Оглавление

Предисловие от издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте *www.dmkpress.com*, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу *dmkpress@gmail.com*; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу *http://dmkpress.com/authors/publish_book/* или напишите в издательство по адресу *dmkpress@gmail.com*.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу *dmkpress@gmail.com*, и мы исправим это в следующих тиражах.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу *dmkpress@gmail.com*.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

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 многих часов 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: FAUN[1], eralabs[2] и Marketto[3]. Вы можете найти Аймена в Twitter[4] и Linkedin[5].

---

1 *https://faun.dev*.

2 *https://eralabs.io*.

3 *https://marketto.dev*.

4 *https://twitter.com/@eon01*.

5 *https://www.linkedin.com/in/elamriaymen/*.

История OpenAI и ChatGPT

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

Сэм Альтман[1], Илон Маск[2], Грег Брокман[3], Рид Хоффман[4], Джессика Ливингстон[5], Питер Тиль[6], Amazon Web Services (AWS), Infosys и YC Research объявили о создании компании OpenAI и пообещали выделить более 1 млрд долларов 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 Universe[7] – 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 долларов 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 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.

---

1 *https://en.wikipedia.org/wiki/Sam_Altman*.

2 *https://en.wikipedia.org/wiki/Elon_Musk*.

3 *https://en.wikipedia.org/wiki/Greg_Brockman*.

4 *https://en.wikipedia.org/wiki/Reid_Hoffman*.

5 *https://en.wikipedia.org/wiki/Jessica_Livingston*.

6 *https://en.wikipedia.org/wiki/Peter_Thiel*.

7 *https://openai.com/blog/universe/*.

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 долларов США в 2023 г. и 1 млрд долларов 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 долларов.

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

Большинство людей не знали 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).

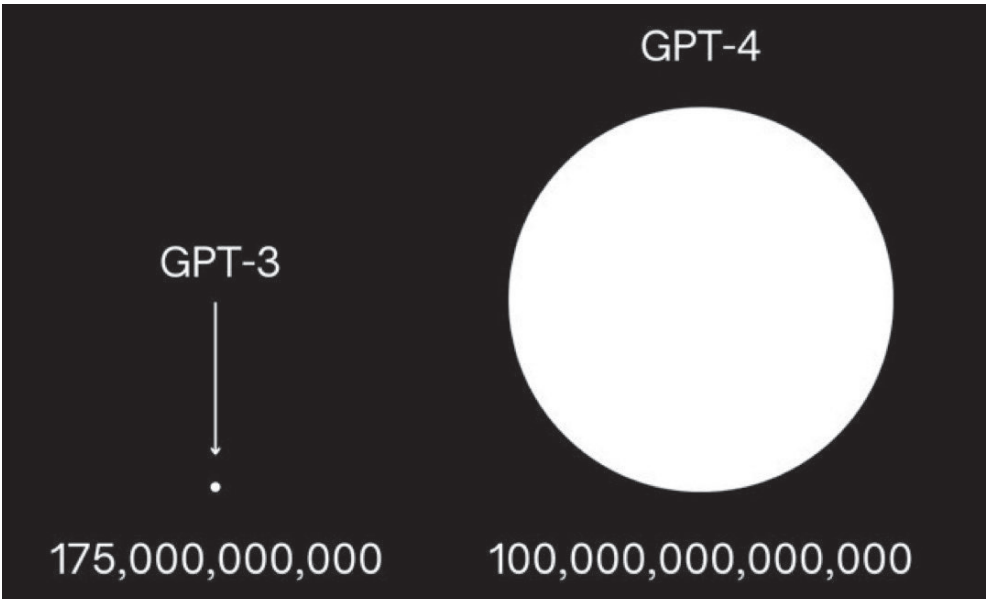
Аббревиатура ChatGPT расшифровывается как Generative Pretrained Transformer (генеративный предварительно обученный трансформер). Эта модель построена на основе семейства *больших языковых моделей* (large language model, LLM). Чат-бот обучен с использованием двух технологий – *обучение с учителем* (supervised learning) и *обучение с подкреплением* (reinforcement learning).

Модель GPT-3 служила основой для ChatGPT до релиза GPT-3.5 15 марта 2022 г., а затем GPT-4 14 марта 2023 г. ChatGPT – это проект, который основан на GPT-3.5 и GPT-4 с добавлением веб-интерфейса, памяти диалогов и других удобных функций. Прочитав это руководство, вы сможете создать свой собственный чат-бот, который потенциально будет лучше, чем универсальный ChatGPT, поскольку вы сможете настроить его в соответствии со своими конкретными потребностями.

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

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».



Наглядное сравнение количества параметров моделей GPT-3 и GPT-4

Интернет полон слухов, спекуляций и некачественного контента. Это верно и в случае искусственного интеллекта. Возможно, вам попадались различные учебные пособия и книги, обещающие научить вас использовать модели OpenAI GPT, но на самом деле предлагающие стандартные наборы запросов к модели (промптов), скопированные в интернете. Эта книга не такая. Это не примитивный набор подсказок; речь идет о понимании принципов и методов, лежащих в основе этих моделей, о том, как использовать их при создании ваших приложений и как их интегрировать с существующими системами и другими инструментами искусственного интеллекта и машинного обучения.

В настоящее время многие проекты используют модели OpenAI в качестве основы для своих продуктов. К наиболее заметным в наши дни относятся:

- GitHub Copilot[1] (с использованием модели OpenAI Codex, потомка GPT-3, настроенной для генерации кода);
- Copy.ai[2] и Jasper.ai[3] (генерация контента для маркетинговых целей);
- университет Дрекселя[4] (выявление ранних признаков болезни Альцгеймера);
- Algolia[5] (улучшение возможностей поисковой системы);
- ваш будущий стартап? Почему бы и нет?

---

1 *https://github.com/features/copilot*.

2 *http://copy.ai*.

3 *http://jasper.ai*.

4 *https://drexel.edu/*.

5 *https://www.algolia.com/*.

Прочитав эту книгу, вы научитесь эффективно использовать модели OpenAI в своих проектах и, возможно, станете следующим, кто присоединится к этому списку.

Об этой книге

Знания, которые вы получите из этой книги, относятся к текущему семейству моделей (GPT-3, GPT-3.5, GPT-4 и т. д.) и, вероятно, также пригодятся для GPT-5, если нам суждено дождаться ее выпуска.

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 применять GPT независимо от его версии.

Цель этой книги – предоставить пошаговое руководство по использованию GPT-3.5 и GPT-4 в ваших проектах с помощью API OpenAI. В руководстве рассмотрены и другие модели, такие как Whispers и Text-to-Speech.

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 следующие приложения:

- точно обученные чат-боты для узкой предметной области;
- интеллектуальная диалоговая система с памятью и контекстом;
- современная семантическая поисковая система, использующая RAG и другие методы;
- интеллектуальная система рекомендации сортов кофе, учитывающая ваши вкусы;
- чат-бот, помогающий освоить команды Linux;
- точно настроенная система прогнозирования категорий новостей;
- автономная диалоговая система AI-AI для имитации человеческих разговоров или решения проблем;
- консультант-психотерапевт на основе ИИ, обученный на большом наборе данных бесед о психическом здоровье,
- и многое другое!

Прочитав эту книгу и следуя примерам, вы освоите следующие навыки:

- 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 других творческих применений;
- 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 стемматизации, лемматизации и сокращения текстов;
- применение заполнения контекста, создание смысловых цепочек и прикладных промптов;
- реализация чат-бота с памятью и сохранением контекста;
- создание алгоритмов прогнозирования и обучения без примеров, оценка их точности;
- использование обучения с несколькими примерами и улучшение его точности;
- тонкая настройка моделей, в том числе собственных;
- понимание и использование передового опыта тонкой настройки моделей;
- применение различных методов обучения и классификации с использованием GPT;
- использование встраивания текста аналогично тому, как это делают компании Tesla и Notion;
- понимание и применение семантического поиска, RAG и других передовых инструментов и концепций;
- интеграция баз данных векторных представлений (например, Weaviate) в вашу интеллектуальную систему.

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 рассылаем еженедельные информационные бюллетени с важными и полезными учебными пособиями, новостями и идеями от экспертов в сообществе разработчиков программного обеспечения.

Как работает GPT?

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

GPT-4 – это просто обновленная и увеличенная модель, она заметно крупнее и производительнее, чем любая другая предыдущая модель GPT, включая GPT-1/2/3/3.5.

Модель четвертого 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-4 заключается в ее впечатляющем размере – с ошеломляющими 1,76 трлн 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* (natural language processing, NLP), который позволяет модели взвешивать важность различных частей предложения или нескольких предложений при прогнозировании. Являясь частью архитектуры Transformer, он дает возможность нейронной сети достигать высокого качества работы, когда речь идет о задачах NLP. Концепция самовнимания основана на идее, что на каждое слово в предложении могут влиять другие слова в предложении и важность каждого слова можно определить на основе его контекста. Впервые механизм самовнимания был представлен в статье «Внимание – это все, что вам нужно»[1].

Представьте себе, что вы присутствуете на большом званом обеде, где все одновременно делятся историями. Вы пытаетесь следить за всеми разговорами вокруг вас, но, естественно, больше сосредоточены на историях, которые имеют к вам какое-то отношение или содержат важную информацию. Когда люди говорят, вы уделяете свое внимание рассказчику, чье повествование в данный момент наиболее интересно или актуально, иногда переключая

[1] *https://arxiv.org/abs/1706.03762*.

фокус на основе новой информации или связей, которые вы устанавливаете с другими рассказываемыми историями.

В этой аналогии вы похожи на модель трансформера в GPT. Званый ужин представляет собой входной текст, где история каждого гостя представляет собой слово или токен в последовательности. Ваша способность слушать несколько историй одновременно и решать, на какой из них сосредоточиться, аналогична механизму самовнимания. Этот механизм позволяет модели *одновременно* обрабатывать все части входных данных и определять релевантность или важность каждого слова в контексте всех остальных, подобно тому, как вы в нужный момент настраиваетесь на самую увлекательную историю.

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

```
from transformers import pipeline
generator = pipeline(
    'text-generation',
    model = 'gpt2'
)
generator(
    "Привет, я языковая модель",
    max_length = 30,
    num_return_sequences=3
)
```

Так выглядит пример вывода предыдущего кода:

```
[
    {
        'generated_text': "Привет, я языковая модель. Когда я писала это, ..."
    },
    {
        'generated_text': " Привет, я языковая модель. Я пишу и поддерживаю ..."
    }
]
```

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')
```

```
# Использование модели в режиме 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-4 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-4, 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 о моделях OpenAI (GPT, встраивание и многие другие понятия) и о том, как обращаться к ним из кода на Python, включая эффективное использование API и инструментов, предоставляемых компанией и сообществом AI/ML, для создания прототипов мощных и креативных инструментов и систем. Это не только улучшит ваши навыки применения API GPT, 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 технических аспектах.

Приятного чтения!

Подготовка среды разработки

Важные примечания

В этом руководстве для создания файлов и вставки контента непосредственно из командной строки или с помощью скриптов и программ часто используется так называемый *heredoc-синтаксис*.

> Heredoc, аббревиатура от «here document» (здесь документ), представляет собой функцию сценариев оболочки, которая упрощает создание многострочного ввода (строк с переносом).

Heredoc-синтаксис начинается с символов <<, за которыми следует маркер-разделитель по вашему выбору, продолжается содержимым, которое вы хотите включить, и завершается тем же маркером-разделителем на новой строке. Этот синтаксис особенно полезен для задач сценариев и программирования, требующих создания файлов или ввода нескольких строк для команды.

Чтобы использовать heredoc-синтаксис для создания файлов или добавления содержимого, используйте следующую структуру:

```
cat  <<  EOF  >  имя_файла
Это строка  #1
Это строка  #2
..
EOF
```

В этом примере `<<EOF` означает начало раздела heredoc, а `EOF` в отдельной строке указывает на его конец. Содержимое между этими маркерами перенаправляется в файл с именем `имя_файла`. Замените `имя_файла` на имя файла, которое вам нужно, и при необходимости измените содержимое.

Основная цель использования синтаксиса heredoc в этом руководстве – дать читателям возможность точно и легко создавать файлы с помощью одного действия копирования и вставки из электронной версии книги. Такой подход гарантирует, что контент автора будет точно воспроизведен в среде читателя, что снижает вероятность ручных ошибок и делает процесс обучения более эффективным.

Я предлагаю копировать примеры heredoc из электронной версии книги прямо в ваш терминал командной строки, чтобы создать файлы и контент, как это задумано в книге[1].

Пользователи Windows могут выполнять примеры heredoc, применяя подсистему Windows для Linux (Windows Subsystem for Linux, WSL). Кроме того, вы можете вручную создать файлы и вставить их содержимое с помощью текстового редактора. Тем не менее для выполнения примеров в этой книге я рекомендую использовать Unix-подобную среду (предпочтительно систему на базе Debian).

---

[1] Читатели русского перевода могут скачать архив с готовыми примерами кода на сайте издательства. В таком случае heredoc-синтаксис не нужен. – *Прим. перев.*

Если вы работатете с другой системой, вам может потребоваться настроить некоторые незначительные детали, такие как менеджер пакетов, процесс установки пакетов и метод создания файлов и каталогов. Это не должно стать проблемой, поскольку примеры просты и могут быть легко адаптированы к любой системе.

Еще одно важное замечание, о котором стоит упомянуть, заключается в том, что это руководство построено таким образом, что каждая глава основывается на предыдущей, а темы и концепции переплетаются между собой, чтобы сформировать всестороннее понимание предмета. Важно последовательно проходить главы, заботясь о том, чтобы полностью усвоить каждый раздел, прежде чем переходить к следующему. Терпение и внимание к деталям существенно помогут вашему обучению, позволяя вам в полной мере извлечь выгоду из пошагового развития.

Код, представленный в этом руководстве, предназначен не для пассивного просмотра, а для активного выполнения. Взаимодействие с кодом – его запуск, изменение и наблюдение за результатами – важно для глубокого понимания обсуждаемых концепций. Этот практический подход поможет вам усвоить материал и развить навыки, необходимые для применения знаний в ваших собственных проектах.

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 работать с установщиком пакетов pip. Вы можете использовать pip для установки пакетов из репозитория пакетов Python и других репозиториев.

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 документации[2].

Самый простой способ получить его – использовать pip:

```
pip install --upgrade pip
pip install virtualenv
pip install virtualenvwrapper
```

Пользователи Windows могут применять следующие команды для установки virtualenvwrapper-win[3]:

[1] *https://github.com/python-virtualenvwrapper/virtualenvwrapper*.

[2] *https://virtualenvwrapper.readthedocs.io/en/latest/install.html*.

[3] *https://github.com/davidmarble/virtualenvwrapper-win/*.

```
pip install --upgrade pip
pip install virtualenv
pip install virtualenvwrapper-win
```

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

Если pip не установлен, проще всего установить его с помощью скрипта `get-pip.py`[1]:

```
curl https://bootstrap.pypa.io/get-pip.py | python3
```

Пользователи Windows могут ввести следующие команды:

```
curl https://bootstrap.pypa.io/get-pip.py
py get-pip.py
```

В итоге в вашей системе должны быть установлены следующие пакеты:

- Python 3.9;
- pip;
- virtualenv;
- virtualenvwrapper (или любой другой менеджер пакетов, который вы предпочитаете).

Я настоятельно рекомендую пользователям Windows создать виртуальную машину с операционной системой Debian Linux, так как большинство примеров, представленных в этой книге, выполнялись и тестировались в системе Debian Linux. Хотя мнигие команды и примеры кода будут работать под Windows, в некоторых случаях могут потребоваться небольшие изменения.

Далее создадим виртуальную среду:

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

После создания виртуальной среды активируйте ее:

```
workon chatgptforpythondevelopers
```

Получение ключа API OpenAI

Следующим шагом является создание ключей API, которые позволят вам получить доступ к официальному API, предоставляемому OpenAI[2].

[1] *https://bootstrap.pypa.io/get-pip.py*.

[2] Сайт и сервисы OpenAI недоступны для пользователей с российским IP-адресом. Для создания учетной записи вам понадобится доступ в интернет с европейским или американским 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 с российских IP-адресов невозможен. 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.

Перейдите по адресу *https://platform.openai.com* и создайте учетную запись[1].

Ключ API должен принадлежать организации, поэтому 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.

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 применение Chronology[2], неофициальной библиотеки, предоставленной OthersideAI. Однако похоже, что эта библиотека больше не обновляется, поэтому не рекомендуется использовать ее для производственных целей.

Чтобы установить официальные средства интеграции Python, выполните следующую команду:

```
pip install openai==1.9.0
```

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

Тестирование ключей API

Прежде всего объявим ключ API и ID организации как переменные окружения (вместо xxx подставьте ваши данные):

```
export API_KEY=xxx
export ORG_ID=xxx
```

Теперь выполните следующую команду для тестирования доступа к API (вводить без переноса строки):

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

Если в вашей учетной записи OpenAI определена только одна организация, вы можете выполнить ту же команду без указания ID организации:

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

Команда должна вернуть список моделей, доступных через API, таких как davinci, ada и многих других.

---

1 *https://platform.openai.com/api-keys*.

2 *https://github.com/OthersideAI/chronology*.

Далее мы выполним тест с применением Python SDK.

Создайте файл с именем `.env` и сохраните в нем ключ API и ID организации:

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

Для тестирования работы в Python SDK выполните следующий код:

```
cat << EOF > test_api.py
import os
from openai import OpenAI
from pprint import pprint

# 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 ID организации
client = OpenAI(
  api_key=os.environ['API_KEY'],
  organization=os.environ['ORG_ID']
)

# Печать объекта client
pprint(vars(client))
EOF
```

Запустите приведенный выше код, выполнив команду

```
python test_api.py
```

Мы будем использовать аутентификацию с помощью ключа API и ID организации в каждом примере кода на протяжении всей книги. Поэтому нам будет полезно упорядочить код.

Создайте папку с именем `src` и внутри нее создайте файл с именем `api.py`:

```
cat << EOF > src/api.py
import os
from openai import OpenAI

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

# Создание объекта client на уровне модуля
client = OpenAI(
```