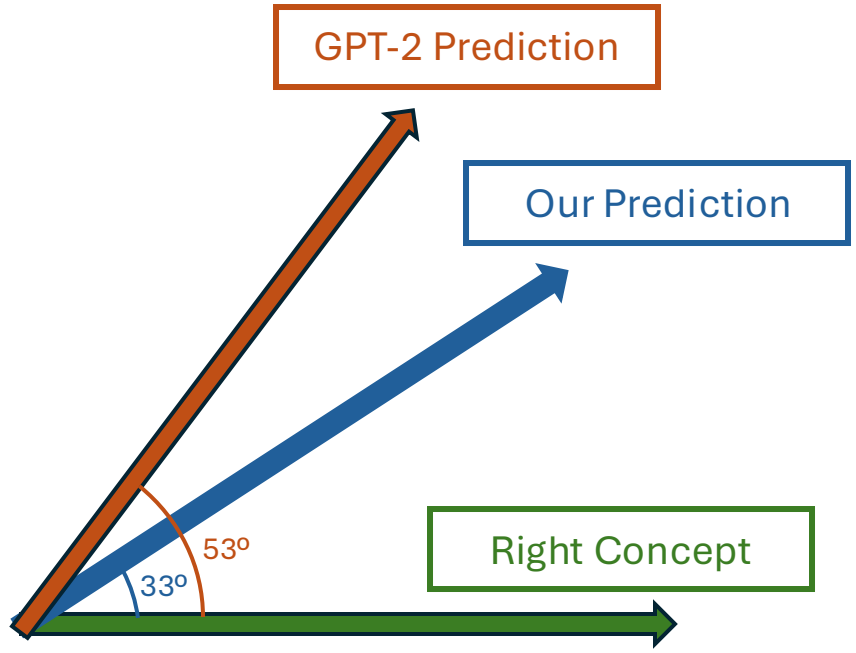
We beated current LLMs in next concept prediction!

Setting

- We built 2/3 pieces of the Architecture

- We **benchmarked** our model to **predict the next concept**, an essential ability for cognitive tasks

Results

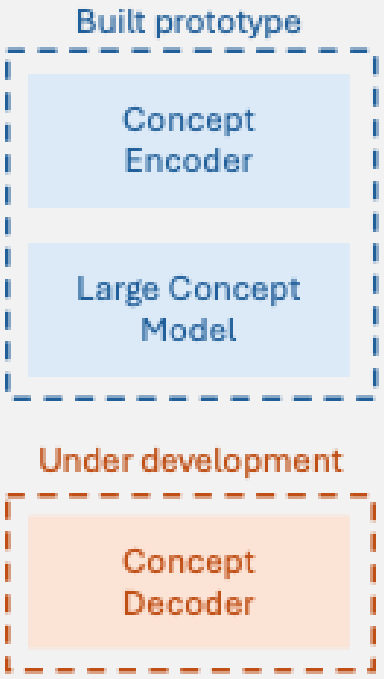


- Our Model outperforms GPT2-small while being **~60x cheaper to train and to run!**

Next steps

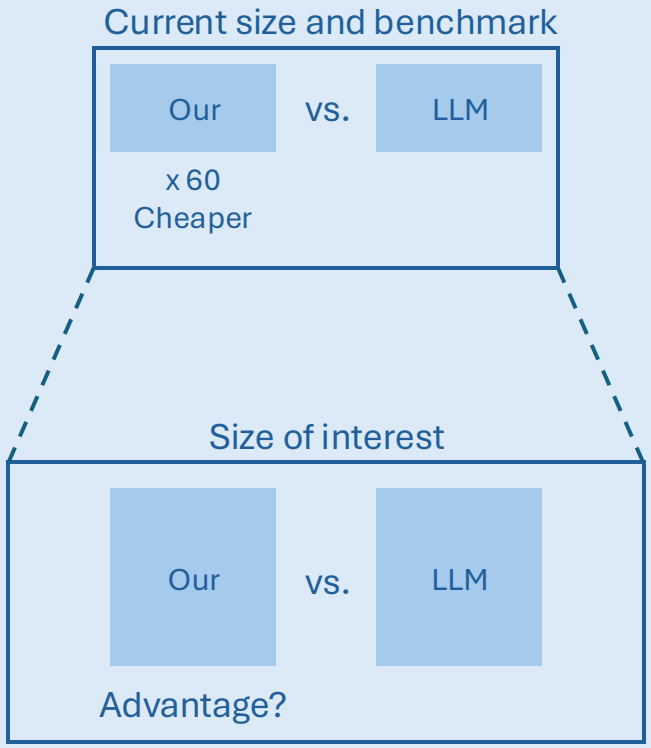
Finish Prototype

And look for **design partners**



Test Scaling

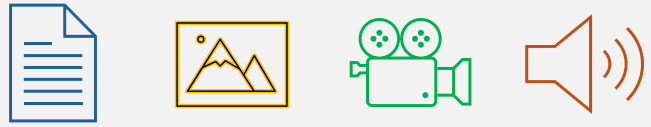
And predict advantages of our model for all sizes



Innovate

Build 6 Innovations allowing:

Native Multimodality



Unlimited memory

Arbitrary thinking time for the hardest problems

Backup

Deep dive on benchmark methodology

Our comparison

After taking a question:

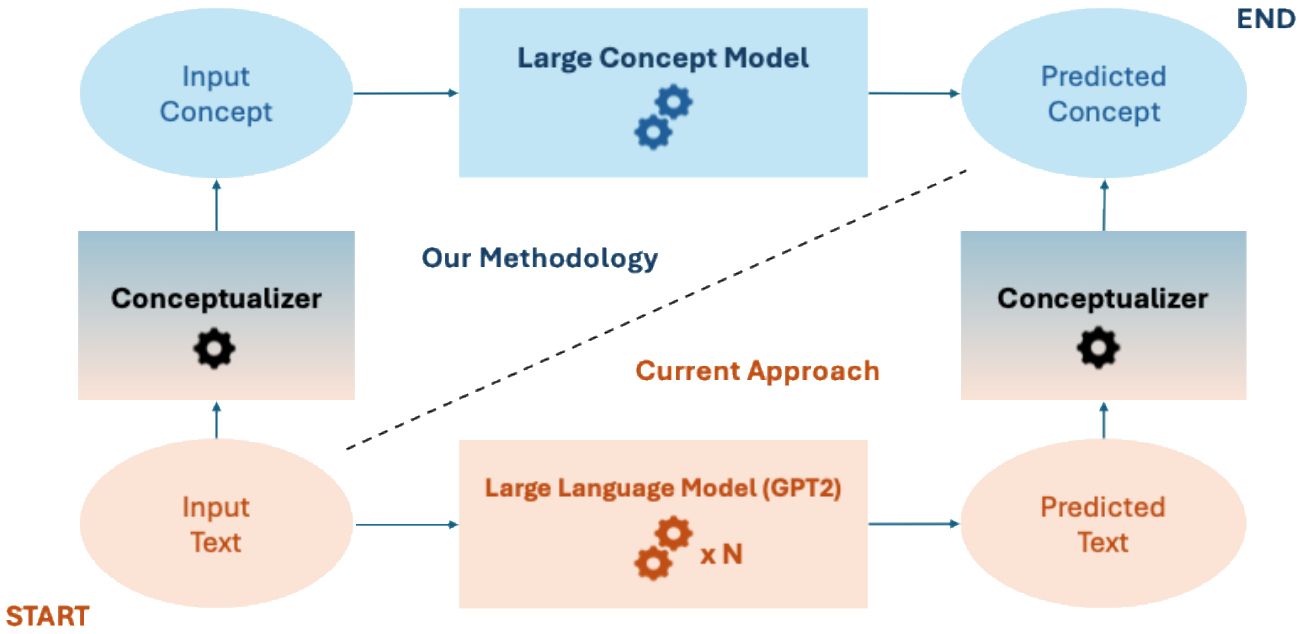
- **Our model** conceptualizes it, and predicts the answer concepts.
- A **small language model** (GPT2-small), predicts the answer words, and we turn them into concepts

Both predictions are **compared with the true answer** expressed in the vector space of concepts

Results

We achieve smaller error while being **~60 x cheaper to train and run.**

Illustration of the methodology

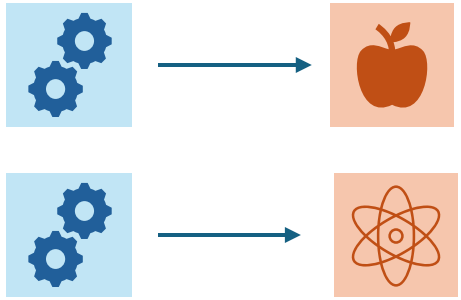


Core ideas of our research

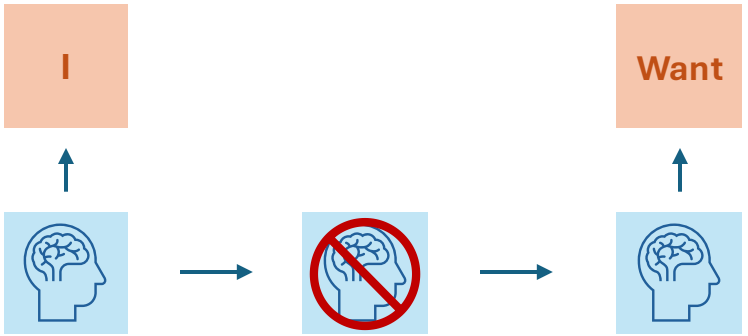
The problems LLMs face today

Inefficient Compute Allocation

- Same computation for simple and difficult words

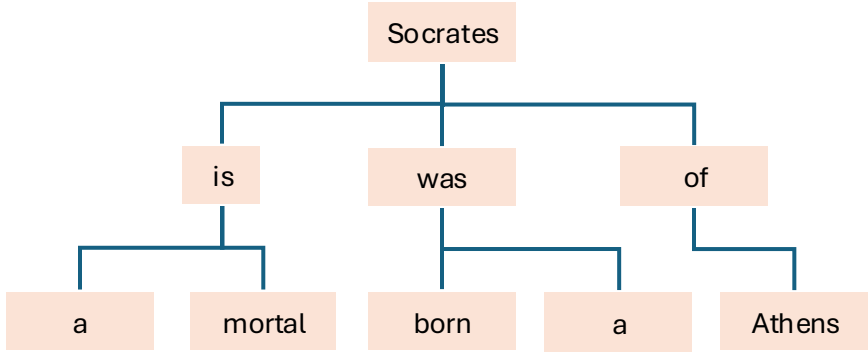


- Forget (hard) computation after generating every single word

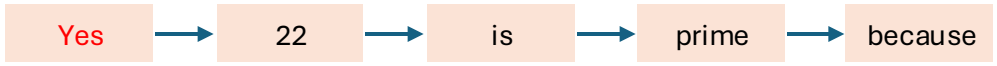


Inefficient Text Generation

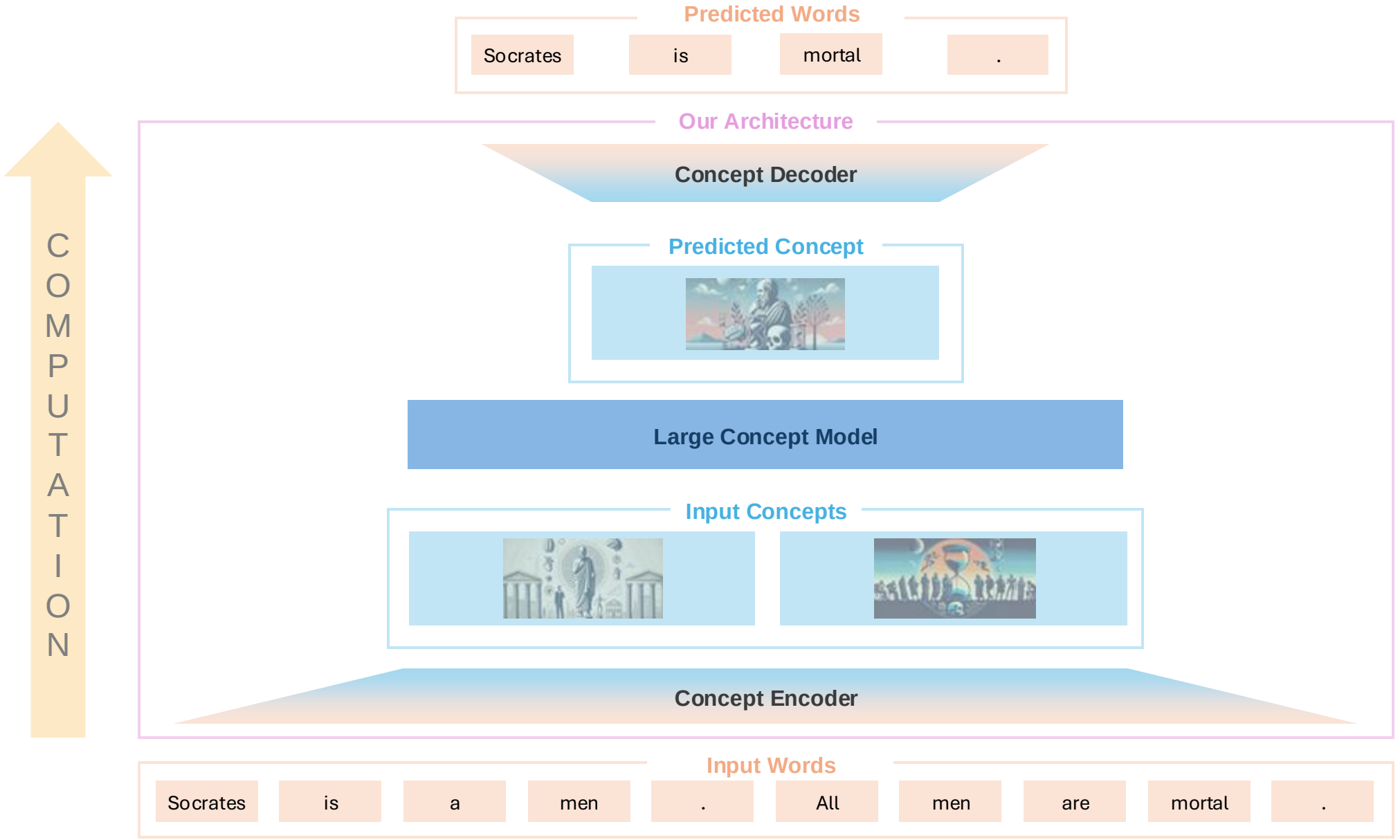
- Don't explore the whole sentence tree



- Cannot backtrack on generated tokens



Large Concept Models Architecture



Large Concept Models Architecture

- Large concept models **solves** the **4 big problems of LLMs**, and is a **framework** enabling:
  - **Reasoning Concepts** => add arbitrary test-time compute
  - **Natively Multimodal** => concepts can be audio, texts, videos or images
  - **Infinite Memory =>** Retrieve information from arbitrary long in the past
  - **Monte Carlo tree search** =>1000 less trees to explore per concept
  - **Differential Search** => optimize decoding at low compute cost
  - **Scaling Laws for Large Concept Models** => Predict big model performance from small models
- Most of the innovation has yet to come!

Move the ceiling of LLM performance!

Chincilla Scaling Laws:

$$Loss = Entropy\,(Text) + \frac{A}{Model\_size^{\alpha}} + \frac{B}{Data\_size^{\beta}}$$

Error | Ceiling | Bigger Models | Train for Longer

We improve the ceiling!

Everybody focused on these

(GPT-2) $Loss = 1.690 + 0.720 + 0.442$

(GPT-4) $Loss = 1.690 + 0.028 + 0.087$

The expected **advantange** of this architecture for **big models** will be **known** after the first 10k$ run.