ГЛАВА 1

Обзор GPT-4 и ChatGPT

Представьте себе мир, в котором вы можете общаться с компьютерами так же быстро, как с друзьями. Как бы это выглядело? Какие приложения вы могли бы создать? Именно такой мир помогает построить OpenAI с помощью своих GPT-моделей, привнося в наши устройства языковые возможности, схожие с человеческими. Являясь последним достижением в области ИИ, GPT-4 и другие модели GPT представляют собой класс *больших языковых моделей* (large language model, LLM), обученных на огромных объемах данных, что позволяет им распознавать и генерировать текст, похожий на созданный человеком.

Последствия применения этих моделей ИИ выходят далеко за рамки простых голосовых помощников. Благодаря моделям OpenAI разработчики теперь могут использовать мощь *обработки естественного языка* (natural language processing, NLP), чтобы создавать приложения, которые предвосхищают наши потребности, как в научной фантастике или фильмах о будущем. GPT-4 и ChatGPT открывают новые горизонты — от внедрения инновационных клиентских систем обслуживания, способных обучаться и адаптироваться, до разработки персонализированных образовательных инструментов, подстраивающихся под каждого студента.

Но что такое GPT-4 и ChatGPT? В данной главе мы рассмотрим базовые принципы, историю и ключевые характеристики этих моделей ИИ. Разобравшись в основах этих моделей, вы сможете создавать приложения следующего поколения на базе LLM.

Введение в LLM

В этом разделе мы рассмотрим основные компоненты и принципы работы GPT-4 и ChatGPT, и, надеемся, вы получите полное представление о языковых моделях и NLP, роли трансформеров, процессах токенизации и предсказания в моделях GPT.

Изучение основ языковых моделей и NLP

GPT-4 и ChatGPT представляют собой новейший тип языковых моделей, полученных в области NLP, которая сама по себе является частью машинного обучения (machine learning, ML) и ИИ. Прежде чем углубиться в GPT-4 и ChatGPT, давайте рассмотрим NLP и смежные области.

Существуют различные определения ИИ, но одно из самых популярных гласит, что искусственный интеллект — это компьютерная система, которая может выполнять задачи, обычно требующие человеческого интеллекта. Согласно такому определению, многие алгоритмы можно отнести к ИИ: например, алгоритм прогнозирования трафика в GPS-приложениях или системы на основе правил, используемые в стратегических видеоиграх. Если посмотреть со стороны, кажется, что для выполнения этих задач машине требуется интеллект.

Машинное обучение является подмножеством искусственного интеллекта. В рамках ML мы не стремимся напрямую реализовать правила принятия решений, используемые системой ИИ. Наша цель — разработать алгоритмы, которые позволят системе самостоятельно обучаться на примерах. С 1950-х годов, когда начались исследования в области ML, в научной литературе было предложено множество алгоритмов ML.

Среди них особо выделяются алгоритмы глубокого обучения. *Глубокое обучение* (deep learning, DL) — это направление в обла-

сти машинного обучения, которое фокусируется на алгоритмах, работающих наподобие человеческого мозга. Такие алгоритмы называются *искусственными нейронными сетями*. Они способны эффективно обрабатывать очень большие объемы данных и успешно решать такие задачи, как распознавание изображений и речи, а также NLP.

Модели GPT-4 и ChatGPT построены на особом типе алгоритмов глубокого обучения, называемых *трансформерами*. Трансформеры — что-то вроде читающих машин. Они обращают внимание на различные части предложения или фрагменты текста, чтобы понять его смысл и выдать связный ответ. Они также способны учитывать порядок слов в предложении и их контекст. Это делает их высокоэффективными в таких задачах, как перевод, ответы на вопросы и генерация текста. На рис. 1.1 показана взаимосвязь между этими понятиями.

Рис. 1.1. Вложенный набор технологий от искусственного интеллекта до трансформеров

NLP — это область искусственного интеллекта, позволяющая компьютерам обрабатывать, интерпретировать и генерировать естественный человеческий язык. Современные решения в сфере NLP основаны на алгоритмах машинного обучения. NLP охватывает широкий круг задач.

- *Классификация текстов* по заранее определенным группам. К ним относятся, например, анализ настроений и категоризация тем. Компании могут использовать анализ настроений, чтобы определить мнение клиентов о своих услугах. Фильтрация электронной почты также является примером категоризации по темам, при которой письма могут быть отнесены к таким категориям, как «Личные», «Социальные», «Рекламные акции» и «Спам».
- *Автоматический перевод* текста с одного языка на другой. Сюда можно отнести не только перевод обычных текстов, но и перевод кода с одного языка программирования на другой, например с Python на C++.
- *Ответы на вопросы* на основе заданного текста. Например, онлайн-портал обслуживания клиентов может использовать модель NLP для ответа на часто задаваемые вопросы о продукте, а образовательное программное обеспечение (ПО) — для предоставления ответов на вопросы студентов по изучаемой теме.
- *Генерация текста*. Формирование связного и релевантного выходного текста на основе заданного входного текста, называемого запросом или чаще всего промтом (prompt).

Как уже говорилось, LLM — это модели машинного обучения, предназначенные для решения, в частности, задач генерации текстов. Языковые модели позволяют компьютерам обрабатывать, интерпретировать и генерировать человеческий язык, обеспечивая более эффективное взаимодействие между человеком и машиной. Для достижения этой цели LLM анализируют огромные объемы текстовых данных или *обучаются* на них, выявляя закономер-

ности и изучая связи между словами в предложениях. Для такого обучения могут использоваться различные источники данных. Это могут быть тексты из «Википедии», Reddit, архивов книг или даже из самого Интернета. При наличии входного текста в процессе обучения LLM учится предугадывать наиболее вероятные последующие слова и таким образом генерировать осмысленные ответы на входной текст. Современные языковые модели, появившиеся в последние несколько месяцев, настолько мощные и обучены на таком большом количестве текстов, что теперь они могут выполнять огромное количество задач NLP, включая классификацию текстов, машинный перевод, ответы на вопросы и многое другое. Модели GPT-4 и ChatGPT — это современные LLM, которые отлично справляются с задачами генерации текстов.

Эволюция LLM охватывает всего несколько лет. Все началось с простых языковых моделей, таких как *n-граммы*, которые пытались предсказать следующее слово в предложении на основе предыдущих. Для этого в *n*-граммных моделях используется *частота*. Предсказываемое слово — это наиболее часто встречающееся слово, следующее за предыдущими в тексте, на котором обучалась модель. Хотя такой подход казался неплохим началом, качество понимания контекста и грамматики у *n*-граммных моделей оставляло желать лучшего, что привело к проблемам с генерацией согласованного текста.

Для повышения производительности *n*-граммных моделей были разработаны более сложные алгоритмы обучения, в том числе рекуррентные нейронные сети (recurrent neural networks, RNN) и сети с долгой краткосрочной памятью (long short-term memory, LSTM). Эти модели могли обучаться более длинным последовательностям и анализировать контекст лучше, чем *n*-граммы, однако им по-прежнему требовалась помощь для эффективной обработки больших объемов данных. Долгое время именно эти типы рекуррентных моделей считались наиболее эффективными и поэтому часто применялись в таких задачах, как автоматический машинный перевод.

Ключевые особенности трансформера и его роль в обработке естественного языка

Трансформеры совершили революцию в NLP, прежде всего благодаря тому, что успешно преодолели одно из критических ограничений предыдущих моделей NLP — рекуррентных нейронных сетей, а именно их неспособность работать с длинными текстовыми последовательностями и запоминать контекст при таких объемах. Другими словами, если RNN склонны забывать контекст при больших объемах информации (печально известное «катастрофическое забывание»), то трансформеры способны эффективно обрабатывать и сохранять его.

Центральным элементом этой технологии является *механизм внимания* — простая, но мощная идея. Вместо того чтобы рассматривать все слова в последовательности текста как одинаково важные, модель «обращает внимание» на наиболее релевантные термины для каждого шага своей задачи. Перекрестное (или кросс-) внимание (cross-attention) и самовнимание (self-attention) — два метода, основанные на этом механизме внимания, которые часто встречаются в LLM и, в частности, в архитектуре трансформеров.

Перекрестное внимание помогает модели определить значимость различных частей входного текста для точного предсказания следующего слова в выходном тексте. Представьте прожектор, который светит на слова или фразы во входном тексте, выделяя релевантную информацию, необходимую для предсказания следующего слова, и игнорируя менее важные детали.

Для наглядности рассмотрим пример с переводом простого предложения. Допустим, у нас есть предложение на английском языке *Alice enjoyed the sunny weather in Brussels*, которое необходимо перевести на французский: *Alice a profité du temps ensoleillé à Bruxelles*. В примере мы сосредоточимся на генерации французского слова *ensoleillé*, которое означает *sunny* («солнечный»). При предсказании

механизм перекрестного внимания присвоит больший вес английским словам *sunny* и *weather*, поскольку они оба имеют отношение к значению слова *ensoleillé*. Фокусируясь на этих двух словах, перекрестное внимание помогает модели создать точный перевод части предложения (рис. 1.2).

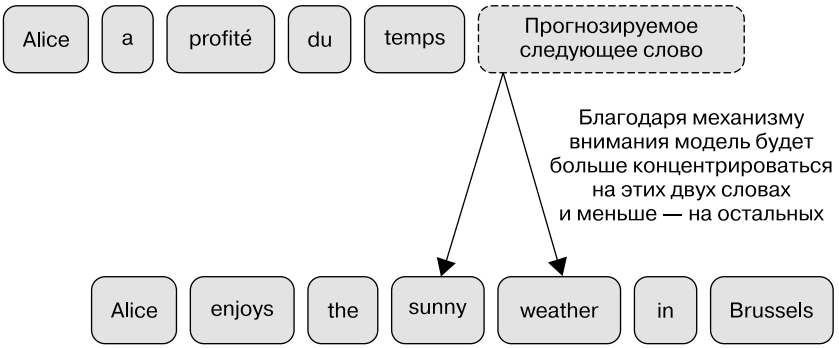

Рис. 1.2. Принцип работы механизма перекрестного внимания

Под *самовниманием* понимается способность модели фокусироваться на различных частях входного текста. В контексте NLP это позволяет модели оценивать значимость каждого слова в предложении по отношению к другим словам. Это помогает ей лучше разбираться в связях между словами и формировать новые *концепции* на основе нескольких слов во входном тексте.

В качестве более конкретного примера можно привести такое предложение: «Алиса получила похвалу от своих коллег». Предположим, что модель пытается понять значение слова *«своих»*. Механизм самовнимания присваивает различные веса словам в предложении и выделяет те, которые относятся к слову «своих» в данном контексте. В нашем случае больший вес придается словам *«Алиса»* и *«коллеги»*. Самовнимание помогает модели строить новые понятия на основе этих слов. В данном примере одним из понятий, которое может появиться, будет «коллеги Алисы», как показано на рис. 1.3.

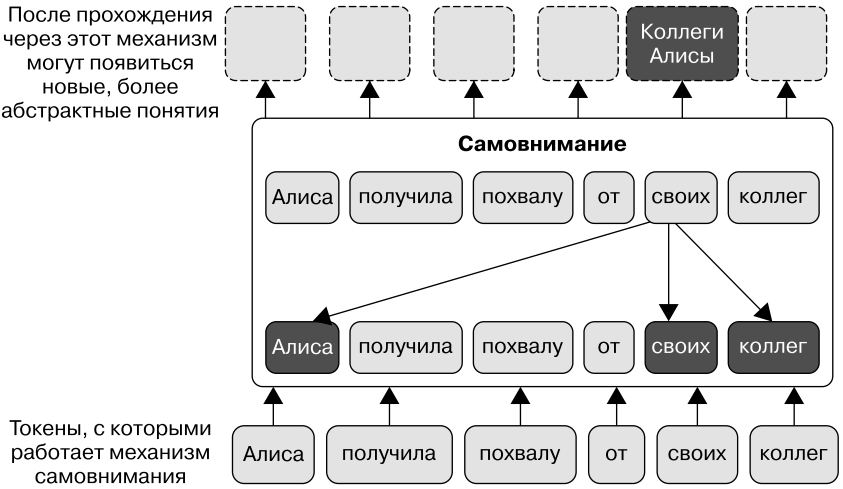

Рис. 1.3. Самовнимание стало причиной появления фразы «коллеги Алисы»

В отличие от рекуррентной архитектуры трансформеры обладают дополнительным преимуществом — их легко *распараллелить*. То есть трансформеры способны обрабатывать несколько частей входного текста одновременно, а не последовательно. Это позволяет ускорить вычисления и процесс обучения, поскольку различные части модели могут работать параллельно, не дожидаясь завершения предыдущих шагов, в отличие от рекуррентных архитектур, где требуется последовательная обработка. Возможность параллельной обработки трансформерных моделей идеально вписывается в архитектуру графических процессоров (GPU), предназначенных для одновременной обработки нескольких вычислений. Благодаря своей высокой параллельности и вычислительной мощности графические процессоры идеально подходят для обучения и запуска трансформерных моделей. Это достижение позволило специалистам по анализу данных обучать модели на гораздо бóльших наборах данных, что стало ключевым моментом на пути к развитию LLM.

Архитектура трансформеров, представленная в 2017 году Васвани и др. из Google в статье *Attention Is All You Need* (https://oreil.ly/jVZW1),

изначально разрабатывалась для решения задач преобразования последовательности в последовательность, таких как машинный перевод. Стандартный трансформер состоит из двух основных компонентов: кодировщика и декодировщика, оба из которых сильно зависят от механизма внимания. Задачами кодировщика являются обработка входного текста, выявление ценных признаков и формирование осмысленного представления этого текста, которое называется *вложением*, или *эмбеддингом (embedding)*. Затем декодировщик использует это вложение для создания выходного результата, например перевода или резюме. В итоге получается эффективная интерпретация закодированной информации.

Генеративные предварительно обученные трансформеры, обычно называемые GPT (generative pre-trained transformers), представляют собой семейство моделей, основанных на архитектуре трансформеров и специально использующих часть оригинальной архитектуры, отвечающую за декодирование. В GPT кодировщик отсутствует, поэтому нет необходимости в механизме перекрестного внимания для интеграции вложений, произведенных кодировщиком. В результате GPT полагается исключительно на механизм самовнимания внутри декодировщика для создания контекстно зависимых представлений и прогнозов. Отметим, что другие известные модели, такие как BERT (bidirectional encoder representations from transformers), основаны на кодирующей части. Данный тип модели в книге не рассматривается. На рис. 1.4 показана эволюция различных моделей.

Рис. 1.4. Эволюция методов NLP от n-грамм до LLM

Суть процесса токенизации и прогнозирования в моделях GPT

Языковые модели в семействе GPT принимают на вход промт, а в ответ генерируют текст. Этот процесс известен как *завершение текста*. Например, промт может звучать так: *«Сегодня хорошая погода, и я решил»*, а выходной текст модели может быть таким: *«Пойти прогуляться»*. Вам может быть интересно, как модель LLM строит этот выходной текст на основе входного промта. Как вы увидите, это в основном просто вопрос вероятностей.

Когда промт отправляется в LLM, вводимый текст сначала разбивается на более мелкие фрагменты, называемые *токенами*. Эти токены представляют собой отдельные слова, части слов, пробелы и знаки пунктуации. Например, предыдущий промт может быть разбит следующим образом: *[«Сегодня», «хорош», «ая», «пого», «да», «,», «по», «этому», «я», «ре», «шил»]*. У каждой языковой модели есть свой токенизатор. Токенизатор GPT-4 на момент написания книги недоступен, но вы можете протестировать токенизатор GPT-3, для этого перейдите по ссылке https://platform.openai.com/tokenizer.

Следует отметить, что 100 токенов примерно равны 75 словам на латинице.

Используя принцип внимания и архитектуру трансформеров, модель LLM обрабатывает эти токены и интерпретирует связи между ними и общий смысл промта. Трансформеры позволяют модели эффективно идентифицировать в тексте критическую информацию и его контекст.

Для создания нового предложения LLM предсказывает наиболее вероятные токены, исходя из контекста промта. Компания OpenAI создала две версии GPT-4 с контекстными окнами с количеством токенов 8192 и 32 768. В отличие от предыдущих рекуррентных моделей, которые с трудом справлялись с обработкой длинных

входных последовательностей, архитектура трансформеров с механизмом внимания позволяет современной LLM рассматривать контекст как единое целое. Основываясь на этом, модель присваивает вероятностные оценки каждому потенциальному последующему токену. Токен с наивысшей вероятностью выбирается в качестве очередного в последовательности. В нашем примере после слов «Сегодня хорошая погода, и я решил» следующим по вероятности токеном может быть «пойти».

Затем этот процесс повторяется, но теперь контекст становится таким: «Сегодня хорошая погода, и я решил пойти», где к исходному предложению добавляется ранее предсказанный токен «пойти». Этот процесс повторяется до тех пор, пока не будет сформировано полное предложение: «пойти прогуляться». Данный метод основан на способности LLM учиться предугадывать следующее наиболее вероятное слово на основе массивных текстовых данных (рис. 1.5).

Рис. 1.5. Процесс завершения текста является итерационным и выполняется токен за токеном