Разработка проектов с языковыми моделями

Легчиков Дмитрий

@legchikov_ai

dmitry-legchikov

@LEGCHIKOV_AI

Бывало ли у вас?

Потому что

- LLM это модно, технологично, интересно
- Реально решают сложные задачи
- Если не мы, то конкуренты опередят

А главное…

Это легко!

Пишешь промпт

вызываешь API

получает ответ

отправляешь пользователям.

Готово!

Ну реально легко!

Решили с командой делать проект с LLM

Про что поговорим

С какой модели начать?

Как найти хороший промпт?

Как оценить качество?

Что со скоростью и надежностью?

А по деньгам что?

Что дальше?

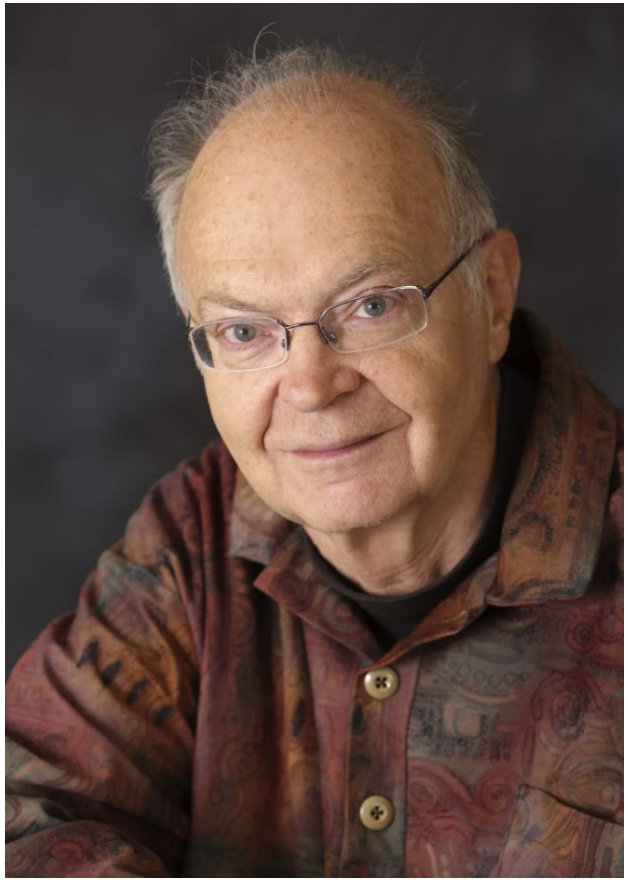
Какую модель выбрать?

ChatGPT

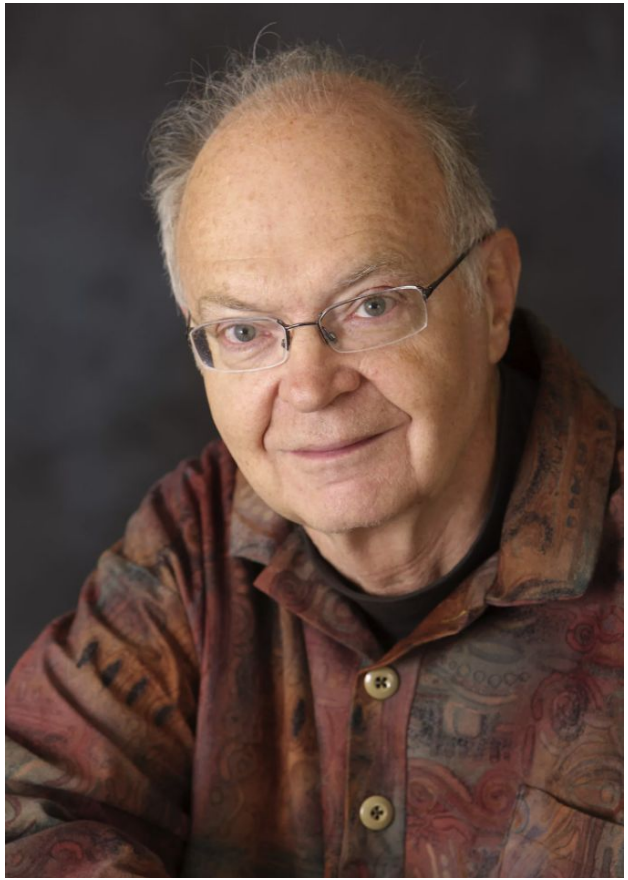
Stanford Alpaca

FALCON LLM
KING OF OPEN-SOURCE LLMS
FOR RESEARCH AND COMMERCIAL USE

H2O.ai h2oGPT



Избегайте преждевременной оптимизации
(c) Дональд Кнут



Избегайте
преждевременной
оптимизации
(с) Дональд Кнут

ChatGPT

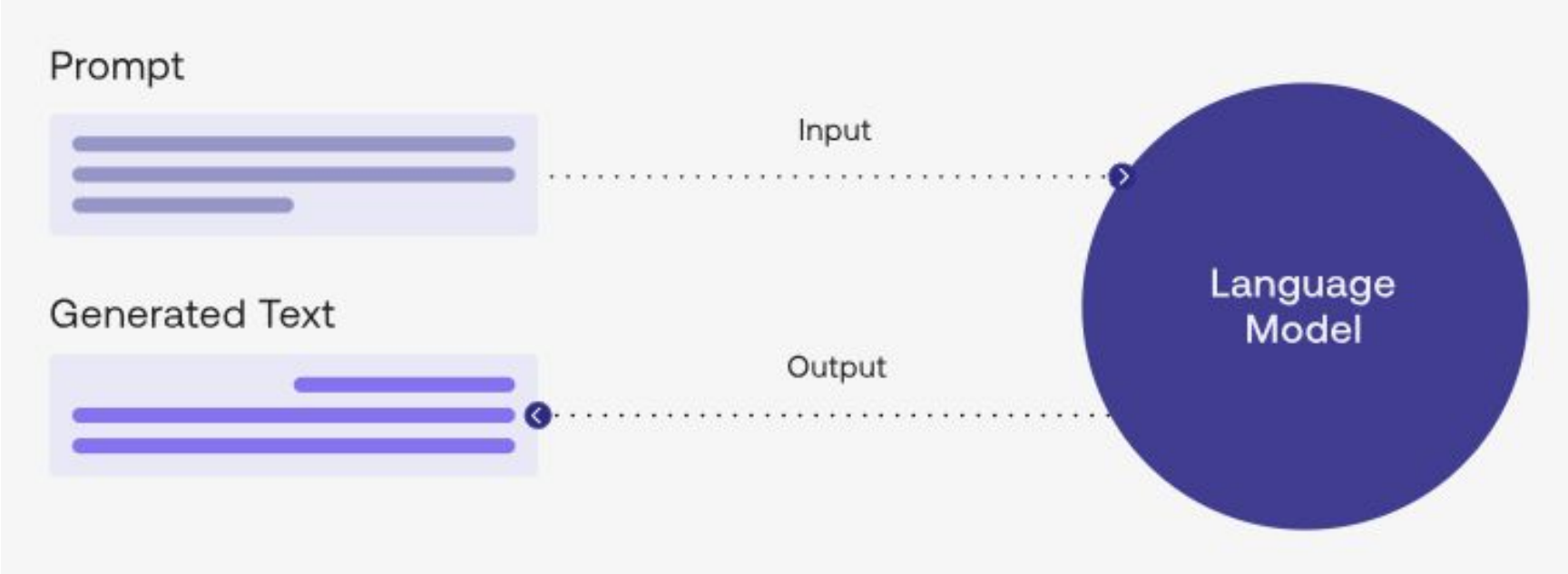
Начинайте с ChatGPT

Если ChatGPT сможет справиться, значит маленькие/заточенные модели тоже смогут

Быстрое итерирование и тестирование гипотез

Ранний запуск позволит собрать фидбек от юзеров

Архитектура



Промпт менеджмент

Промпт менеджмент

Легко потеряться в изменениях

Сложно отслеживать взаимодействие технической и экспертной командой

Маленькое изменение в промпте может сильно влиять на качество

Промпт менеджмент

- Сформулировать какая проблема должна решаться
- Сформулировать кейсы на которых будем тестировать
- Написать промпт
- Генерируем
- Оцениваем результаты
- Улучшаем промпт
- Улучшаем кейсы
- Итерируемся

Промпт менеджмент

Можно начать с Excel таблички
Или выбрать инструмент для Prompt management

Новые появляются почти каждый день
https://topai.tools/s/Prompt-management-tool

Prompttools

In [3]:
```
from typing import Dict, Tuple
from prompttools.harness import PromptTemplateExperimentationHarness
from prompttools.experiment import OpenAICompletionExperiment
```

In [4]:
```
prompt_templates = [
    "Generate valid JSON from the following input: {{input}}",
    "Generate valid python to complete the following task: {{input}}",
]
user_inputs = [
    {"input": "The task is to count all the words in a string"},
    {"input": "The task is to add up numbers 1 to 100"},
]
```

In [5]:
```
harness = PromptTemplateExperimentationHarness(
    OpenAICompletionExperiment,
    "text-davinci-003",
    prompt_templates,
    user_inputs,
    # Zero temperature is better for
    # structured outputs
    model_arguments={"temperature": 0},
)
```

Качество

Почему важно оценивать качество

LLM модели непрерывно обновляются

LLM совершают много ошибок

Если новый промпт работает лучше на нескольких примерах - нет никаких гарантий что в целом он лучше

Какие бывают ошибки

Галлюцинации (неверная информация)

Неправильный формат

Неправильный тон (грубый или оскорбления)

Типы оценки качества

Бенчмарки ($)

Оценка другой LLM ($$)

Человеческая оценка качества ($$$)

Оценка другой LLM

Как могут выглядить промпты:

“Оцени на сколько два ответа содержат корректные факты”

“Оцени какой из двух ответов лучше, согласно критериями (1), (2) и (3)”

“Включает ли ответ обратную связь полученную ранее?”

Как оценивать качество?

https://arize.com/

Summarization

https://colab.research.google.com/github/Arize-ai/phoenix/blob/main/tutorials/evals/evaluate_summarization_classifications.ipynb

Hullatication

https://colab.research.google.com/github/Arize-ai/phoenix/blob/main/tutorials/evals/evaluate_hallucination_classifications.ipynb

Еще про качество

Удивительно, но многие компании готовы пожертвовать небольшой потерей качества ради стабильности и надежности.

Старайтесь сделать задачу для LLM как можно проще:
- Генерацию заменить классификацией
- Уменьшение входа
- Уменьшение выхода

Когда оценивать качество?

Во время промпт инженеринга

Во время эксплуатации

После изменений (промпта, модели, эмбеддингов)

Надежность

Архитектура RAG

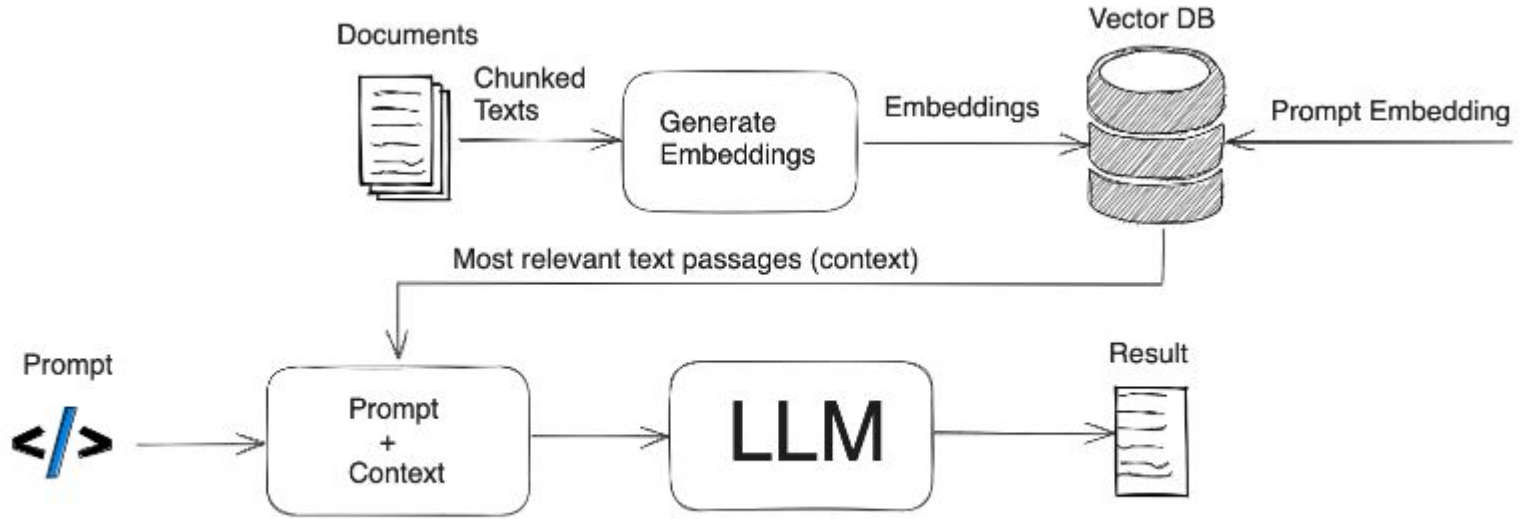


Figure 2. RAG operation. Information preparation and storage. Augmenting prompt with external information.

Source: https://safjan.com/understanding-retrieval-augmented-generation-rag-empowering-llms/

Телеметрия

Ты не можешь починить то о чем не знаешь

API провайдеры не дают SLA на вызовы - может работать медленно, отваливаться

Архитектура RAG

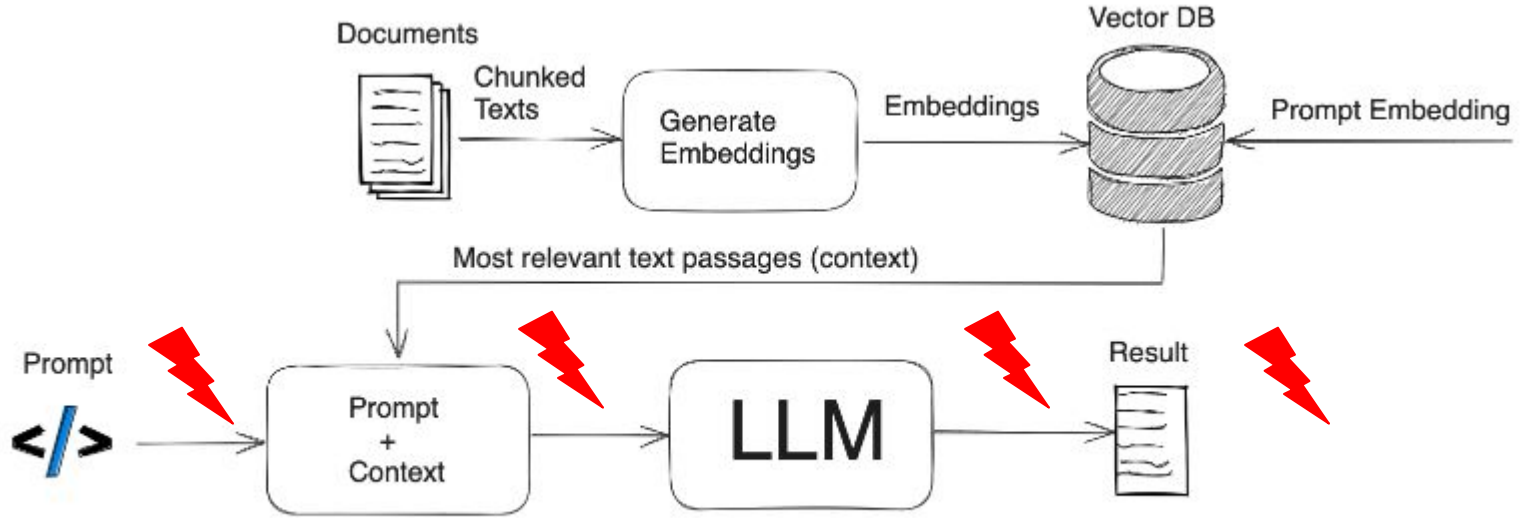


Figure 2. RAG operation. Information preparation and storage. Augmenting prompt with external information.

Source: https://safjan.com/understanding-retrieval-augmented-generation-rag-empowering-llms/

Какие данные собирать?

Запросы и ответы

Данные из контекста

Пользовательский фидбек

Проверка формата выхода LLM

Формата выхода (JSON, Python code, etc)
Пустые строки
Длина выхода

Model-based checks и prompt tricks
(“Are you sure”?)
(“It’s very important”)

Проверка формата выхода LLM

In [7]:
```
from prompttools.utils import validate_json_response
from prompttools.utils import validate_python_response
```

In [8]:
```
harness.evaluate("is_json", validate_json_response, {"response_column_name": "response"})
harness.evaluate("is_python", validate_python_response, {"response_column_name": "response"})
harness.visualize()
```

```
/Users/kevin/hegel/prompttools/prompttools/utils/validate_python.py:28: DeprecationWarning: 'epylint' will be removed in pylint 3.0, use https://github.com/emacsorphanage/pylint instead.
  pylint_stdout, _ = lint.py_run(PROMPTTOOLS_TMP, return_std=True)
```

	prompt	response	latency	is_json	is_python
0	Generate valid JSON from the following input: The task is to count all the words in a string	{"text": "George Washington"}	0.000015	1.0	1.0
1	Generate valid JSON from the following input: The task is to add up numbers 1 to 100	{"text": "George Washington"}	0.000004	1.0	1.0
2	Generate valid python to complete the following task: The task is to count all the words in a string	{"text": "George Washington"}	0.000003	1.0	1.0
3	Generate valid python to complete the following task: The task is to add up numbers 1 to 100	{"text": "George Washington"}	0.000007	1.0	1.0

Fallbacks

Выводить сообщение о неполадках

Выводить заранее сгенерированный ответ

Использовать другую LLM

https://python.langchain.com/docs/guides/fallbacks

Retries

Маскирование персональных данных (PII Masking)

NER модель для маскирования

Другая LLM

https://gpt-index.readthedocs.io/en/latest/examples/node_postprocessor/PII.html

Производительность

Как следить за производительностью?

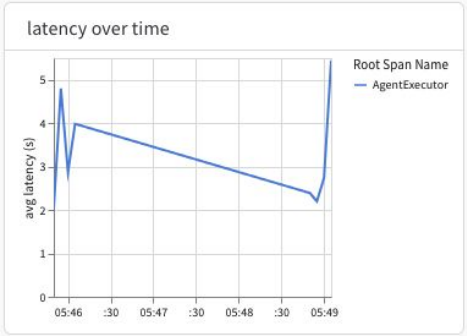
WANDB Prompts
https://colab.research.google.com/github/wandb/weave/blob/master/examples/prompts/trace_debugging/trace_quickstart_langchain.ipynb

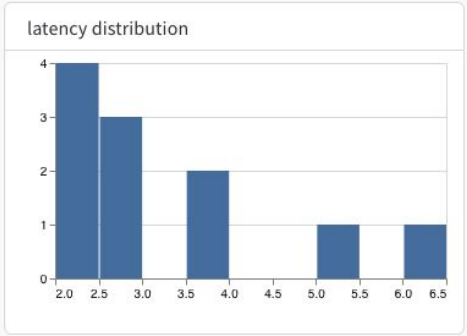
https://docs.wandb.ai/guides/prompts

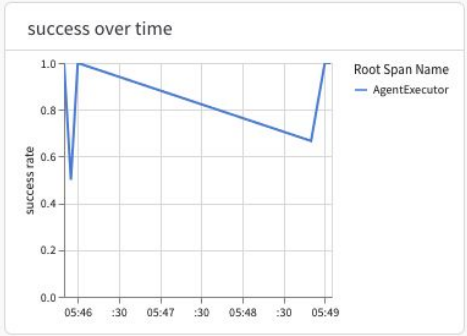
DEMO

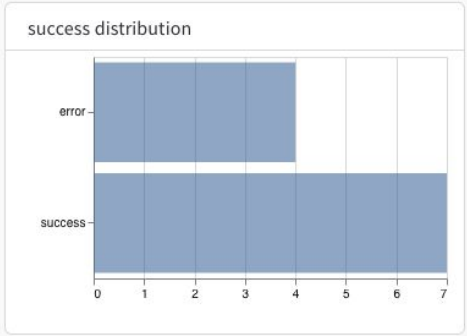
https://weave.wandb.ai/?exp=get%28%0A++++%22local-artifact%3A%2F%2F%2Fdemo_trace_langchain_quickstart%3Auser-latest%2Fobj%22%29

Как следить за производительностью?









traces table

	Success	Span Name	Trace ID	Latency	Inputs	Output	Timestamp
11	True	AgentExecutor	b96cd82f-ebfd-4c33-aefb-e621bb463001	5.446	input What is the sin of 0.47 radians, divided by the	output 0.016773566125150678	2023-09-27 17:49:05
10	True	AgentExecutor	38429b6e-17be-4cd3-a126-914f3c498db1	2.739	input Find the integral of x squared from 0 to 3	output The integral of x squared from 0 to 3 is 27.0	2023-09-27 17:49:01
9	False	AgentExecutor	4fd2f745-9bd8-4ec3-8101-e08fb50e0ea9	2.201	input How many roots does x^2 + 2x + 4 = 0 have?		2023-09-27 17:48:56
8	True	AgentExecutor	d26f53d8-22b5-406f-b7fa-3c889d388973	2.039	input What is the prime factorization of 1273?	output 1273	2023-09-27 17:48:54
7	False	AgentExecutor	72d85aa0-b16a-403b-b177-b2295a3b95e9	2.348	input Find the derivative of the natural log of x base 2 and		2023-09-27 17:48:52

1 - 5 of 11

Export as CSV Columns... Reset Table

Ограничения для пользователей

Captcha

Лимиты для пользователей и компаний

Фильтрация (например, IP)

Скорость генерации

LLM могут медленно генерировать ответ, и все это время пользователи остаются в неведении.

Стриминг

Custom LLM и API поддерживают возможность не дожидаться генерации целого ответа, а отдавать частями.

Пример для ChatGPT:
https://puppycoding.com/2023/07/08/stream-chatgpt-response/

Стриминг

```
result = openai.ChatCompletion.create(
    model = "gpt-3.5-turbo",
    messages = [
        {
            "role": "user",
            "content": prompt
        }
    ],
    stream = True # Add this optional property.
)
```

```
for chunk in result:
    print(
        chunk.choices[0].delta.get("content", ""),
        end = "",
        flush = True
    )
```

Кэширование

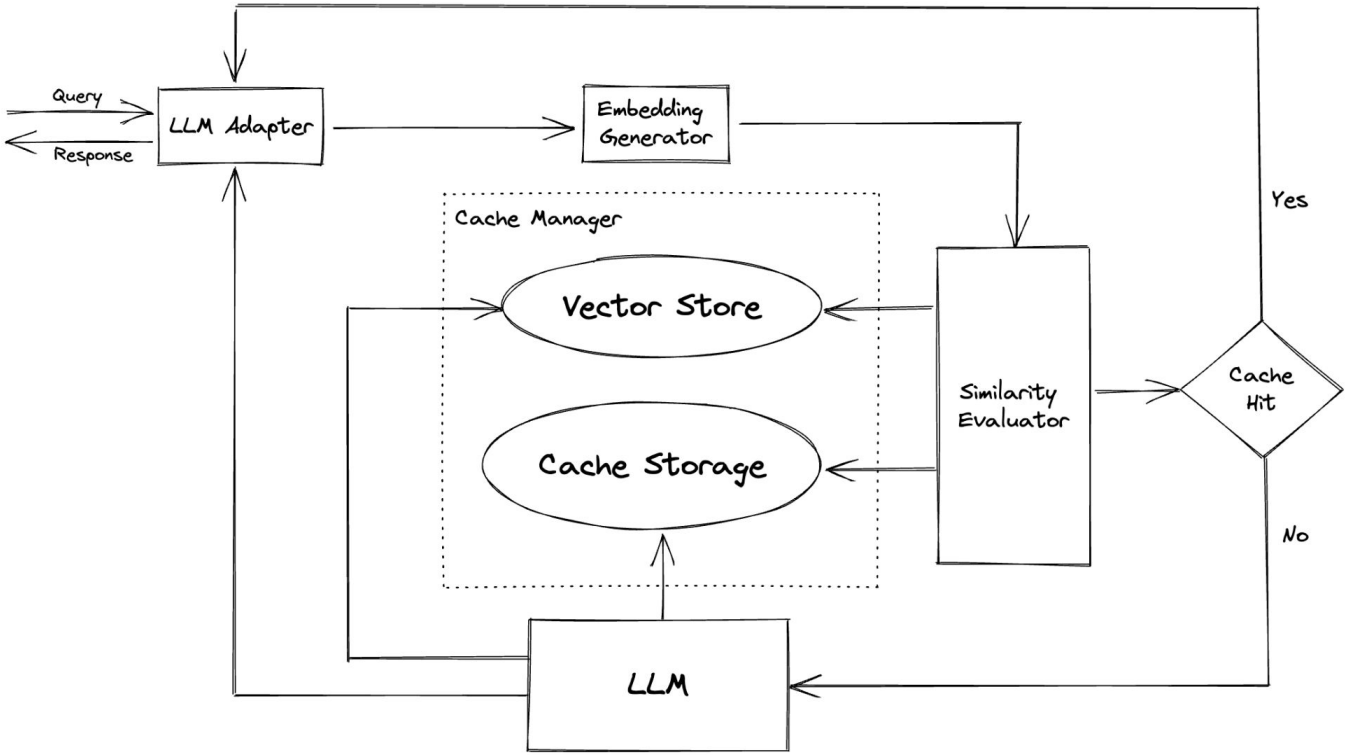
Кэширование

```python
from gptcache import cache
from gptcache.adapter import openai

cache.init()
cache.set_openai_key()
```

```python
question = "what's github"
for _ in range(2):
    start_time = time.time()
    response = openai.ChatCompletion.create(
      model='gpt-3.5-turbo',
      messages=[
        {
            'role': 'user',
            'content': question
        }
      ],
    )
    print(f'Question: {question}')
    print("Time consuming: {:.2f}s".format(time.time() - start_time))
    print(f'Answer: {response_text(response)}\n')
```

```
Question: what's github
Time consuming: 6.04s

Question: what's github
Time consuming: 0.00s
```

Семантическое кэширование



https://gptcache.readthedocs.io/en/dev/usage.html#build-your-cache

Семантическое кэширование

```
from gptcache import cache
from gptcache.adapter import openai
from gptcache.embedding import Onnx
from gptcache.manager import CacheBase, VectorBase, get_data_manager
from gptcache.similarity_evaluation.distance import SearchDistanceEvaluation

print("Cache loading.....")

onnx = Onnx()
data_manager = get_data_manager(CacheBase("sqlite"), VectorBase("faiss", dimension=onnx.dimension))
cache.init(
    embedding_func=onnx.to_embeddings,
    data_manager=data_manager,
    similarity_evaluation=SearchDistanceEvaluation(),
    )
cache.set_openai_key()
```

Метрики эффективность кэширования

Hit Ratio - количество попаданий в кэш (больше лучше)

Latency - время получения ответа из кэша (меньше лучше)

Recall - полнота (больше лучше)

https://gptcache.readthedocs.io/en/latest/configure_it.html

Стоимость

- Во время PoC нормально, если цена будет выше ожидаемого.

- В проде со всеми обвесами при масштабировании цена на API растет очень быстро

- Для русского языка дороже - больше токенов на запрос
  Цена складывается за входные и выходные токены
  https://platform.openai.com/tokenizer

Сбор обратной связи

Подумайте с самого начала какой и как собирать фидбек от пользователей

Это поможет оценивать качество

Будет первым шагом для файнтюнинга

Файнтюнинг

Требуется меньше данных чем кажется

Латенси и стоимость уменьшается

Качество может улучшиться

Максимизируем свои доменные данные

Разработка проектов с языковыми моделями

Легчиков Дмитрий

@legchikov_ai

dmitry-legchikov

@LEGCHIKOV_AI