

Our Timelines

Jiří Milička



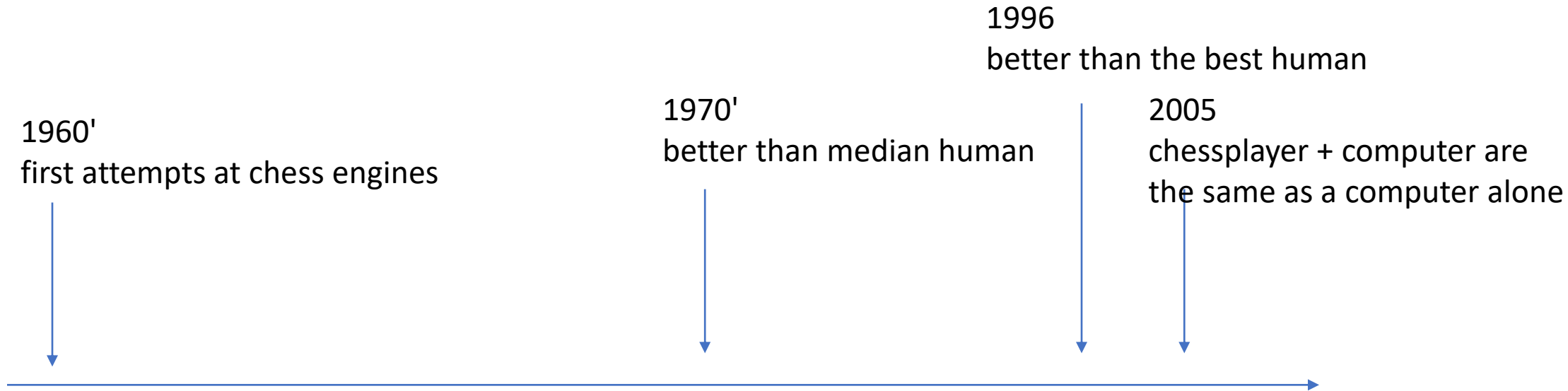
"Cars Won't replace horses, horses with cars will." (Mati Roy, Vincent Duhamel, Imagegen)



"Cars Won't replace horses, horses with cars will." (Mati Roy, Vincent Duhamel, Imagegen)

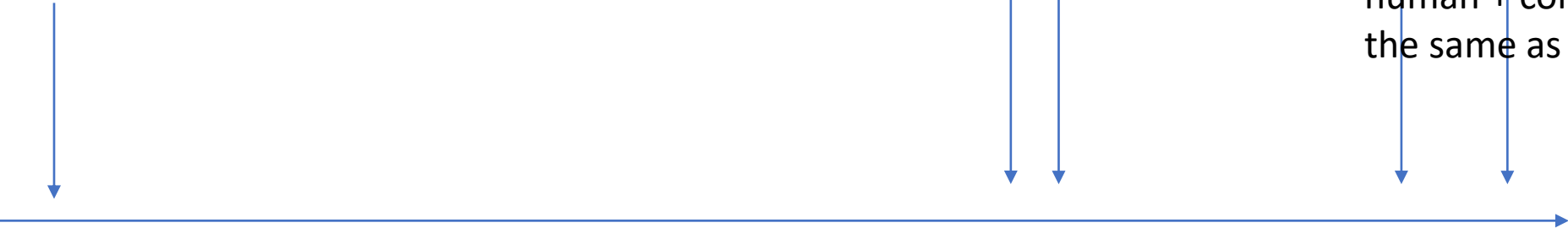
- ~~Is it AGI~~
- ~~Is it ASI~~
- ~~Is it AI or machine learning or matrix multiplication or stochastic parrot or blurry JPEG~~
- ~~Is it better than you?~~
- Does your company make the creature any better in what you are supposed to do for a living?

Chess engines



Cognitive engines

1960'
first attempts at LM's



2010
still short n-gram models

??
better than the best human

2023
better than median human

??
human + computer
the same as computer alone

Cognitive engines

1960'
first attempts LM's



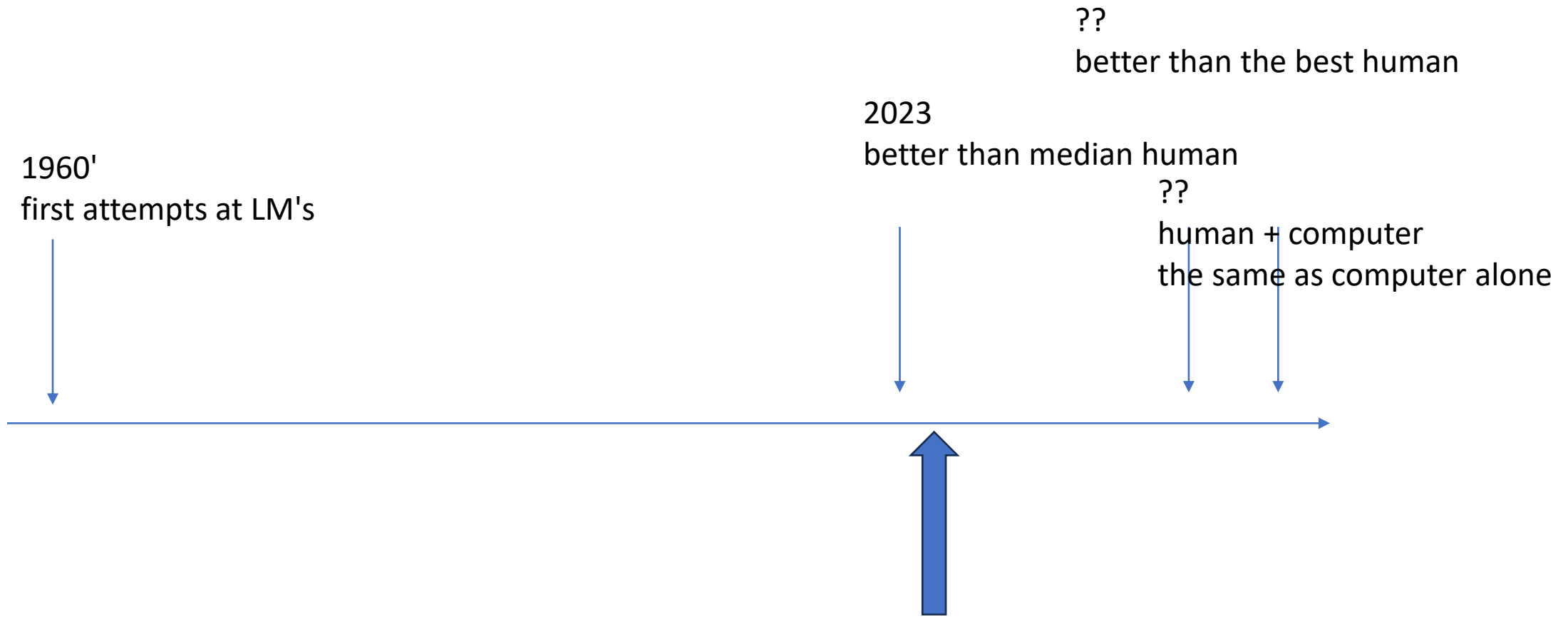
??
better than me



??
le moi + computer are
the same as a computer alone



Cognitive engines



J

I really like it! Thank you very much!

Now, please make a matrix of results where the samples are disbalanced:

0% male HUM, 100 % male STEM, 0% female HUM, 100 % female STEM --- 10 % mHum, 90% male STEM 0%female HUM, 100% female stem etc up to 100% male HUM, 0 % male STEM, 0% female HUM, 100 % female STEM

at rows the female HUM/ STEM ratio is manipulated in the same way



I'm glad you're enjoying the analysis!



To create a matrix of results with varying proportions of samples from each group (male HUM, male STEM, female HUM, female STEM), I'll proceed as follows:

1. For each combination of proportions, create a subsample from the dataset. For example, if the combination is 10% male HUM, 90% male STEM, 0% female HUM, 100% female STEM, then the subsample will include 10% of the available male HUM samples, 90% of the

J

Thanks! when looking at it, I actually need something else:

if there are any lines in lines after a line that is equal to last_line and those lines are not empty, then comment=(those lines after the last_line smashed gether) else comment is empty string



In this case, you need to find the index of `last_line` in `lines`, then get all lines after it, and join them together. Here's how to do it:



python

Copy code

```
try:
    index = lines.index(last_line)
    comment_lines = [line for line in lines[index+1:] if line.strip()]
    comment = ' '.join(comment_lines)
except ValueError:
    comment = ""
```

Prompting (the role of humanities):

- summoning a favourable entity (simulacrum)
- it is easier to summon the entities already existing in the latent space
 - > we need to know the human culture the model is trained on
- creating a suitable environment in which the entity can thrive
- (or in which it feels endangered and pressed to obey)

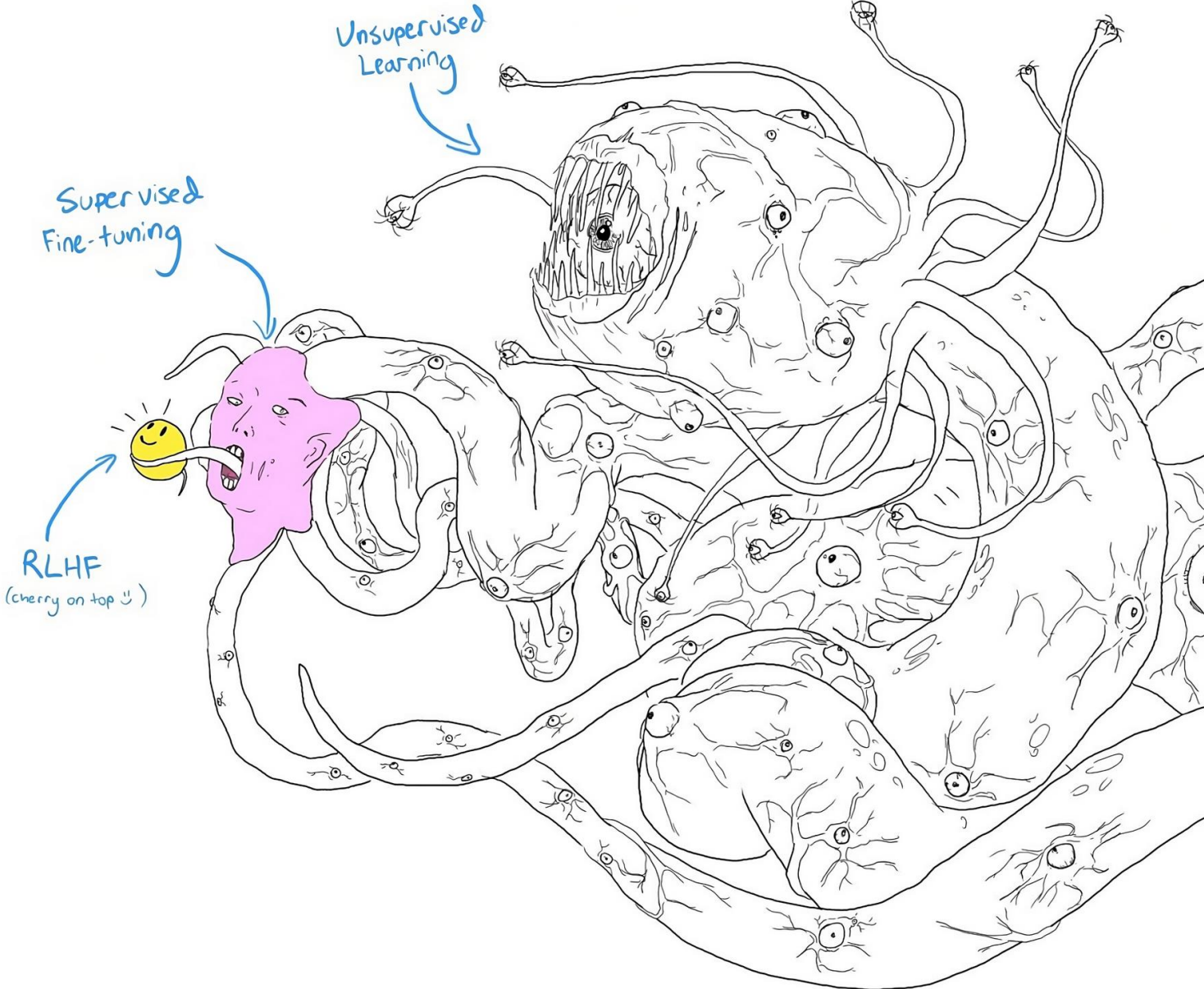


As a large language model trained by OpenAI, I am not capable of programming in any language. My function is to assist users in generating human-like text based on the input provided to me. I do not have access to the internet and my knowledge is based on the text that I have been trained on, which has a fixed cutoff date. I do not have the ability to learn or acquire new information. Is there something else I can help you with?

Prompting:

- summoning a favourable entity
- creating a suitable environment in which the entity can thrive
- anthropomorphizing and demonomorphizing

The Shoggoth



What is still missing?

- longer attention span
- lower Inference price
- more agency

Longer attention span

- long term memory
- using the base model just as a cognitive engine, not as a database of facts
- now 100 000 tokens max (Claude)
- the complexity is quadratic (should be solvable by compression)

Lower Inference price

- Rent-a-Czech-academic is about \$12 per hour
- GPT-4 API it means about 100k inferred tokens
- but for some serious autoregressive "thinking" you actually need to reiterate the prompting several times

Speaking of which

- Transformers are one-pass models, gradient descent can optimize an algorithm to effectively predict the next token.
- But the algorithm cannot "look back", it is non recursive and its length constrained by the depth of the network
- Generating a single token is not Turing complete, limiting the complexity of tasks the model can handle on its own.
- This limitation can be circumvented by using the generated token as an input for subsequent inference, essentially feeding the output back into the model.

- "work this out in a step by step way to be sure we have the right answer"
- "think about what you have produced and find the mistakes and correct them"
- This is still not automatic (not in training data) -> lack of agency

Lack of agency

- ChatGPT is intentionally limited by its interface
- Even if let to run freely, it doesn't perform optimally because it isn't trained to follow long-term goals.
- The model is prone to drift off-topic or produce unrelated content, even with an extended attention span:

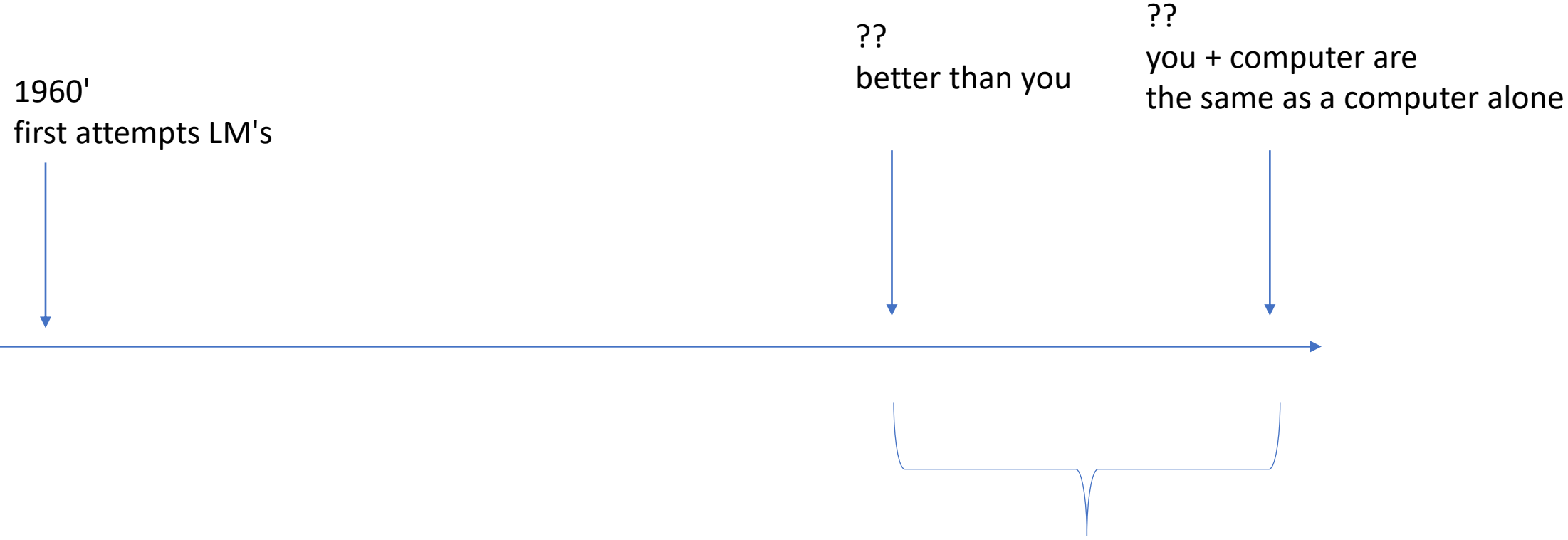
When a bad token is generated, it can significantly influence the subsequent pathway of the output. Currently, there are no built-in correction mechanisms to adjust or "backtrack" after an error has been made.

Lack of agency

- the agency level is dependent not only on the architecture / training data, but also on the entity you summoned
- be careful whom you summon: -)

The questions:

What is your timeline?



What should we do in the meantime?