

Оглавление

Предисловие.....	9
Об авторе	10
Об этой книге.....	11
Глава 1.ChatGPT, GPT, GPT-3, DALL·E, Codex... Что это?	14
Глава 2. Как работает GPT?.....	17
Глава 3. Подготовка среды разработки.....	20
3.1. Установка Python, pip и виртуальной среды для разработки	20
3.2. Получение ключа API OpenAI.....	21
3.3. Установка официальных средств интеграции Python.....	22
3.4. Тестирование ключей API	22
Глава 4. Доступные модели.....	25
4.1. Три основные модели	25
4.2. GPT-3: обработка и генерация естественного языка.....	25
4.3. Codex: понимание и создание компьютерного кода.....	26
4.4. Content Filter.....	27
4.5. Получение списка всех доступных моделей	27
4.6. Какую модель использовать?	31
4.7. Что дальше?	32
Глава 5. Применение GPT для генерации текста	33
5.1. Базовый пример завершения текста	33
5.2. Управление количеством токенов на выходе	35
5.3. Параметр <i>logprobs</i>	36
5.4. Управление креативностью: параметр <i>temperature</i>	41
5.5. Использование параметра <i>top_p</i>	42
5.6. Поточковая передача результатов	43
5.7. Контроль повторов: штрафы за частоту и наличие	46
5.8. Управление количеством выводимых результатов.....	48
5.9. Использование параметра <i>best_of</i>	49
5.10. Управляемое ограничение вывода	50
5.11. Использование суффикса после вывода текста	51
5.12. Пример: извлечение ключевых слов	52
5.13. Пример: генерация твитов	54
5.14. Пример: сочинение песни в стиле рэп	57

5.15. Пример: составление списка дел	58
5.16. Заключение.....	60
Глава 6. Редактирование текста с помощью GPT	61
6.1. Пример: перевод текста.....	61
6.2. Инструкция нужна, но ввод необязателен	63
6.3. Использование конечных точек <i>completions</i> и <i>edits</i>	63
6.4. Форматирование вывода.....	65
6.5. Креативность или определенность?	67
6.6. Создание нескольких правок	70
Глава 7. Примеры более сложной работы с текстом.....	71
7.1. Последовательное использование <i>completions</i> и <i>edits</i>	71
7.2. Apple – это компания или фрукт?	72
7.3. Получение информации о криптовалюте на основе пользовательской схемы (наполнение контекста).....	75
7.4. Создание помощника чат-бота для помощи с командами Linux	77
Глава 8. Встраивание	84
8.1. Что такое встраивание.....	84
8.2. Примеры использования.....	84
8.2.1. Tesla	85
8.2.2. Kalendar AI.....	85
8.2.3. Notion.....	85
8.2.4. DALL-E 2	86
8.3. Подготовка к работе.....	86
8.4. Знакомство со встраиванием текста.....	87
8.5. Встраивания для нескольких вводов	88
8.6. Семантический поиск.....	89
8.7. Косинусное сходство	97
Глава 9. Более сложные примеры встраивания	100
9.1. Предсказание вашего любимого сорта кофе.....	100
9.2. Выполняем «нечеткий» поиск.....	108
9.3. Прогнозирование категории новостей с помощью встраивания	109
9.4. Оценка точности классификатора	113
Глава 10. Тонкая настройка и передовые методы работы	118
10.1. Обучение на ограниченных примерах	118
10.2. Улучшенное обучение с ограниченными примерами.....	119
10.3. Тонкая настройка на практике.....	119
10.4. Наборы данных, запросы и ответы: особые приемы	123
Глава 11. Продвинутая тонкая настройка: классификация лекарств	130
11.1. Набор данных, используемый в примере.....	130
11.2. Подготовка данных и запуск тонкой настройки.....	130
11.3. Тестирование настроенной модели.....	134

Глава 12. Продвинутая тонкая настройка: создание ассистирующего чат-бота	137
12.1. Интерактивная классификация	137
12.2. Как это будет работать?	137
12.3. Создание диалогового веб-приложения	144
Глава 13. Интеллектуальное распознавание речи с помощью Whisper	151
13.1. Что такое Whisper?	151
13.2. С чего начать?	152
13.3. Транскрипция и перевод	154
Глава 14. Контекст и память: как сделать искусственный интеллект более реалистичным	156
14.1. В чем проблема?	156
14.2. Отсутствие контекста = хаос случайности	156
14.3. История = Контекст	157
14.4. Недостатки прямого переноса истории	158
14.5. Память «последний вошел – первый вышел» (LIFO)	159
14.6. Проблемы с памятью LIFO	161
14.7. Выборочный контекст	161
Глава 15. Создание собственного помощника Alexa на основе ИИ	167
15.1. Введение	167
15.2. Запись звука	168
15.3. Расшифровка аудио	169
15.4. Ответ на запрос пользователя	170
15.5. Функция <i>main</i>	171
15.6. Собираем все вместе	172
15.7. Генерация более качественных ответов	174
Глава 16. Классификация изображений с помощью OpenAI CLIP	176
16.1. Что такое CLIP?	176
16.2. Как использовать CLIP	177
16.3. Stable Diffusion наоборот: изображение в текст	181
Глава 17. Генерация изображений с помощью DALL·E	183
17.1. Введение	183
17.2. Базовый пример генерации изображения по запросу	184
17.3. Создание нескольких изображений	185
17.4. Получение изображений разного размера	186
17.5. Улучшенные запросы на создание изображений	187
17.5.1. Подражание художникам	187
17.5.2. Имитация художественных стилей	189
17.5.3. Атмосфера, чувства, эмоции	191
17.5.4. Цвета	194
17.5.5. Разрешение	195
17.5.6. Углы и положения	196

17.5.7. Типы объективов	197
17.5.8. Осветительные приборы	198
17.5.9. Типы пленок и фильтры	199
17.6. Создание генератора случайных изображений	200
Глава 18. Редактирование изображений с помощью DALL·E.....	206
18.1. Пример редактирования изображения	206
Глава 19. Черпаем вдохновение из других изображений	211
19.1. Как создать вариацию имеющегося изображения	211
19. 2. Примеры использования вариативных изображений	214
Глава 20. Что дальше?.....	216
Предметный указатель	217

Предисловие

Когда люди спрашивают меня, чем я занимаюсь, мне всегда сложно дать им простой ответ. Моя карьера продвигалась извилистыми путями, и за эти годы я примерил много разных профессий. Я человек, который страстно любит учиться и пробовать что-то новое, оттого мне и довелось поработать в разных областях.

Я занимался разработкой программного обеспечения, рекламой, маркетингом, сетями и телекоммуникациями, системным администрированием, преподаванием, написанием технических текстов, ремонтом компьютеров и многими другими делами. Мной всегда двигало желание узнать больше и расширить свой кругозор. По мере того как я изучал новые технологии, знакомился с новыми людьми и исследовал новые концепции, мой разум становился более открытым, а кругозор расширялся. Я начал видеть связи и возможности, которых раньше не замечал.

Чем больше я узнавал сам, тем больше мне хотелось учить других, иногда даже бесплатно. Мне нравится это особое чувство, когда видишь, как в чьих-то глазах зажигается огонек понимания. Я всегда был учителем в душе, и мне всегда нравилось делиться своими знаниями с другими.

Именно эти соображения побудили меня написать руководство по работе с большими моделями Open AI.

Во время работы над этой книгой я постоянно думал о людях, которые будут ее читать. Я стремился создать доступное и простое руководство по NLP, GPT и смежным темам для людей, знающих Python, но владеющих ограниченными знаниями в этих областях. Моя цель состояла в том, чтобы предоставить практическую информацию, которую читатели смогут использовать для создания своих собственных интеллектуальных систем, не тратя многие часы на изучение теории, лежащей в основе этих концепций.

В этом практическом руководстве я делюсь своими знаниями и опытом работы с моделями OpenAI, в частности GPT-3 (но также и другими моделями), и тем, как программисты Python могут использовать их для создания собственных интеллектуальных приложений. Книга выстроена как пошаговое руководство, которое охватывает основные понятия и методы использования GPT-3 и дает читателю прочные и разнообразные практические навыки.

Мой почтовый ящик всегда полон, и я получаю много писем. Но наибольшее удовольствие мне доставляют письма с вопросами и пожеланиями от людей, которые прочитали мои онлайн-руководства и курсы и нашли их полезными. Пожалуйста, в любое время обращайтесь ко мне по адресу aymen@faun.dev. Мне важно узнать ваше мнение.

Надеюсь, вы получите от чтения этой книги такое же удовольствие, какое я получил от работы над ней.

Об авторе

Аймен Эль Амри – писатель, предприниматель, преподаватель и инженер-программист, который преуспел в различных профессиях в области информационных технологий, включая DevOps и Cloud Native, облачную архитектуру, Python, NLP, науку о данных и многое другое.

Аймен лично обучил сотни программистов и написал множество книг и курсов, которые прочитали тысячи других инженеров и разработчиков.

У Аймена Эль Амри практичный подход к обучению, который всегда находит отклик у его аудитории. Он разделяет сложные понятия на составные части, рассказывает о них понятным языком и иллюстрирует реальными примерами.

Он основал несколько проектов: FAUN¹, eralabs² и Marketto³. Вы можете найти Аймена в Twitter⁴ и LinkedIn⁵.

¹ <https://faun.dev>.

² <https://eralabs.io>.

³ <https://marketto.dev>.

⁴ <https://twitter.com/@eon01>.

⁵ <https://www.linkedin.com/in/elamriaymen/>.

Об этой книге

Знания, которые вы получите из этой книги, относятся к текущей версии (GPT-3) и, вероятно, также пригодятся для GPT-4, если нам суждено дожидаться ее выпуска¹.

OpenAI предоставляет API (интерфейс прикладного программирования) для доступа к моделям ИИ. Назначение API – абстрагировать базовые модели путем создания универсального интерфейса для всех версий, позволяющего пользователям использовать GPT независимо от его версии.

Цель этой книги – предоставить пошаговое руководство по использованию GPT-3 в ваших проектах с помощью API OpenAI. В руководстве рассмотрены и другие модели, такие как CLIP, DALL-E и Whispers.

Независимо от того, создаете ли вы чат-бот, ИИ-ассистента или веб-приложение, предоставляющее данные, сгенерированные ИИ, это руководство поможет реализовать ваши идеи.

Если вы владеете основами языка программирования Python и готовы изучить еще несколько инструментов, таких как объекты Dataframe библиотеки Pandas и некоторые методы NLP, значит, у вас есть под рукой все необходимое для создания интеллектуальных систем с использованием инструментов OpenAI.

Не беспокойтесь, вам не нужно быть специалистом по данным, инженером по машинному обучению или экспертом по искусственному интеллекту, чтобы понять концепции, методы и учебные примеры, представленные в этой книге. Все объяснения предельно просты для понимания, в них используется базовый код Python, примеры и практические упражнения.

Эта книга сфокусирована на практической части обучения и призвана помочь читателю создавать реальные приложения. Вы найдете здесь множество примеров кода, которые помогут усвоить базовые понятия и применить их в реальных сценариях для решения прикладных задач.

К концу обучения вы создадите такие приложения, как:

- медицинский чат-бот с пользовательской точной настройкой;
- интеллектуальная система подсказок любимого сорта кофе;
- интеллектуальная диалоговая система с памятью и контекстом;
- голосовой помощник с ИИ наподобие Alexa, но более умный;
- чат-бот, помогающий освоить команды Linux;
- семантическая поисковая система;
- система предсказания категории новостей;

¹ Во время работы над переводом этой книги продолжался прием заявок на раннее тестирование GPT-4, но автор и переводчик продолжали ожидать своей очереди, а сроки запуска модели в широкий доступ не были объявлены. – *Прим. перев.*

- интеллектуальная система распознавания изображений (изображение в текст);
- генератор изображений (текст в изображение);
- и многое другое!

Прочитав эту книгу и следуя примерам, вы освоите такие навыки, как:

- выбор подходящей модели из числа доступных, а также правильное использование каждой из них;
- генерация реалистичного текста для различных целей, таких как ответы на вопросы, создание контента и другие творческие применения;
- управление уровнем креативности моделей GPT и применение современных методов создания высококачественного текста;
- преобразование и редактирование текста для выполнения перевода, форматирования и других полезных задач;
- оптимизация производительности моделей GPT с помощью различных параметров и опций, таких как `suffix`, `max_tokens`, `temperature`, `top_p`, `n`, `stream`, `logprobs`, `echo`, `stop`, `presence_penalty`, `frequency_penalty`, `best_of` и др.;
- снижение стоимости использования API за счет сокращения и лемматизации текстов;
- применение заполнения контекста, создание смысловых цепочек и использование передовых технологий;
- использование встраивания текста аналогично тому, как это делают компании Tesla и Notion;
- применение семантического поиска и других передовых инструментов и концепций;
- создание алгоритмов прогнозирования и обучения без примеров и оценка их точности;
- использование обучения с несколькими примерами и улучшение его точности;
- использование тонкой настройки моделей, в том числе собственных;
- использование передового опыта для создания собственных моделей;
- применение различных методов обучения и классификации с использованием GPT;
- создание усовершенствованных моделей с помощью тонкой настройки;
- использование OpenAI Whisper и других инструментов для создания интеллектуальных голосовых помощников;
- классификация изображений с помощью OpenAI CLIP;
- создание и редактирование изображений с помощью OpenAI DALL-E;
- использование готовых изображений для создания собственных версий;

- обратное преобразование изображений из Stable Diffusion (изображение в текст).

Оставайтесь на связи

Если вы хотите быть в курсе последних тенденций в экосистемах Python и ИИ, присоединяйтесь к нашему сообществу разработчиков по адресу www.faun.dev/join. Мы отправляем еженедельные информационные бюллетени с важными и полезными учебными пособиями, новостями и идеями от экспертов в сообществе разработчиков программного обеспечения.

Глава 1

ChatGPT, GPT, GPT-3, DALL·E, Codex... Что это?

В декабре 2015 г. несколько блестящих новаторов объединилось для достижения общей цели: продвигать и развивать дружественный ИИ таким образом, чтобы он приносил пользу человечеству в целом.

Сэм Альтман, Илон Маск, Грег Брокман, Рид Хоффман, Джессика Ливингстон, Питер Тиль, Amazon Web Services (AWS), Infosys и UC Research объявили о создании компании OpenAI и пообещали выделить более 1 млрд долл. США на это предприятие. Новая организация сразу заявила, что будет свободно сотрудничать с другими компаниями и исследователями, делая свои патенты и исследования общедоступными.

Штаб-квартира OpenAI находится в здании Pioneer Building в районе Миссии в Сан-Франциско. В апреле 2016 г. OpenAI выпустила общедоступную бета-версию OpenAI Gym – своей платформы для исследований в области обучения с подкреплением. В декабре 2016 г. OpenAI выпустила Universe¹ – программную платформу широкого применения для измерения и обучения общего интеллекта ИИ в играх, веб-сайтах и других приложениях.

В 2018 г. Илон Маск покинул свое место в совете директоров, сославшись на потенциальный будущий конфликт интересов с разработкой искусственного интеллекта Tesla AI для беспилотных автомобилей, но остался инвестором. В 2019 г. OpenAI перешла от некоммерческой деятельности к бизнес-модели с ограниченной прибылью, при этом для любых инвестиций был установлен предел доходности 1 к 100. Компания распределила акции среди своих сотрудников и стала партнером Microsoft, которая инвестировала в проект 1 млрд долл. США. Затем OpenAI объявила о своем намерении лицензировать свои технологии на коммерческой основе.

В 2020 г. OpenAI анонсировала GPT-3 – языковую модель, обученную на триллионах слов из интернета, – и объявила, что API этой модели станет основой первого коммерческого продукта компании. Модель GPT-3 предназначена для ответов на вопросы на естественном языке, но она также

¹ <https://openai.com/blog/universe/>.

может выполнять перевод между языками и связно генерировать импровизированный текст. В 2021 г. OpenAI представила модель глубокого обучения DALL-E, способную генерировать цифровые изображения из описаний на естественном языке.

Перенесемся в декабрь 2022 г. После запуска бесплатной тестовой версии ChatGPT вокруг OpenAI разгорелась нешуточная шумиха в СМИ. По данным OpenAI, за первые пять дней на предварительное тестирование зарегистрировалось более миллиона человек. Согласно анонимным источникам, на которые ссылалось агентство Reuters в декабре 2022 г., OpenAI прогнозирует выручку в размере 200 млн долл. США в 2023 г. и 1 млрд долл. США в 2024 г. По состоянию на январь 2023 г. велись переговоры о следующем раунде финансирования, перед началом которого компания была оценена в 29 млрд долл.

Такова краткая история OpenAI, исследовательской лаборатории искусственного интеллекта, состоящей из коммерческой корпорации OpenAI LP и ее материнской компании, некоммерческой OpenAI Inc.

Большинство людей не знало о существовании OpenAI до того, как компания запустила свою невероятно популярную модель ChatGPT.

Основная цель ChatGPT заключалась в том, чтобы имитировать человеческое поведение и вести естественные диалоги с людьми. Кроме того, чат-бот может учиться на разговорах с разными пользователями. Этот ИИ не только общается с людьми на естественном языке, он также способен писать учебные пособия и код, сочинять музыку и выполнять другие задачи. Варианты использования ChatGPT весьма разнообразны и почти бесконечны; пользователи доказали это своими примерами. Одни пользователи занимались творчеством (например, сочинением рэпа), другие – вредоносной деятельностью (например, созданием вредоносного кода или команд), а третьи – бизнесом (например, контент-маркетингом, продажами по электронной почте, «холодными» рассылками электронных писем и повышением эффективности бизнеса).

Аббревиатура ChatGPT расшифровывается как Generative Pre-trained Transformer (генеративный предварительно обученный трансформер). Эта модель построена на основе семейства больших языковых моделей OpenAI GPT-3. Точную настройку чат-бота выполняли как с помощью методов обучения с учителем, так и с помощью обучения с подкреплением.

ChatGPT – это просто проект, использующий GPT-3 и добавляющий к базовой модели веб-интерфейс, память и другие удобные функции. Прочитав это руководство, вы сможете создать свой собственный чат-бот, возможно, лучше, чем ChatGPT, поскольку вы сможете настроить его под свои конкретные нужды.

К другим известным проектам, использующим GPT-3, среди прочих относятся:

- GitHub Copilot (с использованием модели OpenAI Codex, потомка GPT-3, настроенной для генерации кода);

- Copy.ai¹ и Jasper.ai² (генерация контента для маркетинговых целей);
- университет Дрекселя (выявление ранних признаков болезни Альцгеймера);
- Algolia (улучшение возможностей поисковой системы).

Вы наверняка уже слышали о GPT-4. Она является преемником GPT-3 и представляет собой нейронную сеть, также созданную OpenAI, но к ней пока не открыт широкий доступ.

Возможно, вам также довелось видеть диаграмму сравнения двух версий GPT, показывающую количество параметров в GPT-3 (175 млрд) и GPT-4 (100 трлн), как показано на рис. 1.1.

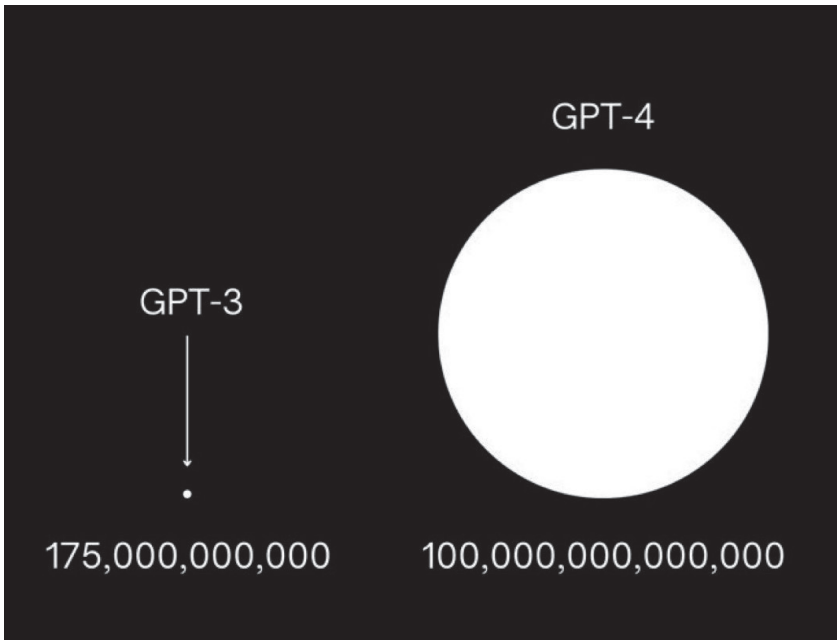


Рис. 1.1. Наглядное сравнение количества параметров моделей GPT-3 и GPT-4

Когда Сэма Альтмана спросили об этой вирусной иллюстрации, он назвал ее полной чушью:

«Невероятные слухи о GPT-4 – полная чушь. Я не знаю, кто все это придумывает. Некоторые люди упорно хотят разочароваться в собственных ожиданиях, и они обязательно разочаруются. Мы не изобрели подлинный искусственный интеллект, хотя от нас ждут именно этого».

В том же интервью Альтман отказался подтвердить, будет ли модель вообще выпущена в 2023 г.

¹ <http://copy.ai>.

² <http://jasper.ai>.

Глава 2

Как работает GPT?

GPT – это *генеративная* текстовая модель. Такая модель способна создавать новый текст, предсказывая его продолжение на основе полученных входных данных.

GPT-3 – это просто обновленная и увеличенная модель, она заметно крупнее и производительнее, чем любая другая предыдущая модель GPT, включая GPT-1 и GPT-2.

Модель третьего поколения обучали на большом массиве текстов, таких как книги, статьи и общедоступные веб-сайты, такие как Reddit и другие форумы. Она использует эти обучающие данные для изучения закономерностей и взаимосвязей между словами и фразами.

Ключевое отличие GPT-3 заключается в ее впечатляющем размере – с ошеломляющими 175 млрд параметров, – что делает ее одной из самых массивных и мощных языковых моделей, когда-либо созданных человеком. Сочетание невиданного ранее количества параметров и обширного набора данных позволяет модели генерировать правдоподобные тексты и выполнять различные задачи обработки естественного языка с впечатляющим качеством.

GPT – это тип нейронной сети, относящейся к архитектуре Transformer, которую специально разработали для задач обработки естественного языка. В повседневной жизни модели с такой архитектурой часто называют просто трансформерами. Архитектура Transformer основана на последовательности блоков самовнимания, которые позволяют модели параллельно обрабатывать входной текст и взвешивать важность каждого слова или токена в зависимости от контекста.

Самовнимание (self-attention) – это механизм, используемый в моделях глубокого обучения для обработки естественного языка (NLP), который позволяет модели взвешивать важность различных частей предложения или нескольких предложений при прогнозировании. Являясь частью архитектуры Transformer, он позволяет нейронной сети достигать высокого качества работы, когда речь идет о задачах NLP.

Вот пример использования трансформеров, разработанных компанией Hugging Face для логического вывода GPT-2:

```
1 from transformers import pipeline
2 generator = pipeline('text-generation', model = 'gpt2')
```

```

3 generator("Привет, я языковая модель", max_length = 30, num_return_
  sequences=3)
4 ## [{'generated_text': "Привет, я языковая модель. Когда я писал это, когда
  я встречался с женой или возвращался домой, она говорила мне, что мой"},
5 ## {'generated_text': "Привет, я языковая модель. Я пишу и поддерживаю
  программы на Python. Я люблю программировать, и это включает в себя"}
6 ...

```

По умолчанию модель не имеет памяти, это означает, что каждый ввод обрабатывается независимо, без переноса какой-либо информации из предыдущих вводов. Когда GPT генерирует текст, у нее *нет априорных представлений* о том, что должно быть дальше на основе предыдущих входных данных. Вместо этого она генерирует каждое слово на основе *вероятности* того, что оно будет следующим словом с учетом предыдущего ввода. Поэтому полученный текст иногда бывает неожиданным и не совсем осмысленным.

Вот еще один пример кода, использующего модель GPT для генерации текста на основе пользовательского ввода.

```

# Импорт необходимых библиотек
from transformers import GPT2Tokenizer, GPT2LMHeadModel

# Загрузка предварительно обученного токенизатора GPT-2 и модели
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
model = GPT2LMHeadModel.from_pretrained('gpt2')

# Настройка модели в режим оценки
model.eval()

# Определение начального запроса, за которым следует завершение
prompt = input("You: ")

# Токенизация запроса и генерация текста
input_ids = tokenizer.encode(prompt, return_tensors='pt')
output = model.generate(input_ids, max_length=50, do_sample=True)

# Декодирование сгенерированного текста и вывод в консоль
generated_text = tokenizer.decode(output[0], skip_special_tokens=True)
print("AI: " + generated_text)

```

GPT-3 целенаправленно разрабатывали как языковую модель общего назначения, поэтому ее можно использовать для различных задач обработки естественного языка, таких как языковой перевод, резюмирование текста и ответы на вопросы. Разработчики из OpenAI предварительно обучили GPT-3, но вы можете создать собственную модель с помощью процедуры тонкой настройки на собственных наборах данных. Это позволяет применять модель для более конкретных творческих задач в дополнение к задачам по умолчанию, таким как сочинение текста, стихов и рассказов. Настроенные модели также можно использовать для создания чат-ботов,

которые являются экспертами в определенной области, диалоговых интерфейсов и многого другого!

В этой книге будет подробно рассказано об OpenAI GPT-3, в том числе о том, как эффективно использовать API и инструменты, предоставляемые компанией и сообществом AI/ML, для создания прототипов мощных и креативных инструментов и систем. Это не только улучшит ваши навыки использования API GPT-3, но и расширит ваше понимание концепций и методов, используемых в обработке естественного языка и других смежных областях.

Благодаря огромному количеству параметров и впечатляющим результатам работы GPT-3 считается значительным достижением в обработке естественного языка. Однако некоторые эксперты выражают обеспокоенность по поводу того, что модель может создавать предвзятый или вредный контент. Как и в случае с любой технологией, здесь очень важно вовремя остановиться и подумать об этических последствиях ее использования. Но в этой книге мы не будем касаться этических вопросов, а сосредоточимся только на практических аспектах.

Приятного чтения!

Глава 3

Подготовка среды разработки

3.1. Установка Python, pip и виртуальной среды для разработки

Очевидно, что вам понадобится Python. В этой книге мы будем использовать Python 3.9.5.

Мы также будем использовать установщик пакетов pip. Вы можете использовать pip для установки пакетов из репозитория пакетов Python и других репозиториях.

Вам не нужно устанавливать Python 3.9.5 непосредственно в вашей системе, так как мы создадим виртуальную среду разработки. Какая бы у вас ни была установлена версия Python на рабочем компьютере, среда разработки будет изолирована от вашей системы и будет использовать Python 3.9.5.

Если Python не установлен, перейдите на страницу www.python.org/downloads/, загрузите и установите одну из версий Python 3.x. В зависимости от вашей операционной системы вам придется следовать разным инструкциям.

Для управления нашей средой разработки мы будем использовать *virtualenvwrapper*¹. Инструкции по установке вы найдете в официальной документации².

Самый простой способ получить его – использовать *pip*:

```
pip install virtualenvwrapper
```

Примечание: если вы привыкли к *virtualenv*, Poetry или любому другому диспетчеру пакетов, вы можете продолжать его использовать. В этом случае нет необходимости устанавливать *virtualenvwrapper*.

Если *pip* не установлен, проще всего установить его с помощью скрипта, представленного в официальной документации.

Загрузите скрипт с <https://bootstrap.pypa.io/get-pip.py>.

Откройте окно терминала / командной строки, перейдите в папку, содержащую файл `get-pip.py`, и выполните команду

¹ <https://github.com/python-virtualenvwrapper/virtualenvwrapper>.

² <https://virtualenvwrapper.readthedocs.io/en/latest/install.html>.

```
python get-pip.py
```

если вы пользователь MacOS или Linux, или команду

```
py get-pip.py
```

если вы пользователь Windows.

В итоге в вашей системе должны быть установлены следующие пакеты:

- Python,
- Pip,
- *virtualenvwrapper* (либо *virtualenv* или любой другой менеджер пакетов, который вы предпочитаете).

Я настоятельно рекомендую пользователям Windows создать виртуальную машину с операционной системой Linux, так как большинство примеров, представленных в этой книге, выполнялись и тестировались в системе Linux Mint.

Далее создадим виртуальную среду:

```
mkvirtualenv -p python3.9 chatgptforpythondevelopers
```

После создания виртуальной среды активируйте ее:

```
workon chatgptforpythondevelopers
```

3.2. Получение ключа API OpenAI

Следующим шагом является создание ключей API, которые позволят вам получить доступ к официальному API, предоставляемому OpenAI. Перейдите на <https://openai.com/api/> и создайте учетную запись.

Примечание к переводу: сайт и сервисы OpenAI недоступны для пользователей с российским IP-адресом. Для создания учетной записи вам понадобится доступ через VPN с европейским IP и временный виртуальный телефонный номер в любой европейской стране, на который вы получите СМС с кодом подтверждения регистрации. В дальнейшем вам этот телефонный номер не понадобится, но доступ к API возможен только через VPN. Мы не будем здесь детально описывать использование виртуальных телефонных номеров, но отметим, что российским пользователям доступно множество сервисов, предоставляющих виртуальные телефонные номера для получения СМС в различных странах с оплатой российскими банковскими картами, включая «Мир», по очень доступной цене. Как показал опыт, для регистрации и входа на сайт OpenAI можно использовать российский аккаунт Google. Процедура регистрации на сайте OpenAI детально описана в различных блогах российских авторов.

Следуйте инструкциям по созданию учетной записи, а затем создайте ключи API по адресу: <https://beta.openai.com/account/api-keys>.

Ключ API должен принадлежать организации, вам будет предложено создать организацию. В этой книге мы будем называть ее LearningGPT.

Сохраните сгенерированный секретный ключ в надежном и доступном месте. Вы не сможете увидеть его снова через свою учетную запись OpenAI.

Примечание к переводу: по состоянию на май 2023 г. новым пользователям на счет зачислялся грант в размере 18 долл., действующий в течение 40 дней, после чего неиспользованный остаток гранта «сгорает». Имейте это в виду при проведении экспериментов за счет гранта. Разумеется, условия предоставления гранта могут в любой момент измениться без предупреждения со стороны OpenAI. К сожалению, на данный момент для российских пользователей не существует простого способа оплаты сервисов OpenAI после исчерпания гранта. Необходимо иметь банковскую карту Visa или Mastercard, выпущенную западным банком, поэтому проблему оплаты каждый пользователь должен решать самостоятельно. Впрочем, начального гранта вполне достаточно для экспериментов и знакомства с API OpenAI.

3.3. Установка официальных средств интеграции Python

Вы можете взаимодействовать с API, используя HTTP-запросы с любого языка через официальные средства интеграции (binding, привязка) Python, или через официальную библиотеку Node.js, или через библиотеку, поддерживаемую сообществом.

В этой книге мы будем использовать официальную библиотеку, предоставленную OpenAI. Другой альтернативой является использование Chronology¹, неофициальной библиотеки, предоставленной OthersideAI. Однако похоже, что эта библиотека больше не обновляется.

Чтобы установить официальные средства интеграции Python, выполните следующую команду:

```
pip install openai
```

Убедитесь, что вы устанавливаете библиотеку в виртуальной среде, которую мы создали ранее.

3.4. Тестирование ключей API

Чтобы убедиться, что все работает правильно, выполним вызов через `curl`. Перед этим нужно сохранить ключ API и идентификатор организации в файле с именем `.env` следующего содержания:

```
cat << EOF > .env
API_KEY=xxx
ORG_ID=xxx
EOF
```

¹ <https://github.com/OthersideAI/chronology>.

Перед выполнением команды `curl` обязательно обновите переменные `API_KEY` и `ORG_ID` на их соответствующие значения из вашей учетной записи. Теперь вы можете выполнить следующую команду (одна строка):

```
source .env
curl https://api.openai.com/v1/models
  -H 'Authorization: Bearer '$API_KEY''
  -H 'OpenAI-Organization: '$ORG_ID''
```

Вы можете обойтись без ссылки на переменные в файле и использовать их непосредственно в команде `curl`:

```
curl https://api.openai.com/v1/models
  -H 'Authorization: Bearer xxxx'
  -H 'OpenAI-Organization: xxxx'
```

Если в вашей учетной записи OpenAI есть только одна организация, можно выполнить ту же команду без указания идентификатора организации.

```
curl https://api.openai.com/v1/models -H 'Authorization: Bearer xxxx'
```

Команда `curl` должна вернуть вам список моделей, предоставляемых API, таких как `davinci`, `ada` и многие другие.

Чтобы протестировать API с помощью кода Python, вы можете выполнить следующий код:

```
import os
import openai

# Чтение переменных из файла .env, а именно API_KEY и ORG_ID
with open(".env") as env:
    for line in env:
        key, value = line.strip().split("=")
        os.environ[key] = value

# Инициализация ключа API и идентификатора организации
openai.api_key = os.environ.get("API_KEY")
openai.organization = os.environ.get("ORG_ID")

# Вызов API и получение списка моделей
models = openai.Model.list()
print(models)
```

Мы будем использовать примерно такой же подход в будущем, поэтому давайте создадим повторно используемую функцию для инициализации API. Она может выглядеть, например, так:

```
import os
import openai

def init_api():
```

```
with open(".env") as env:
    for line in env:
        key, value = line.strip().split("=")
        os.environ[key] = value

openai.api_key = os.environ.get("API_KEY")
openai.organization = os.environ.get("ORG_ID")

init_api()

models = openai.Model.list()
print(models)
```