LLM - Агент NLP

> 🔗 Ссылка на на GitHub проекта с кодом

MVP проактивного агента для работы с задачами и инфой из Вики, GPT-4 + LangChain

Идея

Цель проекта - сделать проактивного агента, который:

- показывает текущие задачи из таск-трекера Todoist, добавляет новые задачи,
- отвечает на вопросы, используя инфу из Википедии,
- самостоятельно понимает в зависимости от запроса, какой `tool` вызвать

Стек

Решил делать на **LangChain**. Вроде модный фреймворк, где и LLM-ку можно подключить, и tools сам будет поддерживать. В качестве LLM-ки - **GPT-4** по API, так как она стабильно работает с лангчейном и человечески даёт адекватные reasoning-ответы. Вики реализовал черел **wikipedia**-пакет

> Оказалось, что ядро LangChain меняется чуть ли не каждый месяц, в новых версиях ломаются импорты и сигнатуры 😕 на Стаке и в обсуждениях GH полно старых примеров, которые не работают 😕😕 даже GPT-5 путается в API LangChain, и предлагает нерелевантные методы 😕😕😕

В итоге я курил доку и собирал рабочий MVP вручную, пришлось дебажить и сравнивать changelog-и. Из-за этого агента пришлось упростить, оставить только 2 требуемых `tools`

Структура проекта

Финальная структура выглядела так:

```
project/
├── src/
├──── main.py
├──── agent/
│     ├── __init__.py
│     └── base_agent.py
├──── local_logger/
│     ├── __init__.py
│     └── local_logger.py
├──── tools/
│     ├── __init__.py
│     ├── todoist_tool.py
```

```
│      └── wiki_tool.py
├──── config/
│      ├──__init__.py
│      └── settings.py
├──── .env
├──── .gitignore
├──── requirements.txt
└──── README.md
```

Изначально задел был на ООП - и на `config/` с `local_logger/` меня хватило. Но проблемы с лангчейном убили время, поэтому под конец я перешёл к функциональщине

Логика агента

Агент в целом строится по схеме React:

- получает запрос от пользователя,
- вызывает нужный инструмент:
    - todoist_manager для задач,
    - wiki_search для гуглинга по Вики сведений,
- возвращает и обрабатывает LLM-кой результат
- пользователь получает итоговый саммари от LLM-ки

Проблемы и примеры

SSL Todoist API

Todoist внезапно падал с ошибкой:

```
SSLError: [SSL: UNEXPECTED_EOF_WHILE_READING]
```

Решить кодом не удалось, проблема судя по всему оказалась в VPN. Без VPN Todoist API работал, но не работал OpenAI API 😭 Такие вот девопс-процессы.

Я сменил VPN endpoint, и Todoist API заработал. На примере ниже - я через агент добавил таску:

```
(.venv) F:\pycode25\mipt-masters\nlp\project>python -m src.main
Агент запущен. Введи запрос ('exit' для выхода):

> list_tasks

Response: Here are the current tasks:

1. **Договориться на встречу по чемодану**
   - Due: 15 августа 2025
   - Created: 14 августа 2025

2. **Проверить функции бук бота - разделить на ветку develop**
   - Due: 15 августа 2025
   - Created: 14 августа 2025

3. **Передеплоить фуд-бота**
   - Due: 15 августа 2025
   - Created: 14 августа 2025

4. **Составить дашборд логов по ru/eng боту**
   - Due: 17 августа 2025
   - Created: 15 августа 2025

Let me know if you need to add any new tasks or require further details on any of these tasks.

> create_task "Сделать агента для университета, дедлайн - 27.10.2025"

Response: Задача "Сделать агента для университета, дедлайн - 27.10.2025" успешно создана.

> list_tasks

Response: Вот список текущих задач:

1. **Договориться на встречу по чемодану** - срок: 15 августа 2025 года.
2. **Проверить функции бук бота - разделить на ветку develop** - срок: 15 августа 2025 года.
3. **Передеплоить фуд-бота** - срок: 15 августа 2025 года.
4. **Составить дашборд логов по ru/eng боту** - срок: 17 августа 2025 года.
5. **Сделать агента для университета, дедлайн - 27.10.2025** - без указанного срока выполнения.

Если нужно добавить новую задачу или получить дополнительную информацию по какой-либо из них, дайте знать!

>
```

С дедлайном вышло не очень, видимо нужно допиливать эндпоинты API-шки в `tool` 🤔

Wikipedia tool

Wikipedia-модуль работал через раз, см. скрин ниже:

```
> Не, расскажи мне что-нибудь про византийских князей. Ис
пользуй википедию

Response: К сожалению, произошла ошибка при попытке получ
ить информацию из Википедии о византийских князьях. Попро
буем ещё раз или исследуем другой запрос?

> wiki_search Византия

Response: Византия, также известная как Восточная Римская
 империя или Византийская империя, представляет собой про
должение Римской империи в её восточных провинциях в пери
од поздней античности и средневековья. Столицей Восточной
 Римской империи был Константинополь. Восточная Римская и
мперия пережила фрагментацию и падение Западной Римской и
мперии в V веке нашей эры и продолжала существовать ещё т
ысячу лет до падения Константинополя перед Османской импе
рией в 1453 году.
```

Византи́я, также известная как **Восточная Римская империя** или **Византийская империя** — продолжение Римской империи в её восточных провинциях в период поздней античности и средневековья, когда столицей Восточной Римской империи был Константинополь. Восточная Римская империя пережила фрагментацию и падение Западной Римской империи в V веке нашей эры и продолжала существовать ещё тысячу лет до падения Константинополя перед Османской империей в 1453 году.

Сверху - пример работы с агентом из консольки. Снизу - оригинальная статья из Вики. На примере также видно, как LLM сначала съела инфу из Вики, а потом передала её нам

Как запустить

Чтобы протестировать агента, нужно:

1. Клонировать проект
2. Заполнить `.env` файл по примеру `.env.example`
3. `python -m venv .venv`. Активировать окружение
4. `pip install -r requirements`
5. `python -m src.main`

Заключение

В целом LangChain - не самый плохой инструмент. Но чтобы делать на нём что-то продуктовое, нужно быть в теме и следить за changelog-ом.

OpenAI-модель ведёт себя стабильно. `Tool` вызывает, результат даёт.

Агент по итогу работает, и даже создаёт задачи. Для учебного MVP можно считать успехом 😅