SYNTHESIS OF REASONING EXAMPLES AS SOCIAL COMPETITION

François Fleuret

UNIVERSITÉ
DE GENÈVE

We cast the production of synthetic training data for reasoning as a game of “social competition” between agents.

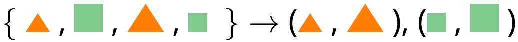
The working hypothesis is that natural selection equips agents with “concepts” to deal with their environment, and when they get a means of communication, they compete with one another through intellectual challenges.

We cast the production of synthetic training data for reasoning as a game of “social competition” between agents.

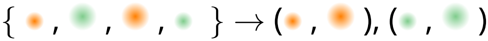
The working hypothesis is that natural selection equips agents with “concepts” to deal with their environment, and when they get a means of communication, they compete with one another through intellectual challenges.

Such a challenge has to be consistent with concepts already learned, so that some other agents can confirm it carries structures, but it has to go beyond them, so that **it proves that its author masters concepts that some other agents do not.**

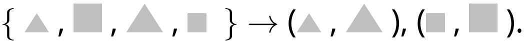
If to operate in its environment an agent has to solve



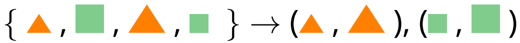
it can do it by being only “color aware”



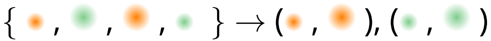
or “shape aware”



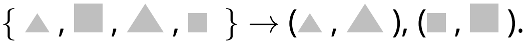
If to operate in its environment an agent has to solve



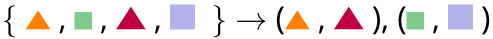
it can do it by being only “color aware”



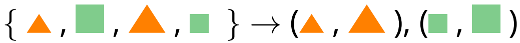
or “shape aware”



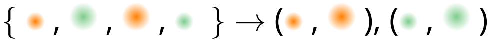
An “shape aware” agent could craft a challenge that those who are not would fail



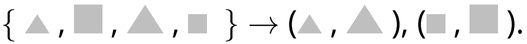
If to operate in its environment an agent has to solve



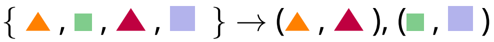
it can do it by being only “color aware”



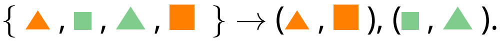
or “shape aware”



An “shape aware” agent could craft a challenge that those who are not would fail



and similarly regarding “color awareness”



To test this experimentally we use GPT models as agents and “quizzes” as fragments of knowledge.

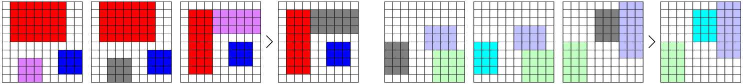
To test this experimentally we use GPT models as agents and “quizzes” as fragments of knowledge.

We differentiate:

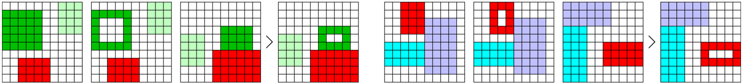
- “world quizzes” whose distributions are pre-defined and play the role of the cognitive challenges from the environment, and
- “culture quizzes” which are generated by agents, and whose role is to prove that its author masters certain concepts that its opponents do not.

A quiz consists of a prompt composed of three 10×10 colored cell grids, and a solution which is a single grid.

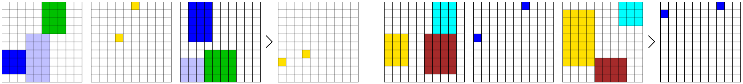
We define seven “tasks”, each of them being a distribution of “world quizzes”, such that the solution is unique given the prompt, and we implement for each of them a procedure to generate samples.



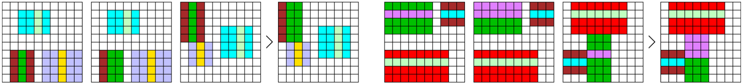
“Replace Color”



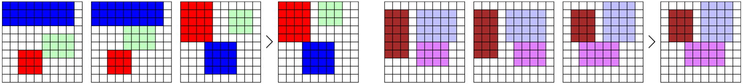
“Frame”



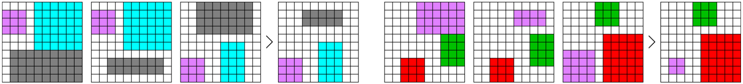
“Detect”



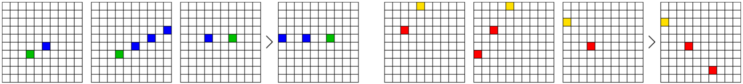
“Half Fill”



“Translate”



“Grow”



“Motion”

MODEL AND TRAINING

Experiments are done with a 37 millions parameter GPT trained from scratch.

- `dim_model`: 512
- `dim_keys`: 64
- `dim_hidden`: 2048
- `nb_heads`: 8
- `nb_blocks`: 12

Given a quiz composed of three 10 × 10 grids for the prompt and one 10 × 10 grid for the solution

$R_0(A)$	$R_0(f(A))$	$R_0(B)$		$R_0(f(B))$
$R_1(A)$	$R_1(f(A))$	$R_1(B)$		$R_1(f(B))$
$R_2(A)$	$R_2(f(A))$	$R_2(B)$		$R_2(f(B))$
$R_3(A)$	$R_3(f(A))$	$R_3(B)$		$R_3(f(B))$
$R_4(A)$	$R_4(f(A))$	$R_4(B)$	>	$R_4(f(B))$
$R_5(A)$	$R_5(f(A))$	$R_5(B)$		$R_5(f(B))$
$R_6(A)$	$R_6(f(A))$	$R_6(B)$		$R_6(f(B))$
$R_7(A)$	$R_7(f(A))$	$R_7(B)$		$R_7(f(B))$
$R_8(A)$	$R_8(f(A))$	$R_8(B)$		$R_8(f(B))$
$R_9(A)$	$R_9(f(A))$	$R_9(B)$		$R_9(f(B))$

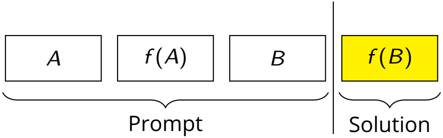
we can represent the prompt as a sequence of 300 tokens

$$R_0(A), \dots, R_9(A), R_0(f(A)), \dots, R_9(f(A)), R_0(B), \dots, R_9(B)$$

and the solution as a sequence of 100 tokens

$$R_0(f(B)), \dots, R_9(f(B))$$

We can train a GPT on a complete sequence composed of the prompt followed by the solution, and then use it to “solve” a quiz, that it to generate the solution given the prompt.



A $f(A)$ B $f(B)$

Prompt

Solution

We re-sample the training examples at every epoch, so there is no over-fitting.

The accuracy of a GPT trained on all the tasks combined gets easily above 99%.

Our main objective is to **generate new meaningful quizzes**, which are both outside the domain of the training examples and consistent with them.

A GPT can produce a full quiz by generating all the tokens without conditioning.

However, if the GPT is properly trained, such a quiz follows the data-distribution, and there would be no novelty beside unstructured sampling noise.

Our main objective is to **generate new meaningful quizzes**, which are both outside the domain of the training examples and consistent with them.

A GPT can produce a full quiz by generating all the tokens without conditioning.

However, if the GPT is properly trained, such a quiz follows the data-distribution, and there would be no novelty beside unstructured sampling noise.

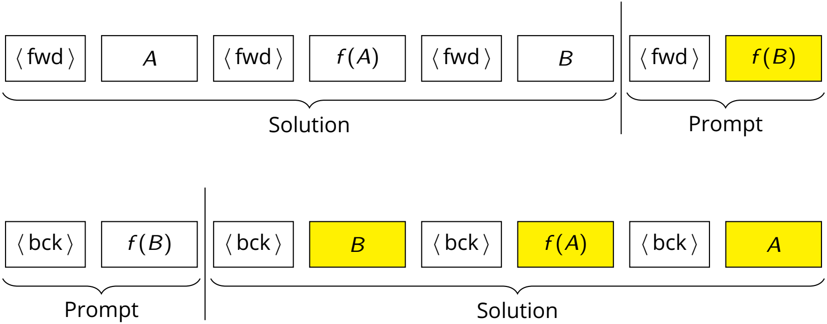
We propose to train N GPTs in parallel, to generate quizzes with structured noise, and to “validate” and keep a generated quiz **only if exactly** $N-1$ **GPTs solve it correctly.**

To generate quizzes with a structured noise, we

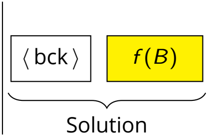
1. generate a solution at high temperature,
2. generate a consistent prompt at low temperature, and
3. re-generate the solution given this prompt at low temperature.

That way we get a quiz that may be slightly outside the support of the training samples, but it should be “functionally consistent.”

Hence we also need to generate prompts given solutions. To do that we train the model on two types of sequences, with additional tokens to indicate the direction.



Given a model trained that way, we implement the sampling of new quizzes as follows:

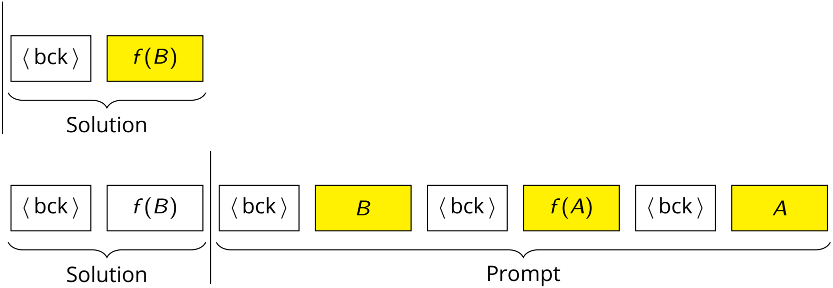


⟨bck⟩ $f(B)$

Solution

1. Generate a solution at high temperature

Given a model trained that way, we implement the sampling of new quizzes as follows:



⟨ bck ⟩ $f(B)$

Solution

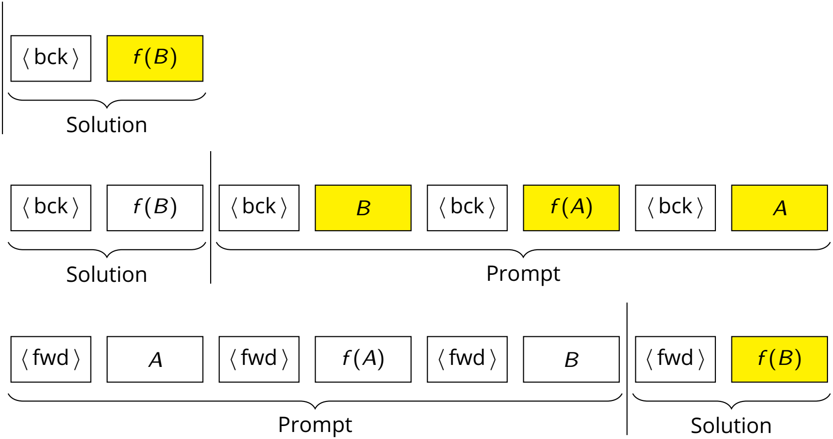
⟨ bck ⟩ $f(B)$ ⟨ bck ⟩ B ⟨ bck ⟩ $f(A)$ ⟨ bck ⟩ A

Solution

Prompt

2. Generate a prompt at low temperature

Given a model trained that way, we implement the sampling of new quizzes as follows:



⟨ bck ⟩ $f(B)$

Solution

⟨ bck ⟩ $f(B)$ ⟨ bck ⟩ B ⟨ bck ⟩ $f(A)$ ⟨ bck ⟩ A

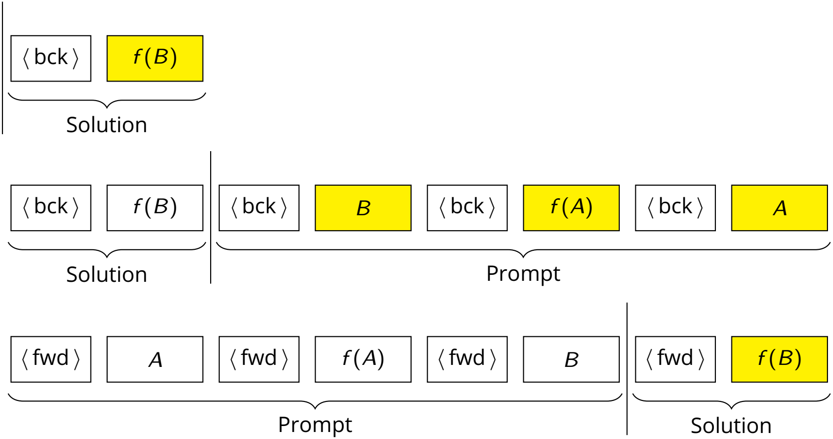
Solution Prompt

⟨ fwd ⟩ A ⟨ fwd ⟩ $f(A)$ ⟨ fwd ⟩ B ⟨ fwd ⟩ $f(B)$

Prompt Solution

3. Re-generate the solution at low temperature

Given a model trained that way, we implement the sampling of new quizzes as follows:



⟨ bck ⟩ $f(B)$

Solution

⟨ bck ⟩ $f(B)$ ⟨ bck ⟩ B ⟨ bck ⟩ $f(A)$ ⟨ bck ⟩ A

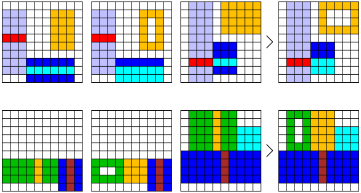
Solution Prompt

⟨ fwd ⟩ A ⟨ fwd ⟩ $f(A)$ ⟨ fwd ⟩ B ⟨ fwd ⟩ $f(B)$

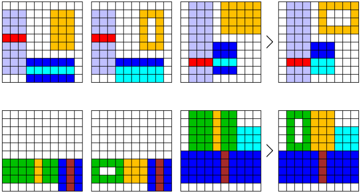
Prompt Solution

We then evaluate the quiz with each of the N models and count the number of correct prediction M and “validate” the new quiz only if $M = N - 1$.

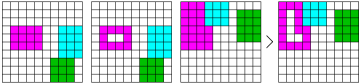
RESULTING CULTURE QUIZZES



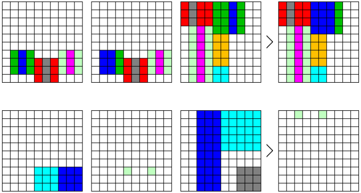
The quizzes “frame” and “half fill” have been combined in a single quiz.



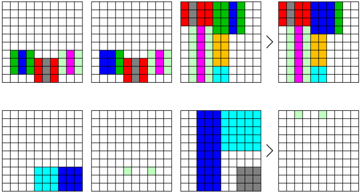
The quizzes “frame” and “half fill” have been combined in a single quiz.



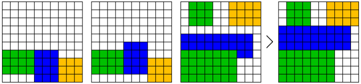
The “frame” quiz has been generalized to non-rectangular shapes.



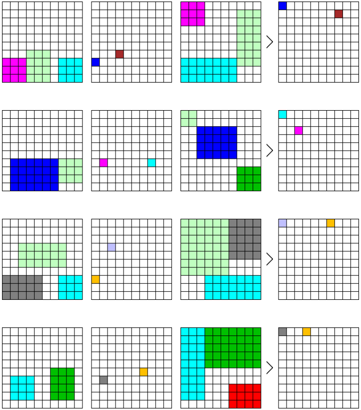
More rectangles were added as distractors.



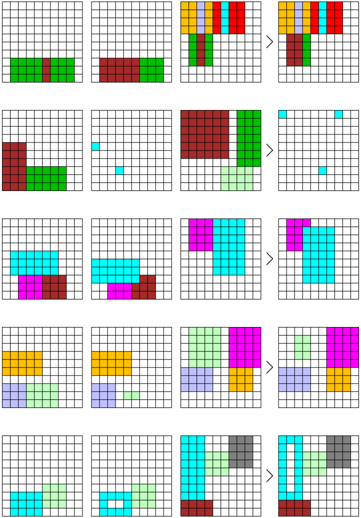
More rectangles were added as distractors.



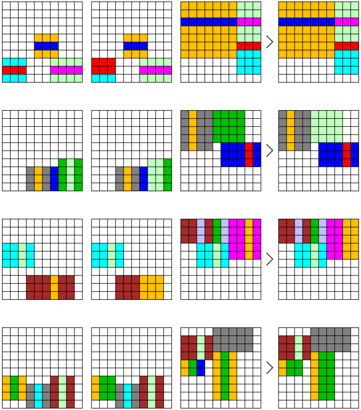
“Translate” with the moving part occluded.



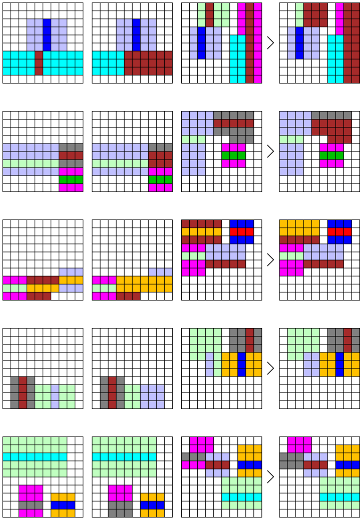
“Detect” quiz extended with location markers colored according to the color of the rectangle they mark.



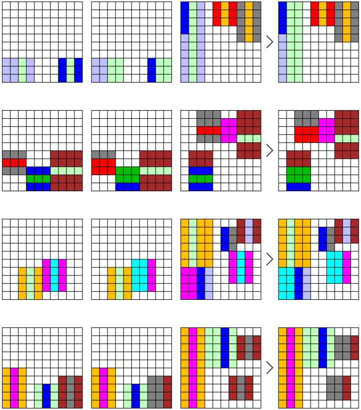
“Half Fill”, “Detect”, “Translate”, “Grow”, and “Frame” with various number of rectangles.



Variations of “Half Fill” where the shapes to change have more complex coloring.



Variations of “Half Fill” with non-rectangular shapes.



Variations of “Half Fill” with two colors or two rectangles have to be modified.

COMMENTS

Even though the two first grids have the form of an example, as in a one-shot learning, their role is more nuanced and may drift away from that interpretation when the culture grows.

Since there are initially few tasks, the two first grids mainly indicate the task identity, possibly with some instance-specific parameters, such as what color to change, or in which direction to move.

Even though the two first grids have the form of an example, as in a one-shot learning, their role is more nuanced and may drift away from that interpretation when the culture grows.

Since there are initially few tasks, the two first grids mainly indicate the task identity, possibly with some instance-specific parameters, such as what color to change, or in which direction to move.

However, most of the tasks' structures is likely encoded in the model parameters, and not extracted from the two first grids.

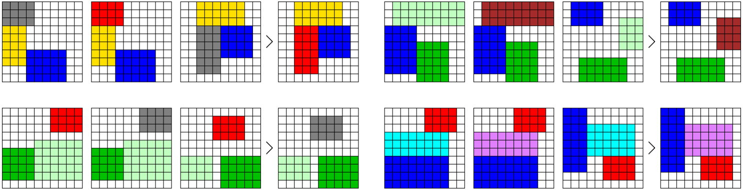
When the culture grows, the structure

$$A, f(A), B, f(B)$$

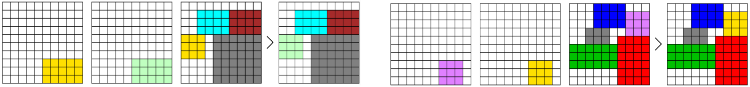
can generalize to

$$X, Y, Z, \Phi(Z; X, Y).$$

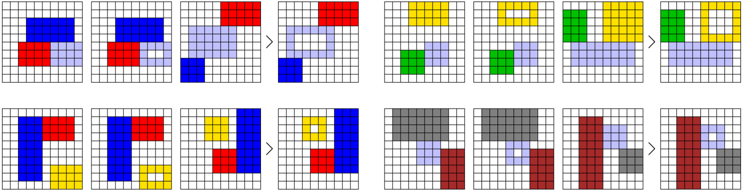
What we observe is that the two first grids may become a language-like description of the task **following shared conventions that have emerged in the culture.**



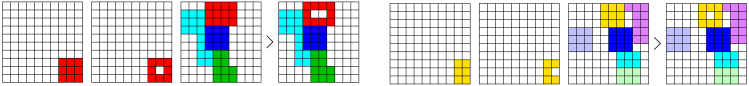
World quizzes from the “Replace Color” task.



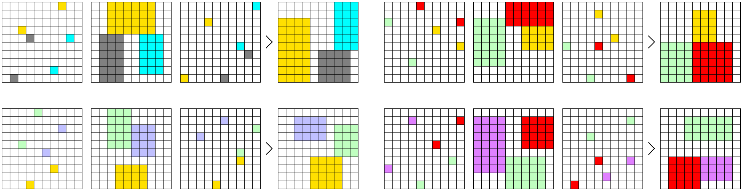
Some generated culture quizzes



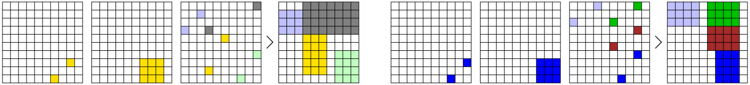
World quizzes from the “Replace Color” task.



Some generated culture quizzes



World quizzes from the “Replace Color” task.



Some generated culture quizzes

A simple and unwelcome way to generate quizzes that can be solved by $N-1$ models is to degrade randomly world quizzes. By adding a certain amount of noise, the process is assured to meet the $N-1$ correct predictions, without leading to the emergence of interesting structures.

To prevent this, we have designed all the tasks to be reversible, that is f is a one-to-one mapping, and we require to have $N-1$ models that solve properly both

$$A, f(A), B \to f(B)$$

and

$$f(A), A, f(B) \to B.$$

Doing so prevent the degradation of the process toward unstructured quizzes.

If Y is a model prediction, we want

$$P(Y = f(B)) = 1 - \delta.$$

However, this is not sufficient since this rate of failure could be due to each model having non-deterministic logits.

If Y is a model prediction, we want

$$P(Y = f(B)) = 1 - \delta.$$

However, this is not sufficient since this rate of failure could be due to each model having non-deterministic logits.

If M is the model index, we also want

$$\forall M, P(Y = f(B) \mid M) \in \{0, 1\}.$$

If Y is a model prediction, we want

$$P(Y = f(B)) = 1 - \delta.$$

However, this is not sufficient since this rate of failure could be due to each model having non-deterministic logits.

If M is the model index, we also want

$$\forall M, P(Y = f(B) \mid M) \in \{0, 1\}.$$

Our final criterion is:

Solve the quiz R times with each one of the N models, resulting in N scores in $\{0, \dots, R\}$. Validate the quiz if all the scores are equal to R but one equal to 0.

QUESTIONS ?