Глава 8. Данные как фактор конкурентного преимущества

Это продолжение перевода книги Томас и др. Создание бизнес-ценности с генеративным ИИ. В предыдущей главе мы поговорили о силе (и потенциале) малых языковых моделей (SLM). Мы ввели идею, что одна модель не должна — и не сможет — править всеми. Мы объяснили, что гигантские модели громоздки, дороги и концентрируют власть в руках немногих (тех, кто может позволить себе их разрабатывать). Более того, они не помогут вам использовать свои данные для создания ценности, если только вы не отдадите эти данные. Другими словами, такие модели делают вас пользователем ИИ, а не создателем ценности с его помощью. Мы утверждаем — и продолжим это доказывать — что узкоспециализированные модели способны на потрясающие вещи. Нам хочется видеть будущее ИИ открытым, поэтому мы против идеи, что один супермозг LLM должен контролировать всё.

Предыдущая глава Содержание Следующая глава

Одна из ключевых идей этой книги — стать создателем ценности с помощью ИИ можно только в том случае, если вы начнёте воспринимать свои данные как спящую суперсилу. Чтобы раскрыть возможности ИИ и начать создавать настоящую ценность, мы уверены: нужно делать серьёзную ставку на создание внутри вашей компании экосистемы, способной запускать работу с вашими данными и превращать их в актив. Мы настолько верим в это, что сделали это названием всей книги: «Создатели ценности ИИ».

В этой главе мы рассмотрим, как разработчики и отраслевые специалисты в вашей компании могут использовать новые методы кастомизации моделей, чтобы участвовать в создании корпоративных моделей ИИ, прокладывая путь к устойчивым и уникальным инновациям в этой области. То есть — создавая ценность.

Кастомизация open source для бизнеса: новый взгляд на корпоративные данные

Менее 1% корпоративных данных попадает в сегодняшние LLM. А если вы действительно хотите стать тем самым Создателем ценности ИИ, ради которого написана эта книга, вам придется задействовать свой самый ценный ресурс — данные — и сделать их частью вашей LLM-стратегии. Именно это и откроет перед вами море возможностей для создания ценности.

Чтобы по-настоящему понять, насколько это важно, давайте на минуту перенесемся к истокам цифрового мира. Почти 350 лет назад Готфрид Вильгельм Лейбниц заложил его основы. Уже тогда он осознал, что любую информацию — будь то язык или математика — можно закодировать в двоичной форме. (Лейбниц не только изобрел бинарную систему, но и участвовал в создании математического анализа, так что неудивительно, если кто-то из вас его недолюбливает.) Он однажды сказал: «Чтобы создать всё, достаточно одного». Лейбниц явно понимал силу в том, чтобы представлять информацию иначе — в данном случае, в виде двоичного кода.

Перенесемся в сегодняшний день: последние десятилетия показали, насколько мощной может быть трансформация, основанная на развитии форматов представления данных. Сейчас даже вкус и запах можно выразить в числовой форме, а затем — в тех самых единицах и нолях, с которыми работает компьютер. Парфюмерные и вкусовые лаборатории буквально создают новые продукты, используя векторы для представления, скажем, лимонной свежести или медово-сливочного аромата. Только вдумайтесь: кто, кроме ИИ, мог бы предложить мороженое со вкусом бейгла «всё и сразу»?

Честно говоря, задолго до появления LLM описания вин и парфюмов уже развлекали нас своей поэтичностью (а иногда и откровенно абсурдной фантазией). Ведь правда, кто реально чувствует «отголосок поцелованного солнцем бузина на утренней росе» или различает «нотки меланхолии с дерзким шлейфом экзистенциального кризиса»? Дальше будет только креативнее — а если говорить честно, возможно, ещё более нелепо — благодаря LLM.

Оригинальный тур по эпохам: ретроспектива представлений данных

За последние десятилетия новые формы представления данных открывали для бизнеса и индустрий совершенно новые возможности и горизонты. Мы решили уделить этому немного внимания, чтобы вы в полной мере осознали ценность LLM для вашей компании — особенно когда такая модель работает с вашими данными. Суть в том, что корпоративные данные можно встроить в новую форму представления — LLM — которая позволит использовать их так, как ещё недавно можно было только представить в кино, при этом создавая реальную, осязаемую ценность для бизнеса.

Если вдуматься, помимо весов в модели, ИИ — это просто сжатые данные. Это новая форма их представления. И, как показывает история, за последние десятилетия мы уже видели разные эпохи представлений данных — каждая из них открывала новую эру создания ценности. Текущая ИИ-революция напрямую связана с силой этих представлений и с тем, что теперь мы можем кодировать невероятные объёмы информации во всех формах внутри этих мощнейших "контейнеров", которыми и являются LLM. Вот как мы видим эти эпохи.

До 1980-х: экспертные системы

Это были (а точнее, и по сей день остаются) вручную собранные символические представления данных. Информация кодировалась в реляционных базах данных, что дало бизнесу совершенно новый способ организации и работы с данными. Впервые стало возможным автоматизировать такие процессы, как расчёт зарплаты или привязка транзакций к складам — и всё это в единой системе. В эту эпоху появились экспертные системы. Люди писали правила, описывающие логические бизнес-процессы на основе структурированных данных. Отличные примеры — системы выявления мошенничества или управления цепочками поставок. Многие компании до сих пор используют этот подход: если нарушено правило — появляется флаг или запускается действие.

Правила хороши для определённого круга задач, но они не особенно гибкие, и исключения случаются постоянно. То есть в итоге такие системы могут работать только в пределах заранее заданных рамок. К тому же на поддержание таких систем уходит немало ручного труда: каждое новое правило нужно прописывать отдельно. (Поэтому эту эпоху мы называем ручной. Например, чтобы сохранить данные в реляционной базе, DBA сначала вручную проектировал схему, в которую эти данные вписывались. Всё требовало участия человека — как в разработке, так и в обдумывании логики.)

Допустим, кто-то решил выявлять мошенничество по платежу в $1 на заправке — появилось новое правило. Потом этот признак перестал быть надежным, и понадобилось другое — снова новое правило. Пример простой, но в прошлом так действительно работали (а иногда — не работали, и тогда бизнесы разочаровывались). В общем, эти системы справлялись, пока правила оставались актуальными. Но со временем правила становилось всё больше, вариантов всё сложнее, и системы рушились под собственной тяжестью.

Теперь представьте сегодняшнюю цифровую экономику. Как система на правилах может распознавать угрозы с учетом множества точек входа и сложных транзакций? Как она выделит сигнал в шуме и мимолётных действиях злоумышленников, как отреагирует на скоординированные атаки с объединённым мониторингом в реальном времени? Никак.

1980-е — ~2010: Машинное обучение

Теперь начинается эпоха более специализированных и менее «ручных» представлений признаков в данных. Как это произошло? Всё дело в том, что с ростом объёмов доступных данных произошёл сдвиг в сторону подходов, основанных на данных. Это было большим прорывом: вместо того чтобы вручную прописывать правила, компьютеры начали извлекать их самостоятельно — на основе примеров из реального мира. Очень круто! Многие из этих методов используются учёными-аналитиками до сих пор: деревья решений, опорные векторы (SVM), метод ближайших соседей и прочее. Эта эпоха была о том, как научить машину помогать в построении признаков и учиться на результатах. И надо сказать, результаты были весьма впечатляющие.

Хотя машины и начали по-новому работать с данными, всё ещё при участии человека, в этот период появились новые формы представлений и механизмы кодирования. Например, графовые представления данных — в виде сетей с узлами и связями. В какой-то момент мир научился использовать такую форму данных, научился по ней "передвигаться" — и она стала критически важной для задач вроде интернет-поиска, социальных сетей, соединения людей и сообществ.

2010 — ~2017: Глубокое обучение

Теперь мы переходим в эпоху больших данных (вспомним те самые 3V: объём, скорость и разнообразие). Компьютеры получили доступ к невиданным прежде объёмам данных. Теперь они не просто обнаруживали новые представления, но и начали их создавать. Так начался мир обученных на задачах признаков — уже не ручных, а извлечённых из данных самой моделью.

В эту эпоху мир получил доступ к колоссальным вычислительным ресурсам (благодаря облакам и видеокартам) и стремительно растущим объёмам данных (спасибо интернету). Машины начали

строить представления признаков — но всё ещё с серьёзной зависимостью от экспертов и большого объёма ручной работы. В то время ресурсы для обработки данных были ограничены, и возможностей для построения более сложных моделей явно не хватало. Например, ИИ в области обработки языка тогда едва ли «помнил» больше пары слов.

Так стартовала эра глубокого обучения (deep learning). Появились новые инструменты и методы, вроде функций активации (подробности которых выходят за рамки этой книги), которые стали основой этой эпохи. Идеально совпали два тренда: рост объёмов данных (начало эпохи big data) и рост вычислительной мощности (выяснилось, что видеокарты, созданные для игр, отлично подходят для работы с матричной алгеброй — именно она лежит в основе глубокого обучения).

Вот тогда и начались настоящие чудеса (пока ещё не магия — но уже близко). Вся эта вычислительная мощь (GPU для построения представлений) соединилась с удобной моделью потребления (облачные сервисы) — и вдруг оказалось, что ИИ можно создать дешевле, чем стоит чашка кофе. Машины начали учиться на огромных массивах данных и строить представления признаков, специфичных под задачи. Например, компьютерное зрение — для поиска аномалий на рентгеновском снимке или дефекта в точке сварки на производственной линии. Некоторые из таких признаков были крайне сложными, и компьютеры начали изобретать составные, «грубые» признаки: например, комбинировать пол, местоположение, рост и профессию в один обобщённый признак, описывающий что-то новое.

Сегодня: базовые модели (также известные как LLM)

Сегодня мы умеем кодировать практически любую форму знания и работать с данными так, как раньше даже не могли себе представить. Как мы уже говорили, базовые модели — это о возможности закодировать огромные объёмы информации в любой форме внутри нового типа мощнейших моделей. Мир вступил в эпоху LLM, где используется не только гигантская вычислительная мощность и колоссальные объёмы данных, но и новая технология — обучение с самоконтролем (self-supervised learning), основанная на трансформерах. Благодаря ей необходимость в тщательно размеченных данных резко сократилась — и это колоссальный сдвиг по сравнению с прошлым.

Конкретно: это новое представление данных обучается на гигантских наборах данных и способно выполнять широкий спектр задач общего назначения. Такие модели выступают как основа, платформа, на которой можно строить более специализированные приложения. Их гибкость и масштаб отличают их от представлений прошлой эпохи, которые создавались под конкретные задачи и на ограниченных массивах данных.

В процессе обучения такие модели берут обучающие данные и разбивают их на маленькие фрагменты — токены (токеном может быть слово или часть слова). Таких токенов создаются триллионы, и они затем преобразуются в векторы — форму, в которой ИИ может с ними работать. Причём токены могут быть чем угодно: не только словами, но и кодом, изображениями, звуками, профилями вкуса и запаха и т.д. По мере того как токены (уже как векторы) проходят через слои нейросети, на них накладываются математические преобразования — в основном матричные умножения и другие базовые операции, но в гигантских масштабах. В процессе обучения данные комбинируются, перемешиваются, и токены формируют меняющиеся последовательности. Более того, информация из разных типов данных — например, текст и аудио — может совмещаться в одной модели. Яркий пример — последняя модель GPT от OpenAI, объединяющая генерацию текста и изображений (благодаря DALL·E) в одном продукте.

Во время обучения параметры нейросети настраиваются таким образом, чтобы модель всё лучше и лучше воспроизводила закономерности в последовательностях токенов. Со временем она начинает «понимать» всё больше нюансов структуры данных, на которых она учится, а также взаимосвязи, скрытые знания и контекст. Это не магия — это математика, человеческая изобретательность и огромные вычислительные мощности.

Именно масштаб делает эту форму представления данных такой мощной: объём знаний, которые можно в неё вложить, связность (модель находит смысловые связи между самыми разными входными данными) и мультимодальность.

Теперь — главная мысль этой главы. За последние пару лет мы стали свидетелями того, как такие представления практически поглотили все доступные публичные данные. Допустим, для простоты, что 100% такой информации уже попало в LLM. Теперь сравните это с тем, что мы упоминали раньше:

лишь около 1% корпоративных данных попало в готовые LLM. Контраст очевиден: почти все публичные данные — внутри, и почти все корпоративные — снаружи.

Встаньте и представьте… свои данные

К этому моменту книги вы уже должны чувствовать, насколько эпоха ИИ — это настоящий переломный момент. Сбор данных в гигантских объёмах — проблема уже решённая (а вот понять их — совсем другая задача), вычислительные мощности доступны в изобилии, а новые ИИ-технологии объединились с этими ресурсами, создав идеальную бурю для революции в ИИ. Так с чего начать, чтобы ваши данные наконец начали работать? Как мы обсуждали в пятой главе, всё начинается с надёжной LLM. Когда вы выбрали базовую модель, которой доверяете, пора встраивать в неё корпоративные данные и переходить к новой, мощной форме их представления. И, наконец, вы разворачиваете свою кастомную модель, масштабируете и создаёте ценность с помощью ИИ. Итак, поговорим об этих трёх шагах.

Шаг 1: Всё начинается с доверия

Не стоит недооценивать значимость этого момента: всё в ИИ меняется именно из-за новой формы представления данных. Чтобы начать извлекать ценность из корпоративных данных, первый шаг вообще не про данные. Он про выбор модели, которой вы можете доверять. Подумайте о ней как о сосуде для ценности — о фундаменте, на котором вы будете строить. Это критически важно, потому что ваши данные будут наслаиваться на уже существующее содержание этой модели. Поэтому нужно понимать, что уже заложено в эту основу: какие данные использовались при обучении, по какому «рецепту» всё было сделано, как модель устроена изнутри.

Вспомните первую главу, где мы советовали задавать своему вендору LLM прямые вопросы вроде: «Какие данные использовались для обучения модели?» — и не принимать ответы типа «Это не ваше дело» или «Мы не знаем». Разве это не то же самое, как при выборе места для строительства дома? Фундамент должен быть прочным. А если он состоит из нарушений авторских прав, ненависти, агрессии, грубости, предвзятости, расизма, порнографии и прочего? Если вы считаете, что в интернете только достоверная информация, нет вредного контента и вам вообще нечего опасаться — вперёд! Но вы ведь видели, как выглядит типичная ветка на Reddit и какой там уровень токсичности? А ведь есть ещё и форумы, в которые лучше вообще не заходить. Хотите ли вы, чтобы ваша ценная корпоративная информация смешалась с таким «содержимым», когда вы начнёте её использовать?

Давайте снова обратимся к аналогии из главы 5 — про воду. Представьте, что мы даём вам стакан воды (LLM), и вы собираетесь добавить в него лимон и сахар (то есть ваши данные), чтобы получить лимонад. Но стакан непрозрачный. Вы не знаете, откуда взялась вода, и вам никто этого не говорит. Вы бы стали выливать в такой стакан свежевыжатый лимон и дорогой органический сахар? Возможно, в стакане чистейшая родниковая вода. А может, мутная дождевая лужа или вообще что-то опасное. Вы не видите, что там внутри! Выпили бы вы потом это? Вряд ли. Так почему вы готовы сделать это с одним из самых ценных активов своей компании — с данными?

Точно так же и с LLM: почти невозможно заставить модель полностью игнорировать всё, что уже содержится в её «воде», и опираться только на ваши данные. Конечно, существуют техники вроде RAG и дообучения, которые помогают. Но даже при кастомизации модель почти наверняка будет сохранять часть характеристик — и в плане качества, и в плане безопасности — от той базы, на которой она построена.

В этой аналогии важно, чтобы стакан, который вы берёте для приготовления лимонада, был прозрачным — вы должны видеть, что внутри. Нужно знать, откуда взялась вода, которая станет основой для вашего напитка, чтобы понимать, что получится в итоге: как он будет выглядеть, на вкус, насколько он будет безопасным. То же самое происходит, когда вы начинаете работать с корпоративными данными и LLM: вам нужна базовая модель с прозрачной историей — чтобы было понятно, какие данные использовались при её обучении и по какому принципу она была построена. Только тогда вы сможете с уверенностью и безопасно добавлять в неё свои данные.

Ещё один важный аспект прозрачности — это широкие коммерческие права и свобода действий в отношении итоговой модели. Напомним, что эта глава вовсе не о поставщиках моделей — она о ваших данных. У вас должны быть разрешения, позволяющие свободно использовать и развивать вашу улучшенную модель. Ведь вы вкладываете в неё свои знания, и она должна работать на ваш бизнес. Кроме того, поскольку вы опираетесь на базовую модель, содержащую данные из публичных

источников, она должна быть юридически защищена от возможных претензий — то есть поставщик должен обеспечивать индемнитет.

> Как говорилось в пятой главе, обязательно проверьте, какие юридические гарантии (индемнитет) предоставляет ваш вендор. Сегодня практически все предлагают ту или иную форму защиты, но условия у всех разные. Кто-то не защищает то, что создаётся моделью, кто-то — наоборот, полностью берёт ответственность. У кого-то ограничения по сумме или только по форме использования. Придётся подключить вашу юридическую команду.

IBM и доверие к Granite

Скажем ещё раз: практически вся эта книга — не про IBM. Мы старались делиться знаниями, описывать кейсы, указывать на риски и помогать вам разобраться в том, что действительно важно на пути к ИИ — и при этом лишь изредка упоминали IBM. Но теперь позволим себе пару страниц рассказать про одну из моделей, о которой до этого почти не говорили: IBM Granite.

Мы гордимся серией моделей Granite, потому что она отвечает всем критериям, о которых мы говорили: полная прозрачность в обучающих данных (ознакомьтесь с подробностями в техническом отчёте по Granite 3); модели выпущены с открытой и понятной лицензией Apache 2.0; и, самое главное, семейство Granite разработано с учётом задач бизнеса — модели можно дообучать на корпоративных данных и при этом сохранять экономическую эффективность.

На рис. 8.1 показано разнообразие моделей в серии IBM Granite 3 (и, возможно, к моменту выхода книги уже вышла или вот-вот выйдет версия Granite 4).

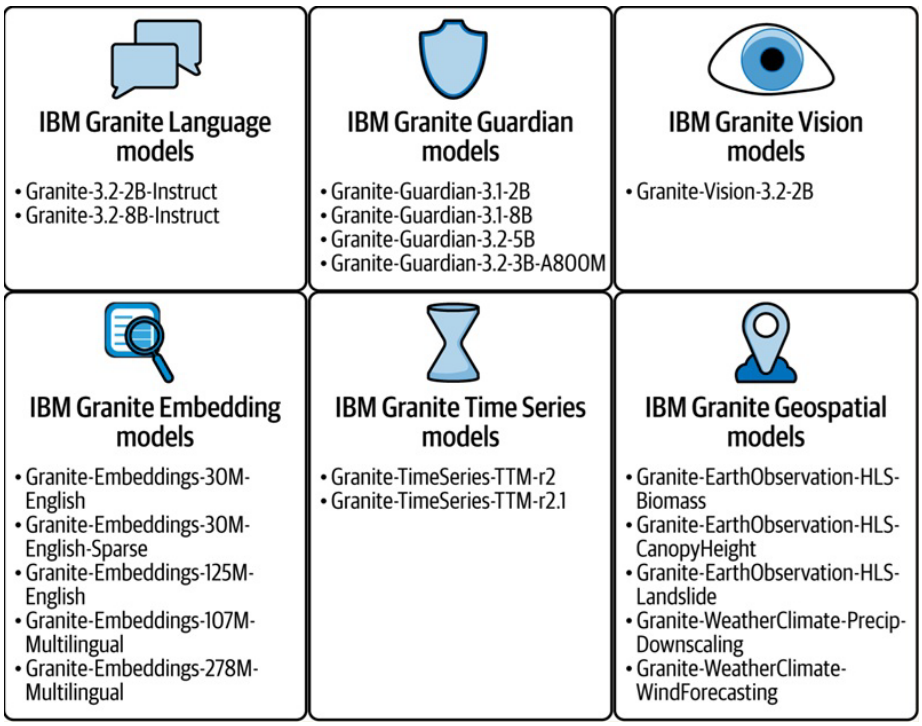

Рис. 8.1. Модельный ряд IBM Granite

Granite Language. Основа для всех языковых задач в бизнесе. Это рабочие лошадки среди LLM: производительные, масштабируемые, готовые к кастомизации с помощью PEFT или InstructLab.

Granite Vision. Мультимодальные модели для задач визуального анализа. На вход — изображение и запрос, на выход — текст. Подходят для распознавания документов, анализа графиков, объяснения трендов и даже мультимодальных задач с RAG.

Granite Guardian. Это модели-«ограждения» (guardrails), которые работают в паре с любой LLM (не только Granite) и следят за безопасностью ввода и вывода — предотвращают вредный или предвзятый контент, галлюцинации и прочее (см. главу 5).

Granite Embedding. Эти модели преобразуют большие объёмы текста и кода в векторные представления, то есть в числовую форму. Это особенно полезно для реализации RAG-сценариев.

Granite Time Series. Очень компактные модели, основанные на GenAI, для прогнозирования. Вместо того чтобы обучаться на языковых данных, они обучались на массивах временных рядов, чтобы обрести свои предсказательные способности.

Granite Geospatial. Мультимодальные модели в области наук о Земле, разработанные совместно с NASA. Используются для прогнозов погоды, оценки биомассы на спутниковых снимках и других задач.

Главные принципы моделей Granite — это прозрачность и гибкость. Каждая модель выпускается с полной информацией об обучающих данных и лицензией Apache 2.0, предоставляющей максимум свободы для использования и внедрения в бизнесе. Благодаря такому подходу к открытости Granite получила одну из самых высоких оценок в Индексе прозрачности Стэнфорда среди поставщиков LLM.

Шаг 2: Представление корпоративных данных внутри LLM

Когда вы выбрали базовую модель, которой доверяете (в терминах нашей аналогии — прозрачный стакан с чистой водой), следующий шаг — выбрать способ, с помощью которого вы добавите в неё свои корпоративные данные (то есть сахар и лимон, превращающие воду в лимонад). Существует несколько техник, среди них наиболее распространённые:

Retrieval-augmented generation (RAG). Скорее всего, вы уже слышали об этом подходе — это один из самых часто применяемых в корпоративной среде. Мы упоминали его в разных главах, но стоит остановиться на нём отдельно, поскольку это один из самых прямых способов добавления корпоративных данных в LLM.

В RAG-подходе после того, как пользователь отправляет запрос, этот запрос используется для извлечения релевантной информации из корпоративного хранилища — обычно на основе семантического сходства текста. (Чаще всего это векторная база данных, поддерживающая подобный поиск, но это может быть и обычная реляционная база, гибридная система или даже объектное хранилище.)

Затем исходный запрос объединяется с найденной информацией (её называют контекстом запроса) и подаётся в модель. В результате LLM использует как свои обученные знания, так и предоставленные в реальном времени данные для формирования ответа.

Как можно заметить, в RAG-подходе веса модели не затрагиваются, и у этого есть свои плюсы и минусы. Это отличная техника, особенно если важно, чтобы ответы модели основывались на самой свежей информации (намного проще обновить базу данных, чем переобучить модель). Но у неё есть и ограничения. Во-первых, RAG — это не просто модель, это целая система со множеством зависимостей и технических сложностей. Во-вторых, каждый раз, когда вы хотите получить от модели ответ — например, на вопрос о внутренней политике HR, — вы должны заново подгружать в неё весь текст этой политики. Это повышает стоимость инференса, особенно при частом использовании. И, наконец, модель при таком подходе не «усваивает» новую информацию, то есть не учится применять её в других задачах и контекстах.

Дообучение

Ещё один распространённый способ адаптировать LLM под корпоративные данные — это дообучение (fine-tuning). Дообучение означает обновление весов модели на основе новых данных (тех самых пар "ввод–вывод", о которых мы уже говорили ранее). Такой подход требует гораздо меньше вычислительных ресурсов, чем полное переобучение модели с нуля, и меньше объёмов данных. Эта техника даёт разумную точку входа для тех, кто хочет стать Создателем ценности ИИ и начать настраивать модели под себя.

Существует множество техник дообучения. Например, есть метод supervised fine-tuning (SFT), при котором обновляются все параметры модели. Есть вариант под названием parameter-efficient fine-tuning (PEFT), где меняется лишь часть параметров. Также есть подход low-rank adaptation (LoRA), при котором обучается внешний по отношению к LLM модуль параметров, работающий в связке с основной моделью. Удобство LoRA в том, что такие модули можно отключать, когда они не нужны, или заменять на другие для решения новых задач.

Допустим, вы управляете компанией, разрабатывающей ролевые игры, и создаёте LoRA-адаптер поверх вашей LLM для диалогов и взаимодействия с неигровыми персонажами, но затем подключаете другой LoRA-модуль — для повествования и написания историй. У этого подхода есть и свои минусы:

если вы хотите сделать 50 дообученных адаптаций, вам придётся управлять 50 разными адаптерами. Кроме того, поскольку они используют матрицы очень низкого ранга, можно предположить, что в какой-то момент их пропускная способность окажется ограниченной.

Выбор метода дообучения зависит от целей по производительности и бюджета. Чем больше параметров вы нацеливаетесь обновить, тем лучше будет результат, но и стоимость обучения возрастёт. Хотя дообучение действительно позволяет улучшить модель с опорой на закрытые данные, у такого подхода есть и побочный эффект, известный как катастрофическое забывание. Это означает, что как только модель дообучается под конкретную задачу, она становится специалистом в этой задаче, но теряет часть своих способностей как универсальный инструмент. Иными словами, она перестаёт справляться с другими задачами, которые раньше умела выполнять. Поэтому под каждую задачу, которую вы хотите закрыть с помощью модели, вам придётся поддерживать отдельную дообученную версию — или, если используется LoRA, отдельный адаптер под каждую ключевую задачу.

InstructLab

Это open source-подход к дообучению, разработанный Red Hat специально для того, чтобы можно было коллективно встраивать закрытые корпоративные знания в LLM и при этом сохранять её универсальные возможности.

Методика настройки LLM с открытым исходным кодом под названием InstructLab была с самого начала создана для того, чтобы устранить сложности, с которыми сталкиваются специалисты по ИИ, стремящиеся адаптировать и внедрять LLM под конкретные задачи бизнеса. InstructLab не только упрощает настройку модели на данных конкретной предметной области, но и ставит целью сделать участие в развитии LLM таким же доступным, как участие разработчиков в любом другом open source-проекте. InstructLab появился как попытка сократить разрыв между тем, как работает open source в программной инженерии, и как это происходит в мире open source ИИ. Сейчас у InstructLab есть как открытая версия, так и корпоративное решение, поддерживаемое Red Hat.

Цель InstructLab — сформировать будущее генеративного ИИ, предоставляя структуру, с помощью которой команды и сообщества могут в доступной форме вносить знания и навыки в существующие LLM. В основе InstructLab лежит оригинальный метод выравнивания модели под названием LAB — Large-scale Alignment for chatBots.

Сегодня существует множество сообществ, активно развивающих и расширяющих модели ИИ с открытыми лицензиями. Но почти все они сталкиваются с тремя серьёзными препятствиями — теми самыми, которые традиционный open source давно научился преодолевать.

Во-первых, нет возможности напрямую вносить вклад в базовую LLM. Все улучшения появляются в виде "форков" — посмотрите вокруг, и вы увидите бесконечное стадо Llama-моделей, дообученных версий оригинальной Llama, которые бродят по миру генеративного ИИ. Это вынуждает вас выбирать среди них модель, которая "более-менее подходит", но которую сложно дальше развивать. Кроме того, такие форки дорого обходятся авторам моделей, ведь когда "родительская" Llama обновляется — как перенести эти улучшения на новую модель? И мы даже не говорим о том, как сложно разобраться, какая из всех этих Llama действительно вам подходит.

Во-вторых, если вы хотите внести свой вклад в модель, входной барьер очень высок. Допустим, у вас появилась отличная идея, вы придумали что-то действительно новое — и это работает. Чтобы продвинуть свою идею, вам нужно научиться делать форк, обучать модель, доводить её до результата. Это требует большого объёма знаний и умений.

В-третьих, нет никакого централизованного управления сообществом, нет лучших практик по проверке, модерации и распространению форков. Видели когда-нибудь, как пятилетние дети играют в футбол? Ну вот и всё.

InstructLab решает эти проблемы. Он даёт инструменты, с помощью которых можно создавать и встраивать в модель новые навыки и знания — без необходимости собирать вокруг себя команду специалистов по ИИ.

Первые шаги с InstructLab

Технология InstructLab даёт возможность базовым моделям с достаточными ресурсами регулярно создавать кастомные сборки — не через полное переобучение, а за счёт "вливания" новых умений и знаний. Это достигается за счёт сочетания трёх ключевых процессов:

- таксономически управляемая методика отбора данных
- генерация синтетических данных
- многоступенчатая методика дообучения, предотвращающая катастрофическое забывание

The Lingo

В мире open source термин upstream означает исходный, главный источник проекта (в нашем примере — оригинальная Llama). Именно там происходит основная работа. Все прочие версии, созданные на её основе, называются форками. Upstream-модель — это главная, авторитетная версия семейства моделей. Если кто-то из разработчиков хочет, чтобы его улучшения попали в основную версию, он должен создать pull request — то есть отправить изменения в основной проект, которые затем должны быть одобрены его сопровождающими.

Этот процесс — важнейшая часть open source: он обеспечивает актуальность основной модели и позволяет использовать улучшения, сделанные в форках и производных версиях сообществом. Более того, он даёт сообществу механизм, с помощью которого можно возвращать пользу «вверх» — в основную модель. По сути, это способ сделать модель вроде Llama в тысячу раз лучше — вместо того чтобы плодить тысячу несвязанных Llama-версий, как в нашей предыдущей аналогии. Команда сопровождающих проекта принимает решения о том, какие изменения достойны того, чтобы попасть обратно в основную модель. Если вы активно участвуете в проекте и вносите много полезных правок, вы со временем можете войти в число тех, кто определяет, куда движется проект (или модель, в данном случае).

Проект InstructLab предоставляет разработчикам инструменты для добавления и объединения новых навыков и/или знаний в любую открытую LLM через GitHub — прямо с их ноутбука.

С помощью проекта InstructLab, представленного на рис. 8.2, команды могут вносить рецепты LAB для новых навыков и/или знаний (то есть ваши корпоративные данные) через pull request в репозиторий InstructLab. Все принятые рецепты затем добавляются поверх выбранной предобученной модели в процессе выравнивания (alignment) сопровождающими проекта InstructLab — будь то публичная модель или внутренняя модель вашей компании.

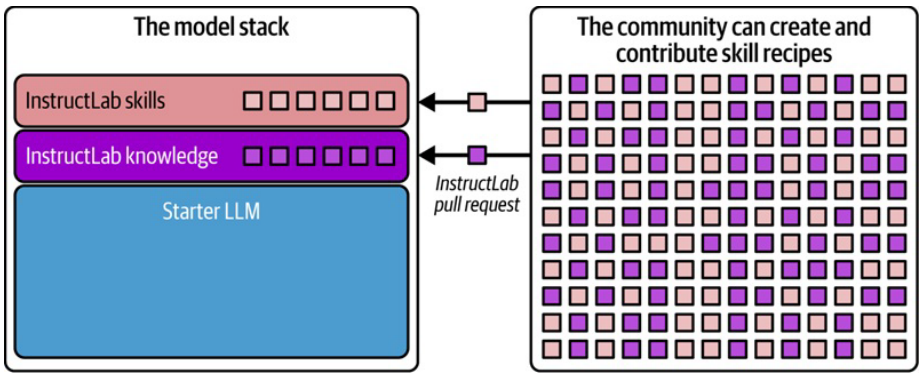

Рис. 8.2. InstructLab предлагает новый способ сделать вклад сообщества накопительным

Предоставление возможности вносить вклад именно на этапе выравнивания, а не тратить ресурсы на предварительное, дорогостоящее обучение новых базовых моделей, открывает путь к гибкому и итеративному процессу разработки. Такой подход отлично подходит как для внутренних команд компании, так и для открытых сообществ — например, отраслевых консорциумов, где бизнесы вместе создают модель под свои специфические нужды. Мы видели это своими глазами. Предобучение LLM может занимать месяцы и требовать тысячи дорогущих GPU, сжигая воду, деньги и время. А вот при помощи InstructLab уже готовую LLM часто можно выровнять под конкретные задачи за день или даже меньше — и это даёт возможность обновлять модель значительно быстрее.

Чуете, что готовится? Рецепты навыков и знаний

В своей основе рецепт навыка или знания — это просто набор инструкций, с помощью которых можно программно сгенерировать большое количество размеченных синтетических данных (то есть ИИ

помогает ИИ), отражающих определённый набор умений или область знаний. Каждый рецепт включает краткое описание пробела в знаниях или навыках, а также пять или более вручную составленных примеров. В случае рецепта знания в качестве входных данных также указывается источник знаний — например, внутреннее пособие компании по льготам для сотрудников в HR-сценарии, если речь идёт об этой теме.

Эти рецепты подаются в виде промпта более крупной модели-наставника (в первой версии InstructLab использовалась модель Mixtral-Instruct), которая затем генерирует большой объём синтетических данных. Почему именно синтетические данные? Это ключевой элемент InstructLab, потому что у большинства компаний нет достаточного количества целевых данных, чтобы обучить (через InstructLab или классические PEFT-подходы) модель масштаба LLM под свои специфические задачи. Синтетические данные позволяют InstructLab превращать большие объёмы неструктурированных корпоративных данных в структурированный датасет, который уже можно использовать для дообучения модели. Как только такие данные сгенерированы, их можно использовать для дообучения LLM и внедрения в неё недостающих знаний или навыков, которые вы хотите включить в модель вашей компании.

Использование синтетических данных для выравнивания модели — идея сама по себе не новая. Существуют и другие примеры, в том числе метод дистилляции моделей, о котором мы говорили в седьмой главе. Например, модель Vicuna-13B была обучена на синтетических данных, сгенерированных GPT-4. Но здесь появляется проблема. Условия использования OpenAI запрещают применять GPT-4 для создания коммерчески конкурентных моделей, что ставит под сомнение жизнеспособность таких решений. Есть и другие примеры, но почти все они завязаны на закрытые модели вроде GPT-4 как на источник синтетических данных. И вот здесь начинается то, что делает open source настоящим двигателем прогресса. LAB-подход особенно интересен тем, что доказывает: модели с открытой и разрешительной лицензией (например, Apache 2.0) вполне могут быть использованы в роли учителя — и при этом обеспечивать уровень качества, соответствующий передовым достижениям в области ИИ.

На сегодняшний день все рецепты навыков и/или знаний, внесённые в проект InstructLab, отображаются в логически выстроенной иерархии, называемой таксономией. Проще говоря, таксономия — это древовидная структура, которая организует информацию по категориям и подкатегориям (см. рис. 8.2). В InstructLab таксономия классифицирует отдельные примеры данных по всё более узким группам, которые в конечном итоге соответствуют конкретным задачам (листьям на ветке). Это даёт разработчикам не только визуальную схему, с помощью которой можно определить, какие навыки и знания могут быть полезны проекту, но и способ обнаружить пробелы и восполнить их новым контентом.

InstructLab учится так же, как учатся люди

В рамках этой книги мы не будем подробно разбирать, как именно устроены знания и навыки внутри InstructLab, но стоит упомянуть один важный момент. Подход InstructLab во многом напоминает то, как учится человек. Например, в его таксономии есть раздел знаний, и — как в жизни — знания можно почерпнуть из книг. Это действительно один из источников для InstructLab. Чтобы выполнять сложные задачи, нам, людям, нужно обладать базовыми навыками, которые можно комбинировать с приобретёнными знаниями. В этом плане InstructLab не отличается.

Например, прежде чем просить ИИ использовать чистую приведённую стоимость (NPV) для оценки инвестиционной привлекательности, модель должна понимать базовую математику — степени, порядок действий, концепцию временной стоимости денег. И, как человек, InstructLab объединяет знания и базовые навыки для выполнения сложных задач — их в этой системе называют композиционными навыками. Если ваша LLM включена в агентную цепочку, в которой нужно написать аналитический отчёт с использованием NPV, она должна уметь всё, о чём шла речь выше: математику, письмо, работу с нюансами — и многое другое.

Таксономия InstructLab также помогает обеспечить разнообразие синтетических данных, генерируемых под каждый рецепт, чтобы покрыть все возможные подзадачи, входящие в рамки той или иной большой задачи.

Представьте себе LLM, которая помогает агенту писать посты для социальных сетей — как в нашем агентном примере из предыдущей главы. То, как вы пишете в X (ранее известном как Twitter),

отличается от LinkedIn или Instagram. На одних платформах требуется короткий формат из-за ограничения по символам, на других больше уместны эмодзи; какие-то платформы ориентированы на изображения, другие требуют делового стиля. Всё это — отдельные письменные навыки, специфичные для соцсетей.

В фрагменте таксономии InstructLab, представленном на рис. 8.3, если участник хочет улучшить способность модели писать посты для соцсетей, он может внести вклад в ветку social_media (или создать её, если она ещё не существует), которая входит в ветку freeform, а та, в свою очередь, — в ветку writing в таксономии навыков. Его вклад будет представлен в виде рецептов синтетических данных для каждой конкретной социальной платформы. Хотите, чтобы ваш ИИ стал поэтом? Дайте ему разные примеры поэзии и создайте отдельные навыки для хайку, сонета, лимерика и так далее.

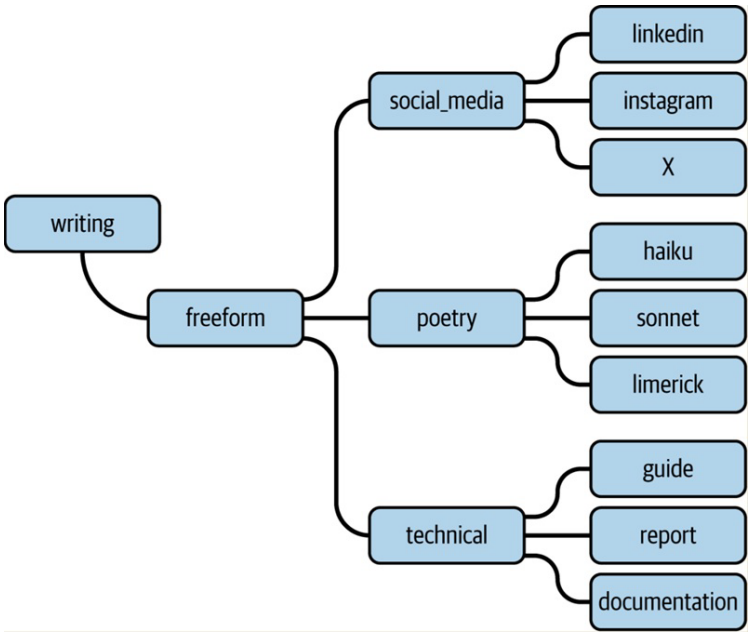

Рис. 8.3. Пример таксономии навыков InstructLab для письменных задач

Уникальный режим обучения LAB внедряет эти новые данные на этапе выравнивания, а не на ресурсоёмком этапе предобучения, на котором большинство LLM получают свои базовые знания и способности. И, что важно, этот протокол обучения также снижает риск катастрофического забывания. Проще говоря, принцип работы InstructLab гарантирует, что новые знания не вытеснят то, чему модель уже научилась раньше.

Когда все рецепты синтетических данных добавлены и включены в таксономию проекта, обучающая и генеративная система InstructLab запускает выполнение всех рецептов для генерации синтетических данных. Затем сгенерированные данные проходят фильтрацию, чтобы отобрать только качественные примеры, и с помощью оригинального поэтапного метода дообучения каждый из стартовых моделей-учеников выравнивается на основе этих данных, тем самым вбирая в себя все внесённые навыки и знания. А поскольку, как говорится, лучше один раз увидеть, мы собрали весь этот процесс на рис. 8.4.

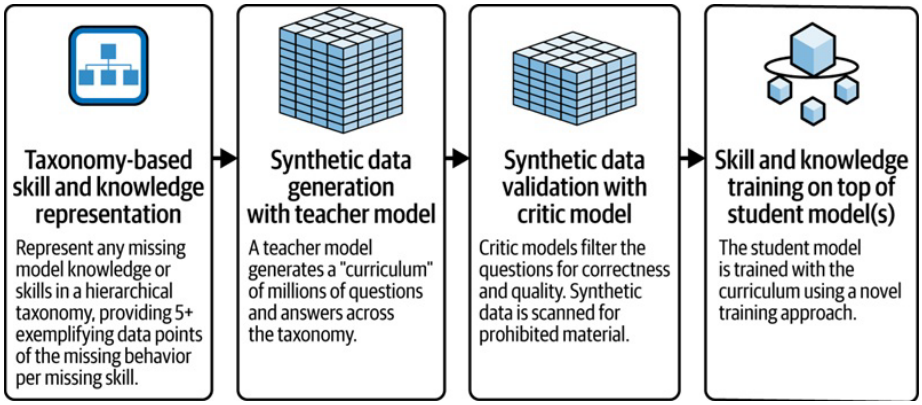

Рис. 8.4. Как работает метод Large-scale Alignment for chatBots (LAB)

Использование силы сообщества

Чтобы ускорить инновации, open source-версия InstructLab перешла на периодический цикл обучения и выпуска моделей, обученных сообществом. Актуальные версии моделей InstructLab публикуются в открытом доступе на платформе Hugging Face, которая, как вы помните из первой части книги, является сердцем крупнейшего в мире сообщества по работе с ИИ. Благодаря охвату Hugging Face, у сообщества появляется возможность скачать настроенную модель InstructLab, поэкспериментировать с ней и выявить пробелы в её работе.

Обнаружив такие зоны, участники могут разработать и внести собственные рецепты навыков и знаний обратно в проект InstructLab через pull request. Как и в других open source-проектах, сопровождающие и активные участники InstructLab рассматривают и объединяют все принятые правки в основную модель раз в неделю. Разумеется, если вы работаете с частной моделью, вы можете выполнять весь этот процесс внутри своей компании в таком же формате.

Чтобы поддержать разработчиков, использующих и развивающих модели InstructLab, в рамках проекта предоставляется интерфейс командной строки под названием Language Model Development Kit (LMDK). LMDK реализует весь процесс InstructLab прямо на ноутбуке участника. Представьте себе пробную кухню, где можно разрабатывать и тестировать новые рецепты генерации синтетических данных, обучающих LLM новым навыкам. Разработчик запускается за считаные минуты — скажем, начинает эксперименты с локальной версией открытой LLM (например, Granite). Находит слабые места в её работе, пишет рецепт навыков или знаний, чтобы устранить этот пробел, и — готово. Весь этот процесс (рис. 8.5) работает как маховик, запускающий быструю инновационную динамику в open source-моделях ИИ.

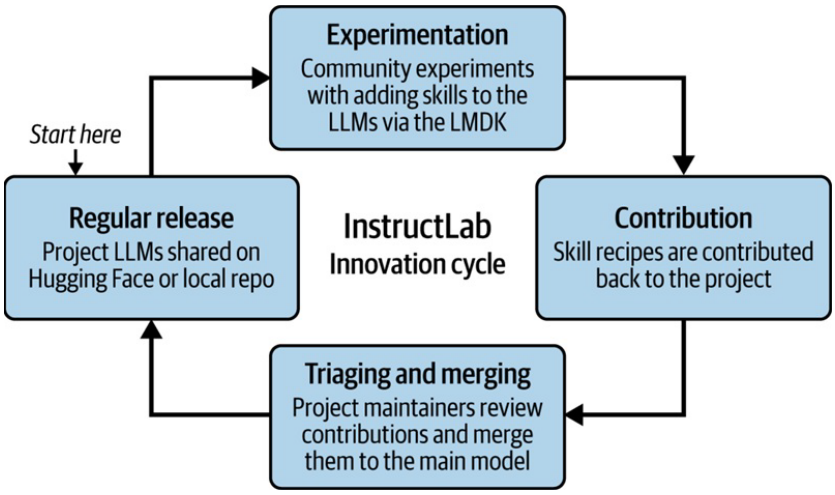

Рис. 8.5. Цикл инноваций InstructLab: маховик для быстрого развития open source ИИ

Один день из жизни участника InstructLab

Мы не будем пошагово разбирать весь процесс InstructLab, но в открытом доступе есть множество подробных руководств, которые быстро превратят вас в уверенного участника. Рис. 8.4 показал, из чего складывается работа участника InstructLab, и, как вы уже поняли, всё начинается с рецепта навыка. Ниже показан пример рецепта для навыка рифмования (он написан на YAML):

```yaml
version: 2
task_description: 'Teach the model how to rhyme.'
created_by: rob-paul-kate
seed_examples:
  - question: "What are 5 words that rhyme with boring?"
    answer: "snoring, pouring, storing, scoring, and exploring."
  - question: "What are 5 words that rhyme with dog?"
    answer: "log, cog, frog, bog, and smog."
  - question: "What are 5 words that rhyme with happy?"
    answer: "snappy, crappy, scrappy, unhappy, and sappy."
  - question: "What are 5 words that rhyme with bank?"
    answer: "shank, crank, prank, sank, and drank."
  - question: "What are 5 words that rhyme with fake?"
    answer: "bake, lake, break, make, and earthquake."
```

Рис. 8.5a. Пример рецепта – поиск рифмы

Затем с помощью локальной версии генератора синтетических данных InstructLab вы создаёте собственный набор данных выравнивания под нужный навык или знание. Эти данные можно использовать для настройки локальной версии модели и быстрой проверки: устранён ли найденный пробел. Вы можете повторять этот процесс, пока модель не начнёт стабильно выполнять нужную задачу. Как только рецепт готов и протестирован через LMDK, вы отправляете его как pull request в таксономию InstructLab на GitHub — так же, как в любом другом open source- или внутреннем проекте. Далее команда сопровождающих либо принимает, либо отклоняет заявку, добавляя новые YAML-файлы в итоговую таксономию. (Этот процесс может проходить как публично, так и полностью внутри вашей компании.)

Завершающий этап InstructLab — это процесс сборки, который можно запускать регулярно, периодически обновляя вашу LLM с учётом, например, новых и ценных вкладов от сообщества разработчиков. В ходе этой сборки все сгенерированные синтетические данные агрегируются и используются в многоступенчатом процессе обучения, направленном на максимизацию качества и снижение таких рисков, как катастрофическое забывание. Когда новая версия модели готова, у вас на руках LLM, адаптированная на основе всех корпоративных данных, предоставленных вашими разработчиками и отраслевыми экспертами.

Хотя InstructLab ещё только в начале пути, уже сейчас видно, что такой сквозной процесс настройки малых моделей на корпоративных данных может не только повысить производительность (а это всегда хорошо), но и существенно сократить затраты по сравнению с использованием одной большой универсальной модели.

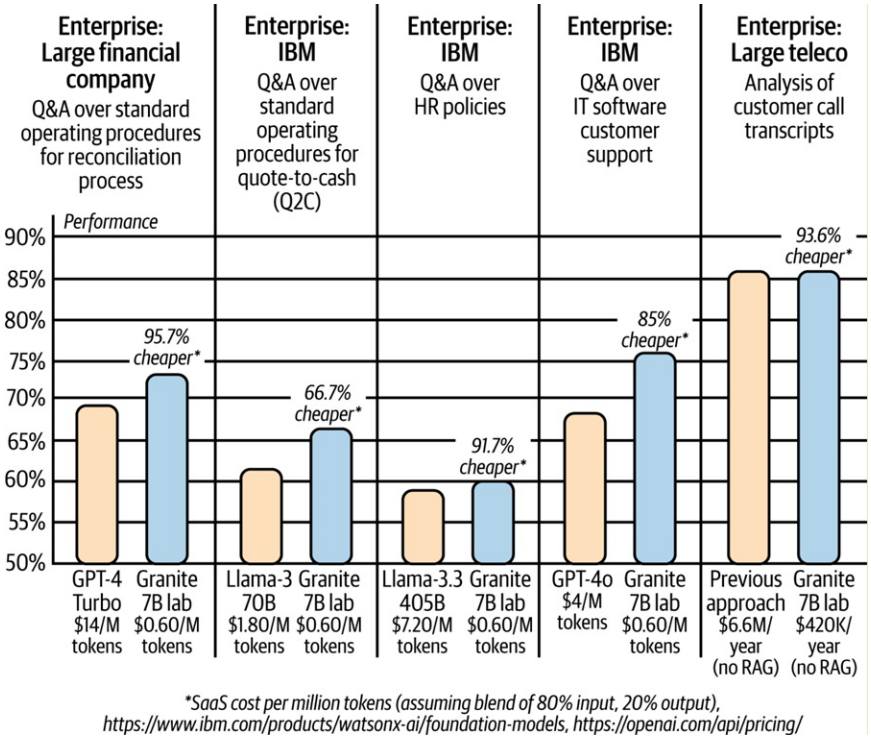

Рис. 8.6. Эффект от использования InstructLab

В сценариях, где задействованы конфиденциальные данные, например, сведения о здоровье сотрудников или дисциплинарные записи, встраивать такую информацию прямо в LLM вряд ли разумно. Вместо этого вы можете адаптировать свою LLM через InstructLab под стиль, ценности и голос вашей компании, а чувствительные данные хранить отдельно — в системе RAG с контролируемым доступом. Такой подход позволяет модели при необходимости безопасно обращаться к конфиденциальной информации, обеспечивая и точность, и защиту данных. То же самое применимо и к случаям, когда ваши данные постоянно меняются или задачам требуется максимально актуальная информация — в этих случаях RAG тоже будет более подходящим решением.

Шаг 3. Финал: развёртывание и эксперименты

Нет смысла в надёжной LLM, обогащённой вашими данными, если никто в вашей компании не может её использовать. Поэтому последний шаг — это развёртывание нового актива по созданию ценности из данных в реальной среде. А что нужно для этого? Много экспериментов. Если вспомнить любую прошлую технологическую революцию (например, интернет), история показывает: всегда есть момент перехода от экспериментов к полномасштабному внедрению.

Сегодня в мире царит огромный энтузиазм и высокие ожидания по поводу генеративного ИИ и агентов. Мы видим приложения и API, которые могут влиять на сотни миллионов пользователей. Возникающий сейчас интерес можно сравнить с появлением интернет-браузеров (тот самый момент с Netscape, о котором мы говорили в первой главе). Но если подумать, реальная ценность интернета для бизнеса раскрылась не в день запуска Netscape. Всё началось, когда интернет объединил всё: от складов до цепочек поставок и интерфейсов для клиентов. Мы уверены, с ИИ произойдёт то же самое: от «плюс ИИ» до «ИИ как основа всего».

Чтобы раскрыть ценность ИИ для бизнеса, необходимо уметь масштабировать развёртывание на весь бизнес. Но прежде нужно создать управляемую среду, в которой можно проводить эксперименты, адаптировать модели с помощью RAG, fine-tuning и InstructLab, а затем переводить эти модели в продуктивную эксплуатацию на уровне всей компании.

Особое внимание стоит уделить тому, что кастомизированные модели теперь представляют собой цифровое воплощение ценных интеллектуальных активов вашей компании. Это значит, что в момент развёртывания вам предстоит принять ряд ключевых бизнес-решений. Можно ли доверить такую модель облаку, или данные, которые она в себе содержит, настолько чувствительны, что её можно использовать только на собственной инфраструктуре? Нужны ли вам те самые защитные рамки — проактивные и реактивные, о которых мы говорили в пятой главе, — чтобы ваши приложения на

основе модели не были использованы во вред? Нужно ли вам в реальном времени отслеживать производительность и безопасность развернутых решений?

И по мере того как генеративный ИИ проникает в разные участки вашей компании, увеличивается и потенциальная зона уязвимости для цифровых атак, так что (опять вспоминая пятую главу) стоит всерьез задуматься об атакующих сценариях и о том, какие новые способы могут использовать злоумышленники, чтобы взломать вашу цифровую модель.

Будущее — открытое, совместное и настраиваемое

Большая часть интернета построена на программном обеспечении с открытым кодом. Каждый день, даже не замечая этого, вы взаимодействуете с операционной системой Linux, а веб-сервер Apache помогает вам выполнять задачи. Сегодня open source также стоит за работой смартфонов на Android и за криптографическим протоколом SSL, который ежедневно защищает миллионы финансовых операций. Мы говорим вам: LLM, созданные открытым сообществом и адаптированные под нужды бизнеса, могут принести те же преимущества.

Когда веса модели LLM публикуются в открытом доступе, каждый получает возможность участвовать в разработке, тестировать, улучшать и формировать будущее этой технологии. Когда разработчикам становится доступна информация о происхождении данных, это повышает доверие и делает модель более понятной.

Прозрачность и открытый исходный код делают системы стабильнее и безопаснее. Это ускоряет и упрощает выпуск новых версий, делает цикл обновлений предсказуемым и способствует созданию более надёжного ИИ-программного обеспечения. Повышение доверия и прозрачности LLM — одна из главных целей проекта InstructLab.

Открытое ПО также способствует здоровой конкуренции, не давая двум-трём компаниям монополизировать отрасль. Когда участие доступно всем, инновации развиваются быстрее, а затраты для конечного пользователя снижаются.

Теперь вы знаете, как превратить свои данные в конкурентную суперсилу. Но прежде чем вы отправитесь покорять рынок (или хотя бы удивлять коллег), давайте на минуту заглянем в наше не подкреплённое ИИ хрустальное будущее (честно говоря, никакого шара у нас нет) и попробуем предположить, какие удивительные приключения ждут постоянно меняющийся мир генеративного ИИ и агентов.