ГЛАВА 1

Основы создания AI-приложений с использованием базовых моделей

Если бы можно было одним словом описать развитие AI после 2020 года, это было бы слово «масштаб». Масштаб моделей AI, лежащих в основе таких приложений, как ChatGPT, Google Gemini и Midjourney, настолько велик, что они потребляют значительную часть вырабатываемого в мире электричества (https://oreil.ly/J0IyO), и существует риск того, что имеющиеся в открытом доступе в Интернете данные, используемые для их обучения, закончатся (https://arxiv.org/abs/2211.04325).

Масштабирование моделей AI приводит к двум серьезным последствиям. Во-первых, увеличивается мощность моделей AI, они становятся способны выполнять больше задач, что открывает возможности для различных приложений. Все больше людей и команд используют AI для повышения производительности, создания экономической ценности и улучшения качества жизни.

Во-вторых, для обучения больших языковых моделей требуются данные, вычислительные ресурсы и талантливые специалисты, а их могут позволить себе лишь немногие организации. Это привело к появлению концепции «*модель как услуга*»: модели, разработанные этими немногими организациями, предоставляются для использования другими как услуга. Теперь любой человек, желающий задействовать AI для создания приложений, может воспользоваться этими моделями, и ему не нужно вкладываться в создание собственной модели.

В общем, спрос на AI-приложения вырос, в то время как барьер выхода на рынок создания AI-приложений понизился. В результате *AI-инженерия* — процесс создания приложений на основе доступных моделей — превратилась в одну из самых быстрорастущих инженерных дисциплин.

Создание приложений на основе моделей машинного обучения (machine learning, ML) не новое направление. Задолго до того, как LLM обрели популярность, AI уже использовался для множества приложений, включая рекомендации продуктов, обнаружение случаев мошенничества и прогнозирование оттока клиентов. Несмотря на то что многие принципы внедрения AI-приложений остаются неизменными, современное поколение масштабных доступных моделей

открывает новые возможности и сталкивается с новыми вызовами, которые мы и рассмотрим в этой книге.

Данная глава начинается с обзора базовых моделей, ставших главным катализатором стремительного развития AI-инженерии. Затем я расскажу о нескольких успешных случаях применения искусственного интеллекта, каждый из которых демонстрирует, с чем он умеет или еще не умеет успешно справляться. Поскольку возможности AI постоянно расширяются, предсказать, на что он станет способен в будущем, становится все сложнее. Однако существующие шаблоны применения могут помочь раскрыть потенциал AI уже сегодня и дать подсказку о возможностях его использования в дальнейшем.

В конце главы я приведу обзор нового технологического стека AI, в том числе расскажу, что изменилось с появлением базовых моделей, что осталось неизменным и чем профессия специалиста по искусственному интеллекту сегодня отличается от работы специалиста по традиционному машинному обучению[1].

Восход AI-инженерии

Базовые модели развились на основе больших языковых моделей, которые изначально были простыми языковыми моделями. И хотя может показаться, что приложения вроде ChatGPT и GitHub Copilot появились неожиданно, однако они стали кульминацией десятилетий технологических достижений, начавшихся с появления первых языковых моделей в 1950-х годах. В этом разделе описываются ключевые открытия, сделавшие возможной эволюцию от языковых моделей к AI-инженерии.

От языковых моделей к большим языковым моделям

Языковые модели существуют уже давно, однако достичь нынешнего масштаба они смогли только благодаря *самообучению*. В этом разделе кратко объясняется, что такое языковые модели и самообучение. Если вы уже знакомы с этими терминами, можете пропустить раздел.

Языковые модели

Языковая модель кодирует статистические закономерности одного или нескольких языков. Интуитивно эта информация позволяет ей прогнозировать вероятность появления того или иного слова в определенном контексте. Например, в контексте *My favorite color is* __ («Мой любимый цвет — __») языковая модель,

[1] В этой книге под *традиционным* машинным обучением я подразумеваю все машинное обучение, существовавшее до появления базовых моделей.

основанная на английском языке, будет чаще предсказывать употребление слова *blue* («синий»), чем *car* («автомобиль»).

Статистическая природа языков была открыта несколько столетий назад. В рассказе 1905 года «Пляшущие человечки» Шерлок Холмс (https://ru.wikipedia.org/wiki/Пляшущие_человечки) применил простую статистику английского языка для расшифровки последовательностей таинственных фигур. Поскольку самой часто встречающейся буквой в английском языке является E, Холмс предположил, что ее обозначает наиболее часто встречающаяся фигура.

Позже Клод Шеннон использовал более сложную статистику для расшифровки сообщений врага во время Второй мировой войны. Его работа по моделированию английского языка была описана в знаковой статье 1951 года «Предсказание и энтропия печатного английского текста» (https://oreil.ly/G_HBp). Многие концепции, представленные в статье, в том числе «энтропия», до сих пор используются в языковом моделировании.

На ранних этапах языковая модель включала в себя только один язык. Однако сегодня языковая модель может охватывать несколько языков.

Основной единицей языковой модели является *токен*. В зависимости от модели токен может быть символом, словом или частью слова — скажем, суффиксом tion[1]. Например, модель GPT-4, лежащая в основе ChatGPT, разбивает фразу *I can't wait to build AI applications* на девять токенов (рис. 1.1). Обратите внимание на то, что в этом примере слово *can't* разделено на два токена — *can* и *'t*. Посмотреть, как различные модели OpenAI токенизируют текст, можно на сайте OpenAI (https://oreil.ly/0QI91).

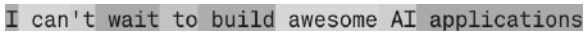

Рис. 1.1. Пример того, как GPT-4 токенизирует фразу

Процесс разбивки исходного текста на токены называется *токенизацией*. В GPT-4 средний токен составляет примерно половину длины слова (https://oreil.ly/EYccr). Так, 100 токенов приблизительно равны 75 словам.

Набор всех токенов, с которыми может работать модель, называется *словарем* модели. С помощью небольшого числа токенов можно составить немало разных слов, подобно тому как с помощью всего нескольких букв алфавита составляют множество слов. Словарь модели Mixtral 8x7B (https://oreil.ly/bxMcW) содержит 32 000 токенов, а модели GPT-4 — 100 256 токенов (https://github.com/openai/tiktoken/blob/main/tiktoken/model.py). Метод токенизации и размер словаря определяются разработчиками модели.

---

[1] В других языках, не в английском, один символ кодировки Unicode порой может обозначаться несколькими токенами.

Почему языковые модели используют в качестве единицы *токен*, а не *слово* или *символ*? Есть три основные причины.

1. В отличие от символов токены позволяют модели разбивать слова на значимые компоненты. Например, слово *cooking* можно разделить на *cook* и *ing*, при этом оба компонента будут содержать часть значения исходного слова.
2. Поскольку уникальных токенов меньше, чем уникальных слов, это уменьшает размер словаря модели, повышая ее эффективность (это обсуждается в главе 2).
3. Токены помогают модели обрабатывать незнакомые слова. Например, вымышленное слово *chatgpting* можно разделить на *chatgpt* и *ing*, что поможет модели понять его структуру. Токены обеспечивают баланс: их меньше, чем слов, но они несут больше смысла, чем отдельные символы.

Существует два основных типа языковых моделей: *маскированные* и *авторегрессионные*. Они различаются тем, какую информацию могут использовать для прогнозирования токена.

Маскированная языковая модель

Маскированная модель обучена предсказывать недостающие токены в любой части последовательности, используя контекст, *находящийся как перед пропущенными токенами, так и после них*. По сути, маскированная модель обучается заполнять пробелы. Например, имея контекст *My favorite __ is blue* («Мой любимый __ синий»), маскированная модель должна предсказать, что пропущено, вероятно, слово *color* («цвет»). Известный пример маскированной модели — двунаправленная нейронная сеть-кодировщик (bidirectional encoder representations from transformers, BERT) (Девлин и др., 2018, https://arxiv.org/abs/1810.04805). На момент написания книги маскированные модели, как правило, использовались для решения негенеративных задач, таких как анализ эмоциональной окраски высказываний и классификация текстов. Они эффективны и для решения задач, требующих понимания общего контекста, таких как отладка кода, где модель, чтобы выявить ошибки, должна понять код, находящийся как перед ошибкой, так и после нее.

Авторегрессионная языковая модель

Авторегрессионная языковая модель обучена прогнозировать следующий токен в последовательности с помощью только предыдущих токенов. Она предсказывает, какое слово будет следующим в предложении *My favorite color is __* («Мой любимый цвет __»)[1]. Авторегрессионная модель может генерировать токены непрерывно один за другим. Сегодня авторегрессионные языковые модели лучше всего подходят для генерации текста, и именно поэтому они намного популярнее маскированных моделей[2].

---

[1] Автогрессионные языковые модели еще называют *каузальными* (https://oreil.ly/h0Y8x).

[2] Маскированную языковую модель, такую как BERT, тоже можно использовать для генерации текста, если очень постараться.

На рис. 1.2 представлены два типа языковых моделей.

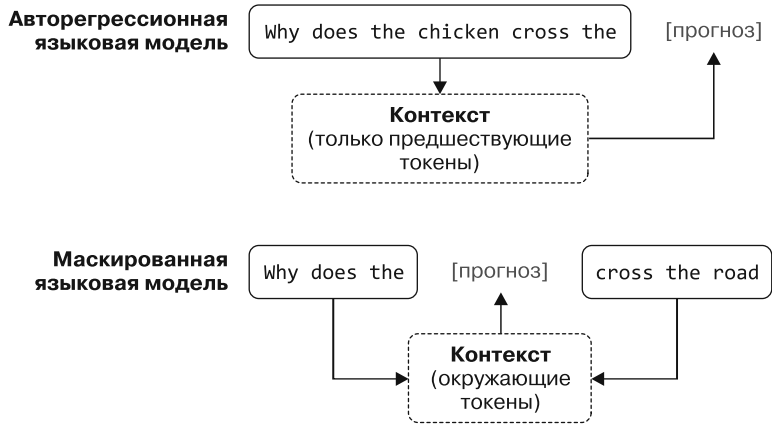

Рис. 1.2. Авторегрессионная и маскированная языковые модели

В этой книге, если не указано иное, под термином «языковая модель» будет подразумеваться авторегрессионная модель.

Вывод языковых моделей — открытого типа. Языковая модель может использовать свой фиксированный словарь с конечным числом токенов для создания бесконечного количества возможных выходных данных. Модель, способная генерировать результаты открытого типа, называется *генеративной*, отсюда и термин *«генеративный AI»*.

Языковую модель можно представить как *машину-завершатель*: получив текст (запрос), она пытается завершить текст, например:

текстовый запрос (от пользователя) — `To be or not to be;`
завершение (от языковой модели) — `, that is the question.`

Важно отметить, что завершения — это прогнозы, основанные на вероятностях, и нет гарантии, что они верны. В силу вероятностной природы языковых моделей работа с ними одновременно оказывается увлекательной и приносит разочарование. Мы поговорим об этом подробнее в главе 2.

Несмотря на простоту, завершение — невероятно мощный инструмент. Многие задания, в том числе связанные с переводом, реферированием, программированием и решением математических задач, можно сформулировать как задания на завершение. Например, получив запрос *How are you in French is...* («Как дела по-французски...»), языковая модель может дополнить его фразой *Comment ça va* («Как дела?»), эффективно выполнив перевод с одного языка на другой.

Другой пример. Получив запрос:

```
Вопрос: Скорее всего, это спам? Вот письмо: <содержание письма>
Ответ:
```

языковая модель может дополнить его фразой «Вероятно, это спам» и таким образом станет классификатором спама.

Завершение — мощный инструмент, однако оно не равноценно участию в беседе. Например, если задать модели-завершателю вопрос, она может дополнить ваш запрос, задав еще один вопрос, но не ответит на первый. В разделе «Последующее обучение» главы 2 обсуждается, как заставить модель адекватно отвечать на запросы пользователя.

Самообучение

Языковое моделирование — лишь один из множества алгоритмов машинного обучения. Существуют также модели для обнаружения объектов, тематического моделирования, рекомендательных систем, составления прогнозов погоды, предсказания цен на акции и т. д. Чем так примечательны языковые модели, что они оказались центральным элементом подхода к масштабированию, который стал причиной популярности ChatGPT?

Дело в том, что языковые модели способны обучаться *самостоятельно*, в то время как многим другим моделям необходим *учитель*. Обучение с учителем — это процесс обучения алгоритмов машинного обучения с использованием размеченных данных, и он может быть дорогостоящим и долгим. Самообучение помогает преодолеть узкое место, связанное с разметкой данных, путем создания более крупных наборов данных для обучения моделей, что фактически позволяет моделям масштабироваться. Вот как оно работает.

При обучении с учителем вы размечаете примеры, чтобы показать модели, какие типы поведения ей следует освоить, а затем обучаете ее на этих примерах. После обучения модель можно будет применять к новым данным. Например, для обучения модели обнаружения случаев мошенничества вы задействуете примеры транзакций, каждый из которых имеет пометку «мошенничество» или «не мошенничество». После того как модель их изучит, ее можно будет использовать для прогнозирования того, является ли транзакция мошеннической.

В основе успеха моделей AI в 2010-х годах лежало обучение с учителем. Модель, с которой началась революция глубокого обучения, AlexNet (Крижевский и др., 2012, https://oreil.ly/WEQFj), была обучена с учителем. Ее научили классифицировать более миллиона изображений в наборе данных ImageNet. Она относила каждое изображение к одной из 1000 категорий, таких как «автомобиль», «воздушный шар» или «обезьяна».

Недостаток управления заключается в том, что процесс разметки данных — дорогостоящий и занимает много времени. Если разметить одно изображе-

ние стоит 5 центов, то разметка миллиона изображений для набора данных ImageNet обойдется в 50 000 долларов[1]. Если необходимо, чтобы каждое изображение разметили два человека для перекрестного контроля качества разметки, стоимость удвоится. Поскольку в мире гораздо больше 1000 объектов, чтобы расширить возможности моделей для работы с большим количеством объектов, нужно добавлять разметку для новых категорий. При масштабировании до 1 млн категорий стоимость одной только разметки данных возрастет до 50 млн долларов.

Большинство людей способно справиться с разметкой обычных объектов без предварительного обучения. Поэтому ее можно получить за относительно небольшую стоимость. Однако не все задачи разметки настолько просты. Например, создание латинских переводов для модели перевода с английского на латынь будет стоить дороже. Стоимость разметки, обозначающей, показывает ли снимок КТ признаки рака, будет астрономической.

Самообучение помогает преодолеть узкое место, связанное с разметкой данных. При самообучении вместо того, чтобы требовать явную разметку, модель сама выводит метки, опираясь на входные данные.

Языковое моделирование происходит в режиме самообучения, поскольку каждая входная последовательность предоставляет как метки — токены, которые нужно прогнозировать, так и контекст, который модель может использовать для прогнозирования этих меток. Например, предложение *I love street food.* («Я люблю уличную еду») дает шесть обучающих примеров (табл. 1.1).

Таблица 1.1. Обучающие примеры на основе предложения I love street food. для работы языковой модели

Ввод (контекст)	Вывод (следующий токен)
`<BOS>`	`I`
`<BOS>, I`	`love`
`<BOS>, I, love`	`street`
`<BOS>, I, love, street`	`food`
`<BOS>, I, love, street, food`	`.`
`<BOS>, I, love, street, food, .`	`<EOS>`

[1] Фактическая стоимость разметки данных зависит от нескольких факторов, в том числе от сложности задачи, масштаба (при больших наборах данных стоимость разметки одного образца обычно снижается) и поставщика услуги разметки. Например, в сентябре 2024 года компания Amazon SageMaker Ground Truth (https://oreil.ly/EVXJl) брала 8 центов за изображение при разметке менее 50 000 изображений и всего 2 цента — при разметке более 1А млн изображений.

В табл. 1.1 маркеры <BOS> и <EOS> обозначают начало и конец последовательности. Они необходимы, чтобы языковая модель могла различить несколько последовательностей при обучении. Каждый маркер обычно обрабатывается моделью как отдельный специальный токен. Маркер конца последовательности особенно важен, поскольку он помогает языковым моделям понять, когда следует завершить ответы[1].

Самообучение отличается от обучения без учителя. При самообучении метки выводятся на основе входных данных. При обучении без учителя метки вообще не требуются.

Самообучение означает, что языковые модели могут учиться на текстовых последовательностях и им не нужна разметка. Поскольку текстовые последовательности встречаются повсюду — в книгах, блогах, статьях и комментариях на Reddit, — можно собрать огромный объем обучающих данных, который позволит языковым моделям масштабироваться и становиться большими языковыми моделями (large language models, LLM).

Термин «большая языковая модель», однако, вряд ли является научным. Насколько велика должна быть языковая модель, чтобы считаться *большой*? То, что сегодня называют большим, завтра может быть признано крошечным. Размер модели обычно измеряется количеством ее параметров. *Параметр* — это переменная в модели машинного обучения, которая обновляется в процессе обучения[2]. В целом, хотя это не всегда верно, чем больше параметров у модели, тем выше ее способность обучаться необходимому поведению.

Когда в июне 2018 года появилась первая модель генеративного предобученного трансформера (Generative Pre-trained Transformer, GPT) от компании OpenAI, она содержала 117 млн параметров и считалась большой. В феврале 2019-го, когда компания OpenAI представила модель GPT-2 с 1,5 млрд параметров, статус модели с 117 млн понизился и ее стали считать маленькой. На момент написания книги большой считается модель с 100 млрд параметров. Возможно, однажды и это значение назовут маленьким.

Прежде чем перейти к следующему разделу, хочу затронуть вопрос, который обычно воспринимается как само собой разумеющийся: *почему большим моделям требуется больше данных?* Большие модели обладают более высокой способностью к обучению, следовательно, им нужно больше обучающих данных, чтобы

---

[1] Как и у людей, важно понимать, когда следует замолчать.

[2] Меня учили, что параметры модели включают в себя ее веса и смещения. Однако сегодня для обозначения всех параметров модели, как правило, используется термин «веса».

максимально реализовать свой потенциал[1]. Вы можете обучать большую модель на небольшом наборе данных, но это будет неэффективно с точки зрения вычислительных ресурсов. С этим набором данных можно было бы достичь таких же или даже лучших результатов в моделях меньшего размера.

От больших языковых моделей к базовым

Языковые модели способны выполнять невероятные задачи, однако их применение ограничено текстом. Мы, люди, воспринимаем мир не только через язык, но и с помощью зрения, слуха, осязания и других чувств. Чтобы функционировать в реальном мире, AI необходима способность обрабатывать не только текстовые данные.

Именно поэтому языковые модели расширяются с целью охватить больше типов данных (модальностей). GPT-4V и Claude 3 могут понимать как изображения, так и текст. Некоторые модели даже понимают видео, 3D-объекты, строение белка и т. д. Включение большего числа типов данных в языковые модели повышает их мощность. Компания OpenAI в аннотации к своей системе GPT-4V (https://oreil.ly/NoGX7) в 2023 году отметила, что «включение дополнительных модальностей (таких как ввод изображений) в LLM рассматривается некоторыми как ключевой рубеж в исследованиях и разработках в области AI».

И хотя многие все еще называют Gemini и GPT-4V большими моделями, их лучше характеризует термин *«базовые модели»* (https://arxiv.org/abs/2108.07258). Слово «базовый» указывает одновременно на важность этих моделей для AI-приложений и на то, что на их основе можно строить другие модели для разных нужд.

Базовые модели — это прорыв в традиционной структуре исследований AI. Долгое время исследования AI были разделены по модальностям данных. Обработка естественного языка (natural language processing, NLP) занималась только текстом, компьютерное зрение — только зрением. Модели, работающие только с текстом, могут использоваться для решения таких задач, как перевод и определение спама. Модели, работающие только с изображениями, можно применять для обнаружения объектов и классификации изображений. Модели, работающие только с аудио, могут заниматься распознаванием речи (speech-to-text, STT) и ее синтезом (text-to-speech, TTS).

Модель, способная работать с несколькими модальностями данных, называется *мультимодальной*. Генеративную мультимодальную модель также обозначают

---

[1] Тот факт, что чем больше модель, тем больше обучающих данных ей необходимо, кажется контринтуитивным. Если модель мощнее, разве ей не должно требоваться меньше примеров для обучения? Однако мы не пытаемся заставить большую модель соответствовать по уровню производительности маленькой модели, использующей те же данные. Мы стараемся максимизировать ее результативность.

термином «большая мультимодальная модель» (large multimodal model, LMM). Если языковая модель генерирует следующий токен, основываясь только на текстовых токенах, то мультимодальная модель генерирует следующий токен, основываясь на текстовых и графических токенах или токенах любых других модальностей, которые она поддерживает (рис. 1.3).

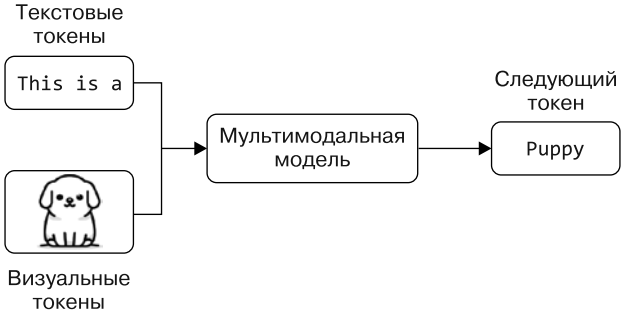

Рис. 1.3. Мультимодальная модель способна генерировать следующий токен на основе информации как из текстовых, так и из визуальных токенов

Как и языковым моделям, мультимодальным необходимы данные для масштабирования. Самообучение применимо и к мультимодальным моделям. Например, компания OpenAI для обучения своей модели CLIP (OpenAI, 2021), работающей с языком и изображениями, использовала вариант самообучения, названный *естественно-языковым самообучением* (https://oreil.ly/zcqdu). Вместо того чтобы вручную генерировать метки для каждого изображения, они нашли пары «изображение, текст», которые встречались в Интернете. Компании удалось создать набор данных из 400 млн таких пар, что в 400 раз превышает размер набора ImageNet, без затрат на разметку вручную. Благодаря этому набору данных CLIP стала первой моделью, способной обобщать задачи классификации изображений без необходимости дополнительного обучения.

В книге термин «базовые модели» применяется для обозначения как больших языковых моделей, так и больших мультимодальных моделей.

Обратите внимание на то, что модель CLIP не генеративная — ее не обучали генерировать вывод открытого типа. CLIP — это *модель эмбеддинга*, обученная создавать гибридные эмбеддинги текста и изображений. О них подробно рассказывается в подразделе «Знакомство с эмбеддингом» в главе 3. Пока можете считать эмбеддинги векторами, которые стремятся отразить смысл исходных данных. Мультимодальные модели эмбеддинга, такие как CLIP, служат основой для генеративных мультимодальных моделей наподобие Flamingo, LLaVA и Gemini (ранее Bard).