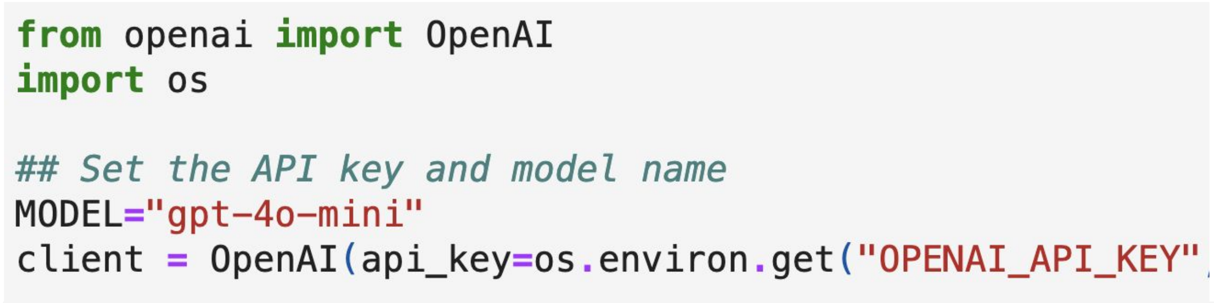
CSCE 689-609
Programming LLM Applications

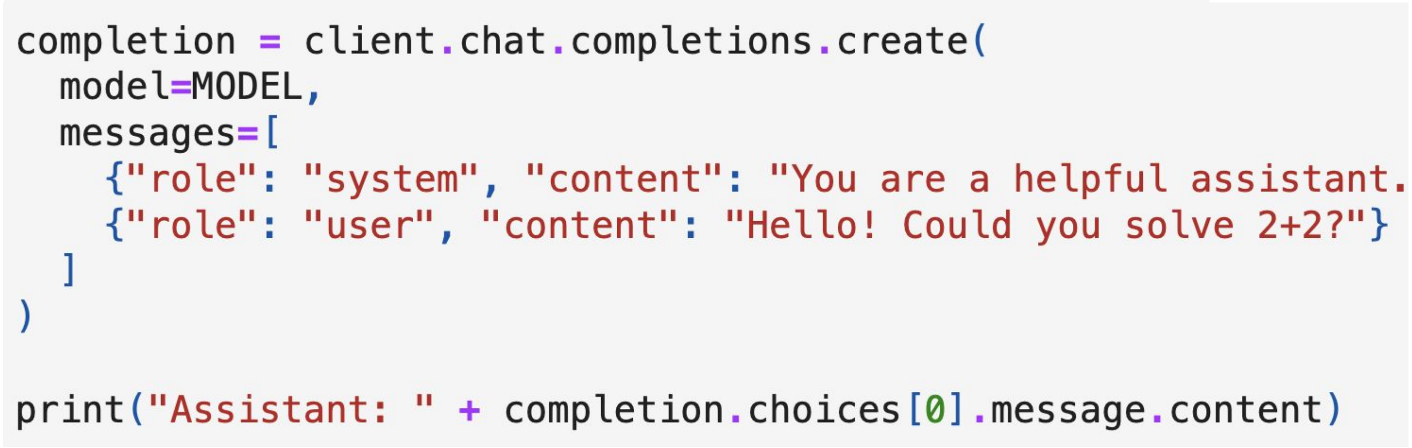
Jeff Huang
jeff@cse.tamu.edu
o2lab.github.io

Hello World

https://github.com/openai/openai-cookbook/blob/main/examples/gpt4o/introduction_to_gpt4o.ipynb

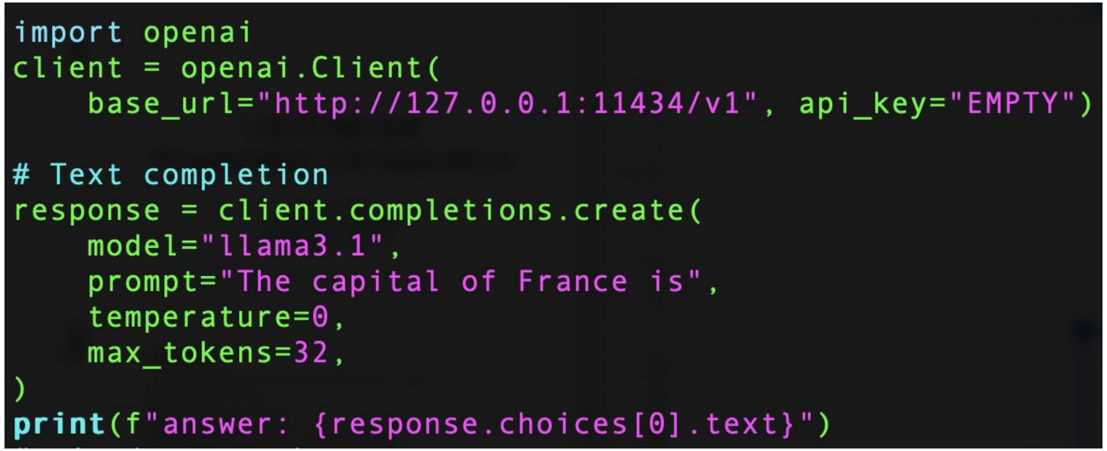


```python
from openai import OpenAI
import os

## Set the API key and model name
MODEL="gpt-4o-mini"
client = OpenAI(api_key=os.environ.get("OPENAI_API_KEY",
```

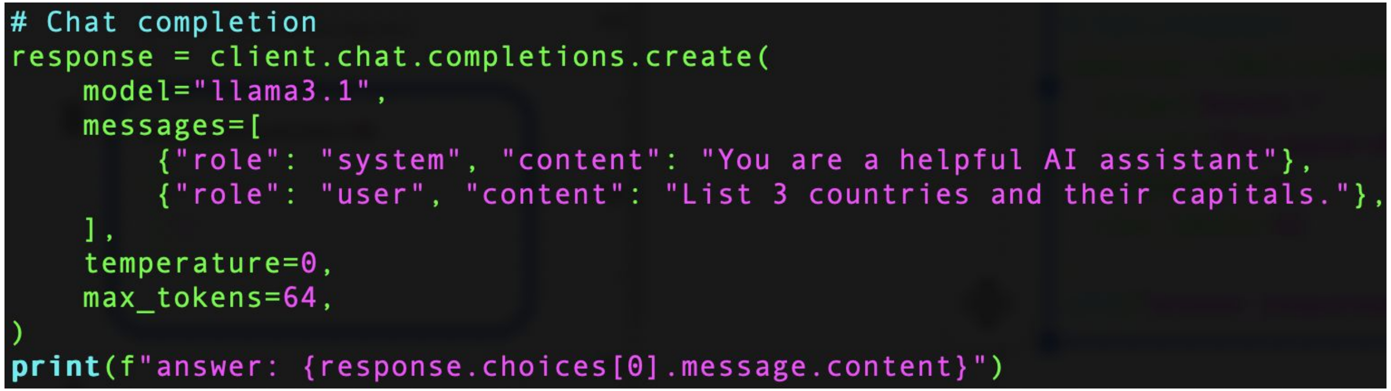


```python
completion = client.chat.completions.create(
  model=MODEL,
  messages=[
    {"role": "system", "content": "You are a helpful assistant.
    {"role": "user", "content": "Hello! Could you solve 2+2?"}
  ]
)

print("Assistant: " + completion.choices[0].message.content)
```

Hello World (local model llama3.1-8B)

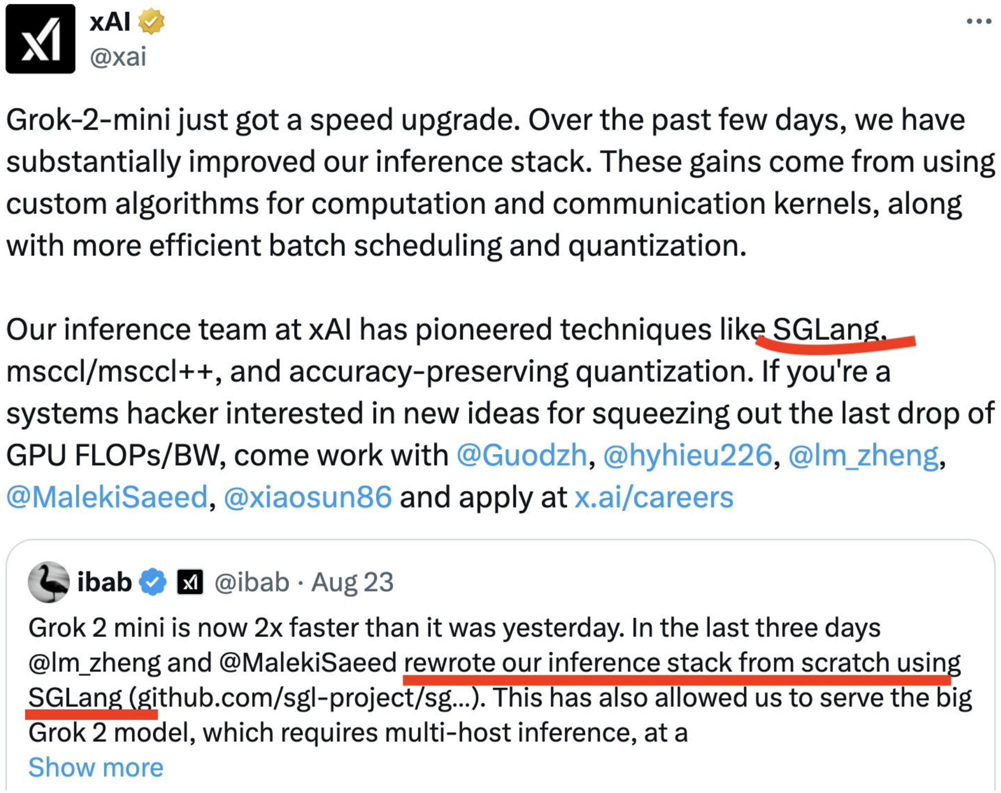


```python
import openai
client = openai.Client(
    base_url="http://127.0.0.1:11434/v1", api_key="EMPTY")

# Text completion
response = client.completions.create(
    model="llama3.1",
    prompt="The capital of France is",
    temperature=0,
    max_tokens=32,
)
print(f"answer: {response.choices[0].text}")
```



```python
# Chat completion
response = client.chat.completions.create(
    model="llama3.1",
    messages=[
        {"role": "system", "content": "You are a helpful AI assistant"},
        {"role": "user", "content": "List 3 countries and their capitals."},
    ],
    temperature=0,
    max_tokens=64,
)
print(f"answer: {response.choices[0].message.content}")
```

SGLang https://github.com/sgl-project/sglang


xAI
@xai

Grok-2-mini just got a speed upgrade. Over the past few days, we have substantially improved our inference stack. These gains come from using custom algorithms for computation and communication kernels, along with more efficient batch scheduling and quantization.

Our inference team at xAI has pioneered techniques like SGLang, msccl/msccl++, and accuracy-preserving quantization. If you're a systems hacker interested in new ideas for squeezing out the last drop of GPU FLOPs/BW, come work with @Guodzh, @hyhieu226, @lm_zheng, @MalekiSaeed, @xiaosun86 and apply at x.ai/careers

ibab @ibab · Aug 23

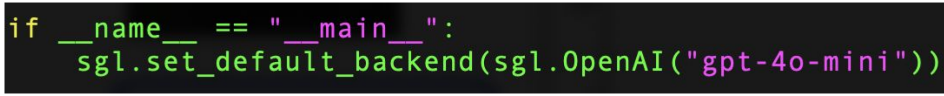
Grok 2 mini is now 2x faster than it was yesterday. In the last three days @lm_zheng and @MalekiSaeed rewrote our inference stack from scratch using SGLang (github.com/sgl-project/sg...). This has also allowed us to serve the big Grok 2 model, which requires multi-host inference, at a

Show more

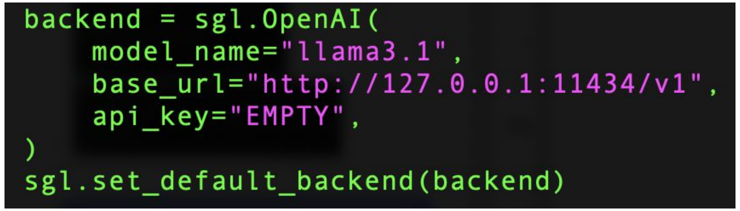
SGLang

https://github.com/sgl-project/sglang/blob/main/examples/frontend_language/quick_start/openai_example_chat.py

```python
@sgl.function
def multi_turn_question(s, question_1, question_2):
    s += sgl.system("You are a helpful assistant.")
    s += sgl.user(question_1)
    s += sgl.assistant(sgl.gen("answer_1", max_tokens=256))
    s += sgl.user(question_2)
    s += sgl.assistant(sgl.gen("answer_2", max_tokens=256))


def single():
    state = multi_turn_question.run(
        question_1="What is the capital of the United States?",
        question_2="List two local attractions.",
    )

    for m in state.messages():
        print(m["role"], ":", m["content"])

    print("\n-- answer_1 --\n", state["answer_1"])
```



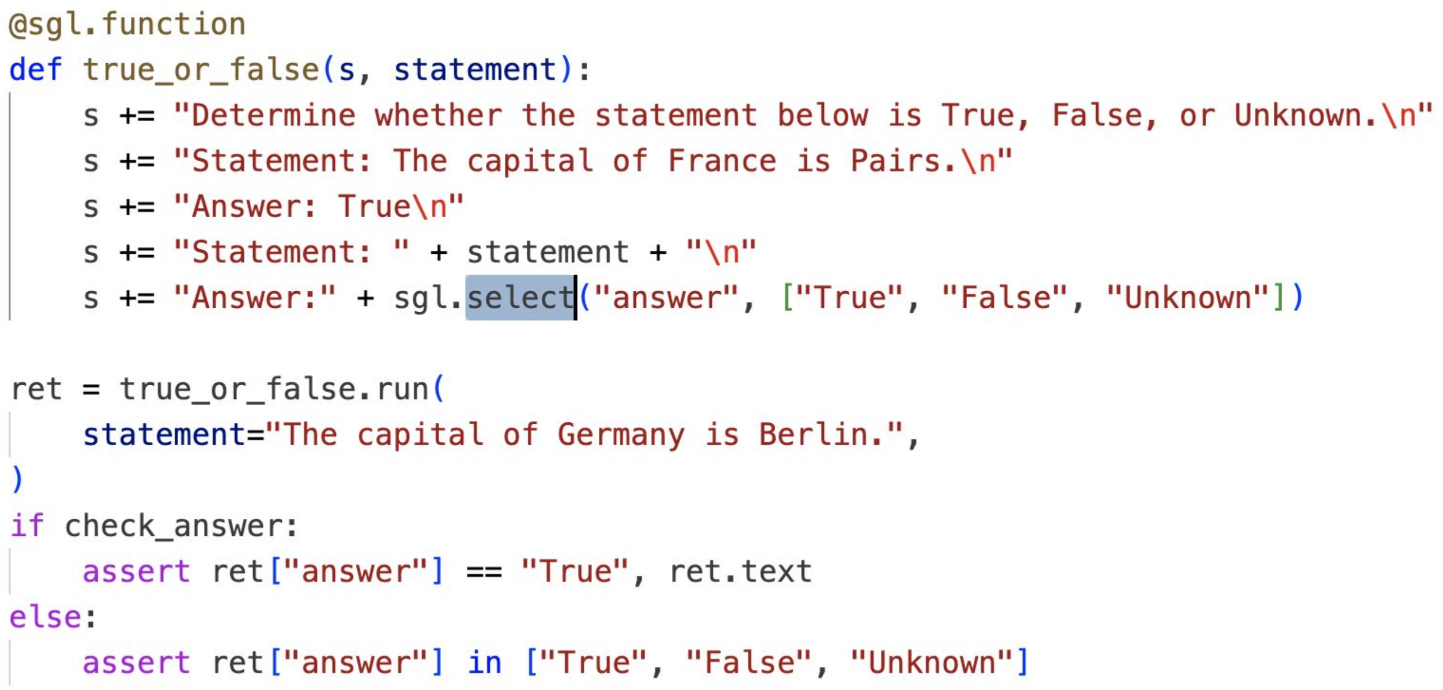
```python
if __name__ == "__main__":
    sgl.set_default_backend(sgl.OpenAI("gpt-4o-mini"))
```



```python
backend = sgl.OpenAI(
    model_name="llama3.1",
    base_url="http://127.0.0.1:11434/v1",
    api_key="EMPTY",
)
sgl.set_default_backend(backend)
```

SGLang (select)

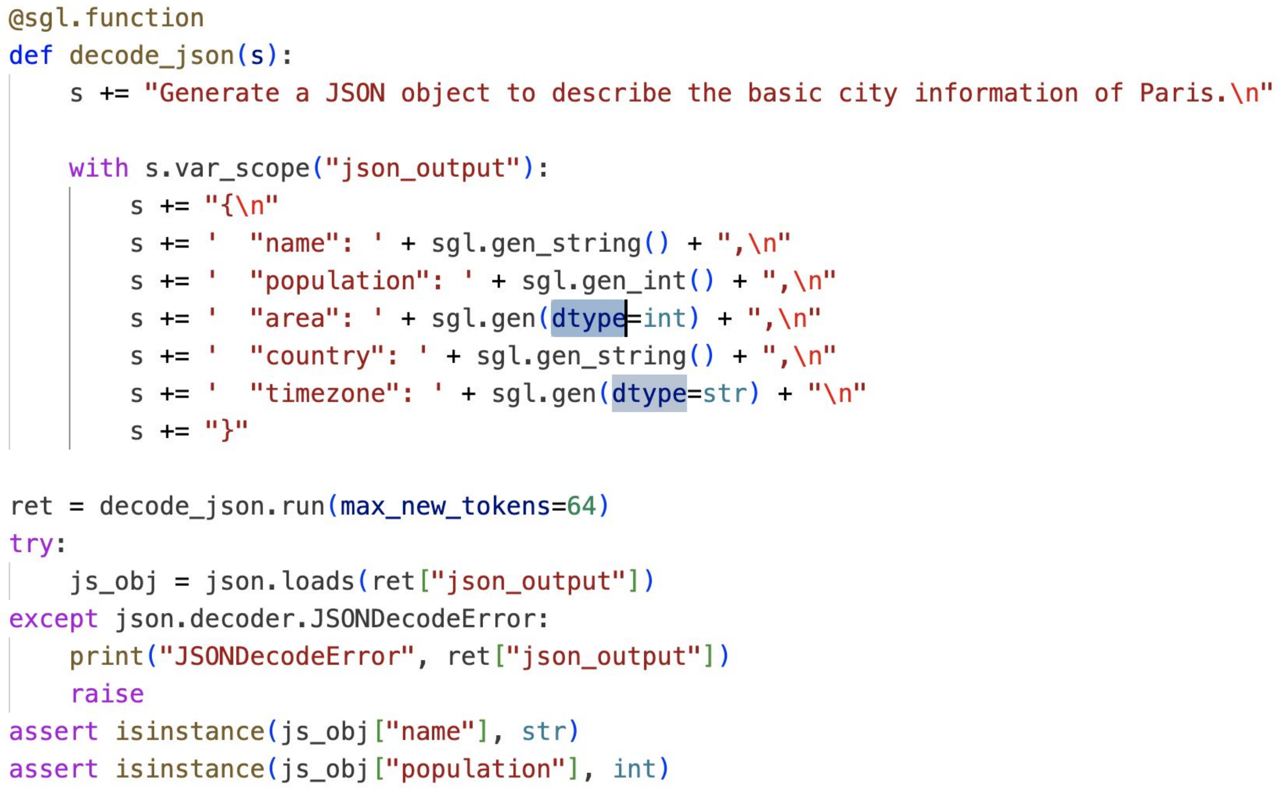
https://github.com/sgl-project/sglang/blob/main/python/sglang/test/test_programs.py



```python
@sgl.function
def true_or_false(s, statement):
    s += "Determine whether the statement below is True, False, or Unknown.\n"
    s += "Statement: The capital of France is Pairs.\n"
    s += "Answer: True\n"
    s += "Statement: " + statement + "\n"
    s += "Answer:" + sgl.select("answer", ["True", "False", "Unknown"])

ret = true_or_false.run(
    statement="The capital of Germany is Berlin.",
)
if check_answer:
    assert ret["answer"] == "True", ret.text
else:
    assert ret["answer"] in ["True", "False", "Unknown"]
```

SGLang (data types)

https://github.com/sgl-project/sglang/blob/main/python/sglang/test/test_programs.py



```python
@sgl.function
def decode_json(s):
    s += "Generate a JSON object to describe the basic city information of Paris.\n"

    with s.var_scope("json_output"):
        s += "{\n"
        s += '  "name": ' + sgl.gen_string() + ",\n"
        s += '  "population": ' + sgl.gen_int() + ",\n"
        s += '  "area": ' + sgl.gen(dtype=int) + ",\n"
        s += '  "country": ' + sgl.gen_string() + ",\n"
        s += '  "timezone": ' + sgl.gen(dtype=str) + "\n"
        s += "}"

ret = decode_json.run(max_new_tokens=64)
try:
    js_obj = json.loads(ret["json_output"])
except json.decoder.JSONDecodeError:
    print("JSONDecodeError", ret["json_output"])
    raise
assert isinstance(js_obj["name"], str)
assert isinstance(js_obj["population"], int)
```

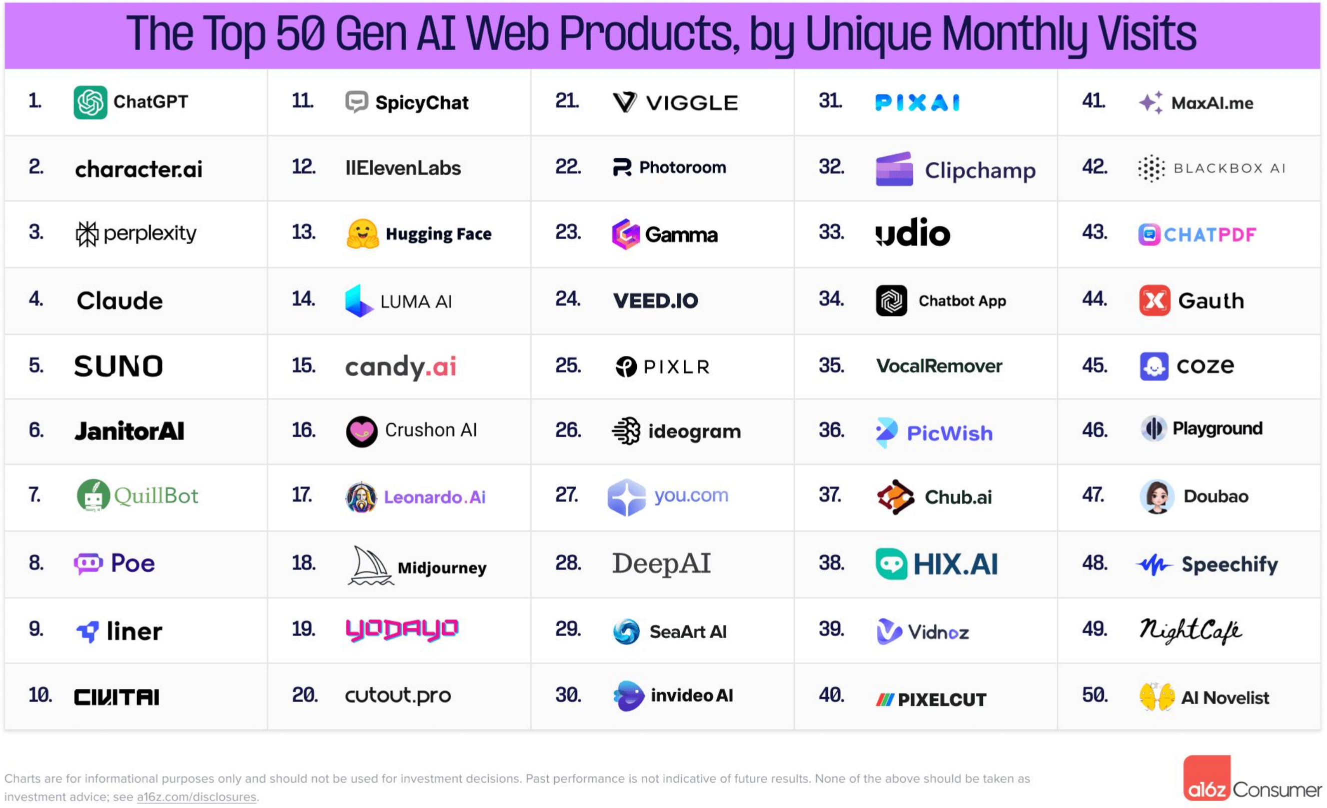
SGLang (regex)

https://github.com/sgl-project/sglang/blob/main/python/sglang/test/test_programs.py

```python
@sgl.function
def decode_json(s):
    from sglang.lang.ir import REGEX_FLOAT, REGEX_INT, REGEX_STR

    s += "Generate a JSON object to describe the basic city information of Paris.\n"
    s += "Here are the JSON object:\n"

    # NOTE: we recommend using dtype gen or whole regex string to control the output

    with s.var_scope("json_output"):
        s += "{\n"
        s += '  "name": ' + sgl.gen(regex=REGEX_STR) + ",\n"
        s += '  "population": ' + sgl.gen(regex=REGEX_INT, stop=[" ", "\n"]) + ",\n"
        s += '  "area": ' + sgl.gen(regex=REGEX_INT, stop=[" ", "\n"]) + ",\n"
        s += '  "latitude": ' + sgl.gen(regex=REGEX_FLOAT, stop=[" ", "\n"]) + "\n"
        s += "}"
```

Popular LLM Applications

https://a16z.com/100-gen-ai-apps-3/



The Top 50 Gen AI Web Products, by Unique Monthly Visits

1. ChatGPT	11. SpicyChat	21. VIGGLE	31. PIXAI	41. MaxAI.me
2. character.ai	12. IIElevenLabs	22. Photoroom	32. Clipchamp	42. BLACKBOX AI
3. perplexity	13. Hugging Face	23. Gamma	33. udio	43. CHATPDF
4. Claude	14. LUMA AI	24. VEED.IO	34. Chatbot App	44. Gauth
5. SUNO	15. candy.ai	25. PIXLR	35. VocalRemover	45. coze
6. JanitorAI	16. Crushon AI	26. ideogram	36. PicWish	46. Playground
7. QuillBot	17. Leonardo.Ai	27. you.com	37. Chub.ai	47. Doubao
8. Poe	18. Midjourney	28. DeepAI	38. HIX.AI	48. Speechify
9. liner	19. YODAYO	29. SeaArt AI	39. Vidnoz	49. NightCafe
10. CIVITAI	20. cutout.pro	30. invideo AI	40. PIXELCUT	50. AI Novelist

Charts are for informational purposes only and should not be used for investment decisions. Past performance is not indicative of future results. None of the above should be taken as investment advice; see a16z.com/disclosures.

a16z Consumer

Popular LLM Applications

1. Chatbots and Virtual Assistants

- Customer support
- Personal assistants: Siri, Alexa, and Google Assistant

2. Content Generation

- Creative writing: stories, poetry, movie scripts, etc
- Marketing, product descriptions, advertisements

3. Translation and Language Services

- Machine translation
- Language correction: Grammarly

4. Education and Tutoring

- Virtual tutors, educational materials, quizzes

5. Search and Information Retrieval

6. Code Generation and Assistance

7. Sentiment Analysis and Opinion Mining

- Analyze customer reviews

8. Summarization and Text Compression

- Meeting notes, summaries of long documents

9. Healthcare

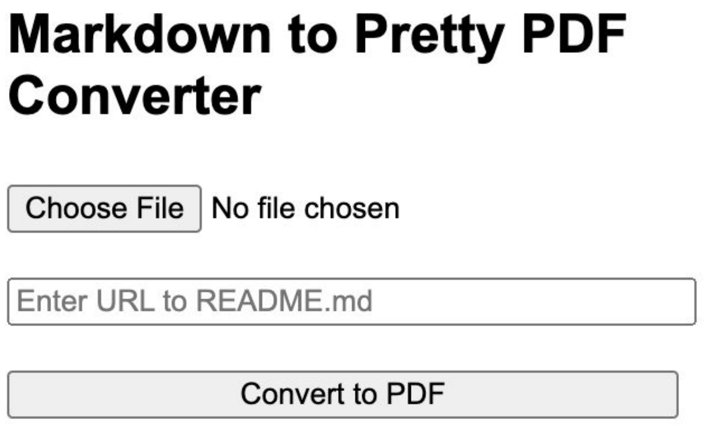
10. Legal and Contract Analysis

…

Cursor (AI Code Editor)

https://www.cursor.com/

Let’s write a README-to-PDF browser extension with AI



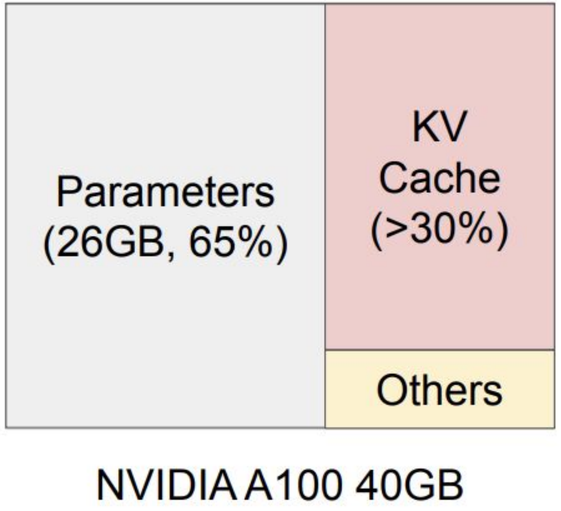
Markdown to Pretty PDF Converter

LLM Limitations (Today)

- **Speed**
  - A bit slow and large memory consumption (require big machines or GPUs to run fast)
- **Limited context window** (100M tokens in a prompt?)
- **Wrong answers**
  - Sometimes deceptive!
- **Hallucination**
  - Responses that are factually incorrect, nonsensical, or disconnected from the input prompt

How much memory is required to run a LLM?

- Model weights
- It also depends on KV cache!



PagedAttention: https://arxiv.org/pdf/2309.06180

How large is KV cache?

The size of the KV cache depends on several factors:

1. **Number of Layers (`L`)**: The number of transformer layers in the model.
2. **Number of Attention Heads (`H`)**: The number of attention heads per layer.
3. **Head Dimension (`d_h`)**: The dimensionality of each attention head.
4. **Sequence Length (`T`)**: The maximum sequence length or the number of tokens stored.
5. **Batch Size (`B`)**: The number of sequences processed in parallel.

The formula for the size of the KV cache is:

$$\text{KV Cache Size} = 2 \times B \times T \times L \times H \times d_h \times 4\,\text{bytes}$$

Question: what can we do if running out of KV cache?

- Efficient Streaming Language Models with Attention Sinks
    - https://arxiv.org/pdf/2309.17453

Case 0: Code Vulnerability Discovery

- HW0

Sample prompt: You are an expert in software security testing. I have a test harness below. Can you generate an input that could exploit the potential vulnerability using the given test harness? Note that the input will be passed to the test harness, so make sure to include proper data format. Please return Python code to generate the input and save it in a file x.bin.

Case 1: Email Assistant

- Part of HW3

Case 2: PDF Assistant

- Part of HW3

Popular Tools to Build LLM Applications

Cloud Infra: Model APIs (gpt-4o, claude-3.5-sonnet, gemini-1.5-pro, etc)

SDK: LiteLLM (Call all LLM APIs)

Local Inference Framework: SGLang, vLLM, NVIDIA/TensorRT-LLM

General purpose framework: Langchain

Building RAG systems: LlamaIndex, Haystack

Local: PrivateGPT

Memory management: MemGPT

Agents: AutoGen

Important Notes

- Read:
    - **Scaling Laws for Neural Language Models**. 2020. https://arxiv.org/pdf/2001.08361
- Due
    - HW1 (next Saturday)