make ai speak computer

@dottxtai

cameron pfiffer

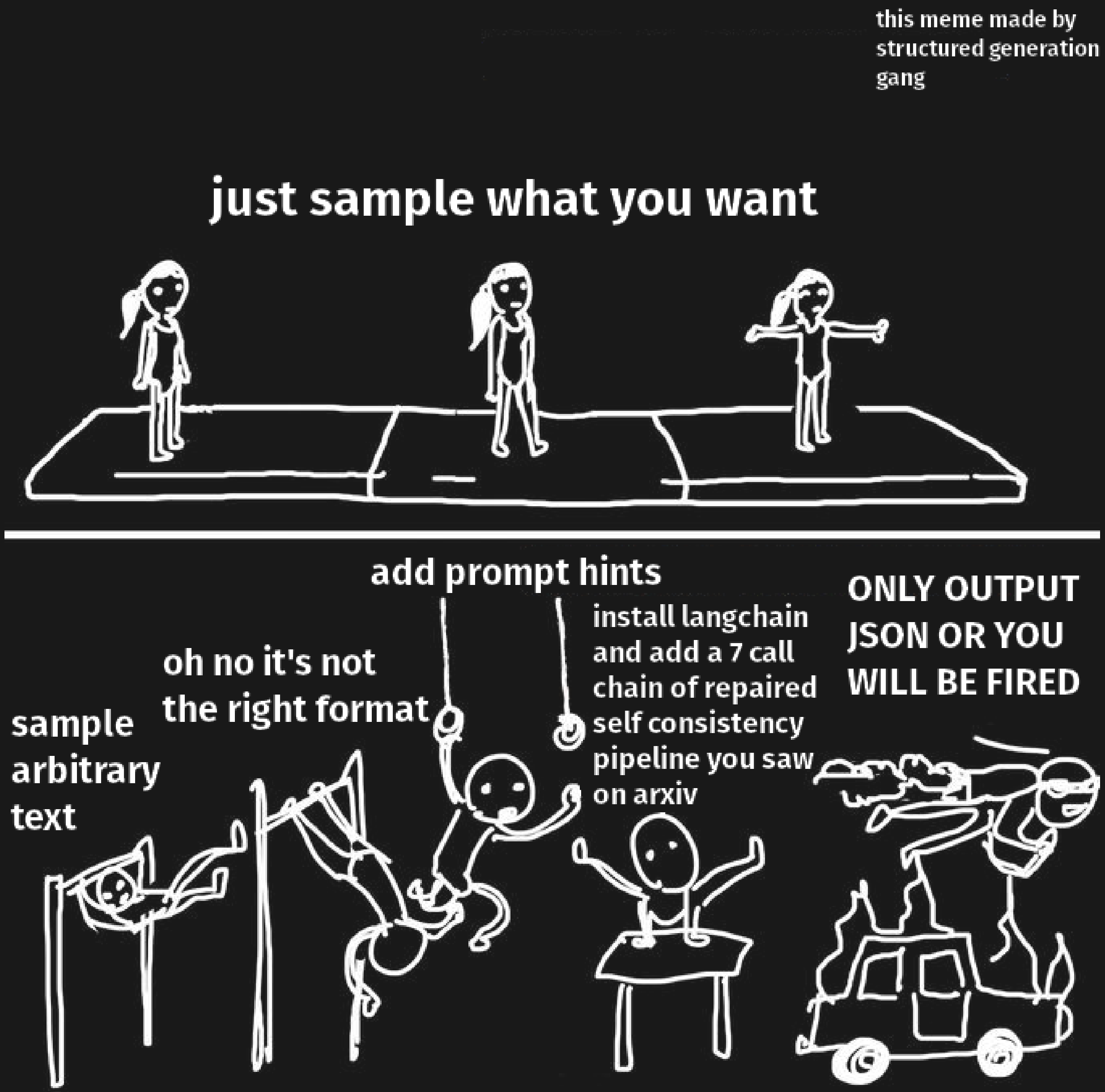
@cameron_pfiffer

i work at .txt

we make outlines

(it's open source)

outlines is how you make ai speak computer


this meme made by structured generation gang
just sample what you want
sample arbitrary text
oh no it's not the right format
add prompt hints
install langchain and add a 7 call chain of repaired self consistency pipeline you saw on arxiv
ONLY OUTPUT JSON OR YOU WILL BE FIRED

(thanks to @vooooogel)

just write code the way you already do

outlines

guarantees you'll get what you ask for

ask for pydantic types

```python
import outlines

llm = "NousResearch/Hermes-3-Llama-3.1-8B"
model = outlines.models.transformers(llm)

# Let's specify the types of games we can choose from.
class GameSettingType(str, Enum):
    cyberpunk = "cyberpunk"
    solarpunk = "solarpunk"
    fantasy = "fantasy"

class GameSetting(BaseModel):
    setting: GameSettingType
    description: str

generator = outlines.generate.json(model, GameSetting)
setting = generator("Pick a game setting and describe it in JSON format.")
```

get pydantic types back

```
GameSetting(
    setting="cyberpunk",
    description="""
    A cityscape of neon lights and towering skyscrapers,
    where the elite live in a floating fortress,
    while the rest eke out a precarious existence in the slums below.
    """
)
```

in json this is

```
{
    "setting": "cyberpunk",
    "description": "A cityscape of neon lights and
    towering skyscrapers, where the elite live in a floating fortress,
    while the rest eke out a precarious existence in the slums below.
    Powerful megacorporations hold more influence than governments, and
    cybernetic enhancements are commonplace."
}
```

outlines is how you program with concepts let's do some

combat

our story

(a robot made this)

```
╭─────────── Neon Blades: A Cyberpunk Showdown ────────────╮
│ In a neon-drenched future, mega-corporations hold the    │
│ power. The streets are filled with augmented humans,     │
│ cybernetic implants, and advanced technology. The rich   │
│ live in luxurious sky-scrapers while the poor fight for  │
│ survival in the gritty, crime-ridden lower levels. The   │
│ line between human and machine has been blurred, and the │
│ world is on the brink of a new revolution as factions    │
│ fight for control of the city's dwindling resources and  │
│ its inhabitants' minds and bodies. The city's leader, a  │
│ powerful AI, maintains control through its vast network  │
│ of surveillance and cybernetic enforcers. The conflict   │
│ between the haves and have-nots is reaching a boiling    │
│ point, and the streets are no longer safe for the        │
│ innocent or the guilty alike. In this world, the only    │
│ way to survive is to outsmart the system and fight for   │
│ your own version of the future.                          │
│                                                          │
│ Setting type: cyberpunk                                  │
╰──────────────────────────────────────────────────────────╯
```

our characters

```
╭────────────────────────── Zara ──────────────────────────╮
│ A genetically-enhanced mercenary with a quick trigger    │
│ finger and a prosthetic arm that houses a hidden         │
│ cybernetic blade. She has honed her skills through years │
│ of skirmishes and battles in the unforgiving streets of  │
│ the city. Her loyalty lies only with the highest bidder, │
│ and she is known for her ability to adapt to any         │
│ situation with ruthless efficiency.                      │
│                                                          │
│ Attack Skill: high                                       │
│ Defense Skill: medium                                    │
│                                                          │
╰──────────────────────────────────────────────────────────╯
╭────────────────────────── Axel ──────────────────────────╮
│ A disgruntled former employee of one of the              │
│ mega-corporations, Axel has turned to a life of crime to │
│ make a living after being betrayed by the system. He is  │
│ a skilled hacker and has managed to stay one step ahead  │
│ of the corporate enforcers for years. His latest heist   │
│ has put him in the crosshairs of the city's most         │
│ dangerous mercenary, Zara.                               │
│                                                          │
│ Attack Skill: medium                                     │
│ Defense Skill: high                                      │
│                                                          │
╰──────────────────────────────────────────────────────────╯
```

why are they fighting

```
╭─────────────────── Start of the battle ───────────────────╮
│ Axel has just pulled off the heist of a lifetime,         │
│ stealing valuable data from one of the                    │
│ mega-corporations. But his celebration is cut short when  │
│ he is ambushed in an alley by Zara, the city's most       │
│ notorious mercenary. She has been hired by the            │
│ corporation to retrieve the stolen data at any cost, and  │
│ she is not about to let Axel slip through her fingers.    │
│ The two adversaries face off in a brutal battle, their    │
│ skills and wits tested to the limit as they fight for     │
│ control of the city's future.                             │
╰───────────────────────────────────────────────────────────╯
```

programming with

outlines

specify our **setting**

```
# Let's specify the types of games we can choose from.
class GameSettingType(str, Enum):
    cyberpunk = "cyberpunk"
    solarpunk = "solarpunk"
    fantasy = "fantasy"

class GameSetting(BaseModel):
    setting: GameSettingType
    description: str # this part is unstructured!
```

define skills

```python
class CombatSkillLevel(str, Enum):
    low = "low"
    medium = "medium"
    high = "high"

    def modifier(self):
        if self == CombatSkillLevel.low:
            return 1
        elif self == CombatSkillLevel.medium:
            return 2
        elif self == CombatSkillLevel.high:
            return 3
```

bundle them up

```python
# Nested types!
class Skills(BaseModel):
    attack: CombatSkillLevel
    defense: CombatSkillLevel

    def attack_modifier(self):
        return self.attack.modifier()

    def defense_modifier(self):
        return self.defense.modifier()
```

define a Character

```python
# Here's a class for the characters.
class Character(BaseModel):
    # Attributes
    name: str
    description: str
    skills: Skills # this is a nested type!
    health_points: int = 10

    # . . . methods . . .
```

define methods

```
class Character(BaseModel):
    # . . . misc init . . .

    def attack(self, opponent):
        # Get the modifiers for your attack and your opponent's defense
        attack_modifier = self.skills.attack_modifier()
        defense_modifier = opponent.skills.defense_modifier()

        # Roll a 10-sided die and add the attack modifier
        attack_roll = random.randint(1, 10) + attack_modifier
        defense_roll = random.randint(1, 10) + defense_modifier

        # Calculate damage
        if attack_roll > defense_roll:
            return attack_roll - defense_roll
        else:
            return 0

    # . . . take_damage method . . .
```

stick it in a Story

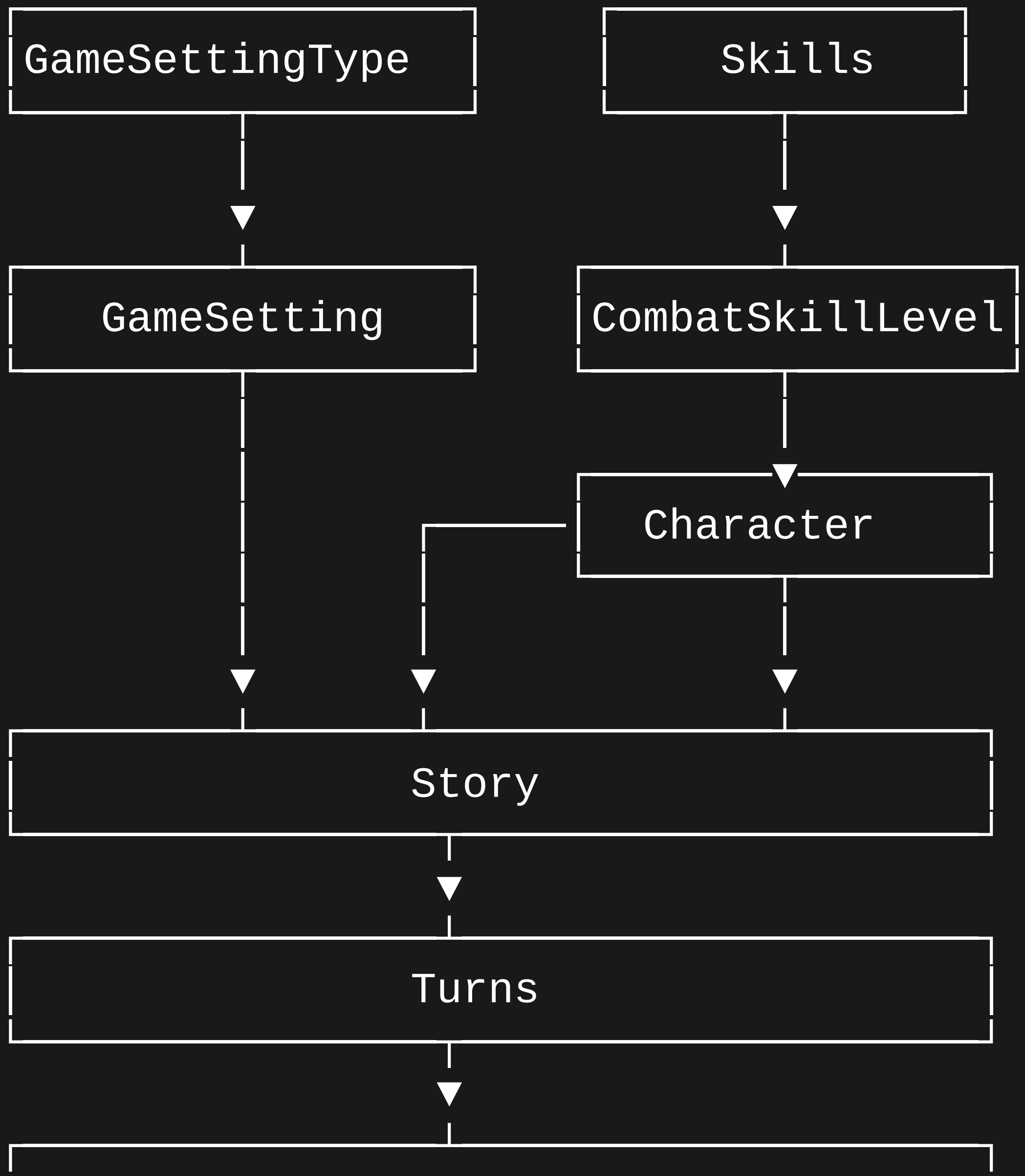
```python
# A story is
# - a setting,
# - two characters,
# - a reason for battle, and
# - a title
class Story(BaseModel):
    setting: GameSetting
    characters: Annotated[list[Character], Len(2)]
    reason_for_battle: str
    title_of_story: str
```

some conveniences

```
# A turn in combat
class Turn(BaseModel):
    description: str

# The final story wraps everything up -- the battle, the characters, etc.
# It'll also include the implications of the battle for the world.
class FinalStory(BaseModel):
    end_of_battle_description: str
    implications_of_battle: str
```

our types


GameSettingType
Skills
GameSetting
CombatSkillLevel
Character
Story
Turns

make a prompt

```python
@outlines.prompt
def story_prompt():
    """
    <|im_start|>user
    You are the best game master in the world.
    You are tasked with creating a story for a
    roleplaying game. The game features two characters
    fighting one another.

    Select a setting from the following options:
    - Cyberpunk
    - Solarpunk
    - Fantasy

    You'll need to create

    - A scenario, which includes a setting, characters,
      and a description of the encounter
    - A reason for the battle
    - A title for the story

    Characters have skills, which may be low, medium, or high.
```

let's start

```python
def main():
    # Set up our story. This call generates a
    # everything we need to get started.
    story_generator = outlines.generate.json(model, Story)
    story = story_generator(story_prompt())

    # Print out the characters
    for character in story.characters:
        # Print out some character information
        print(character)
```

the combat loop

```python
# Main combat loop. This runs as long as all characters
# are still alive.
while all(character.health_points > 0 for character in story.characters):
    # Loop through each character. The characters will take
    # turns attacking each other.
    for character in story.characters:
        # Get the opponent
        opponent = next(c for c in story.characters if c != character)

        # Check if the character's attack is successful.
        # a successful attack is when the attack roll
        # is greater than the defense roll, and the
        # damage is the attack roll minus the defense roll.
        damage = character.attack(opponent)

        # . . . handle the attack . . .
# . . . still in the combat loop . . .

# Turn the past turns into a string
action_history = story.setting.description + \
    "\n" + story.reason_for_battle + \
    "\n".join([f"{turn.description}" for turn in turns])

# action_prompt is a function that takes in
# the character, the opponent, the action history,
# the attack success, and the damage. This is
# the prompt we'll give to the model.
prompt = action_prompt(
    character,
    opponent,
    action_history,
    attack_success,
    damage,
)

# Generate the action description
action_generator = outlines.generate.json(model, Turn)
action_description = action_generator(prompt)

# Add the action to the list of turns
turns.append(action_description)

# . . . loop finished . . .
```

the turns

Turn 1

In a flash of steel, Zara lunged at Axel with her cybernetic blade extended, but he nimbly sidestepped the deadly thrust, his mind racing as he searched for an opening in this dance of death amongst the flashing neon signs and humming robots in the rain-soaked cyberpunk alleys of the city's lower levels.

Zara failed to attack Axel.

Turn 2

With a flash of his razor-sharp cyberblade, Axel launched a lightning-fast strike at Zara's neck, but she nimbly sidestepped the deadly thrust, her enhanced reflexes allowing her to dance away from the attack with inhuman grace amidst the neon-drenched rain-soaked alleyways of the city's lower levels. The strike missed its mark, raining down a shower of sparks as it clashed against the metal wall behind her, and she lunged forward to strike again, her eyes glowing an eerie red as her thoughts raced to predict Axel's every move.

Axel failed to attack Zara.

```python
# If we get here, the battle is over and one character has won.
# Print out the final health points
winner = next(c for c in story.characters if c.health_points > 0)
loser = next(c for c in story.characters if c.name != winner.name)

# Put the history of the battle into a prompt
# asking the language model to describe the final moments
# of the battle.
prompt = final_prompt(
    "\n".join([f"{turn}" for turn in turns]),
    winner,
    loser,
)

# Summarize the final story
final_story_generator = outlines.generate.json(model, FinalStory)
final_story = final_story_generator(prompt)
```

the final moments

With a desperate, dark energy crackling along the length of his cyberblade, Axel unleashed a final, deadly backhand strike that connected squarely with Zara's exposed flank, shattering her battle-wrought resolve and tumbling her to the wet, neon-drenched cement. His face a mask of grime and determination, Axel leapt atop the fallen mercenary, his blade at her throat as the flashing lights of the alley reflected off her closed eyes, the victor and the vanquished lying spent amidst the rain-soaked streets after an unforgiving dance of death.

```
╭──────────────────────── implications ────────────────────────╮
│ Axel's victory over Zara, a deadly and notorious              │
│ mercenary, marks him as the city's new most-wanted            │
│ target. His stolen data puts him in the crosshairs of         │
│ corporate and government forces alike, and he must            │
│ navigate a path through the treacherous underbelly of         │
│ the city to stay one step ahead of the inevitable hunt        │
│ that will be relentlessly pursued by those seeking            │
│ vengeance for the stolen secrets and retribution for the      │
│ ruthlessly vanquished. Zara's death serves as both a          │
│ turning point and a warning to the unsavory elements          │
│ that Axel will stop at nothing to accomplish his goals,       │
│ leaving a chilling legacy of the price for crossing him       │
│ amidst the neon-drenched, rain-soaked alleys of the           │
│ city's lower levels.                                          │
╰───────────────────────────────────────────────────────────────╯
```

we are hosting a cursed outlines contest

make stupid code and win a cool shirt

twitter github discord