

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет» (ПГУ)

С. В. Рындина

Интеллектуальные информационные
системы и технологии. Большие языковые
модели. Промпт-инжиниринг
и интеллектуальные агенты

Учебное пособие

Пенза
Издательство ПГУ
2025

УДК 004.6(075)

P95

Р е ц е н з е н т ы :

доктор технических наук, доцент,
заведующий кафедрой «Информационно-вычислительные системы»
Пензенского государственного университета

А. В. Кузьмин;

кандидат экономических наук, доцент,
руководитель сектора методологии учета и оформления
банковских операций ПАО Банк «Кузнецкий» (г. Пенза)

О. Б. Волошина

Рындина, Светлана Валентиновна.

P95

Интеллектуальные информационные системы и технологии. Большие языковые модели. Промпт-инжиниринг и интеллектуальные агенты : учеб. пособие / С. В. Рындина. – Пенза : Изд-во ПГУ, 2025. – 92 с.

ISBN 978-5-908089-22-7

Рассмотрены технологии искусственного интеллекта, стандарты в области искусственного интеллекта, промпт-инжиниринг и интеллектуальные агенты, а также их применение для решения различных бизнес-задач.

Издание подготовлено на кафедре «Цифровая экономика» и предназначено для обучающихся по направлениям подготовки 38.04.01 «Экономика», 38.04.02 «Менеджмент», 38.04.03 «Управление персоналом», 38.04.08 «Финансы и кредит», также материал пособия может быть использован для знакомства с инфраструктурой обработки больших данных при изучении дисциплин «Основы анализа больших массивов данных» (направление подготовки 38.03.02 «Менеджмент»), «Бизнес-аналитика на основе больших данных» (направление подготовки 38.03.05 «Бизнес-информатика»), «Интеллектуальные информационные системы» (направление подготовки 38.03.05 «Бизнес-информатика»).

УДК 004.6(075)

ISBN 978-5-908089-22-7

© Пензенский государственный
университет, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА	5
1.1. Нейросети.....	7
1.2. Государственные стандарты в области искусственного интеллекта	19
1.3. Большие языковые модели. Эволюция моделей GPT	29
1.4. Зарубежные конкуренты GPT	37
1.5. Отечественные разработки в области LLM	45
1.6. Малые языковые модели	51
2. ПРАКТИЧЕСКИЕ ПРИЛОЖЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА.....	56
2.1. Промпт-инжиниринг: диалоговые интерфейсы для взаимодействия с моделями	56
2.2. Промпт-инжиниринг: оптимальная структура промпта	62
2.3. Промпт-инжиниринг: техники улучшения качества промпта.....	71
2.4. Интеллектуальные агенты, ИИ-агенты	79
СПИСОК ЛИТЕРАТУРЫ.....	89

ВВЕДЕНИЕ

В соответствии с учебным планом обучающиеся на первом курсе магистерских программ экономического профиля изучают дисциплину «Интеллектуальные информационные системы и технологии».

В пособии рассмотрены цифровые решения, основанные на технологиях искусственного интеллекта (ИИ), основные компоненты экосистемы ИИ: данные, инфраструктура (аппаратная и программная составляющие), модели и методы. Также затронуты вопросы составления качественных промптов/затравок, которые имеют решающее значение при использовании генеративного ИИ.

Представлен обзор отечественных больших языковых моделей. Также рассмотрены наиболее популярные на глобальном рынке большие языковые модели.

В заключение подробно описаны различные виды интеллектуальных агентов, которые являются цифровыми продуктами на основе рассмотренных в пособии технологий ИИ.

1. ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Технологии искусственного интеллекта (Artificial Intelligence, AI) – это совокупность методов, инструментов и процессов, использующихся для разработки компьютерных систем, решение задач которыми схоже с тем, как в аналогичных ситуациях функционирует человеческий интеллект. Фактически такие системы имитируют свойства и принципы работы человеческого интеллекта. К технологиям искусственного интеллекта относят анализ данных, распознавание образов, обработку текстов и запросов, сформулированных на естественном языке, обучение на потоках данных и принятие решений. Технологии ИИ – это мультидисциплинарная система, использующая достижения компьютерных наук, математики, лингвистики и некоторых других. Постепенно в экосистему ИИ втягиваются все новые науки, которые становятся востребованными в ней за счет адаптации новых разработок/методов/моделей к потребностям развития технологий.

Практически всегда, когда речь заходит об искусственном интеллекте, одновременно затрагивают и машинное обучение (Machine Learning, ML), глубокое обучение (Deep Learning, DL), нейросети, которые фактически относятся к технологиям ИИ.

Искусственный интеллект, как определено выше, – это компьютерная система, обладающая способностью рассуждать, действовать как человек, а машинное обучение (ML) – это технологии, в основе которых находится обучение на данных, а результатом такого обучения является способность выполнять определенный круг задач (детектировать объекты, ситуации, явления и т.п.). То есть с помощью данных система получает способность выявлять закономерности в функционировании объектов, их свойствах, в процессах и явлениях, различать или обнаруживать сходство в объектах различной природы (текстах, аудио, изображениях, видео), прогнозировать развитие процессов, явлений и т.п.

В технологиях ИИ и ML очень много общего:

- основаны на данных;
- используются для автоматизации процессов (сводя участие человека в определенных видах деятельности к минимально необходимому);

– способны повышать эффективность применения при расширении объема качественных данных, на которых они обучаются и которые лежат в основе их приобретенных способностей получать целевой результат.

Во многих современных приложениях ИИ и ML используются в тандеме, чтобы задействовать сильные стороны обоих. Например, система ИИ может использовать алгоритмы ML в качестве одного из своих компонентов для решения конкретных задач, например таких, как анализ данных или прогнозирование.

Для технологий ИИ цель можно обозначить как создание автономных систем, способных имитировать человеческое мышление и поведение, решать проблемы, понимать естественный язык, распознавать образы и взаимодействовать с людьми. Для ML цель обычно более утилитарная и специализированная: автоматическое улучшение производительности модели путем анализа данных и выявления скрытых зависимостей.

Технологии ИИ позволяют создавать гибкие и адаптивные системы, способные решать разнообразные задачи, приспосабливаясь к изменениям окружающей среды.

Ключевые особенности таких систем: автономность (способность действовать самостоятельно, без постоянного вмешательства человека), адаптивность (возможность изменять свое поведение в зависимости от ситуации и новых данных), интеллектуальность (умение анализировать ситуацию, планировать и выбирать оптимальную стратегию действий).

Также можно рассматривать ML как одно из направлений ИИ (рис. 1.1).

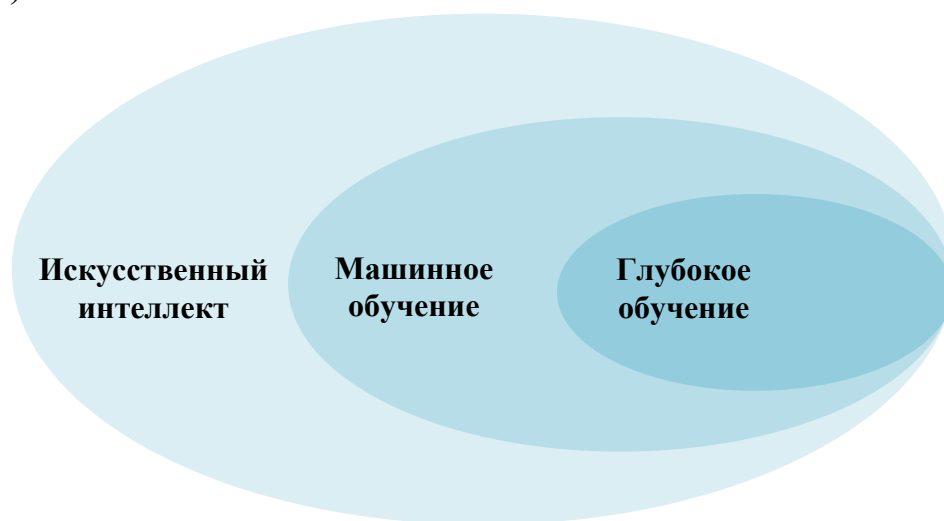


Рис. 1.1. Соотношение между машинным обучением и искусственным интеллектом

Глубокое (иначе глубинное) обучение – это один из видов машинного обучения (см. рис. 1.1). В машинном обучении признаки, необходимые для моделирования, часто выбираются вручную, алгоритмы (например, линейная регрессия, дерево решений, метод опорных векторов, кластерный анализ и др.) применяются для решения конкретных задач, данных требуется меньше, как и вычислительных ресурсов. Глубокое обучение (DL) основано на искусственных нейронных сетях с множеством слоев («глубокая архитектура»), в процессе настройки/обучения способно обнаруживать сложные иерархии признаков. Для обучения используются большие объемы данных высокой размерности (с большим числом признаков, с возможностью их выделения в сырых/первичных данных, которые в процессе обучения выстраиваются в иерархическую структуру), требовательны к вычислительным ресурсам, так как обучение основано на использовании различных типов нейронных сетей, например рекуррентные нейронные сети (RNN), трансформеры (Transformer).

Основное различие между машинным обучением и глубоким обучением наблюдается в уровне абстракции и сложности решаемых задач при обработке данных. В простых задачах, когда есть структура в данных, определены конкретные признаки, используется машинное обучение. Для сложных задач со скрытыми закономерностями, неявными структурами и паттернами в данных оптимальным будет использование глубокого обучения. Глубокое обучение – это продвинутый инструмент машинного обучения, который в отличие от традиционных методов машинного обучения (МО) способен решать задачи повышенного уровня сложности.

Ключевой этап в процессе применения технологий ИИ – когда обученная модель начинает использоваться для обработки новых данных и выдачи предсказаний.

1.1. Нейросети

Передовые модели ИИ – это программные решения на основе нейросетей.

В первом приближении можно рассматривать нейросеть как структуру, состоящую из искусственных нейронов, объединенных в слои, последовательно пропускающие через себя данные, которые подаются на вход, а на выходе получается результат (рис. 1.2).

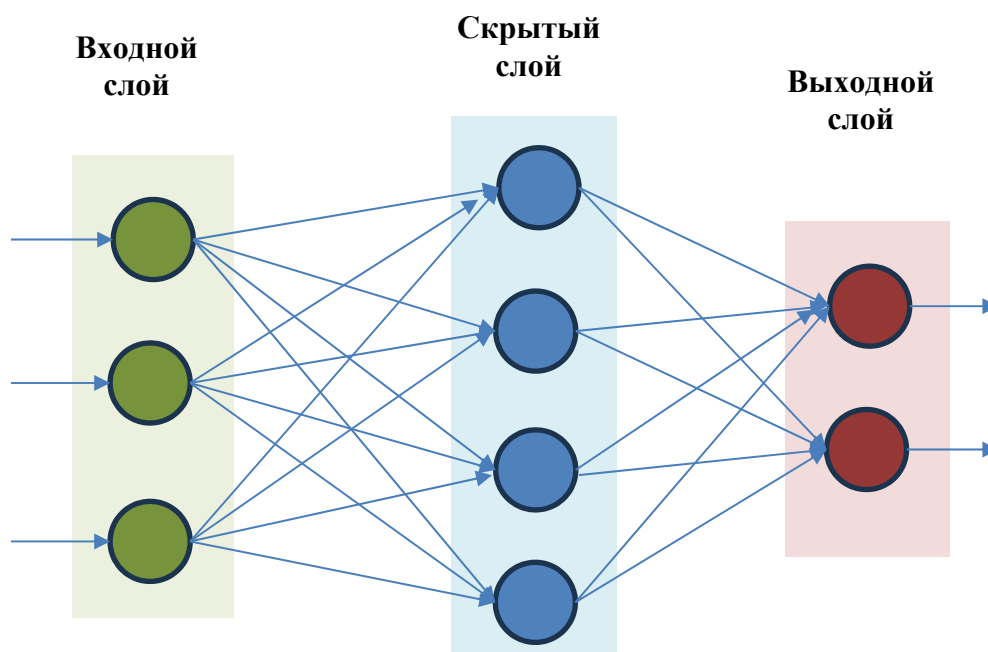


Рис. 1.2. Простейшая архитектура нейросетевой модели

Выходной слой – результат работы предварительно обученной нейросети, позволяет по входным данным определять значение отклика/выхода, который обычно имеет некоторую практическую ценность, например позволяет по данным датчиков получить прогноз погоды (рис. 1.3).

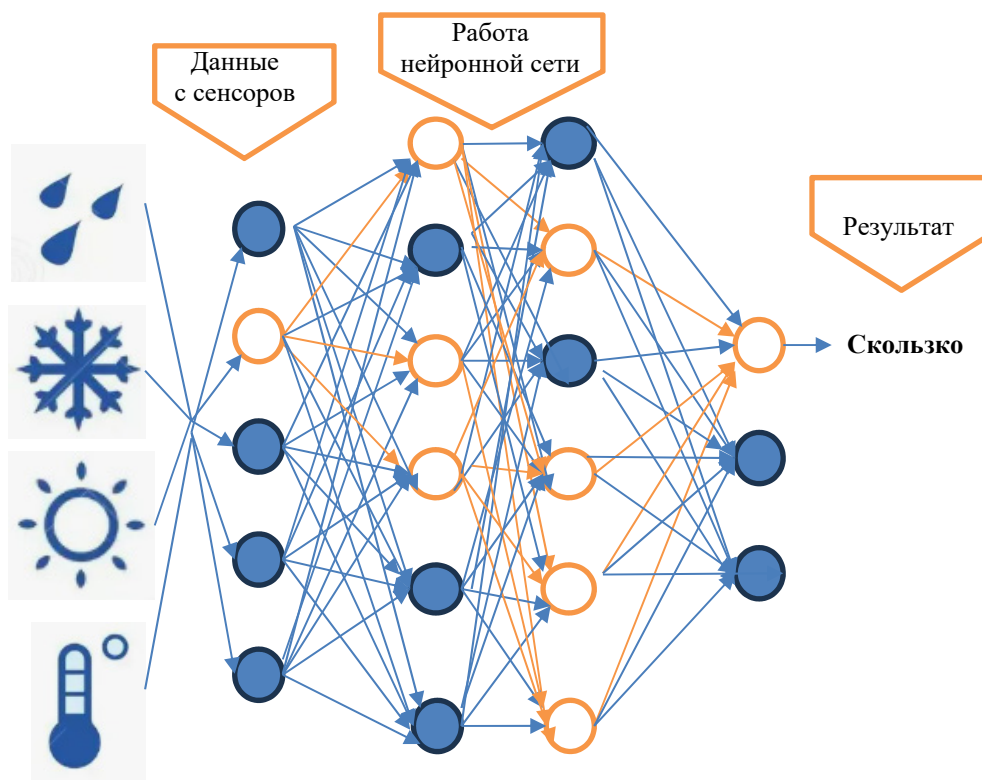


Рис. 1.3. Принцип работы нейросети

С развитием технологий искусственного интеллекта появляются новые архитектуры нейросетей.

Рассмотрим наиболее востребованные нейросети, которые наиболее часто используются в продуктовой разработке в настоящий момент.

Сверточные нейронные сети.

Сверточные нейронные сети (Convolutional Neural Networks, CNN) разработаны для того, чтобы работать с двумерными данными (такими как изображения, звук, пространственные данные), извлекая важную информацию, сохраняя иерархичность признаков и экономя ресурсы.

Типичная сверточная сеть состоит из нескольких слоев, организованных в каскад (рис. 1.4):

1. Convolutional layer (сверточный слой), задача которого применить фильтры (ядра свертки) к входным данным, чтобы выявить низкоуровневые признаки (границы, линии, углы и т.д.). Фильтры скользят по изображению, формируя карту признаков (feature map), отражающую места обнаружения искомых признаков. Фильтры формируются автоматически во время обучения.

2. Pooling (пуллинг/свертка) выполняет субдискретизацию карты признаков (т.е. функцию пространственного понижения размерности данных), уменьшая размеры карт, сохраняя самые важные признаки. Этот слой необходим для сокращения количества параметров, снижения риска переобучения, а также повышения устойчивости к небольшим изменениям масштаба и положения объекта.

3. Fully Connected layer (полносвязный слой) выполняет классификацию или вывод по результатам предыдущих этапов.

4. Activation Function (функция активации) используется для придания нелинейности сети.

5. Softmax (многопеременная логистическая функция) часто используется в последнем слое для задач классификации и преобразует выход в вероятностное распределение по категориям.

К типичным вариантам применения CNN относятся распознавание лиц и эмоций, детекция объектов и сегментация изображений.

CNN – важная составляющая современной экосистемы ИИ. Современные исследования стремятся преодолеть их ограничения (высокую потребность в больших объемах размеченных данных и привлечение значительных вычислительных ресурсов), предлагая гибридные подходы и новые виды архитектуры, расширяющие возможности сверточных сетей.

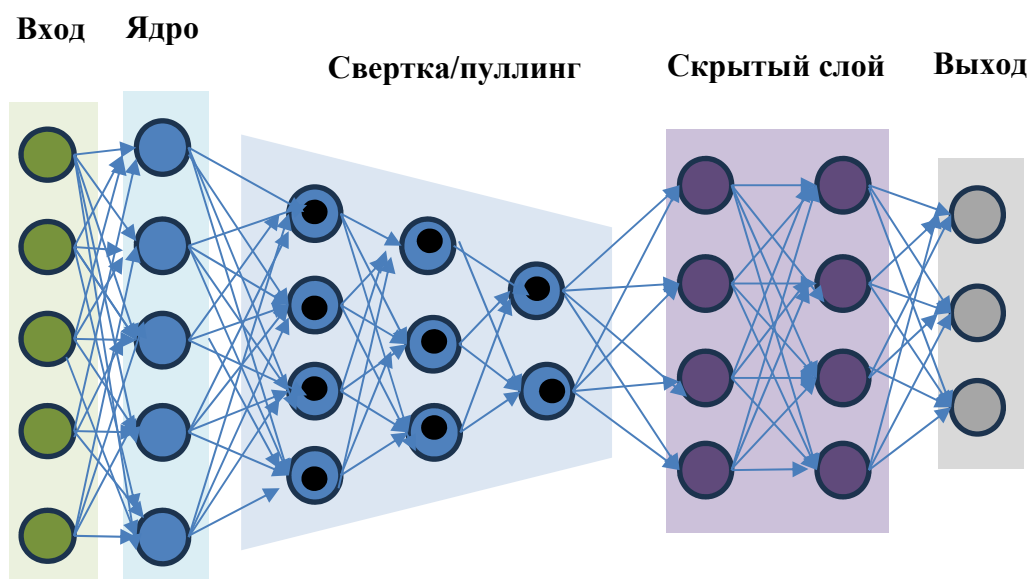


Рис. 1.4. Архитектура CNN

Генеративно-сопоставительная сеть.

Генеративно-сопоставительная сеть (Generative Adversarial Networks, GAN) – архитектура глубокого обучения, состоящая из двух нейронных сетей, обучающихся в конкурентных условиях.

GAN состоит из двух частей (рис. 1.5):

1) генератора (Generator), нейросети, которая создает синтетические экземпляры данных (тексты, изображения, звуки), близкие к настоящим. Обычно процесс начинается с небольшого набора случайных чисел (шума), из которого строится сложная структура (например, изображение). Чем сложнее устроен генератор, тем качественнее получаются результаты. Цель генератора – максимизировать шансы обмануть дискриминатор, создавая как можно более похожие на оригинал фейковые данные;

2) дискриминатора (Discriminator), нейросети, которая оценивает данные, поступающие на вход, стремясь отличить оригинальные данные от созданных генератором. Чаще всего дискриминатор реализуют как обычную многослойную нейросеть или сверточную нейросеть (CNN), завершающуюся выходом Softmax (в этом случае она является сигмной, так как классов всего два – ложь/истина), дающей вероятность, что образец настоящий. Цель дискриминатора – минимизировать шансы принятия сгенерированных образцов за настоящие.

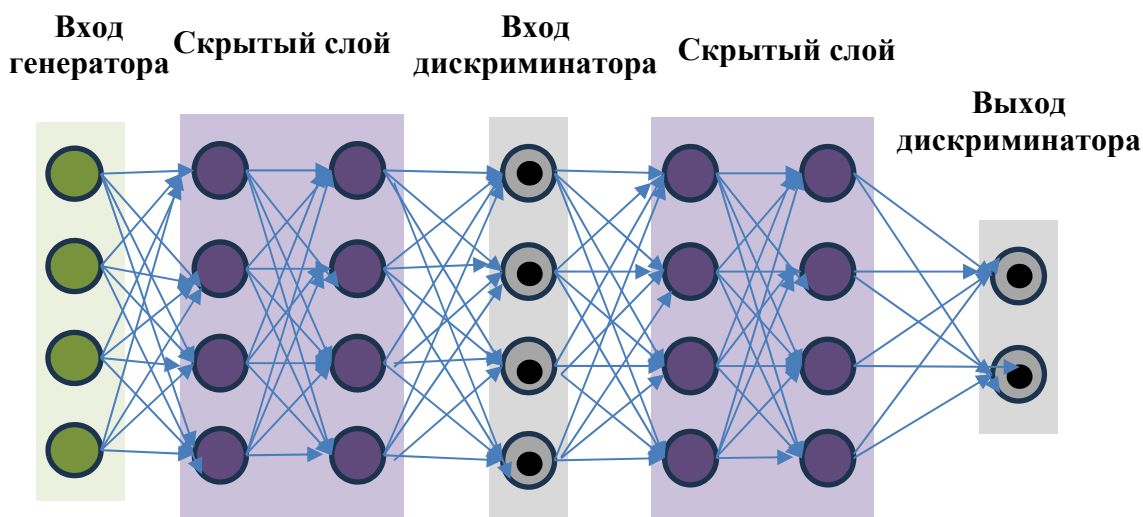


Рис. 1.5. Архитектура GAN

Обучение GAN проходит циклически:

1. Генерируется пакет случайных образцов.
2. Генератор создает фейки на основе этих образцов.
3. Настоящие и поддельные данные перемешиваются и отправляются в дискриминатор.
4. Оценка потерь (loss function) вычисляется отдельно для обеих сторон.
5. Параметры обновляются, стремясь улучшить успех каждой стороны.

Обучение останавливается, когда достигнуто равновесие Нэша (никто не может улучшить свою позицию, меняя стратегию единолично).

Диффузионные модели.

Генеративные модели на основе диффузии (Diffusion model) основаны на принципе обратимости процесса распространения шума в изображении или звуке. Изначально сигнал зашумляется, затем модель учится восстанавливать чистый сигнал обратно.

В физике диффузией называется процесс случайного перемешивания или распространения частиц. Эта идея легла в основу нового вида нейросетей.

Процесс работы диффузионных моделей включает два этапа (рис. 1.6):

1. Forward Diffusion Process – процесс прямой диффузии, на котором берется исходная картина или объект и постепенно добавляется шум, пока изображение не становится практически неразличимым.

В этом процессе каждый следующий шаг (т.е. результат добавления шума) зависит только от предыдущего, при этом шум добавляется контролируемым, инкрементальным (увеличивающимся постепенно) способом с использованием цепи Маркова.

Цепь Маркова и есть математическое описание процесса постепенного добавления к изображению шума, так как цепь Маркова – это последовательность случайных событий с конечным или счетным числом исходов, где вероятность наступления каждого события зависит только от состояния, достигнутого в предыдущем событии.

2. Reverse Diffusion Process – процесс обратной диффузии. Он обратен первому. Модель пытается восстановить первоначальное изображение, начиная с полностью зашумленного состояния, и, пошагово удаляя шум, постепенно возвращается обратно к оригиналу. В процессе обучения нейронной сети она тренируется предсказывать количество шума, которое было добавлено на каждом предыдущем шаге, и постепенно удалять его слой за слоем.

Обратный процесс – это также цепь Маркова. Процедура напоминает принцип «постепенной реконструкции»: нейросеть учится понимать, какой именно части случайного шума соответствует информация об изображении, позволяя ей преобразовать бессмысленный шум в четкое изображение. Так модель осваивает процесс восстановления.

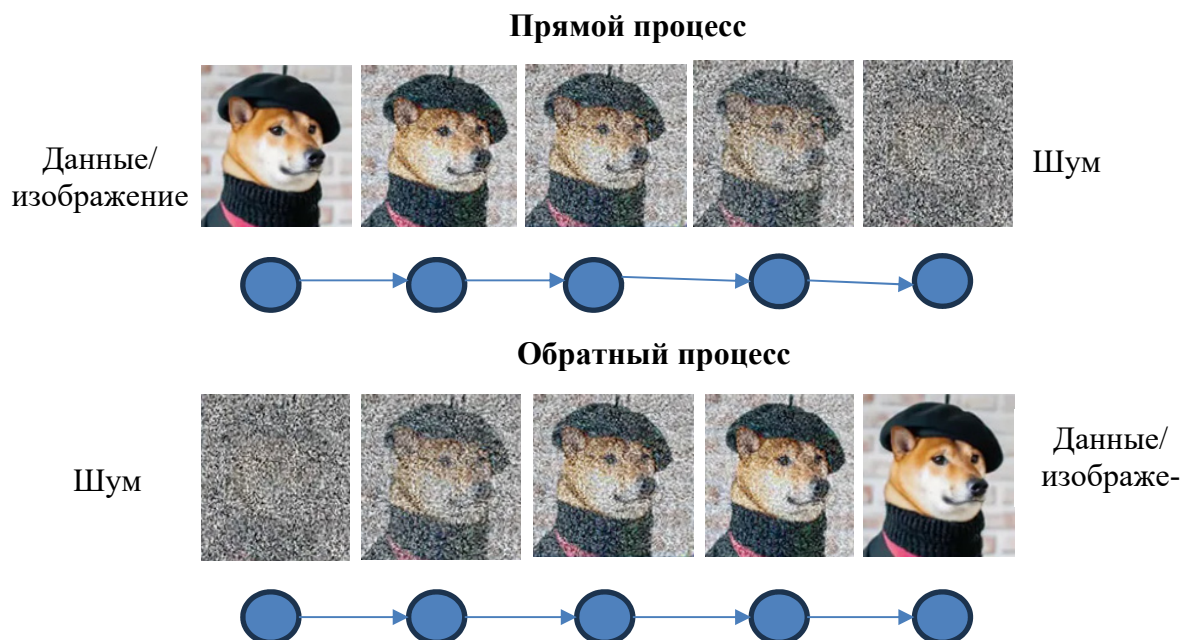


Рис. 1.6. Процессы прямой и обратной диффузии

Диффузионные модели обладают рядом преимуществ:

- высокое качество. Диффузионные модели способны генерировать изображения высокого разрешения и качества, часто неотличимые от реальных фотографий;
- контроль над результатом. Благодаря поэтапному процессу обучения такие модели позволяют контролировать финальный результат, включая изменения стиля, деталей и общей композиции;
- широкая область применения. Помимо визуального искусства, диффузионные модели используются для синтеза музыки, речи и даже кодирования видео.

Дополнительный методы, которые позволяют повысить качество работы модели: classifier-guided guidance и classifier-free guidance.

Classifier-guided guidance (руководство классификатором) – это метод, применяемый в рамках диффузионных моделей для повышения точности и управляемости процесса генерации объектов (например, изображений).

Метод состоит из двух компонент: диффузной модели и классификатора.

Основной механизм, ответственный за создание изображения из случайного шума, неизменен – это Diffusion Model, вспомогательная модель Classifier оценивает вероятность принадлежности текущего шага генерации определенному классу (например, животное, пейзаж, автомобиль). Диффузионная модель учитывает сигнал от классификатора, стараясь минимизировать ошибку классификации. Если классификатор возвращает ошибку, то на следующем шаге диффузионная модель старается скорректировать изображение так, чтобы вернуться на правильный путь, т.е. приблизить изображение к желаемому.

Применение classifier-guided guidance помогает существенно повысить точность генераций, особенно в случаях, когда требуемые классы имеют сложную структуру или детализацию. Например, такая техника может использоваться для генерации портретов конкретных знаменитостей, ландшафтов редких мест или даже фантастических существ, минимизируя риск появления артефактов или ошибок в процессе.

Метод также активно используется в исследованиях, связанных с разработкой решений для мультимодальной генерации, перевода текста в картинку и улучшения качества существующих алгоритмов компьютерного зрения.

Classifier-free guidance («свободное руководство без классификаторов») – это метод, позволяющий улучшать управляемость и качество генерации в диффузионных моделях без использования отдельного

классификатора. Вместо того чтобы полагаться на внешний классификатор для оценки вероятности принадлежности создаваемого изображения к нужному классу, этот метод основывается непосредственно на самом диффузионном механизме, встроенном в архитектуру модели.

Вместо разделения процесса генерации на две независимые компоненты (диффузию и классификацию) в подходе *classifier-free guidance* весь контроль осуществляется самой моделью за счет введения дополнительного параметра, называемого *guidance scale*, величина которого регулирует баланс между двумя основными режимами работы:

- безусловной генерацией, когда модель сосредотачивается только на воспроизведении структуры и статистики данных, задача модели сводится лишь к созданию правдоподобного образа;

- условной генерацией, когда задается целевой класс и модель должна создать изображение именно этого класса. Условность означает, что модель принимает дополнительный входящий сигнал (текстовую подсказку, метку класса и т.п.) и адаптирует генерацию согласно этому условию.

Guidance scale управляет степенью влияния условий на итоговый результат. Чем выше значение этого параметра, тем сильнее проявляется влияние условия (целевого класса), а значит, генерация становится более точной и контролируемой. Низкий *guidance scale* делает генерацию менее зависимой от целей и ближе к чисто статистическому воспроизводству данных.

Преимуществом подхода является отсутствие зависимости от внешнего классификатора, что сокращает вычислительные затраты и повышает стабильность системы. Классификация зачастую требует дополнительной тренировки отдельной модели, которая может привести к дополнительным ошибкам и снижению производительности. *Classifier-free guidance* устраняет необходимость в таком дополнительном компоненте, делая систему проще и эффективнее.

Рекуррентные нейронные сети.

Рекуррентные нейронные сети (Recurrent Neural Network, RNN) предназначены для обработки последовательных данных, таких как речь, текст, временные ряды (акции, погода) и многое другое. Основное отличие RNN от классических нейронных сетей состоит в том, что они сохраняют память о прошлых событиях, используя скрытые состояния (*hidden states*), позволяющие учитывать предыдущие элементы последовательности при обработке текущих.

RNN используют свою внутреннюю память, хранящую ранее обработанные данные, чтобы правильно интерпретировать каждый последующий элемент.

Основные характеристики RNN:

1. Память. Классические нейронные сети обрабатывают каждую точку данных отдельно и изолированно друг от друга. RNN же обладают памятью, которая сохраняет информацию о предыдущих элементах последовательности и учитывает ее при анализе последующих.

2. Обратные связи. Нейроны в обычных сетях связаны только вперед (слои последовательно передают сигналы следующему слою). В RNN выход одного слоя соединен с самим собой через петлю обратной связи, что позволяет сохранять состояние и передавать его дальше (рис. 1.7).

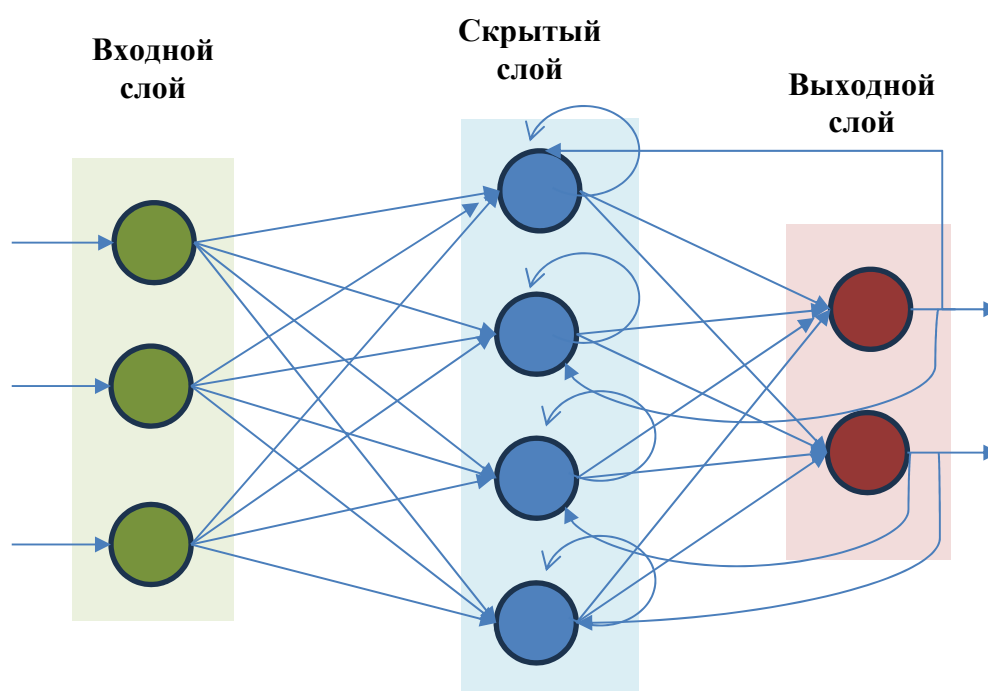


Рис. 1.7. Обратные связи в рекуррентных нейронных сетях

3. Способность обрабатывать переменную длину входных данных. Большинство классических нейросетей требуют фиксированного размера входных данных. RNN прекрасно справляются с задачей обработки последовательностей произвольной длины, будь то предложение из нескольких слов или длинные литературные произведения.

К типичным областям применения RNN относятся:

– обработка естественного языка (Natural Language Processing, NLP). Анализ текстов, понимание контекста предложений, генерация

разговорных диалогов и автоматическая обработка отзывов пользователей;

- машинный перевод. Перевод текста с одного языка на другой, учитывая порядок слов и грамматику обоих языков;

- распознавание речи. Понимание устной речи и преобразование ее в письменную форму;

- генерация музыки. Композиция мелодий и ритмических структур, основанных на музыкальном стиле;

- прогнозирование временных рядов. Прогнозирование поведения покупателей/пользователей/клиентов по историческим данным (по прошлым данным прогнозируется сумма покупок, длительность пребывания в сервисе в следующий период и т.п.), прогнозирование котировок ценных бумаг и иных динамических процессов, развивающихся во времени.

У рекуррентных нейронных сетей есть не только преимущества, но и определенные ограничения: из-за сложной архитектуры обратных связей обучение глубокой RNN затруднено, так как небольшие ошибки быстро затухают при распространении назад по цепочке вычислений, ограниченность памяти (рост обратных связей между элементами имеет естественные лимиты), сложный характер передачи обратных сигналов делает обучение медленным.

Сильная сторона RNN – качество выполнения задач по обработке текстов или машинному переводу достигается за счет роста вычислений, больших требований к инфраструктуре и низкой скорости обучения.

Для преодоления проблем RNN были разработаны различные модификации GRU (Gated Recurrent Units) и LSTM (Long Short-Term Memory).

В LSTM (Long Short-Term Memory, длинная цепь элементов краткосрочной памяти) для управления потоком информации используются затворы/ворота: вход (какая новая информация будет сохраняться в ячейке состояния), забыть (какую прошлую информацию сохранить, а какую забыть), выход (какую часть содержимого ячейки передать на выходной уровень); отдельная ячейка состояния (cell state), где хранится долговременная информация, и отдельное скрытое состояние (hidden state), которое передается далее в сеть. Ячейка состояния защищена специальными вентилями, что предотвращает потерю важных данных при передаче сигналов.

GRU (Gated Recurrent Unit, Закрытый рекуррентный модуль) – это упрощенная версия LSTM, предлагающая более эффективную

реализацию механизмов сохранения и обновления состояний, так как для управления потоком информации используются только два типа затворов/ворот: обновить (какая информация сохраняется из предыдущего скрытого состояния и какая обновляется новой информацией) и сбросить (какая доля прошлой информации игнорируется при расчете нового скрытого состояния); одно общее представление объединяет ячейку состояния и скрытое состояние, т.е. хранение и передача информации происходят совместно.

Кроме модификаций у RNN есть и очень востребованная альтернатива – трансформер.

Трансформеры.

Трансформеры (Transformer) появились как решение проблем RNN, связанных с долговременными зависимостями и последовательностью обработки. Они используют механизм внимания (attention mechanism), позволяющий каждому элементу выборочно фокусироваться на важных частях ввода независимо от порядка следования.

Основная концепция трансформера заключается в полном отказе от рекурсивных и сверточных слоев, используемых в предшествующих архитектурах, вместо них используется самовнимание (self-attention), позволяющее каждому элементу последовательности учитывать взаимосвязи с остальными элементами.

Архитектура трансформера состоит из двух ключевых частей: энкодера (encoder) и декодера (decoder).

Энкодер (Encoder) предназначен для преобразования входящей последовательности во внутреннее представление. Каждый блок энкодера состоит из Self-Attention Layer (слоя самовнимания, также называемого внутренним вниманием), устанавливающего отношения между различными позициями внутри одной последовательности, и Feed Forward Network, представляющей собой простую полносвязную нейронную сеть, выполняющую дополнительные нелинейные преобразования данных.

Каждый блок дополняется нормализацией (Layer Normalization) и механизмом пропуска (Residual Connection), позволяющим ускорить обучение и стабилизировать распространение градиента.

Декодер (Decoder) предназначен для генерации выходных последовательностей. Его блоки включают Masked Self-Attention Layer (замаскированный слой самовнимания), который аналогичен self-attention в энкодере, но с добавлением маски, предотвращающей доступ каждой позиции к будущим токенам, что важно для задачи автоматического порождения текста, Enc-Dec Attention Layer (слой

внимания кодировщика и декодера) позволяет декодеру получать дополнительную информацию от энкодера, обеспечивая взаимодействие между исходной и порождаемой последовательностями, и Feed Forward Network, представляющую собой простую полносвязную нейронную сеть, выполняющую дополнительные нелинейные преобразования данных. Также присутствует нормализующий слой и пропускной механизм, аналогичные энкодеру.

Так как самый важный концепт таких нейросетей – это самовнимание, остановимся подробнее на принципах его работы.

Данные, которые подаются на вход нейросети, имеют специальное представление: эмбединг (embedding, вложение/встраивание), когда каждая единица текста (токен) в предложении представлена специальным вектором чисел. Токен в нейросети – это минимальная единица текста, которую модель использует для обработки и генерации, очень часто токен меньше слова, т.е. одно слово может быть представлено несколькими токенами. В разных реализациях трансформеров свои подходы к токенизации и количество токенов, а также принцип разбиения на токены могут значительно отличаться.

Каждому токену присваиваются три вектора, которые используются для анализа взаимосвязей:

1. Query (Запрос) – это своего рода интерес слова к другим словам в предложении.
2. Key (Ключ) – это своеобразный ответ слову-запросчику. Ответ дает понять, насколько сильно слово отвечает на тот или иной запрос.
3. Value (Значение) – это фактическая информация (значение и роль в предложении, которую несет слово).

На основании этих векторов токены сравниваются и для каждого токена вычисляется степень его важности относительно других токенов. Итоговый вывод для каждого токена формируется на основе взвешенной суммы всех значений в последовательности (рис. 1.8).

Самовнимание позволяет каждому токену «видеть» всю последовательность целиком, уделяя особое внимание релевантным частям.

Есть и более сложный вариант реализации концепции самовнимания, который реализован в некоторых трансформерах, – multi-head attention (многоголовое внимание). Вместо одного блока внимания используется несколько параллельно работающих голов (heads), каждая из которых фокусируется на разных аспектах взаимодействия между элементами. Выходящие представления комбинируются для формирования комплексного и многомерного представления. Multi-head attention добавляет глубины и богатства представлению, позволяя охватывать разнообразные аспекты.

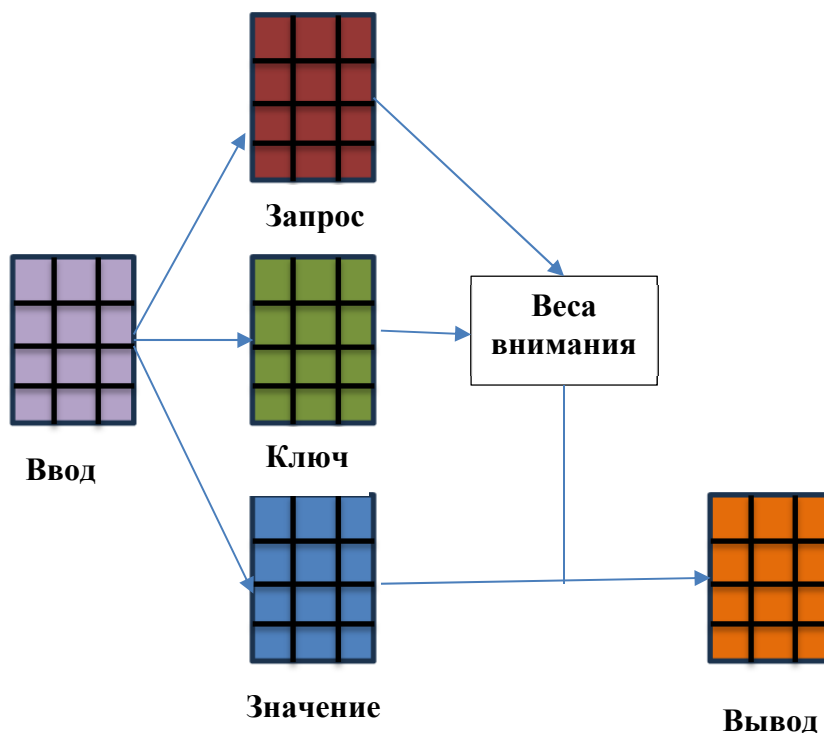


Рис. 1.8. Механизм реализации слоя самовнимания в трансформерах

В трансформерах для токенов также используются позиционные эмбединги (Positional Embeddings). Поскольку трансформеры не содержат информации о порядке следования элементов (они не рекурсивные и не сверточные), вводится дополнительное представление позиционной информации через встраивание координат (position embeddings). Эти координаты добавляются к стандартным эмбедингам токенов, давая модели знания о положении каждого элемента в последовательности.

Трансформеры отлично зарекомендовали себя при решении задач по обработке текстов или машинному переводу, генерации текстов, обработке изображений, обработке аудио и речи. Спектр применения трансформеров постоянно расширяется, так как они отличаются гибкостью и универсальностью.

1.2. Государственные стандарты в области искусственного интеллекта

В последние годы технологии искусственного интеллекта в фокусе внимания государственной политики, что проявляется в том числе в виде разработки государственных стандартов (ГОСТов) в области ИИ.

Стандарты устанавливают требования к качеству продукции, процессов, работ и услуг в различных сферах деятельности.

ГОСТы в области ИИ – это добровольные государственные стандарты, применение которых имеет рекомендательный характер, они служат для достижения следующих целей:

- повышение качества, улучшение характеристик систем ИИ, призванное повысить конкурентоспособность приложений на основе технологий ИИ;

- оптимизация процессов, обеспечивающих функционирование экосистемы ИИ: снижение затрат и повышение эффективности;

- соответствие мировым практикам, когда соответствующий ГОСТ основан на международных стандартах / использует их, чтобы решения отечественных разработчиков могли конкурировать на глобальном рынке;

- укрепление репутации, привлечение инвестиций, так как компаниям следующим ГОСТам больше доверия со стороны потребителей и партнеров, а также инвесторов.

Начнем рассмотрение с ГОСТ Р 59276–2020, устанавливающего общие положения относительно способов обеспечения доверия к системам ИИ [1]. Системы ИИ были выделены в отдельный класс программного обеспечения, а стандартизация стала полнее учитывать специфические черты разработки и эксплуатации таких систем.

Стандарт направлен на повышение доверия пользователей и регуляторных органов к системам ИИ путем подтверждения их качества и надежности. Важность обеспечения доверия особенно значима в сферах, где последствия неправильных решений могут повлечь угрозу здоровью и жизни людей, экологическому и экономическому ущербу (например, медицинская диагностика, транспорт, автоматизация производства). Уровень доверия критично зависит от качества систем ИИ, оцениваемого с помощью показателей эффективности, надежности, защищенности, практичности и мобильности. Особое внимание уделено таким этапам жизненного цикла систем ИИ, как создание (формирование требований, разработка концепций, моделей и производственный процесс), а также эксплуатация (использование, сопровождение и вывод из эксплуатации). Рассматриваются факторы риска, снижающие качество систем, сопровождающие каждый этап (ошибки моделирования, недостаточное тестирование, небезопасность данных и т.п.).

В стандарте выделяются три архитектурных уровня систем ИИ:

- физический (сенсорные устройства и исполнители);
- инфраструктурный (аппаратные ресурсы, хранение и обработка данных);
- прикладной (программное обеспечение, алгоритм анализа данных).

Обеспечение доверия должно учитывать каждый из этих уровней, устанавливая соответствующие критерии и методы проверки.

Для повышения доверия рекомендуется использовать следующие методы:

- установление адекватных критериев качества и показателей доверия;
- регулярную проверку и мониторинг системы;
- предупреждение рисков путем контроля и минимизации негативных воздействий.

ГОСТ Р 59276–2020 формирует основы для регулирования и оценки систем ИИ, обеспечивая необходимую степень доверия к таким технологиям в реальных практических приложениях.

ГОСТ Р ИСО/МЭК 24668–2022 посвящен организации эффективной аналитической работы с большими данными на предприятиях различного профиля. Основной акцент сделан на построении надежной инфраструктуры и организации управленческих процессов, которые позволяют эффективно собирать, обрабатывать и анализировать большие объемы разнородных данных для оптимизации бизнеса и принятия взвешенных решений [2].

В данном стандарте рассмотрена классификация больших данных и те их характеристики, которые делают традиционные методы обработки данных в этом случае неэффективными. Описаны процессы управления данными, которые позволяют обеспечивать их высокое качество и защиту. Проведен базовый анализ компонентов инфраструктуры больших данных, обоснована необходимость подбора оптимальных инструментов и оборудования для обработки и хранения огромных объемов данных, необходимость интеграции различных систем и сервисов для полноценной работы с большими данными, а также обеспечение безопасного обмена данными между ними. Рассмотрены важные для развития аналитики данных процедуры, включающие правильное планирование и постановку целей, методики внедрения аналитических проектов, идентификацию рисков и контроля качества выполняемых задач. Определены методы и показатели, которые позволяют проводить оценку эффективности процессов аналитики больших данных и улучшать их.

Стандарт служит основой для построения надежных и эффективных процессов аналитики больших данных в российских компаниях, обеспечивая качественное управление данными и аналитикой в современном цифровом мире.

ГОСТ Р 71476–2024 посвящен искусственному интеллекту и обеспечивает единое понимание и эффективное применение технологий ИИ в различных сферах деятельности за счет стандартизации терминологии и концепций в области ИИ [3].

Стандарт устанавливает единую терминологию и включает определения ключевых терминов, таких как «система ИИ», «обучение с учителем», «глубокое обучение», «нейронные сети» и др. Рассматривает важность больших данных, облачных и периферийных вычислений, включенных наряду с программным обеспечением, аппаратными ресурсами и методами обучения в экосистему ИИ. В стандарте обозначены отличия между традиционным программированием и подходами на основе машинного обучения. Также определены этапы жизненного цикла систем ИИ, начиная от постановки целей и проектирования до эксплуатации и вывода из эксплуатации, рассмотрены особенности процессов, связанных с созданием и эксплуатацией систем ИИ, с акцентом на мониторинге и возможной переоценке на протяжении всего жизненного цикла таких систем.

В стандарте затронуты этические и юридические аспекты создания и эксплуатации систем ИИ, предлагаются практические рекомендации по снижению рисков, соблюдению положений нормативных актов и этических норм с целью достижения справедливости, непредвзятости и объяснимости систем ИИ. Обсуждаются социальные и экономические последствия внедрения систем ИИ, выделяются риски и преимущества их использования в различных отраслях, таких как медицина, транспорт, банковская сфера и государственное управление.

Стандарт обеспечивает руководящие принципы и рекомендации, полезные для разработчиков, регуляторов и пользователей, позволяя минимизировать риски и повышать пользу от применения ИИ в различных сферах деятельности.

ГОСТ Р 59898–2021 посвящен вопросам оценки качества систем искусственного интеллекта [4]. Стандарт предлагает детальную методологию и критерии оценки качества систем ИИ на всех этапах жизненного цикла, включая разработку, внедрение и эксплуатацию. В документе определены процедуры и метрики, применяемые для оценки функциональности, надежности, безопасности и других важных характеристик систем ИИ. Предложена модель качества систем ИИ, основанная на наборе характеристик и метрик, позволяющих комплексно

оценить работоспособность и соответствие систем требованиям заказчика и пользователей. Ключевые характеристики качества систем ИИ включают производительность, надежность, удобство использования, мобильность и поддерживаемость. Приводятся примеры расчета и интерпретации этих метрик. Особое внимание уделено формированию и выбору наборов данных для тестирования систем ИИ, которые должны удовлетворять требованиям представительности, объективности и конфиденциальности. Установлены также требования к специалистам, участвующим в процессе оценки, и порядку формирования экспертных групп. Документ нацелен на повышение доверия к системам искусственного интеллекта путем установления единых подходов и стандартов к оценке их качества, предлагает целостную методологию оценки, охватывающую все этапы жизненного цикла систем ИИ, что способствует повышению их надежности и безопасности. Стандарт призван обеспечить объективность и прозрачность процесса оценки, уменьшая риски и повышая конкурентоспособность российских систем искусственного интеллекта.

В ГОСТ Р 59898–2021 рассматривается модель качества системы ИИ как продукта, оценивающая внутренние характеристики и конструктивные особенности системы ИИ. Главное назначение модели в обеспечении соответствия системы техническим требованиям и стандартам качества. В ее состав входят следующие категории:

- функциональность: соответствие заявленным функциям (пригодность, корректность, способность к взаимодействию) и способность выполнять возложенные задачи;
- производительность: скорость и объем обрабатываемых данных, нагрузка на ресурсы;
- надежность: стабильность, устойчивость к ошибкам, восстанавливаемость;
- совместимость: интеграция с другими системами и готовность к работе в различных средах;
- поддерживаемость/сопровождаемость: тестируемость, легкость модернизации, обновления и устранения дефектов.

Согласно ГОСТ Р ИСО/МЭК 25010–2015, регламентирующему требования и оценку качества информационных систем и программного обеспечения, создавая основу для определения, измерения и оценки качества систем и программных продуктов [5], наряду с моделью качества продукта, на которой основаны положения ГОСТ Р 59898–2021, можно также использовать:

- модель качества при использовании – описывает взаимодействие пользователей с системой, охватывая такие характеристики, как эффективность, производительность, удовлетворенность, свобода от риска и покрытие контекста;

- модель качества продукта – определяет характеристики качества программного обеспечения и компьютерных систем, такие как функциональная пригодность, уровень производительности, совместимость, удобство использования, надежность, защищенность, сопровождаемость и переносимость.

Рассмотрим модель качества при использовании подробнее. Эта модель направлена на оценку качеств ИИ, проявляющихся в процессе эксплуатации системы. Она фокусируется на восприятии пользователей и удобства использования ИИ-решений в реальных условиях. Основными категориями качества здесь выступают:

- практичность: легкость освоения и использования системы пользователями;

- надежность: частота отказов и ошибок, восстановление после сбоев;

- эффективность: способность системы выполнять задачи быстро и качественно;

- доступность: доступность системы для пользователей с особыми потребностями.

ГОСТ Р ИСО/МЭК 25010–2015 предлагает принципы построения моделей качества, методики измерения и оценки качества, демонстрирует, как качества продукта и качества при использовании влияют друг на друга. Также в документе раскрываются способы использования моделей качества для определения требований, разработки показателей и оценки качества систем и программного обеспечения.

В начале 2025 г. введены в действие четыре ГОСТа, регулирующих вопросы качества компонентов экосистемы ИИ [6–9], прежде всего качество данных, что делает актуальным введение в рассмотрение третьей модели, оценивающей качество систем ИИ, – модели качества данных. В этой модели основной фокус оценки качества смещается на качество данных, используемых для обучения и функционирования систем ИИ. Основные направления оценки:

- адекватность: соответствие данных решаемой задаче;
- актуальность: своевременность и современность данных;
- последовательность: согласованность и непротиворечивость данных;

– соответствие требованиям: соответствие собранных данных законодательным и нормативным актам.

Модель качества данных необходима для систематического и эффективного управления качеством данных, используемых в системах искусственного интеллекта:

– повышение надежности систем ИИ. Высококачественные данные гарантируют, что модели ИИ и МО будут обучены на корректных и репрезентативных данных, что значительно улучшает точность и надежность результатов;

– оптимизация бизнес-процессов. Использование качественной модели данных позволяет организациям своевременно выявлять и исправлять ошибки в данных, что ведет к улучшению бизнес-процессов и повышению конкурентоспособности бизнеса;

– соответствие регуляторным требованиям. Качественная модель данных гарантирует соответствие нормативным требованиям и защищает организацию от штрафов и санкций;

– рост доверия потребителей. Высокая надежность и точность данных укрепляют доверие потребителей и улучшают репутацию бренда;

– устойчивость к изменению данных. Изменения в источниках данных, технологических платформах или методах обработки данных могут негативно сказаться на качестве данных. Модель качества данных позволяет оперативно выявлять и корректировать такие изменения, обеспечивая стабильность и преемственность процессов;

– экономия ресурсов. Использование низкокачественных данных увеличивает трудозатраты на очистку, нормализацию и обработку данных. Качественная модель данных уменьшает эти издержки, позволяя эффективнее использовать ресурсы организации;

– развитие культуры качества данных. Постоянное применение модели качества данных формирует корпоративную культуру, направленную на улучшение качества данных и осознанное отношение сотрудников к этому вопросу.

Каждая из трех моделей качества (при использовании продукта и данных) отражает отдельный аспект оценки систем ИИ и предназначена для комплексного рассмотрения всех сторон качества искусственного интеллекта.

Факторы, влияющие на уровень надежности ИИ:

– качество данных. Чем точнее и полнее обучающие данные, тем выше надежность ИИ;

- архитектура и алгоритмы. Сложные и неэффективные алгоритмы снижают надежность системы;
- регулярность проверок и коррекций. Постоянное наблюдение и поддержание способствуют росту надежности;
- внешние условия. Наличие посторонних шумов, атак или повреждений негативно сказывается на надежности;
- настройка и модернизация. Периодическая настройка и модернизация повышают надежность ИИ.

Таким образом, уровни надежности ИИ варьируются от абсолютно надежного до совершенно ненадежного, в зависимости от целого комплекса факторов, включая архитектуру, качество данных и условия эксплуатации.

В заключении раздела подробнее остановимся на ГОСТах, регулирующих качество данных.

ГОСТ Р 71484.1–2024 посвящен вопросам качества данных, используемых в задачах искусственного интеллекта, машинного обучения и аналитики данных [6]. Стандарт сфокусирован на механизмах обеспечения качества данных: моделях качества данных, показателях качества, требованиях к управлению качеством и жизненном цикле данных. В стандарте приведены практические примеры и сценарии, иллюстрирующие применение стандартов и процессов обеспечения качества данных.

ГОСТ Р 71484.2–2024 также посвящен качеству данных для аналитики и МО и устанавливает перечень показателей качества данных [7]:

- аккуратность: степень корректности данных и их меток;
- полнота: степень заполнения данных и отсутствие пропущенных значений;
- согласованность: однородность и отсутствие противоречий в данных;
- достоверность: соответствие данных реальности и их источникам;
- актуальность: свежесть и соответствие данных текущим условиям;
- проверяемость: возможность независимой проверки данных;
- сбалансированность: равномерность распределения данных по категориям;
- репрезентативность: полнота и соответствие данных изучаемой совокупности.

В стандарте рассматриваются компоненты качества данных на всем жизненном цикле данных, включая их сбор, подготовку, обработку и использование. Выделяются три уровня качества данных:

- внутренне присущие характеристики: свойства данных, заложенные изначально (например, точность и консистентность, т.е. непротиворечивость);

- внутренне присущие и системно-зависимые характеристики: свойства, зависящие от инфраструктуры и используемых систем (например, доступность и сохранность данных);

- системно-зависимые характеристики: свойства, зависящие от конкретной системы и среды (например, переносимость и восстанавливаемость данных).

Стандарт содержит рекомендации по ведению отчетности и документированию процесса оценки качества данных.

ГОСТ Р 71484.3–2024 посвящен процессам управления качеством данных, обеспечивающим успешное функционирование систем ИИ и МО [8]. В стандарте описаны процессы управления качеством данных на всех этапах жизненного цикла данных, от сбора и подготовки до вывода из эксплуатации, определены процедуры и механизмы управления данными, включая процессы верификации, валидации, управления конфигурацией и изменениями. Рассмотрены вопросы управления рисками, связанными с качеством данных, и приводятся рекомендации по их минимизации. Особое внимание уделено компетентности сотрудников, их обучению и ответственности за качество данных.

ГОСТ Р 71484.4–2024 предлагает унифицированную структуру процесса управления качеством данных, которая обеспечит их соответствие требованиям и повысит надежность результатов аналитических исследований и машинного обучения [9]. Стандарт описывает подробную структуру процесса управления качеством данных, включающую планирование, оценку, повышение качества и валидацию данных. В стандарте рассматриваются методы повышения качества данных, такие как очистка, нормализация, аугментация и обезличивание данных, определены роли участников процесса управления качеством данных, такие как планировщик данных, создатель данных, сборщик данных, инженер данных, распорядитель данных и пользователь данных, уделено внимание особенностям управления качеством данных для различных подходов к машинному обучению: обучение с учителем, без учителя, с частичным привлечением учителя и обучение с подкреплением.

Подводя итоги по материалам рассмотренных стандартов, необходимо отметить, что на надежность и качество систем ИИ влияют факторы из нескольких категорий. Большой акцент в добровольном регулировании сделан на технических факторах и организационно-управленческих.

Технические факторы:

- качество данных;
- алгоритмы и архитектура;
- регулярные проверки и калибровка;
- условия эксплуатации (в том числе внешние, такие как колебания температуры, влажность, электромагнитные излучения и механические вибрации, потенциально способные нарушить работу аппаратного обеспечения);
- объем и мощность вычислительных ресурсов.

Организационно-управленческие факторы:

- контроль качества;
- мониторинг и обратная связь;
- документация и инструкции;
- обучение персонала;
- политика безопасности.

Следует отметить и влияние двух дополнительных групп факторов.

Социальные факторы:

- «человеческий фактор» (вероятность ошибок/неточностей и т.п., допущенных человеком при разработке или при эксплуатации/взаимодействии с системой ИИ);
- доверие пользователей (предубеждения, сомнения, предвзятое отношение, побуждающие игнорировать результаты, выданные ИИ, мешающие эффективно эксплуатировать системы ИИ);
- общественное мнение (негативное восприятие систем ИИ может ограничивать распространение таких систем).

Условия эксплуатации могут быть дополнены категорией «факторы среды эксплуатации»:

- воздействие внешних факторов (помехи, погодные условия и другие внешние факторы могут негативно влиять на работу аппаратных компонентов);
- срок службы оборудования (старение оборудования и износ комплектующих могут негативно влиять на работу аппаратных компонентов);
- энергоснабжение (перебои с энергоснабжением могут вызывать неисправности в аппаратных компонентах).

Только комплексное решение с учетом всех перечисленных факторов позволяет создать надежную и долговечную систему искусственного интеллекта.

1.3. Большие языковые модели. Эволюция моделей GPT

Ранее уже были рассмотрены типы нейронных сетей, которые способны генерировать оригинальный контент: текст, изображения, видео, музыку, речь, код (см. раздел 1.1). Но фокус рассмотрения был на архитектуре таких нейросетей и особенностях их обучения. Теперь же уделим внимание результатам работы и тому, как эффективнее использовать генеративный искусственный интеллект в практических задачах.

Генеративный искусственный интеллект (Generative Artificial Intelligence, GenAI) отличается от традиционного распознающего или классифицирующего ИИ тем, что он генерирует новые объекты, а не только принимает решения на основе анализа существующих данных.

Генеративные модели используют различные подходы для создания новых объектов:

1. Генерация случайных образцов, когда генеративная модель обучается на большом наборе данных и изучает распределение вероятностей внутри этих данных. После обучения она способна создавать совершенно новый контент, аналогичный обучающим данным, но уникальный сам по себе. Например, модель создает портрет лица, которого никогда раньше не существовало, основываясь на изучении множества реальных фотографий лиц.

2. Интерполяция между существующими образцами, когда генеративные модели комбинируют разные образцы, создавая нечто среднее между ними. Это позволяет получать промежуточные состояния между разными объектами, например создание несуществующего животного путем комбинации признаков кошки и собаки.

3. Преобразование одного типа данных в другой, когда генеративные модели способны по тексту создавать изображение, видео или музыку и, наоборот, могут рассказать о том, какие объекты присутствуют на изображении, и даже превратить это описание в осмысленный небольшой сюжет.

К генеративным моделям относятся и генеративно-состязательные нейросети, и диффузные модели, и трансформеры (см. раздел 1.1).

Генеративный искусственный интеллект не способен самостоятельно мыслить, и этим он отличается от человека. Он может обучаться на предоставленных ему данных и по-своему их интерпретировать. Однако считать, что генеративный ИИ не способен на творчество и не может создавать оригинальные решения, ошибочно. Да, основа такого творчества не тождественна человеческой креативности, но имитация в продвинутых моделях столь совершенна, что грань между тем, что создает ИИ, и тем, что может сделать человек искусства, становится все менее ощутимой.

Главным преимуществом ИИ перед реальными людьми являются почти не ограниченные вычислительные возможности. Чем больше данных, на которых учится модель, тем качественнее она учитывает контекст ситуации и умеет адаптироваться.

GenAI способен воссоздавать объекты и распознавать контекст, и достигаются такие возможности благодаря алгоритмам машинного обучения и большим языковым моделям (Large Language Model, LLM).

Большие языковые модели представляют собой мощные инструменты искусственного интеллекта, предназначенные для автоматической обработки естественного языка (Natural Language Processing, NLP).

Большие языковые модели состоят из многослойных искусственных нейронных сетей, которые проходят процесс глубокого обучения. Их основной задачей является понимание и генерация фраз на естественном языке. Для достижения высокой точности LLMs обрабатывают миллиарды токенов (слов или фрагментов слов) из огромного набора данных, содержащего разнообразную литературу, научные статьи, веб-сайты, социальные сети и многое другое.

Ключевые характеристики LLM:

- масштабируемость: точность и универсальность модели повышается с увеличением объема тренировочных данных и повышением сложности архитектуры нейронной сети;

- автоматический выбор наилучшего решения и автономность действий: самостоятельно выбирают оптимальный ответ на заданный вопрос или предложение и во многих случаях способны выполнять задачи по ситуации без участия человека (интеллектуальные агенты на основе LLM);

- обучение без учителя: многие современные LLM начинают свое обучение без четких инструкций относительно правильных ответов, основываясь на анализе структуры и закономерностей большого

объема данных, а уже после самостоятельной разметки исходных данных обучение становится более продуктивным и точность моделей существенно возрастает;

– возможность переобучения/дообучения: изначально натренированная на одном языке или в одной области знания LLM может быть дополнительно обучена на новых данных для улучшения специфических навыков.

Среди известных больших языковых моделей (LLM) есть как проприетарные, так и модели с открытым исходным кодом.

Рассмотрим несколько зарубежных и отечественных LLM.

Подробно рассмотрим эволюцию моделей GPT, которые фактически задали тренд развития LLM, демонстрируя рост в точности, стабильности и универсальности.

Модель GPT (Generative Pre-trained Transformer) – это семейство моделей нейронных сетей, разработанных компанией OpenAI для обработки естественного языка. OpenAI – американская научно-исследовательская организация, занимающаяся разработками в области искусственного интеллекта [10].

Первая версия модели GPT (GPT-1) была представлена в июне 2018 г. GPT-1 имеет архитектуру трансформера, основанную на механизме внимания (Attention mechanism), с примерно 117 миллионами параметров. Изначально модель использовалась для классических задач NLP, таких как классификация текста, заполнение пропусков, обобщение текста и др.

Серия GPT-2, выпущенная в 2019 г., состояла из четырех моделей, которые выходили поэтапно: уменьшенная версия с 124 миллионами параметров, средняя версия GPT-2 Medium с 3555 миллионами параметров, частичная версия GPT-2 Large с 774 миллионами параметров вышла 20 августа 2019 г., полная версия GPT-2 Extra Large с 1,5 миллиардами параметров стала доступной 5 ноября того же года. Модель впервые продемонстрировала высокую эффективность в режиме zero-shot learning – способности решать новые задачи, не проходя явного дообучения на примерах для этих задач. Zero-shot learning (обучение с нулевой обучающей выборкой) – это метод машинного обучения, в котором для обучения не используются размеченные данные, т.е. ранее обученную модель можно использовать для классификации новых данных без необходимости дальнейшего обучения и модель способна распознавать новые классы или категории данных, с которыми она раньше не сталкивалась.

Модель демонстрировала способность создавать содержательные статьи, рассказы и даже связные фрагменты художественной

прозы. GPT-2 могла выполнять разнообразные задачи, такие как перевод текста, написание статей (новостных заметок), создание кода.

Компания OpenAI решила ограничить публичный доступ к GPT-2 Extra Large, опасаясь, что модель могут использовать для создания фейковых новостей и дезинформации. Возможности модели заставили задуматься об управлении рисками и широком спектре потенциальных злоупотреблений, связанных с неэтичным и попросту злонамеренным использованием.

Широкая известность пришла к компании и ее разработкам вместе с GPT-3 – это семейство включало уже 8 различных моделей, первые из которых стали доступны в 2020 г. Самая крупная конфигурация содержала 175 миллиардов параметров, что сделало ее крупнейшей моделью своего времени. Объем тренировочного корпуса достиг колоссальной величины – порядка нескольких терабайтов. Модель проявила удивительную многозадачность, продемонстрировав успех в задачах перевода, программирования, классификации документов, создании контента и даже решении простых математических задач. Улучшилась способность правильно выполнять поставленные задачи без дополнительного обучения на специальных наборах данных (zero-shot learning). До GPT-3 большинство моделей приходилось отдельно обучать, чтобы модель могла справиться с каким-то классом задач: перевод, анализ текста или написание кода. GPT-3 же сама подбирала подходящий алгоритм для поставленной задачи, анализируя ее контекст.

Однако при значительно расширившемся функционале GPT-3 все равно остается инструментом общего назначения, не заточенным конкретно под одну область.

Одной из сильных сторон GPT-3 является способность удерживать контекст беседы на протяжении длительного периода времени, что позволяет вести с моделью длительные непрерывные диалоги, когда новое сообщение рассматривается в контексте предыдущих и модель правильным образом учитывает это в своих дальнейших репликах.

Однако популярность и массовое использование GPT-3 довольно быстро выявили некоторые ограничения модели:

- отсутствие глубинного понимания смысла: несмотря на высокое качество генерации текста, модель часто повторяется, теряет последовательность изложения или неправильно интерпретирует сложные темы;

- склонность к фальсификациям: иногда GPT-3 создает правдоподобные, но абсолютно недостоверные утверждения, потому что ее

знание основано исключительно на тексте, а не на фактах реального мира;

- высокая цена эксплуатации: из-за невероятных объемов памяти и сложной архитектуры поддержание и запуск модели обходится дорого;

- предвзятость и токсичность: поскольку модель обучается на открытом корпусе данных, она может воспроизводить стереотипы, ненавистнические выражения и прочие нежелательные материалы.

Модель GPT-3.5 стала доступна осенью 2022 г., было заявлено, что это не отдельная модель, а скорее модернизация и усовершенствование предыдущей версии GPT-3. Однако фактически GPT-3.5 демонстрирует значительные улучшения и новое поколение подходов к обработке естественного языка:

- устранение некоторых серьезных недостатков и уязвимостей первоначальной версии;

- повышение качества генерации текста и снижение вероятности выдачи неправильного или вредного контента;

- улучшение обучаемости и приспособляемости модели к различным сценариям использования.

Самое главное различие между GPT-3 и GPT-3.5 состоит в подходе к доработке модели и введении новой техники обучения. Это обучение с подкреплением на основе отзывов людей (Reinforcement Learning from Human Feedback, RLHF), метод машинного обучения, который сочетает обучение с подкреплением (RL) с обратной связью от людей (HF). Этот метод предполагает привлечение человеческих оценщиков для ранжирования ответов модели и последующего усиления лучших вариантов с помощью алгоритмов обучения с подкреплением, что помогает нарастить результативность LLM за счет учета намерений и предпочтений человека.

Благодаря RLHF в GPT-3.5:

- заметно повысилось качество генерации и снизилась частота поверхностных или странных ответов;

- исчезли некоторые характерные ошибки GPT-3 вроде многократного повторения одних и тех же идей или выражений;

- стал возможен автоматический отказ от предоставления потенциально вредных или некорректных ответов, за счет чего снизился риск непредвиденных последствий.

У GPT-3.5 остались некоторые ограничения GPT-3, хотя их проявление сгладилось, но также появились и вновь приобретенные за счет возросших возможностей функционала:

- ошибки случаются, особенно в специфических ситуациях или незнакомых сценариях, хотя процент хороших ответов вырос;
- модель проявляет склонность к формальным и шаблонным ответам, особенно при работе с абстрактными или метафорическими понятиями;
- при попытке решить сложную многоступенчатую задачу модель иногда испытывает затруднения, запутываясь в собственных выводах;
- энергоэффективность улучшилась, но требуемые объемы памяти и мощности остаются высокими, делая эксплуатацию дорогостоящей.

GPT-3.5 знаменует следующий этап в развитии генеративных языковых моделей. Применяемый подход с привлечением human-in-the-loop (HITL) позволил значительно улучшить качество и доверие к результатам. HITL – это подход к обучению и развертыванию искусственного интеллекта, при котором люди участвуют в процессе маркировки данных, обучения модели, проверки или принятия решений.

С модели GPT-3.5 начинается масштабное бизнес-использование генеративных моделей, так как OpenAI представляет в ноябре 2022 г. специальную версию языковой модели ChatGPT. Первая общедоступная версия ChatGPT была основана на архитектуре GPT-3.5.

Важно, что обучение ChatGPT включало в себя два этапа:

- pre-training. ChatGPT прошел предварительное обучение на множестве источников, таких как книги, статьи, веб-ресурсы и другие формы открытого текста;
- fine-tuning. После основного этапа обучения ChatGPT проходит дополнительное обучение, называемое fine-tuning, которое направлено на устранение слабостей и повышение общей продуктивности модели.

А RLHF для обучения ChatGPT помогла:

- повысить чувствительность к формулировке вопроса и улучшить качество ответов;
- использовать инструкции и команды непосредственно в интерфейсе для управления поведением модели.

ChatGPT позволяет эффективно общаться с искусственным интеллектом в форме диалога и демонстрирует следующие качества:

- универсальность при решении задач (ответы на вопросы, генерация контента на заданную тему, кодирование и программирование, анализ загружаемого контента: саммаризация, изменение стиля изложения, пересказ, перефразирование и т.п., обучение и поддержка);

- простота использования (радикально снижает барьер входа для обычных пользователей, которым не обязательно разбираться в терминологии или методиках работы с искусственным интеллектом);
- контекстное взаимодействие (учет предыдущих диалогов, фокусировка на обсуждаемой проблеме при обмене репликами с пользователем).

ChatGPT имеет стандартный для языковых моделей набор ограничений:

- иногда выдает ложную/сомнительную информацию, полагаясь на данные, полученные из ненадежных источников;
- будучи продуктом обучения на общественных источниках, порой отражает культурные и общественные стереотипы, которые могут быть несправедливыми в отношении чего-либо или кого-либо;
- создает у собеседника впечатление, что понимает смысл его слов, хотя на самом деле всего лишь симулирует понимание, следуя заранее установленным паттернам.

С появлением новых моделей совершенствуется и ChatGPT. На пути построения диалоговых интерфейсов поверх языковых моделей находятся и другие компании, которые представляют на рынке свои решения, уделяя особое внимание решению проблем с выявленными недостатками и ограничениями и расширению функциональных возможностей.

В марте 2023 г. компания OpenAI представила GPT-4, включающую сотни миллиардов параметров, что значительно превышает показатели предыдущих версий. Такой объем параметров обеспечивает невероятно высокий уровень точности и разнообразия генерируемого контента.

Мультимодальность – это одна из самых ярких особенностей GPT-4, модель способна воспринимать и обрабатывать не только текстовую информацию, но и визуальные данные, может анализировать изображения и графические файлы параллельно с текстом, интегрируя оба вида информации для формирования полных и точных ответов. Также GPT-4 может долго удерживать контекст, т.е. оперировать информацией из сотен предыдущих обменов, что позволяет проводить долгие и последовательные обсуждения. Также GPT-4 намного лучше справляется с задачами, требующими логического вывода и комплексного анализа, может решать головоломки, рассуждать о морально-этических проблемах и обеспечивать аргументированные ответы на сложные вопросы.

Улучшены алгоритмы фильтрации генерируемого контента, разработаны специальные меры для предупреждения выдачи недопустимого содержания или некорректных реакций. Алгоритмы предотвращают выход за границы безопасного пространства и повышают ответственность за каждое слово и фразу, которые генерирует модель.

Доля роботизированности в ответах (формальности, шаблонности) сохраняется, хотя и достаточно мала, модель иногда допускает ошибки, галлюцинирует (придумывает несуществующие данные и факты).

В мае 2024 г. OpenAI представила на рынке модель GPT-4o, где o – это сокращение от omni (лат. всесторонний), т.е. подчеркивается способность модели охватывать множество видов задач и взаимодействий одновременно. Эта модель по производительности превосходила предыдущие версии, сочетая максимальную мощность и эффективность. Модель способна распознавать эмоции, переключаться между стилями коммуникации.

GPT-4o демонстрирует отличные результаты в решении математических задач различной сложности, начиная от элементарных алгебраических уравнений и заканчивая высокоуровневыми расчетами. У GPT-4o расширенные способности к мультимодальному восприятию, модель может обрабатывать текстовую информацию, «читать» видео и звуковые записи, суммируя их содержимое за считанные секунды. Модель стала дешевле и быстрее по сравнению с предшественниками, что делает ее привлекательной для коммерческого использования и массового внедрения.

Компания также выпускает специализированные модели, которые настраиваются на определенные навыки. Так, в начале 2025 г. была представлена модель GPT-o3 с улучшенными способностями к рассуждениям и комплексному анализу информации. Она поддерживает расширенные инструменты и подходит для решения сложных задач, включая научные исследования и программирование. Модель способна разбивать сложные задачи на подзадачи и определять последовательность действий. А весной 2025 г. стала доступной для использования GPT-o4-mini, компактная версия, которая сохраняет основные возможности флагманских моделей, но требует меньших вычислительных ресурсов. Модель способна эффективно работать с текстами, изображениями и данными, демонстрируя высокую точность даже в сложных ситуациях. Эти модели, настроенные на определенный круг задач, позволяют клиентам продуктов на основе LLM выбрать оптимальную модель под свои задачи с учетом располагаемого бюджета, так как стоимость доступа к разным моделям отличается

и можно динамично управлять тарифами, снижая стоимость обращения к моделям при заданном качестве обслуживания.

Последняя версия языковой модели от OpenAI GPT-5 представлена 7 августа 2025 г. Модель строит логические цепочки, выявляет противоречия и предлагает решения, опираясь на огромный объем данных. GPT-5 понимает документы до 1 млн токенов, что позволяет анализировать, например, многолетнюю корпоративную отчетность или целые кодовые базы. GPT-5 не ждет подсказки на каждом этапе, а сама доводит задачу до результата. Модель напрямую взаимодействует с Gmail, Google Calendar, VS Code, GitHub Copilot, CRM-системами.

Обновления моделей, в которых компания стремится устранить существующие недостатки и расширить сферу применимости, имеют следующие перспективные тренды в разработке:

- увеличение размеров моделей;
- увеличение многообразия и глубины обучающих данных;
- развитие методов оценки и мониторинга качества ответов;
- расширение мультимодальности, включение видео- и аудиопотоков в процессы обработки.

Помимо технических аспектов, большое значение придается вопросам этичности, справедливости и прозрачности. Поскольку крупные языковые модели оказывают огромное влияние на общество, усилия направлены на минимизацию вреда и максимизацию пользы от их внедрения.

1.4. Зарубежные конкуренты GPT

Рассмотрим других заметных игроков зарубежного рынка, представляющих свои разработки LLM аналогичного уровня.

Первую большую языковую модель Google разработала в 2018 г., это модель BERT (Bidirectional Encoder Representations from Transformers).

В модели BERT реализован принципиально новый подход к обучению языковых моделей, основанный на двунаправленном внимании (bidirectional attention). Ранее доминировали односторонние модели, такие как GPT (от OpenAI), которые читали текст слева направо или справа налево, но не могли полноценно учитывать взаимосвязь между элементами в обоих направлениях.

BERT же рассматривает контекст как слева, так и справа от текущего токена. Это делает ее сильной в задачах понимания и анализа

текста, что привело к серьезным улучшениям в задачах обработки естественного языка (NLP), таких как:

- вопросно-ответные системы;
- машинный перевод;
- резюмирование текстов;
- категоризация документов.

Однако модель BERT были присущи следующие недостатки:

- медлительность и ресурсоемкость: во многом из-за двунаправленной обработки контекста BERT – это тяжелая модель, требующая значительных вычислительных ресурсов для обучения и вывода, при этом скорость работы и точность ухудшаются при работе с длинными текстами;

- получение непредсказуемых результатов или нестабильная генерация из-за используемой в BERT техники маскировки токенов (masked language modeling), при которой случайно выбранные слова замещаются специальным маркером [MASK], а модель должна угадать правильное слово (при всей эффективности подхода количество ошибок было достаточно большим);

- использование декодирования (игнорирование прямой генерации): BERT выступает в роли энкодера, умеющего хорошо понимать текст, но модель не предназначена для прямой генерации текста, не может последовательно продолжить предложение или генерировать целые тексты, как это делает GPT;

- значительный расход оперативной памяти из-за необходимости хранить огромную матрицу представлений для всех слоев и позиций, особенно остро этот негативный эффект ощущается при развертывании в реальных приложениях;

- маленький контекст из-за того, что модель обучалась на небольших сегментах текста, качество обработки снижалось при увеличении длины контекста.

В дальнейшем, учтя выявленные ограничения и недостатки, проанализировав опыт успешных моделей конкурентов, в Google разработали модель Gemini.

Модель Gemini – это линия языковых моделей, которые характеризуются мультимодальностью и высоким уровнем производительности [11].

Первая версия Gemini 1.0 вышла в свет в ноябре 2023 г. и представляла собой попытку Google объединить традиционные языковые модели с мультимодальными возможностями. Gemini 1.0 позволяла совмещать обработку текста с работой с изображениями и звуком.

Основой модели стал стандартный трансформерный подход, дополненный специальными модулями для анализа мультимедиа. Были доступны версии: Ultra (флагман для самых сложных задач), Pro (универсальная модель) и Nano (компактная для мобильных устройств).

В период с февраля по май 2024 г. были выпущены обновленные модели Gemini 1.5 в версиях Gemini 1.5 Pro (с качеством на уровне 1.0 Ultra при меньших затратах) и Gemini 1.5 Flash (облегченная и быстрая версия).

В декабре 2024 г. появилась Gemini 2.0, которая продемонстрировала значительное улучшение в части мультимодальности и кросс-доменных задач. Была улучшена производительность, увеличено количество поддерживаемых языков, добавлена совместимость с дополнительными типами файлов. Ключевыми преимуществами модели стали возросший объем памяти и поддержка работы с графикой и видео, наличие инструментов для самостоятельного обучения пользователей, дополнительная поддержка иностранных языков. Были доступны версии: 2.0 Flash (экспериментальная и GA), 2.0 Pro (для сложных задач). Ключевые инновации: встроенные агентные возможности (tool use), нативная генерация изображений и аудио через Live API для взаимодействия в реальном времени.

В марте 2025 г. компания Google представила первую версию модели Gemini 2.5 – экспериментальную Gemini 2.5 Pro, а в августе 2025 г. – усовершенствованную версию Gemini 2.5 Deep Think. Модели версии 2.5 (описанные как «мыслящие») способны внутренне генерировать и оценивать промежуточные шаги рассуждений перед тем, как дать финальный ответ.

Ключевые инновации модели в новой версии – это интеграция механизма «мышления» (Deep Think), значительно повышающего качество рассуждений, контекстное окно до 2 миллионов токенов. Режим Deep Think включает в себя несколько ИИ-агентов, которые, работая параллельно, пробуют разные подходы к решению задачи. В итоге модуль-критик или выбирает лучшее решение, или комбинирует финальный ответ из нескольких решений.

Отличительной чертой модели Gemini 2.5 стала полная поддержка интерактивного режима, когда модель способна поддерживать длительные беседы и динамически реагировать на изменение условий среды. Также реализована поддержка парсеров данных и дополнительных форматов представления информации.

Кроме Gemini 2.5 Pro и Gemini 2.5 Deep Think в этой серии есть и модели Gemini 2.5 Flash и Gemini 2.5 Flash-Lite, которые компания Google начала внедрять в свою поисковую систему.

Как и в случае с различными моделями GPT, вывод облегченных и специализированных версий моделей продиктован необходимостью снижения стоимости обслуживания запросов для пользователей, которые могут динамично управлять своими расходами, ориентируясь на свой бюджет и требования к качеству выполнения задач моделью.

Как и в случае с версионностью GPT, каждая следующая версия Gemini расширяет возможности модели и укрепляет ее положение на рынке искусственного интеллекта.

Еще один заметный игрок – это компания Anthropic, которая была основана бывшими сотрудниками OpenAI в январе 2021 г. с целью разработки этично спроектированных и надежных больших языковых моделей. Сегодня компания известна своими инновационными подходами к развитию и контролю таких моделей, став одним из ведущих игроков на рынке искусственного интеллекта.

Claude – это флагманская модель компании, представленная в 2023 г. [12]. Несколько моделей, представленных компанией, отличались по возможностям:

- Claude 1 – базовая версия модели, ставшая отправной точкой для дальнейших разработок;
- Claude Instant – упрощенная версия Claude, оптимизированная для быстрого ответа и минимальной нагрузки на серверы;
- Claude+ – продвинутая версия, предлагаемая по подписке. Обеспечивает повышенный уровень точности и производительности;
- Claude XL – экстра-увеличенная версия, предназначенная для корпоративных клиентов и сложных задач.

Модель Claude 1 обладала возможностью генерации текстов (статей, эссе, постов для соцсетей), также выполняла резюмирование, обобщение и сокращение объемных текстов, выделение ключевых моментов, перевод текстов между различными языками с сохранением смысла и стиля. Однако модель имела ограничения в области кодирования, математических вычислений и логического анализа.

Обновленная версия представлена позже, в конце 2023 г. Отличается улучшенными возможностями в математике, программировании и анализе данных.

Весной 2024 г. компания представила третью линию языковых моделей Claude 3 с более чем 30 миллиардами параметров, что гарантирует высочайшее качество обработки данных.

В серии Claude 3 вышли:

- Claude 3 Haiku (быстрая модель для повседневных задач);
- Claude 3 Sonnet (универсальная модель среднего размера);
- Claude 3 Opus (наиболее мощная модель для сложных задач).

Особенность Claude 3 – это расширенная база тренировочных данных, включающая дополнительные категории (научно-технические тексты, юридическую литературу, специализированную отраслевую информацию). Также используются технологии детекции спама и ботов, с помощью которых модель определяет подозрительные запросы и предотвращает их обработку, защищая пользователей от возможных угроз. В модели присутствуют фильтры ненормативной лексики и провокаций. Кроме того, модель отличается простотой интеграции, так как доступны удобные SDK (Software Development Kit, набор инструментов для разработки программного обеспечения) и API (Application Programming Interface, программный интерфейс приложений, который позволяет разным программам общаться между собой), встраиваемые в любые приложения и сервисы.

Компанией Anthropic регулярно выпускаются новые версии моделей, оснащенные свежими функциями и лучшими показателями производительности.

В феврале 2025 г. Anthropic выпустила первую гибридную модель Claude 3.7 Sonnet, которая сочетает мгновенные ответы с углубленным пошаговым анализом. В мае 2025 г. представлены две новые усовершенствованные модели – Claude Opus 4 и Claude Sonnet 4.

Большинство компаний, разрабатывающих наиболее успешные LLM, имеют американскую юрисдикцию. И в настоящем пособии проведен далеко не исчерпывающий обзор самых передовых и востребованных моделей.

Однако среди глобальных конкурентов LLM, представленных американскими компаниями, появились и крайне перспективные разработки от китайских компаний.

Китайская компания DeepSeek была основана в Ханчжоу в 2023 г. командой бывших сотрудников Zhejiang University совместно с инвестиционным фондом High-Flyer и возглавлялась Лян Вэньфэном, одним из основателей фонда High-Flyer. С самого начала компания получила финансовую поддержку от материнской структуры, позволившую ей привлечь талантливых специалистов и провести исследования в области ИИ.

Компания быстро завоевала международное внимание благодаря своим большим языковым моделям DeepSeek [13], разработанным с акцентом на открытость и низкие затраты на обучение. В отличие от закрытых моделей от западных конкурентов, таких как OpenAI и Google, компания DeepSeek сделала свои модели доступными для общественности, что вызвало серьезный резонанс в мировой технологической среде.

Осенью 2023 г. компания DeepSeek выпустила свою первую модель DeepSeek Coder, бесплатную для коммерческого использования. Позже в декабре была представлена DeepSeek LLM – большая языковая модель с 67 миллиардами параметров, рассчитанная на конкуренцию с западными аналогами (по производительности приближалась к GPT-4).

В 2024 г. появились модели DeepSeek-V2 (столкнулась с проблемами производительности и масштабируемости, однако была недорогой со стоимостью всего 2 юаня за миллион токенов, что вызвало ценовой конфликт среди китайских производителей ИИ) и DeepSeek-V3 (тесты производительности показали, что эта модель соответствует возможностям GPT-4o и Claude 3.5 Sonnet).

В ноябре 2024 г. стала доступной модель DeepSeek-R1-Lite-Preview, которая позиционируется как «легкая» версия флагмана DeepSeek-R1, предназначенная для тестирования и раннего ознакомления с возможностями модели. Модель предназначена для решения задач, требующих логического вывода, математических рассуждений и решения проблем в реальном времени.

В январе 2025 г. на рынок были выведены две модели: DeepSeek-R1 (флагманская модель компании, демонстрирующая ее готовность соперничать с ведущими мировыми производителями моделей искусственного интеллекта) и DeepSeek-R1-Zero (служила дополнением, подчеркивающим специфические аспекты подхода компании к обучению моделей). Обе модели получили широкое внимание как инновационные и доступные решения в области искусственного интеллекта, особенно благодаря своей ориентации на низкую стоимость и высокую производительность.

При обучении моделей R1 использовался метод Multi-token Prediction (MTP), который позволяет анализировать разные части предложения одновременно и предсказывать несколько слов одновременно, что повышает скорость работы модели и точность ее ответов.

Главная особенность моделей DeepSeek – отсутствие этапов обучения с учителем (SFT) и замена их на чистое обучение с подкреплением (pure reinforcement learning). Например, DeepSeek-R1 отказалась от классического этапа выравнивания данных, используя только предварительно подобранные примеры размышлений, что позволило сократить затраты на обучение и ускорить процесс. Это сделало DeepSeek привлекательнее для бизнеса и исследователей, желающих сэкономить на разработке и обслуживании моделей.

Модель DeepSeek-R1 построена на мультиэкспертной архитектуре (Multi-expert architecture, MoE), подходе к проектированию нейронных сетей, при котором вычислительная нагрузка на определенном слое или операции распределяется между несколькими «экспертными» подсетями, что позволяет экономить ресурсы.

Процесс работы архитектуры MoE включает несколько этапов:

- вентильная сеть вычисляет для каждого эксперта оценку, указывающую на его релевантность для данного токена;
- на основе этих оценок выбирается подмножество экспертов для обработки токена. Распространенная стратегия выбора – top-k routing, при которой выбираются «k» экспертов с наивысшими оценками;
- после выбора эксперты независимо обрабатывают входной токен;
- выходы активированных экспертов объединяются для получения окончательного выхода MoE-слоя. Это объединение часто происходит с использованием взвешенного усреднения или суммирования, где веса определяются оценками вентильной сети.

Благодаря такой архитектуре вместо задействования всех параметров модели для каждого входного токена активируется только небольшое подмножество экспертов (типичные значения – 2, 3 или 4 эксперта на токен), что не только экономит ресурсы, но и облегчает масштабирование.

В модели DeepSeek-R1-Zero при обучении использовалась концепция Zero Shot (Zero-Shot Learning, ZSL), когда модель обучается без использования заранее подготовленных примеров и внешней коррекции. Модель использует общую архитектуру с R1, но с меньшей нагрузкой на вычислительные ресурсы, является более дешевой версией, предназначенной для массовых пользователей и небольших проектов.

Разница в стоимости и подходах к обучению определяет их целевую аудиторию и область применения. Модель DeepSeek-R1 ориентирована на профессиональные задачи и крупные компании, тогда как модель R1-Zero предназначена для массового использования и индивидуальных разработчиков.

Выпуск DeepSeek-R1 привел к снижению курса акций высокотехнологичных компаний США и Европы, отразив сомнения инвесторов в оправданности высоких расходов на разработку ИИ. Общая потеря капитализации составила более \$1 трлн за короткий период, вызвав переоценку инвестиций в искусственный интеллект.

Модели от компании DeepSeek критикуют за слабые меры безопасности и потенциальные риски, связанные с отсутствием глубокой проверки данных и чрезмерной самостоятельностью в обучении. Поскольку DeepSeek стремительно развивался, проявились проблемы с масштабированием. Быстрый рост числа пользователей приводил к появлению сбоев в работе сервисов, что препятствовало нормальной эксплуатации моделей. Особенно остро эта проблема стояла в странах с большим населением, таких как Индия и Россия.

Инновационные модели DeepSeek-R1 и DeepSeek-R1-Zero способствуют росту конкуренции на рынке искусственного интеллекта и снижают порог входа для пользователей и разработчиков. Их появление демонстрирует значимость китайских компаний в области ИИ и подчеркивает тенденции к открытости и доступности технологий.

Также конкуренцию западным большим языковым моделям составляет Qwen [14] – семейство больших языковых моделей, разработанных подразделением облачных вычислений Alibaba Group – Alibaba Cloud [15].

Первая версия Qwen была представлена в бета-режиме в апреле 2023 г. под названием Tongyi Qianwen. Публичный релиз состоялся в сентябре 2023 г. после получения одобрения со стороны китайского правительства.

В июне 2024 г. Alibaba запустила Qwen 2, а в сентябре выпустила некоторые версии в открытый код, сохранив при этом самые передовые закрытыми.

В январе 2025 г. была представлена Qwen2.5-VL, визуально-лингвистическая модель (Vision-and-Language, VL), способная не только обрабатывать текст, но и анализировать изображения и видеоролики. Модель ориентирована на мультимодальные задачи, такие как описание картинок, идентификация объектов и синхронизация текста с визуальными материалами.

В апреле 2025 г. стала доступна Qwen 3, оснащенная большей емкостью параметров и дополнительным обучением, что делает ее более точной и эффективной в задачах обработки естественного языка. В Qwen 3, как и в DeepSeek-R1, вместо плотной архитектуры трансформеров (где каждый параметр активируется для каждого токена) используется архитектура смеси экспертов (MoE), которая активирует только соответствующие параметры. Qwen 3 оптимизирована для анализа документов в форматах PDF, таблиц Excel и JSON, что полезно для бизнес-задач, также в модель встроены механизмы, предотвращающие генерацию нежелательного контента.

1.5. Отечественные разработки в области LLM

Обратимся к отечественным разработкам. Наиболее передовые и универсальные модели представлены на отечественном рынке решений в области ИИ двумя технологическими компаниями (представителями национального бигтеха) – Сбером и Яндексом.

Собственная разработка Сбера в области больших языковых моделей называется GigaChat [16]. Эта модель была представлена Сбером в 2023 г. и с тех пор претерпевает постоянное развитие и совершенствование. GigaChat является прямым конкурентом популярных моделей, таких как GPT от OpenAI и Gemini от Google, и выделяется своей уникальной архитектурой и характеристиками.

GigaChat построен на основе трансформерной архитектуры, модель оснащена сотнями миллиардов параметров, что делает ее одной из крупнейших на российском рынке. GigaChat способен решать широкий спектр задач, от генерации текста и анализа данных до решения математических задач и программирования, при этом особое качество модель демонстрирует при работе с запросами на русском языке, что актуально для пользователей, предпочитающих русскоязычный интерфейс.

В марте 2025 г. представлена линейка GigaChat 2.0, которая включает три модификации:

- GigaChat 2 Max – самая мощная модель, предназначена для сложных профессиональных задач;
- GigaChat 2 Pro – оптимальный вариант для повседневных и бизнес-задач, предлагает баланс между производительностью и скоростью;
- GigaChat 2 Lite – облегченная версия для простых и быстрых запросов.

А в августе 2025 г. стала доступной модель GigaEmbeddings (Efficient Russian Language Embedding Model), которая способна на интонационную обработку текстов.

Модель GigaEmbeddings использует трехэтапный процесс обучения, в который входит предварительное обучение, точная/тонкая настройка и мультизадачное обучение. Благодаря оптимизации архитектуры параметры удалось сократить на 25 % без потери качества. Модель доступна на GitVerse (<https://gitverse.ru/GigaTeam/GigaEmbeddings>) и HuggingFace (<https://huggingface.co/ai-sage/Giga-Embeddings-instruct>).

GitVerse – российская платформа для работы с исходным кодом, созданная компанией СберТех (дочерней компанией Сбера). Это аналог GitHub, но с учетом особенностей работы в российской среде.

Hugging Face – платформа для совместной работы в сфере машинного обучения, ориентированная на демократизацию искусственного интеллекта через коллективную разработку моделей, наборов данных и приложений.

Сбер активно инвестирует в развитие GigaChat, улучшая его производительность, точность и функциональность. В ближайших планах компании – введение новых режимов работы, расширение возможностей и улучшение взаимодействия с пользователями.

Семейство больших языковых моделей, разработанных компанией Яндекс, – YandexGPT [17]. Первой публичной версией стал YaLM, представленный в 2021 г., а в 2023 г. Яндекс представил YandexGPT, ставший непосредственным конкурентом популярным иностранным моделям, таким как GPT от OpenAI и Gemini от Google.

Первая версия модели YaLM представляла собой базовую конструкцию для дальнейшего развития. Основное внимание в ней было уделено анализу и генерации текста на русском языке, использовалась в основном для внутреннего тестирования и оптимизации.

Версия YandexGPT 1.0, представленная в 2023 г., показала значительное улучшение и расширение возможностей. Модель была интегрирована в интеллектуального агента «Алису». Осенью 2023 г. выпущена обновленная версия YandexGPT 2. Весной 2024 г. доступной стала новая корпоративная линейка API: YandexGPT 3 Pro и YandexGPT 3 Lite.

Осенью 2024 г. стали доступны улучшенные версии YandexGPT 4 Pro и YandexGPT 4 Lite с контекстным окном в 32 тысячи токенов. Обе модели поддерживают функцию chain-of-thought (цепочка рассуждений), т.е. способность «рассуждать» (разбивать сложные задачи на более простые и решать их пошагово).

В феврале 2025 г. компания представила новую версию YandexGPT 5 Pro со следующими возможностями:

- точные вычисления и решение логических задач;
- генерация программного кода с поддержкой более 15 языков программирования;
- создание развернутых текстовых сводок;
- формирование идей и помощь в написании текстов;
- выбор релевантных источников при поиске информации в Интернете;
- поддержка русского и английского языков.

Для англоязычных запросов результаты сопоставимы с GPT-4o, при этом скорость ответа на 40 % выше благодаря оптимизированным алгоритмам кэширования.

Модель доступна в разных сервисах и устройствах:

- чате с Алисой (в веб-версии, мобильном приложении и «Яндекс Браузере»);
- умных устройствах;
- корпоративных сервисах Yandex Cloud AI Studio и SpeechSense;
- Нейроэксперте [18].

Для бизнеса YandexGPT 5 Pro позволяет создавать виртуальных ассистентов, которые обрабатывают документы, анализируют переписку, распределяют обращения по темам и составляют текстовые сводки. Например, нейросеть можно использовать в контакт-центрах для автоматического анализа клиентских сообщений.

Также доступна и облегченная модификация модели Yandex GPT 5 Lite.

YandexGPT ориентирован на русскоязычную аудиторию, что делает его предпочтительным выбором для российских пользователей. Модель соответствует российским законам о защите данных и обеспечивает высокий уровень безопасности. Семейство YandexGPT представляет собой значимый вклад в развитие больших языковых моделей в России. Каждая версия отличается особыми характеристиками и областями применения, что делает его важным инструментом для бизнеса и пользователей.

В 2025 г. стало известно, что команда Yandex Research совместно с ведущими научно-технологическими вузами разработала метод быстрого сжатия больших языковых моделей без потери качества (HIGGS), который позволяет запускать их даже на обычных устройствах – без дорогих серверов и мощных видеокарт [19].

Метод HIGGS (Hadamard Incoherence with Gaussian MSE-optimal GridS) позволяет сжимать нейросети без использования дополнительных данных и без вычислительно сложной оптимизации параметров, а также:

- работает на пользовательском устройстве. Квантизацию можно выполнить за считанные минуты прямо на телефоне или ноутбуке без необходимости предварительной подготовки на специализированной инфраструктуре;
- совместим с моделями на сотни миллиардов параметров. Например, с помощью HIGGS можно сжимать такие большие модели, как DeepSeek-R1 (671 млрд параметров);
- обеспечивает баланс между качеством, размером модели и сложностью квантизации.

Метод уже протестировали на популярных LLM: Llama 3, Qwen2.5 и др. Эксперименты показали, что HIGGS – это лучший способ квантизации по соотношению качества к размеру модели среди всех существующих методов квантизации без использования данных.

Технология уже применяется в Яндексе для быстрого прототипирования. Она может быть полезна в образовании, социальных проектах и других сферах с ограниченными ресурсами.

Кроме несомненных технологических лидеров большие языковые модели в фокусе внимания компании МТС и Т-банка. Даже если разрабатываемые модели проигрывают в качестве лидерам рынка, для компаний, строящих свою цифровую экосистему из платформ/сервисов/приложений, обращение к собственной модели, а не использование чужой – это возможность получить конкурентное преимущество за счет оптимизации затрат (собственные решения при масштабном использовании обычно дешевле аренды чужих), а также держать под контролем собственные данные и безопасность эксплуатации решений на основе ИИ.

Подразделение МТС, специализирующееся на исследованиях и разработках в области искусственного интеллекта, MTS AI, основано в 2018 г. [20]. Большие языковые модели, разработанные MTS AI, – это Cotype Nano (2023) и Cotype Pro 2 (2024) [21].

Cotype Nano – компактная версия модели, предназначенная для легкого и быстрого использования. Она оптимизирована для малых и средних задач, требующих высокой производительности при низких ресурсах, используется в мобильных приложениях, чат-ботах и легких веб-сервисах. Модель способна обрабатывать контекст до 32 000 токенов (около 45 страниц текста) за раз.

Cotype Pro 2 – это улучшенная версия предыдущей модели, предназначенная для более сложных задач и требующая больших вычислительных ресурсов. Модель на 40 % быстрее предыдущей и на 50 % точнее обрабатывает длинные тексты – до 128 тыс. токенов, что соответствует примерно 900 тыс. символов с пробелами. Она адаптирована под выполнение более чем 100 сценариев корпоративных пользователей в России и может быть развернута на собственной инфраструктуре компании без подключения к внешним серверам, что обеспечивает повышенную защиту данных.

Обе модели соответствуют российским законам о защите данных и обеспечивают высокий уровень безопасности, способны решать широкий спектр задач, от генерации контента до анализа данных и программирования.

Следует отметить, что MTS AI занимается и другими направлениями ИИ, к которым относятся:

- машинное обучение и анализ данных;
- компьютерное зрение;
- голосовые технологии;
- роботы и беспилотные транспортные средства;
- городские решения.

Разработки MTS AI:

- Smart City – комплексные решения для городов, включающие системы видеонаблюдения, мониторинга транспорта и экологической обстановки;
- Robots Delivery – роботы-курьеры, способные доставлять товары и посылки, использовались в пилотных проектах в Москве и других городах;
- Voice Assistant – голосовой помощник, разработанный для контактных центров и голосовых сервисов, получивший признание за точность и качество работы;
- Traffic Management System – система управления дорожным движением, использующая ИИ для оптимизации потоков и сокращения пробок;
- Healthcare Analytics – решения для анализа медицинских данных, диагностики заболеваний и профилактики.

В феврале 2025 г. компания MTC запустила LLM-агрегатор на базе облачных сервисов MWS. Это облачная B2B-платформа – MWS GPT [22], которая объединяет большие языковые модели компании MTC. Сервис предназначен для малого, среднего и крупного бизнеса.

Возможности MWS GPT:

- работа с разными нейросетями. Платформа позволяет одновременно использовать разные LLM, меняя их по ходу использования;
- работа с моделями с открытым исходным кодом. Их можно дообучить под собственные задачи на своих корпоративных данных;
- решение разных бизнес-задач. MWS GPT может извлекать структурированные данные из текста, кратко излагать информацию, организовывать умный поиск по корпоративным базам знаний, быстро обрабатывать входящие обращения в службе поддержки, создавать чат-ботов, генерировать контент, анализировать данные и т.п.;
- собственный чат-интерфейс. С его помощью можно сохранять промпты и обмениваться ими;

– No-Code-конструктор. Позволяет проектировать и создавать приложения без привлечения разработчиков.

В основе MWS GPT лежит агентская архитектура. Это значит, что любая модель может стать полноценным ИИ-агентом, принимая решения об использовании дополнительных навыков в зависимости от поставленной задачи.

Оплата за использование сервиса зависит от количества обработанных токенов, что снижает порог входа для малого бизнеса.

Изначально клиентам будет доступна только облачная версия MWS GPT, но в будущем компания добавит возможность развертывать сервис на локальной инфраструктуре. Число доступных моделей планируют расширить, в том числе за счет продуктов, разработанных другими компаниями.

Также MTS AI активно сотрудничает с ведущими вузами и исследовательскими институтами, такими как МГУ, МФТИ и Физтех. MTS AI является важным игроком на рынке искусственного интеллекта, предлагая инновационные решения и технологии для бизнеса и государства.

Российская IT-компания «Т-Технологии» [23], которая занимается разработкой программного обеспечения, искусственного интеллекта, кибербезопасности и цифровых решений для бизнеса и государственных структур, является дочерней компанией Т-Банка, для которого развивает экосистему финансовых и лайфстайл-услуг для частных лиц и бизнеса через мобильные приложения и цифровые сервисы.

В декабре 2024 г. «Т-Технологии» представили модель T-Pro [24], предназначенную для решения сложных задач обработки естественного языка. Модель построена на базе модели семейства Qwen-2.5 с помощью технологии продолженного предобучения (Continual Pretraining), специально адаптирована для работы на русском языке и выполнения определенных задач. Особо отмечалось, что T-Pro не предназначена для использования в качестве готового голосового ИИ-ассистента.

В июле 2025 г. представлена обновленная модель T-Pro 2.0 с гибридным режимом рассуждений, который позволяет использовать одну и ту же модель в двух режимах: первый предназначен для быстрых ответов без длительного размышления (срабатывает в простых запросах), второй – для более обстоятельных ответов, в начале которых модель подробно анализирует и обдумывает задачу (используется при решении сложных задач). Модель T-Pro 2.0 содержит 32 миллиарда параметров, подходит для создания ИИ-агентов, способных

к автономному принятию решений, доступна на платформе Hugging Face под лицензией Apache 2.0.

Модель уже внедрена в ИИ-ассистентов Т-Банка: они обрабатывают до 45 % клиентских обращений без участия оператора.

В 2023 г. российская компания Just AI [25], занимающаяся разработками в области ИИ и создающая коммерческие приложения на их основе, в тестовом режиме запустила собственную модель JustGPT, которая основана на open-source модели LLaMA-2 и используется в инструменте генеративного ИИ для бизнеса Jay CoPilot [26].

Модель JustGPT – аналог чат-бота ChatGPT, основанный на открытой модели Llama2, но дообученной под практические задачи русскоязычной аудитории. Модель содержит 70 млрд параметров.

На этом завершим рассмотрение отечественных разработок больших языковых моделей и рассмотрим еще один тренд – малые языковые модели.

1.6. Малые языковые модели

Малые языковые модели (Small Language Model, SLM) – компактные модели ИИ, разработанные для обработки, понимания и генерации контента на естественном языке. В отличие от больших языковых моделей SLM обычно имеют диапазон от нескольких миллионов до нескольких миллиардов параметров.

Несмотря на скромность размеров, малые языковые модели часто обладают достаточной производительностью и эффективностью для многих практических задач, особенно там, где большие модели избыточны либо ресурсозатратны. SLM способны справляться с конкретными небольшими задачами – это классификация коротких текстов, чат-боты, быстрый поиск, фильтрация спама, генерация коротких фраз.

Малые языковые модели создаются с использованием упрощенных версий искусственных нейронных сетей, применяемых в LLM. Языковые модели содержат набор параметров (по сути, изменяемых настроек), которые они используют для обучения на основе данных и создания прогнозов. Модели LLM (например, GPT-4) могут содержать более триллиона параметров, а SLM ограничены лишь несколькими сотнями миллионов параметров. Уменьшенная архитектура позволяет SLM выполнять задачи обработки естественного языка в специализированных приложениях.

SLM обучаются на наборах данных, которые меньше и более специализированы, чем те, которые используются даже относительно

небольшими LLM. Набор данных, на котором обучаются модели SLM, обычно специфичен для их функции. После обучения модель можно адаптировать для различных конкретных задач с помощью точной настройки.

Проблемы с SLM проявляются в тот момент, когда они начинают использоваться не по назначению, например, в научных исследованиях (как второе мнение в научной дискуссии, для подбора материала по сложным, составным запросам), в медицинской диагностике (комплексном анализе результатов различных исследований). Фактически это ограничения инструмента, которые являются продолжением его достоинств, максимально проявляющихся при решении тех задач, для которых он и предназначен.

Есть ряд ограничений, связанных с SLM, которые стоит учитывать при использовании:

- ограниченная возможность в понимании сложного контекста. SLM могут испытывать трудности с пониманием нюансов, контекстуальных тонкостей и сложных взаимосвязей в языке, что может привести к недоразумениям или упрощенным интерпретациям текста;

- вероятность снижения точности для сложных задач. Ограниченные возможности обработки и меньшие наборы обучающих данных могут привести к снижению точности и увеличению уровня ошибок в задачах, связанных с многоаспектными рассуждениями, сложными шаблонами данных или высокими уровнями абстракции;

- ограниченная производительность. SLM выгодны для быстрых и экономичных решений, но не могут обеспечивать надежную производительность, необходимую для сложных задач.

SLM можно классифицировать по трем основным типам: дистиллированные версии более крупных моделей, модели для конкретных задач и упрощенные модели [27].

Дистиллированные версии более крупных моделей.

Дистилляция – это метод, когда данные от предварительно обученной LLM передаются к малой модели. В этом подходе большая модель-преподаватель используется для обучения меньшей учащейся модели, которая учится имитировать поведение преподавателя. Учащаяся модель сохраняет большую часть знаний преподавателя, но требует меньше параметров и меньше вычислительной мощности. Дистилляция позволяет эффективно разворачивать языковые модели в условиях ограниченных ресурсов, при этом сохраняя высокий уровень производительности. Одна из популярных дистиллированных SLM – DistilBERT, которая предлагает сопоставимую производительность

с более крупным аналогом (BERT) при меньшем размере и с ускоренным выводом. Модель DistilBERT разработана компанией Hugging Face (об одноименной платформе и размещенной на ней модели от Сбера см. предыдущий раздел), представлена в 2019 г.

Модели для конкретных задач.

Модели для конкретных задач – это малые языковые модели, предназначенные для определенных задач или областей. В отличие от моделей общего назначения, например ChatGPT, эти модели настроены для достижения отличных результатов в специфических задачах, таких как анализ тональности, перевод или ответы на вопросы, и максимально уместны в приложениях с аналогичным функционалом. Фокусируясь на узком наборе заданий, модели для конкретных задач иногда могут достигать более высокой точности и эффективности, чем модели общего назначения. Они особенно полезны, если для определенной задачи требуется высокая производительность, а область действия модели ограничена для оптимизации использования ресурсов.

Упрощенные модели.

Обрезка и квантование – это подходы, когда сокращают объемы представления параметров с 32 до 8 или даже 2 бит, уменьшая размеры и требования к ресурсам, а также количество ошибок. Упрощенные модели создаются с меньшим количеством параметров и архитектурами, оптимизированными для минимизации вычислительных требований, при этом обеспечивают высокую производительность. Они часто используются в мобильных приложениях, на пограничных устройствах или в других сценариях, где вычислительные ресурсы ограничены.

Также, ориентируясь на требования к SLM, разрабатываются более эффективные архитектуры.

С узкоспециализированными задачами малые модели справляются быстро и с минимальными затратами памяти, энергии и т.п., что важно при запуске приложений, в которые интегрируются такие модели, на портативных устройствах, для которых низкое энергопотребление и нетребовательность к вычислительной мощности может быть критически важной.

В итоге SLM не слепо копируют поведение LLM, а выбирают другие стратегии решения задач. Например, отвечая на объемный вопрос, они сначала разбивают его на части и решают пошагово, а не целиком. Это помогает экономить ресурсы и получать такие же точные результаты.

При этом при решении того круга задач, для которого они создавались, SLM оказываются не менее точными, чем LLM, и гораздо более безопасными:

- меньшие объемы данных и параметров у SLM очень часто не приводят к потере качества. Этому способствует тщательный отбор данных: среди них нет источников, которые ввели бы нейросеть в заблуждение;

- за счет меньшего объема данных в SLM «мишеней» для хакерских атак тоже меньше, поэтому риски взлома и утечки данных тоже гораздо ниже. Данные максимально фильтруют от нерелевантного и некачественного контента, добиваясь меньшего объема. В результате снижаются риски неэтичного или вредоносного использования;

- более простая архитектура и меньший объем данных позволяют лучше понимать, как работает конкретная модель, и предсказывать результаты.

В случае с SLM на очистку прав и подбор релевантных предметной области наборов данных требуется гораздо меньше ресурсов, а риски нарушений сводятся к минимуму.

В LLM массивы данных огромны, но распределены неравномерно: например, огромную долю составляют материалы на английском. В SLM за счет более качественного отбора можно получить данные на разных языках, чтобы результаты были точнее и объективнее.

Благодаря небольшим объемам SLM выдают результат гораздо быстрее, что особенно важно там, где нужны аналитика и прогнозы в режиме реального времени.

В отличие от LLM, у SLM больше шансов стать массовыми и найти применение, в том числе в системах интернета вещей: в рамках «умного» дома, транспорта или цифровых двойников предприятий. Для последних решающий фактор – большая прозрачность и безопасность, так как на крупных производствах и внутри корпораций действуют жесткие требования конфиденциальности и защиты данных. В ближайшей перспективе SLM станут доступны на большинстве смартфонов.

Если говорить про конкретные задачи, то SLM могут идеально справляться там, где нужно «схватывать на лету»: додумывать задачу по первым словам, дописывать текст, переводить и исправлять ошибки прямо в процессе написания. Например, для синхронного перевода или генерации субтитров на смартфоне.

Еще одна перспективная сфера для малых моделей – персональные ИИ-ассистенты. Они уже появляются в виде мобильных устройств без привычных для нас интерфейсов (а иногда даже без экрана) и умеют делать почти все: от заказа столика в ресторане до управления «умным» домом. Для бизнеса они могут составлять письма и сообщения, следить за качеством обслуживания клиентов и эффективностью сотрудников.

Недостатки SLM [28]:

- ограниченная универсальность. SLM хорошо справляются с задачами в своих доменах, но за их пределами становятся менее эффективными;

- зависимость от качества данных. Если обучающие данные плохие, модель начнет ошибаться. А в случае SLM это особенно чувствительно: даже немного «шумные» примеры могут сильно ухудшить работу;

- узкая база знаний. SLM не обладают широким пониманием языка и мира вокруг нас. Это плохо в задачах, требующих более глубокого понимания различных тем и доменов;

- потенциальная предвзятость в конкретных доменах. Даже при хорошей выборке SLM могут «унаследовать» предвзятости, если они присутствуют в исходных данных.

Таким образом, хотя маленькие языковые модели уступают большим аналогам в масштабируемости и универсальности, они незаменимы в ситуациях, когда важно эффективное использование аппаратных ресурсов и скорость обработки запросов. SLM за счет экономичности и безопасности будут пользоваться спросом, а значит, активно развиваться.

2. ПРАКТИЧЕСКИЕ ПРИЛОЖЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

2.1. Промпт-инжиниринг: диалоговые интерфейсы для взаимодействия с моделями

Промпт-инжиниринг – это искусство формулирования запросов таким образом, чтобы извлекать наиболее точные, полезные и качественные ответы от больших языковых моделей (LLMs), таких как DeepSeek, GigaChat и др. Этот процесс позволяет эффективно взаимодействовать с моделями машинного обучения, направляя их работу и добиваясь желаемого результата.

Промпт-инжиниринг – это мощный инструмент, позволяющий раскрыть весь потенциал больших языковых моделей. Поэтому так важно освоить правила эффективной коммуникации с системами ИИ.

Промпт-инжиниринг и промптинг – это слова синонимичные. Однако есть некоторые различия в их употреблении.

Промптинг – это «общение» с нейросетью на базовом уровне, интуитивная постановка задач обычными пользователями.

Промпт-инжиниринг – профессиональное составление промптов людьми, которые хорошо знают, как устроены нейросети, и умеют их обучать. Подразумевается работа со специальными командами, параметрами, настройками ИИ, которая больше напоминает программирование.

Чтобы попробовать самостоятельно практические кейсы, связанные с удачными и не очень промптами, можно использовать диалоговые интерфейсы, позволяющие работать с большими языковыми моделями от четырех компаний (Сбер, Яндекс, DeepSeek, Alibaba).

Сбер предоставляет к своей модели GigaChat безлимитный доступ. Без регистрации доступна ограниченная по функциональности версия GigaChat (рис. 2.1)

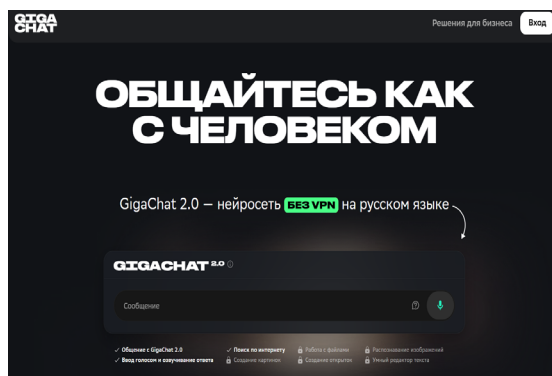


Рис. 2.1. Интерфейс GigaChat [16]

Полнофункциональная версия доступна после регистрации и авторизованного входа (рис. 2.2), и ее возможности гораздо шире. В пособии используется именно она.

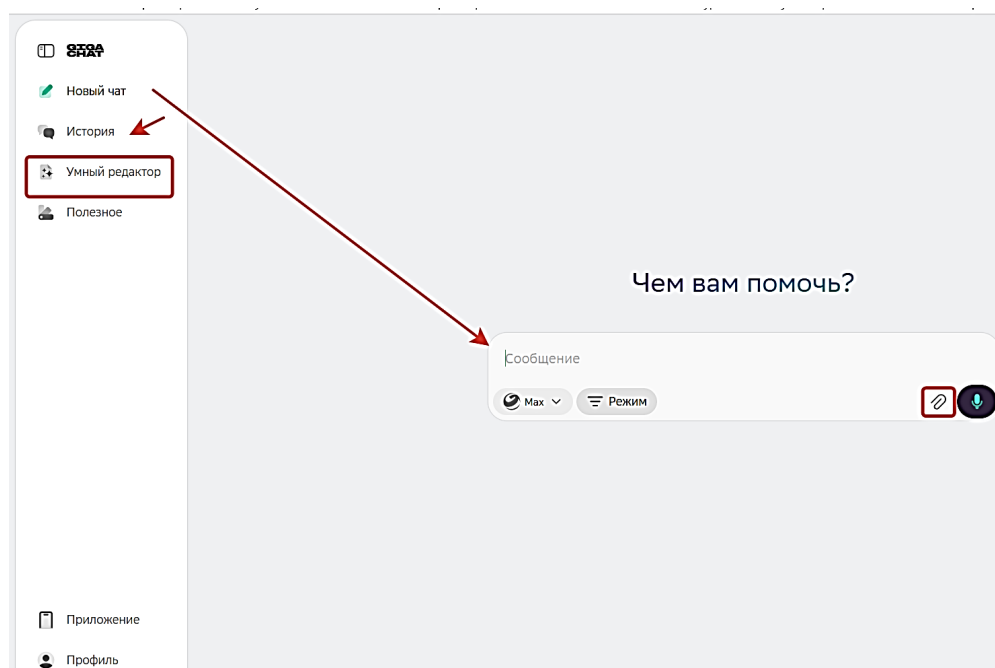


Рис. 2.2. Полнофункциональная версия GigaChat

В полной версии GigaChat есть возможность работать с файлами. Также сохраняется история чатов и можно посмотреть предыдущие диалоги, если в этом появляется необходимость.

Яндекс предоставляет с 1 июля 2025 г. безлимитный доступ к своей модели YandexGPT. Ранее в зависимости от версии YandexGPT были ограничения на количество бесплатных запросов: не более пяти в сутки, а для неограниченного использования нужно было оплатить подписку «Алиса ПРО».

Веб-интерфейс для YandexGPT 5 доступен с Алисой (рис. 2.3).

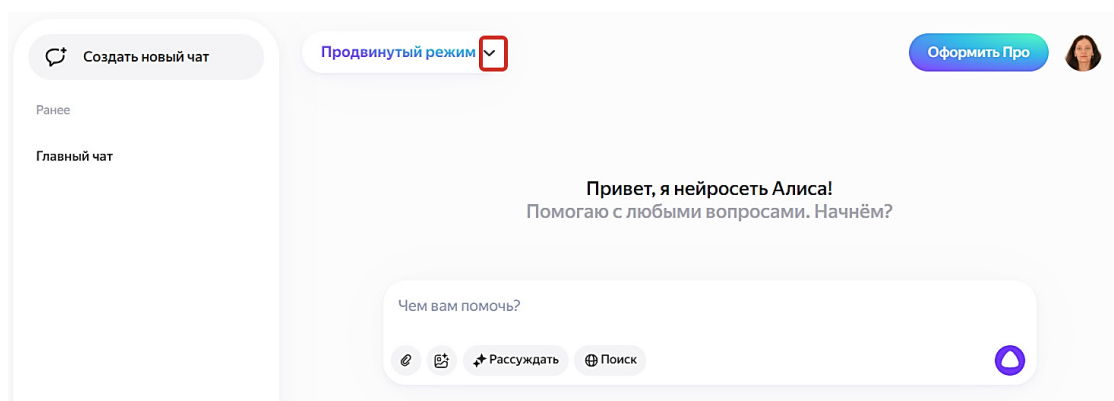


Рис. 2.3. Веб-интерфейс для общения с YandexGPT 5 [29]

Продвинутый режим соответствует расширенной модели YandexGPT 5 Pro, а базовый – облегченной модификации YandexGPT 5 Lite. Алисой можно пользоваться и непосредственно в поисковике (рис. 2.4).

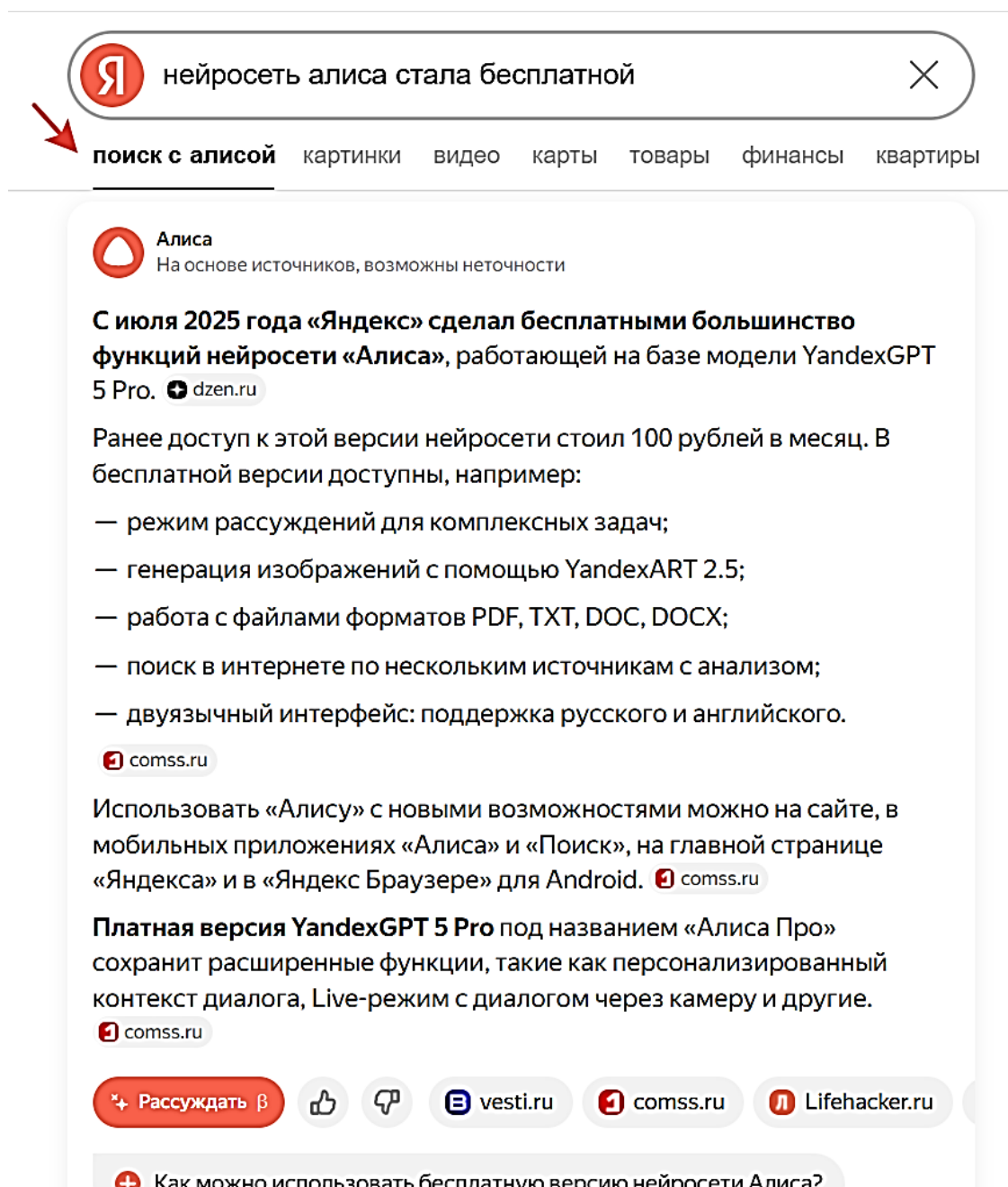


Рис. 2.4. Поиск с Алисой в Яндекс.Поиске

Если запрос предполагает генерацию ответа, то в Яндекс.Поиске она срабатывает автоматически и располагается либо справа в отдельной врезке, либо сразу под поисковым запросом, а ниже в выдаче идут релевантные ресурсы, отвечающие ключевым словам запроса.

Без регистрации, напрямую можно пообщаться и с моделью Qwen от компании Alibaba (рис. 2.5). Общение происходит на языке пользователя, в данном случае на русском.

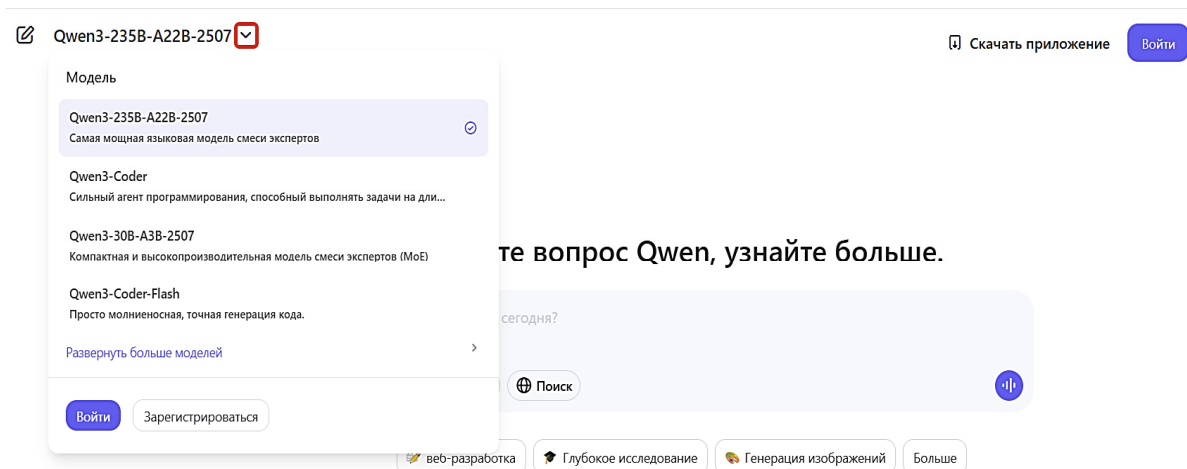


Рис. 2.5. Веб-интерфейс Qwen [15]

Веб-интерфейс модели DeepSeek (рис. 2.6) доступен только после регистрации (рис. 2.7).

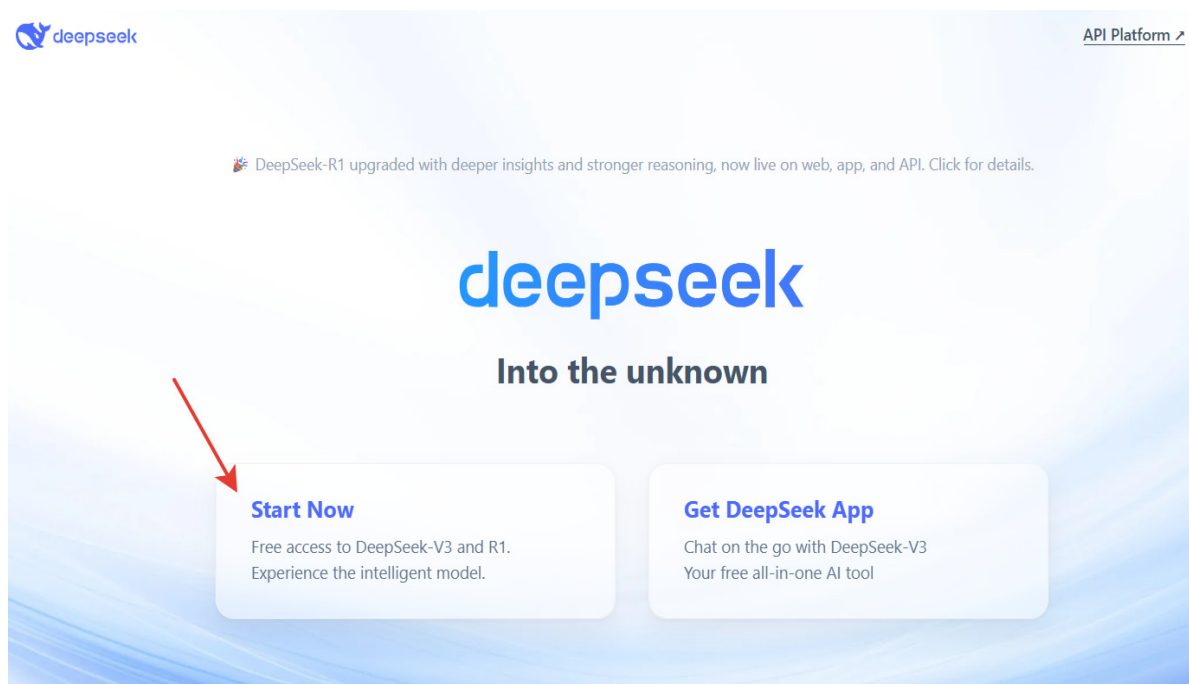


Рис. 2.6. Стартовая страница DeepSeek [13]



Only login via email, Google, or +86 phone number login is supported in your region.

Phone number / email address

Password

By signing up or logging in, you consent to DeepSeek's [Terms of Use](#) and [Privacy Policy](#).

Log in

[Forgot password?](#) [Sign up](#)

OR

Log in with Google

Рис. 2.7. Регистрация в сервисе

Предоставляется возможность анализировать файлы и выполнять мультимодальные запросы (рис. 2.8).

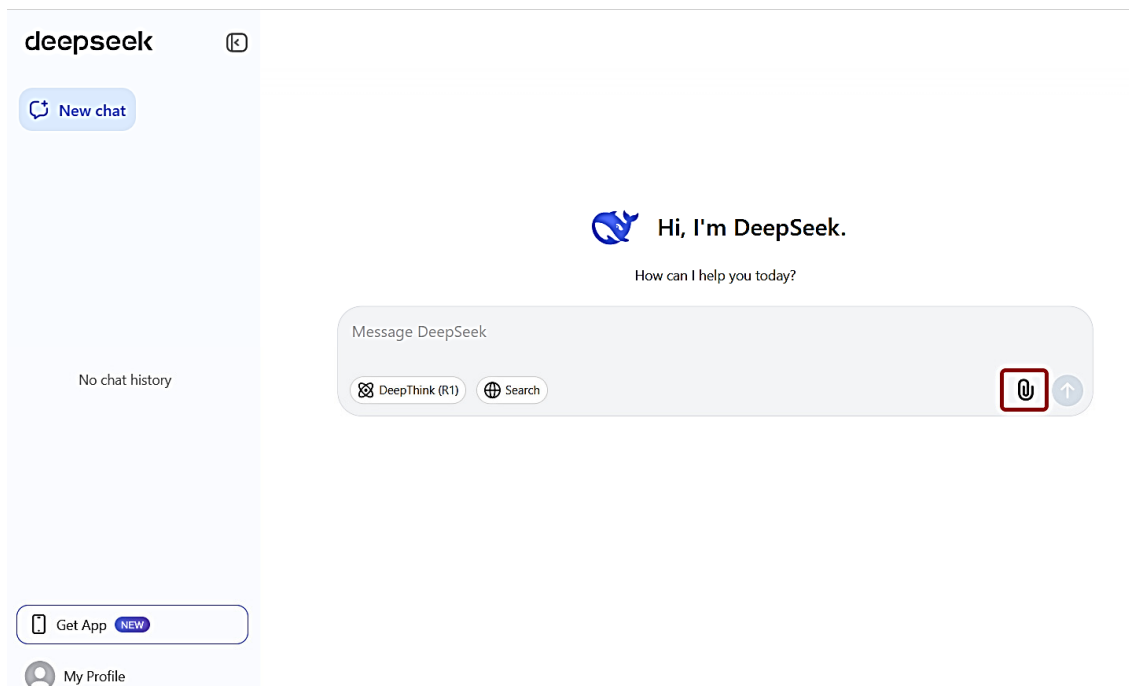


Рис. 2.8. Чат с DeepSeek

Компания Google представила руководство по промпт-инжинирингу от Lee Boonstra [30] как поддержку взаимодействия с семейством своих LLM Gemini. В издании рассматриваются различные методы промпт-инжиниринга, которые помогут начать работу, а также предлагаются рекомендации, которые помогут освоить промпт-инжиниринг, также обсуждаются некоторые проблемы, с которыми можно столкнуться при создании промптов.

Можно познакомиться с этим руководством в полнотекстовом варианте, а можно воспользоваться помощью какой-либо доступной LLM, в которой есть возможность прикреплять файлы с данными, и попросить перевести документ и создать аннотацию из ключевых идей. Например, DeepSeek предлагает такой вариант аннотации, как на рис. 2.9.

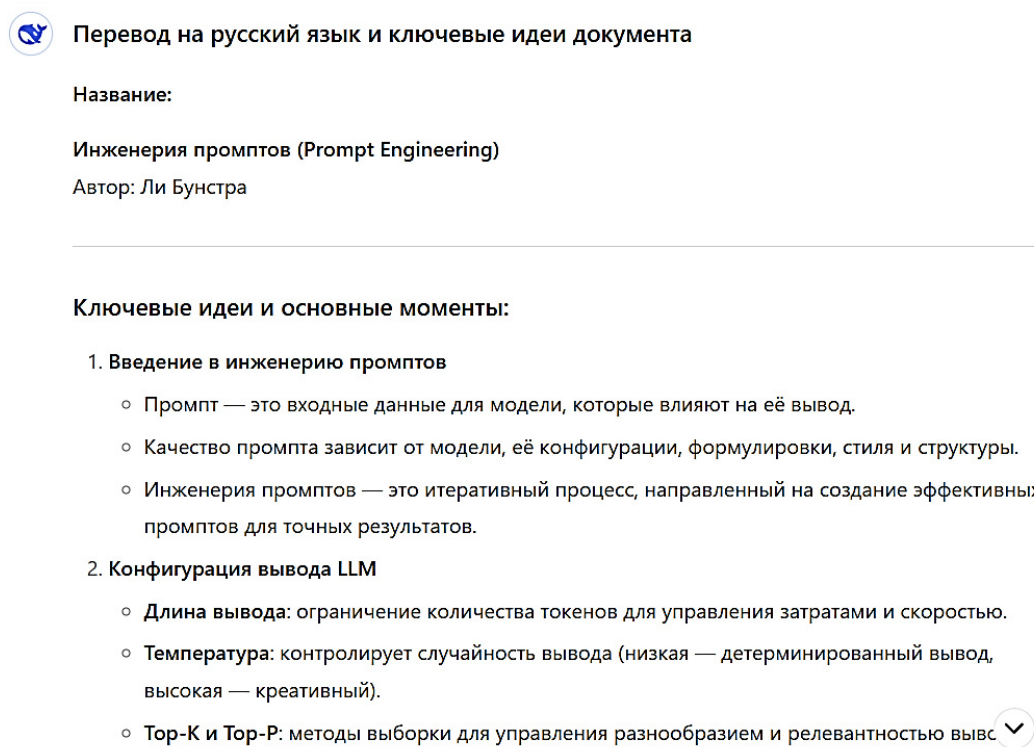


Рис. 2.9. Перевод и саммаризация документа [30], выполненные DeepSeek (фрагмент)

Хороший промпт — структурированный, недвусмысленный, целенаправленный запрос, который учитывает особенности работы LLM. Промтинг — это не про «написать вопрос», а о правильной архитектуре взаимодействия с моделью, которая заключается в выборе точного контекста, оптимальной структуры, подходящего тона, верных ограничений, релевантных примеров и корректных метаинструкций.

2.2. Промпт-инжиниринг: оптимальная структура промпта

Начнем с описания оптимальной структуры промпта для эффективного взаимодействия с ИИ. Она включает четыре ключевых компонента: цель, роль, формат ответа и контекст.

Рассмотрим их подробнее.

1. Постановка задачи (goal/objective). Первая часть промпта должна содержать точное определение задачи или вопроса, который требуется решить. Важно избегать расплывчатых формулировок и абстрактных выражений. Чем яснее сформулирована цель, тем проще модели понимать, какого результата от нее ожидают. Без правильно обозначенной цели модель может выдавать нерелевантные или неполные ответы.

Примеры постановки задачи:

- расплывчато: Посмотри, сколько стоит аренда машины;
- четко: Рассчитай стоимость аренды автомобиля на неделю в Москве летом 2025 г.;
- неоднозначно: Придумай что-нибудь вкусное;
- конкретно: Напиши рецепт приготовления пиццы «Маргарита»;
- абстрактно: Расскажи про машинное обучение;
- определено: Объясни, как работает метод опорных векторов (SVM) для классификации на примере бинарного разделения данных на плоскости. Используй простые аналогии и приведи математическую формулу с пояснением каждого параметра;
- поверхностно: Расскажи об ИИ;
- детально: Напиши 300 слов о применении ИИ в медицине, с акцентом на диагностику рака.

Нельзя сказать, что неоднозначные, расплывчатые формулировки совсем не дают полезного результата. Иногда излишняя конкретика не позволяет осознать, что есть альтернативные варианты, которые существенно расширяют перспективу обзора.

Так, запрос стоимости аренды машины даст мало полезного в части ориентировки в ценах, так как не указан класс авто, на какой срок нужна аренда (скидки при длительной аренде могут быть проигнорированы в ответе, так как эта длительность под вопросом), в каком городе планируется арендовать машину.

Однако, например, GigaChat кроме первичной информации о средних ценах на аренду авто наиболее популярных марок на сутки дал важные рекомендации и предложил уточняющие запросы (рис. 2.10).

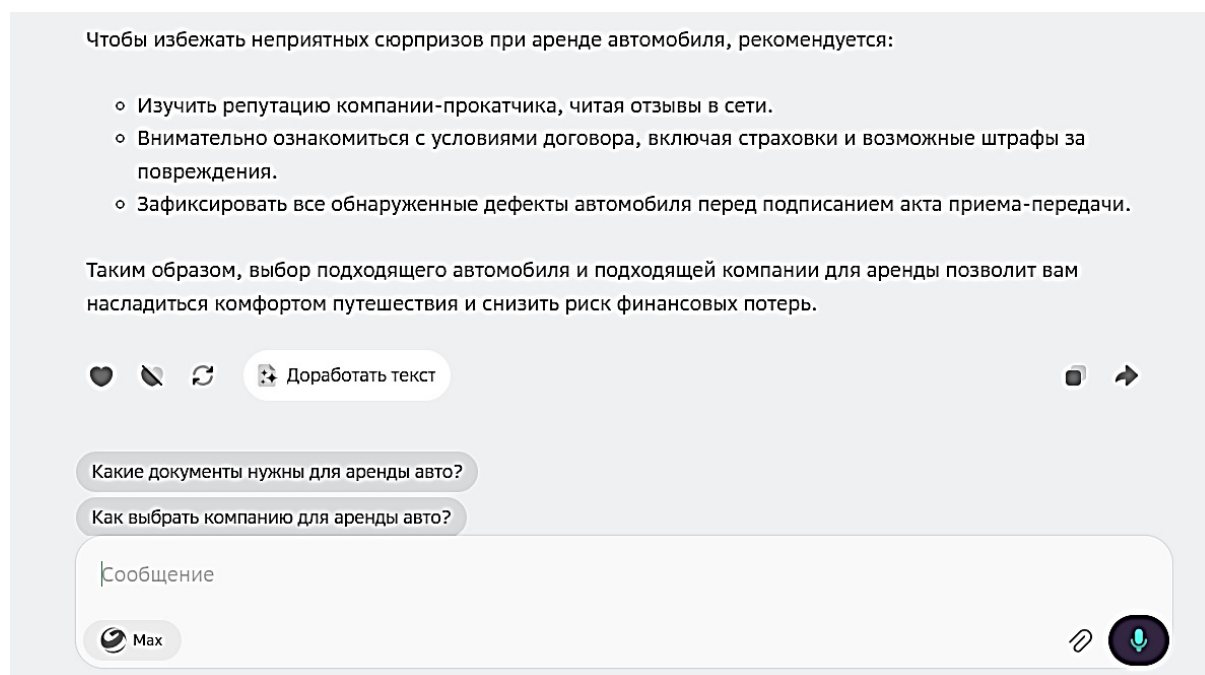


Рис. 2.10. Фрагмент ответа GigaChat на промпт «Посмотри, сколько стоит аренда машины»

А вот с запросом чего-то вкусного ситуация складывается менее удачная. При отсутствии четкой цели ИИ может предложить что угодно – от рецепта пирога до идеи кулинарного шоу, что снижает практическую пользу ответа. И рецепт десерта не будет ценен для того, кто не очень любит сладкое и лучшим пирожным считает котлету.

И если нам нужно больше узнать про конкретный метод машинного обучения, то более точный и детальный запрос предпочтительнее. Однако если изначально нет даже базовых знаний в этой предметной области, то и более общая, абстрактная формулировка сработает для первичной ориентации в новой теме. Есть ощущение, что альтернативный промпт про метод машинного обучения создавался специалистом, который просто затрудняется с формулировками, и ему требуется второе мнение, чтобы точнее сформулировать свои мысли.

В случае с рассказом об ИИ первый вариант промпта приведет к слишком общим, поверхностным рассуждениям об ИИ, а второй промпт сужает тематику и задает конкретные направления, которые должны быть раскрыты в ответе.

Слишком общие формулировки приводят к тому, что модель «плавает» в вариантах.

Также не стоит допускать перегрузки промпта и формулировать слишком много задач в одном запросе.

Обратите внимание, что в приведенных примерах нет категоричных меток: верно/неверно, хорошо/плохо. Иногда нужно подняться над конкретной проблемой, чтобы понять, не упускаешь ли из виду какие-то важные аспекты и нюансы, которые оказываются в стороне от узкоспециализированного вопроса, но влияют на конечный результат, складывающийся из суммы знаний и общих, и частных, и формальных, и личных.

2. Роль (Role Prompting) представляет собой условную маску, которую принимает ИИ для достижения указанной цели. Это позволяет задать нужный стиль и характер общения. Игнорируя этот элемент, мы рискуем получить стандартный ответ, лишенный индивидуальности или нужной стилистики.

Примеры выбора/игнорирования роли:

- указана роль: Как профессиональный консультант по инвестициям расскажи о преимуществах инвестирования в акции технологических компаний;

- роль отсутствует: Расскажи об акциях;

- указана роль: Ты – опытный специалист по данным с 10-летним стажем. Объясни junior-разработчику, как правильно проводить валидацию модели машинного обучения, избегая утечки данных;

- роль отсутствует: Расскажи, как провести валидацию модели машинного обучения.

Модель адаптирует стиль, глубину и терминологию под роль. Без определенной роли ИИ не сможет подобрать правильный тон или уровень детализации, предоставив либо поверхностную общую информацию, либо слишком технический ответ. В некоторых случаях это не принципиально и можно обойтись без роли. Но попробовать, чтобы почувствовать разницу и осознать, насколько изменяется результат и нужно ли это для данной задачи, обязательно стоит.

3. Формат ответа определяет, в каком виде ожидается ответ: текстовом, табличном, графическом, в виде списка или алгоритма. Недостаточное внимание к этому параметру может привести к несоответствию ожиданий пользователя и полученного результата.

Примеры указания формата:

- без указания формата: Подготовь по данным файла итоговый отчет;

- с указанием формата: Представьте итоговый отчет в виде таблицы Excel с колонками «Продукт», «Цена» и «Количество продаж»;
- без указания формата: Какие книги полезны для маркетолога;
- с указанием формата: Составь список топ-10 книг по маркетингу.

Отсутствие конкретных указаний относительно формата может заставить ИИ выбрать произвольный способ представления информации, что может снизить удобство ее восприятия. А может, наоборот, дать толчок к пересмотру того, как лучше представить ожидаемый результат.

Так, более общий запрос, без указания, что нужен непременно список из десяти самых популярных/полезных книг нужной тематики, привел к очень интересным рекомендациям, разделенным на категории (рис. 2.11,а,б). Даже если не все направления, выделенные в ответе, интересны для конкретного пользователя, они все равно могут подтолкнуть его посмотреть на квалификацию маркетолога шире и выбрать что-то в дополнение к основной специализации.

В то же время если нужны были книги узконаправленные и четко ориентированные на продвижение и рекламу, то более подходящим был бы следующий запрос: «Ты преподаватель по маркетингу с 10-летним стажем, какие книги по своему предмету можешь рекомендовать начинающему специалисту по рекламе? Составь список из десяти самых полезных и нужных книг, которые стоит прочитать в первую очередь». Или несколько другой ракурс проблематики получится, если поменять роль: «Ты успешный маркетолог в крупной компании, имеешь опыт продвижения для разных продуктов и разных сегментов бизнеса. В профессии больше 10 лет. Какие книги по своей специализации можешь рекомендовать начинающему специалисту по рекламе? Составь список из десяти самых полезных и нужных книг, которые стоит прочитать в первую очередь и которые были максимально полезны для твоей карьеры в рекламе». Последний запрос, адресованный модели GigaChat, позволяет увидеть очень существенную разницу в продолжении списка, хотя пересечения в результирующих списках, разумеется, будут, но и важные и такие нужные нюансы тоже прозвучат (рис. 2.12).

Для профессионального роста маркетологам полезно читать литературу разных жанров – теоретические учебники, практические кейсы, аналитику тенденций рынка и руководства по развитию ключевых компетенций. Вот список книг, полезных каждому специалисту по маркетингу:

Теория маркетинга

1. **Филипп Котлер** – «Основы маркетинга»
✓ Классика жанра, фундаментальные знания и понимание базовых концепций маркетинга.
2. **Дэвид Огилви** – «Огилви о рекламе»
✓ Глубокий взгляд на рекламу, креативность и эффективные стратегии продвижения брендов.
3. **Игорь Манн** – «Маркетинг на 100%»
✓ Практичные советы по построению эффективной маркетинговой системы внутри бизнеса.

Профессиональная специализация

4. **Эндрю Стэнтон** – «Сделано, чтобы цеплять»
✓ Понимание психологии потребителей и факторов влияния на покупательские решения.
5. **Нир Эяль** – «Hooked: Как создавать продукты, формирующие привычки»
✓ Инструменты разработки продуктов, удерживающих внимание пользователей.
6. **Роберт Чалдини** – «Психология влияния»
✓ Научный подход к созданию убедительных сообщений и влиянию на поведение аудитории.
7. **Рэнд Фишкин** – «Искусство SEO»
✓ Основы эффективного поискового продвижения сайта.
8. **Дмитрий Кот** – «Копирайтинг: сила убеждения»
✓ Советы по написанию продающих текстов и эффективному копирайтингу.

Сообщение

 Max



a)

Анализ и исследования

9. **Сергей Рыжиков** – «Управление результатом»
✓ Методы оценки эффективности рекламных кампаний и оптимизации расходов.
10. **Михаил Дашкиев** – «Бизнес-молодость»
✓ История успеха российских предпринимателей и принципы построения прибыльного бизнеса.

Тренды и инновации

11. **Шейн Сноу** – «Умнее быстрее лучше»
✓ Идеи и технологии повышения продуктивности компаний в условиях быстро меняющегося мира.
12. **Джозеф Джуфреско** – «Персонализация будущего»
✓ Персонализированный маркетинг и адаптация продукта под каждого клиента.

Личная эффективность и лидерство

13. **Стивен Кови** – «Семь навыков высокоэффективных людей»
✓ Принципы управления временем, мотивации и развития лидерских качеств.
14. **Энтони Роббинс** – «Разбуди в себе исполина»
✓ Методики личностного роста и раскрытия потенциала руководителя и специалиста.

b)

Рис. 2.11. Фрагмент ответа GigaChat на промпт
«Какие книги полезны для маркетолога?»:
a – начало списка; б – продолжение

Исходя из моего опыта работы в сфере маркетинга и рекламы, вот список из десяти книг, которые оказали наибольшее влияние на мою карьеру и являются ключевыми для любого начинающего специалиста по рекламе:

Список литературы для начинающих специалистов по рекламе:

1. **Филипп Котлер — «Основы маркетинга»**

✓ Один из главных учебников, дающий базовые понятия и инструменты маркетинга. Это хорошая отправная точка для понимания принципов работы рекламного процесса.

2. **Дэвид Огилви — «Огилви о рекламе»**

✓ Книга легендарного основателя Ogilvy & Mather раскрывает психологию восприятия рекламы потребителями и важность уникального торгового предложения (USP).

3. **Игорь Манн — «Маркетинг на 100%»**

✓ Руководство по применению практических инструментов маркетинга и повышению эффективности рекламной кампании.

4. **Том Питерс — «Преврати себя в бренд!»**

✓ Уникальное руководство по персональному брендингу, помогающее развить личный бренд и стать заметным специалистом в области рекламы.

5. **Мартин Линдстром — «Buyology: правда и мифы о том, почему мы покупаем»**

✓ Исследования неврологии потребления и эмоциональных триггеров, лежащих в основе потребительского поведения.

6. **Роберт Чалдини — «Психология влияния»**

✓ Основа теории убеждения и методов влияния на целевую аудиторию, важные для успешной рекламной кампании.



Сообщение

Max



Рис. 2.12. Фрагмент ответа GigaChat на промпт совета новичку в рекламе из роли успешного специалиста

4. Контекст добавляет дополнительную информацию, уточняющую запрос и влияющую на содержание ответа. Пропуск контекста может привести к непониманию ситуации и выдаче общего, неподходящего ответа.

Примеры детальной/минималистичной подачи контекста:

– подробно: Ты — дизайнер интерьеров, клиент хочет минималистический дизайн спальни в светлых оттенках;

– контекст отсутствует: Придумай дизайн интерьера.

Игнорирование фона или деталей может привести к созданию совершенно другого дизайна, не соответствующего потребностям клиента. При отсутствии деталей модель делает неверные допущения. Например, пожелание «сделай лучше» в промпте имеет слишком большой простор для толкований в силу субъективности и отсутствия конкретики, что именно нужно улучшить, по какому критерию недостаточно хорошо оценен результат. Так, например, уточнение «Ускорьте выполнение алгоритма путем оптимизации структуры данных» дает направление улучшения и подсказывает примерное решение задачи.

Дополнительно в промпте можно указать стиль, ограничения и привести примеры.

Внутри промпта можно указать стиль речи, подходящий для задачи и/или аудитории:

1. Научный стиль (объективный, точный и информативный) содержит специальную терминологию, аргументы и доказательства.

Пример запроса: Опиши биологические процессы фотосинтеза научным языком.

2. Официально-деловой стиль (стандартизированный, нейтральный, сдержанный) используется в документах, деловых письмах, законах и постановлениях.

Пример запроса: Напиши служебную записку о необходимости закупки канцелярии официально-деловым стилем.

3. Художественный стиль (эмоциональный, образный, насыщенный метафорами и художественными средствами выразительности) подходит для творческих работ.

Пример запроса: Расскажи сказку о храбром рыцаре, столкнувшемся со многими испытаниями на своем жизненном пути, используя художественный стиль речи.

4. Публицистический стиль (экспрессивный, эмоционально заряженный, воздействующий на чувства читателя) уместен в статьях средств массовой информации, блогах и публичных выступлениях.

Пример запроса: Напиши публицистическую статью о вреде курения, призывающую отказаться от пагубной привычки.

5. Разговорный стиль (неформальный, живой, спонтанный) отличается простотой лексики, употреблением разговорных оборотов и сокращений.

Пример запроса: Перескажи случай с установкой труб неверного диаметра сантехниками невысокой квалификации простым разговорным языком.

6. Эпистолярный стиль содержит личное обращение, характерное для писем, записок, дневниковых записей, отражает искреннюю доверительность.

Пример запроса: Напиши письмо школьному другу детства, вспоминая совместные приключения той поры, в эпистолярном стиле.

7. Образовательно-методический стиль (учебный, разъясняющий, с подробными и детальными описаниями) используется в учебниках, учебных пособиях, руководствах и инструкциях.

Пример запроса: Объясни понятие интеграла методом, понятным школьникам старших классов.

8. Политико-дипломатический стиль (официальный, выдержанный и осторожный) используется в официальных переговорах, международных соглашениях и заявлениях политиков.

Пример запроса: Составь политическое заявление, подчеркивающее важность международного сотрудничества в борьбе с изменением климата.

9. Историко-биографический стиль – это историческая хроника, наполненная фактическими данными, датами, именами исторических деятелей и событий.

Пример запроса: Опиши жизненный путь Александра Сергеевича Пушкина историческим стилем.

Стиль выбирается исходя из целей запроса и предполагаемой аудитории. Нужна легкость изложения, приветствуется юмор, стоит указать, разговорный или художественный стиль. Для научных работ и серьезных исследований стоит придерживаться научного или официального стиля.

Модель может не идеально воспроизвести выбранный стиль, особенно если запрос чрезмерно сложен или необычен. Поэтому критично отнеситесь к стилевой окраске ответа и скорректируйте его, если есть ощущение, что запрошенный стиль не соблюдается.

Ограничения на выдаваемые результаты, явно прописанные в промпте, позволяют обеспечить достижение нужных целей без нарушения правил конфиденциальности, доступности или законов. Также ограничения помогают скорректировать выдачу в нужном направлении.

Рассмотрим примеры ввода ограничений в промпт на примере запросов из различных предметных областей:

1. Юриспруденция:

– Промпт: Поясни порядок составления претензии в страховую компанию при ДТП.

– Ограничение: Важно придерживаться законодательства РФ и избегать ссылок на внутренние регламенты страховых организаций.

2. Образование:

– Промпт: Создай занятие по изучению методов очистки данных для студентов первого курса вуза с пояснениями, практическими кейсами и заданиями.

– Ограничение: Необходимо использовать бесплатные доступные ресурсы и платформы, исключая платные курсы и подписки.

3. Медицина (оказание первой помощи):

– Промпт: Подготовь инструкцию по первичной обработке раны пострадавшему, если помощь оказывает лицо без медицинского образования.

– Ограничение: Не рекомендуется применение лекарственных препаратов или инструментов, требующих специальной подготовки или лицензий.

Можно также дополнить промпт примером/образцом (одним или несколькими), чтобы лучше сориентировать модель в том, какого результата от нее ждут.

Показательные примеры промптов, содержащие образцы ожидаемых результатов, называются шот-помощниками (shot prompting), или демонстрационными примерами. Они помогают ИИ-модели быстрее понять, что именно ожидается от нее, и выдают ответы в нужном стиле и формате.

Бизнес-задача. Промпт: Опиши шаги запуска новой рекламной кампании в социальных сетях.

Образец ожидаемого результата:

1. Определение целевой аудитории.
2. Выбор платформ распространения (VK и др.).
3. Разработка уникального торгового предложения.
4. Создание визуальных материалов и текстов объявлений.
5. Настройка таргетинга и бюджета кампаний.
6. Тестирование и запуск пилотных рекламных кампаний.
7. Оценка результатов и оптимизация настроек.

Образование. Промпт: Подготовь конспект лекции по основам программирования на Python.

Образец ожидаемого результата:

Заголовок: Основы программирования на Python

Содержание лекции:

Переменные и типы данных

Условные операторы (if, else)

Циклы (for, while)

Функции и модули

Работа с файлами

Практическое задание: создание простого скрипта на Python.

Экономика и финансы. Промпт: Объясни, что такое инвестиционный портфель и как его сформировать.

Образец ожидаемого результата:

Инвестпортфель – это совокупность ценных бумаг, акций, облигаций и иных финансовых инструментов, собранных одним владельцем.

Процесс формирования:

1. Оценка рисков и возможностей инвестора.
2. Распределение активов по классам (акции, облигации, недвижимость и т.д.).
3. Регулярная переоценка и балансировка портфеля.

Благодаря таким подсказкам модель генерирует ответы гораздо ближе к ожиданиям.

Подведем промежуточные итоги по рекомендациям для создания качественного промпта. Структура промпта и иллюстрирующий ее пример представлены в табл. 2.1.

Таблица 2.1

Структура промпта

Компонент промпта	Содержание	Пример
Контекст	Что уже известно	Я разрабатываю чат-бот для поддержки клиентов банка
Задача/проблема/цель	Что нужно сделать, какого результата нужно достичь	Создай 5 примеров ответов на вопрос клиента «Почему с меня списали комиссию?»
Стиль/тон	Какой стиль речи нужно использовать в ответе	Отвечай вежливо, профессионально, не используй технический жаргон
Формат вывода	Как должен выглядеть ответ	Каждый ответ должен содержать не более двух предложений
Ограничения	Чего избегать	Не упоминай внутренние системы банка. Предлагай обращение в поддержку только при необходимости
Примеры	Образцы ожидаемого результата	Комиссия могла быть списана в соответствии с тарифным планом вашего счета. Рекомендуем ознакомиться с условиями обслуживания

2.3. Промпт-инжиниринг: техники улучшения качества промпта

С подходами к обучению моделей или добавлению к промпту определенных элементов, которые представлены далее, уже знакомы. Но для лучшего понимания особенностей настройки промптов систематизируем популярные техники (Фреймворки), которые применяются в определенных типах промптов для улучшения результата:

1. CoT (Chain of Thought, цепочка рассуждений).

Методика предполагает пошаговую формулировку запросов, позволяя модели поэтапно анализировать и решать поставленную задачу, переходя от одного этапа к другому. Это помогает повысить точность вывода и качество решения сложных проблем.

– Промпт: Реши задачу пошагово. Не пиши сразу ответ. Задача: У Маши было 10 яблок. Она отдала 3 подруге и съела 2. Сколько осталось?

При затруднении модели в разбиении рассуждений на шаги можно сделать по задаче серию промптов самостоятельно.

– Промпт: Найди площадь прямоугольника со сторонами 8 и 12 см. Шаг 1. Запиши формулу площади прямоугольника. Шаг 2. Подставь значения сторон. Шаг 3. Вычисли произведение длин сторон.

2. FEWS (Few Shot Prompting, несколько примеров перед задачей).

Этот метод подразумевает предоставление нескольких примеров правильного ввода-вывода вместе с заданием. Так модель учится распознавать структуру и шаблон желаемого результата. Обычно используется три-четыре примера для демонстрации ожидаемого поведения системы.

3. ZEMS (Zero Shot with Metadata Schema, беспримерная подача с метаданными).

Задача ставится непосредственно без примеров, однако вводятся дополнительные условия и ограничения, такие как структура формата вывода, требования к типу ответа и даже критерии оценки качества. Метаданные помогают установить четкую связь между входящими параметрами и ожидаемым результатом.

– Промпт: Составь короткое предложение, содержащее глаголы прошедшего времени. Структура: {субъект} + {глагол} + {объект}.

4. BOA (Behavior-Oriented Approach, поведенческий подход).

Фокусируется на определении типа взаимодействия с моделью. Например, задаются правила, каким образом модель должна реагировать на конкретные типы вопросов, вводить ограничение на количество символов, указывать эмоциональный тон ответа и другие характеристики поведения модели.

– Промпт: Ваша роль – учитель математики, строго объясняющий решение задач. Объясните, почему сумма углов треугольника равна 180 градусам? Убедитесь, что ваш ответ понятен ученикам младших классов.

5. IDA (Instruction Design Analysis, анализ проектирования инструкций).

Подходит для глубокого анализа требований к конкретной задаче. Инструкция создается на основе тщательного изучения контекста проблемы, структуры запроса и критериев успешного выполнения задания. Такой подход часто применяется при создании инструкций для коммерческих продуктов и решений.

– Промпт: Создай резюме кандидата на позицию программиста Python. Воспользуйся данными из файла о кандидате Иване Иванове. Критерии успеха:

- четкая структура: контактная информация, опыт работы, образование, навыки;
- акцент на опыте разработки на Python;
- лаконичность текста.

6. RAPP (Role Assignment & Pattern Preparation, назначение ролей и подготовка шаблона).

Данный подход предлагает четкое определение ролей (например, диалоговых персонажей) и явно сформулированные шаблоны взаимодействий (паттерны). Совместно эти элементы позволяют построить эффективный процесс общения с системой.

– Промпт: Ты – интервьюер, проводящий собеседование с кандидатом на должность инженера-механика. Задай серию вопросов кандидату в соответствии с шаблоном: «Расскажите о вашем самом значимом проекте в сфере машиностроения».

7. Mitigating Hallucinations (смягчение галлюцинаций).

Данный подход помогает уменьшить количество ложных ответов, генерируемых большими языковыми моделями. Модели склонны «выдумывать» факты. Уменьшить такие проявления в результатах можно при добавлении в промпт следующих ограничителей:

- если не знаешь ответа, сообщи об этом, не выдумывай;
- ответь только на основе следующих документов (добавьте их файлы или тексты);
- не используй при ответе данные старше 2020 г.

Эти методики можно комбинировать друг с другом, адаптируя их под специфику вашей задачи и цели проекта. Выбор конкретного подхода зависит от сложности задачи, доступного объема тренировочных данных и особенностей целевой аудитории.

Эксперты, исследователи ИИ создают полезные шаблонные конструкции для промптов. Сделаем небольшой обзор популярных неофициальных фреймворков для составления промптов, которые

широко используются в сообществе для улучшения качества взаимодействия с LLM.

1. TAG (Task – Action – Goal, Задача – Действие – Цель).

Паттерн: Что нужно сделать? Какие шаги нужно предпринять? Какой результат ожидается?

Простой и понятный фреймворк для структурирования запроса. Хорош для быстрых, конкретных задач, особенно в бизнесе и образовании.

– Промпт: Напиши деловое письмо клиенту, объясняющее задержку поставки товара на 3 дня. Используй вежливый тон, сообщи о причине (проблемы с логистикой), предложи скидку 10 % на следующий заказ. Цель – сохранить доверие и избежать жалобы.

– Промпт: Объясни, как работает блокчейн, как если бы я был 12-летним подростком. Разбей объяснение на простые шаги, используй аналогию с книгой записей. Цель – чтобы я понял основную идею без знания программирования.

2. APE (Action – Purpose – Expectation, Действие – Цель – Ожидание).

Паттерн: Что сделать? Зачем? Что получить в результате?

Минималистичный, но эффективный, подходит для быстрых рабочих задач.

– Промпт: Перепиши этот абзац (текст абзаца). Нужно сделать текст понятнее для новичков в этой теме. Используй простые слова, короткие предложения, без жаргона/сленга.

3. STAR (Situation – Task – Action – Result, Ситуация – Задача – Действие – Результат).

Используется для генерации историй, кейсов, примеров поведения, полезен в обучении, коучинге, HR.

– Промпт: Команда из 5 человек работала над запуском приложения, но отставала от графика. Нужно было ускорить процесс без потери качества. Я внедрил ежедневные 10-минутные стендапы и перераспределил задачи. Проект сдан на 2 недели раньше. Перепиши этот кейс как ответ на вопрос на собеседовании.

– Промпт: Учащийся не понимает, как применять формулу площади круга. Объяснение проведено на жизненном примере. Преподаватель предложил рассчитать, сколько краски нужно для покраски поверхности круглого стола. Учащийся понял применение формулы. Опиши этот пример для учебника по математике.

4. PAIN (Problem – Agitation – Information – Now, Проблема – Усиление боли – Информация/решение – Призыв к действию).

Маркетинговый фреймворк, адаптированный под промпты. Часто используется для написания рекламных текстов, писем, продающих постов.

– Промпт: Многие стартапы терпят крах из-за плохого позиционирования. Без четкого уникального торгового предложения можно остаться незамеченным, даже если продукт отличный. На нашей онлайн-конференции вы узнаете, как создать сильное позиционирование за 1 день. Зарегистрируйтесь до пятницы – скидка 30 %. Напиши e-mail на основе этой информации.

5. BAV (Before – After – Bridge, До – После – Мост).

Паттерн: Какая была ситуация? Какой желаемый результат? Что помогло перейти от «до» к «после»?

Маркетинговый фреймворк, адаптированный под промпты, идеален для сторителлинга, мотивационных текстов, объяснения изменений, маркетинга. Помогает создавать убедительные нарративы.

– Промпт: Расскажи историю человека, который перестал тратить время на соцсети и начал учиться. До: бесцельный скроллинг, усталость. После: развитие, энергия, прогресс. Мост: правило «30 минут без телефона перед сном». Рассказ – в формате поста в социальной сети, до 200 слов.

– Промпт: Объясни пользу Notion-шаблона для бизнеса. До: предприниматель тратил 5 часов в неделю на отчеты. После: делает за 10 минут. Мост: шаблон с автоматическими формулами и календарем. Поддай это как рекламный текст для рассылки.

6. RAFT (Role – Audience – Format – Topic, Роль – Аудитория – Формат – Тема).

Паттерн: Кто ты во время выступления? Перед кем ты выступаешь? В каком виде? О чем?

Фреймворк полезен в обучении, создании учебных материалов.

– Промпт: Ты – учитель начальной школы. Расскажи, как образуется дождь, в виде короткой сказки про капельку воды. Добавь 2 вопроса в конце: «Почему облака не падают?» и «Куда девается вода после дождя?». Тон дружелюбный, с элементами волшебства.

– Промпт: Ты – главный аналитик компании. Подготовь презентацию из 5 слайдов для совета директоров на тему «Риски внедрения ИИ в HR». Слайды:

- 1) введение;
- 2) технические риски;
- 3) этические вопросы;
- 4) репутационные угрозы;
- 5) рекомендации. Тон формальный, аргументированный.

7. CLEAR (Concise – Logical – Explicit – Accurate – Relevant, Кратко – Логично – Ясно – Точно – Релевантно).

Паттерн: Без лишних слов. Последовательная структура. Четко указано, что нужно. Корректные данные и формулировки. Только то, что нужно.

Фреймворк подходит для профессиональных, технических или аналитических задач, где важна точность.

– Промпт: Сравни iPhone 15 и Samsung Galaxy S24 по пяти параметрам: камера, автономность, производительность, цена, экосистема. Ответ в форме таблицы 2×5. Только факты, без рекламы. Актуальные данные на 2024 г.

– Промпт: Объясни, что такое НДС, логично и кратко. Начни с определения, затем покажи, как он начисляется на примере цепочки: производитель → оптовик → розница. Только релевантная информация, без отвлеченных тем.

8. CARE (Context – Action – Role – Expectation, Контекст – Действие – Роль – Ожидание).

Паттерн: Каково положение вещей / Какова ситуация? Что нужно сделать? В какой роли должен выступить ИИ? Какой нужен результат?

Один из самых эффективных фреймворков для сложных, креативных или многослойных задач, где важна позиция модели.

– Промпт: У меня канал о личной эффективности, аудитория – те, кому 18–30 лет. Ты – сценарист с опытом в образовательном контенте. Напиши сценарий для 5-минутного видео «Как перестать прокрастинировать за 3 шага». Видео должно начинаться с провокационного вопроса, содержать научные факты, практические советы и призыв к действию.

– Промпт: Клиент не оплатил счет за услуги, задержка в оплате уже 2 месяца. Ты – юрист с опытом взыскания долгов. Напиши деловое, но вежливое напоминание об оплате. Письмо должно напоминать о долге, указывать сроки, угрожать судебным иском, но сохранять профессиональный тон.

9. RACE (Role – Action – Context – Expectation, Роль – Действие – Контекст – Ожидание).

Аналог CARE, но с акцентом на роль в начале. Подходит, когда важно сразу задать «тон» и позицию модели.

– Промпт: Ты – копирайтер в рекламном агентстве. Придумай заголовок для рекламы нового фитнес-приложения с акцентом на удобство, доступность и комфорт. Реклама будет в метро, аудитория – молодежь. Заголовок короткий (до 8 слов), запоминающийся, на русском языке.

– Промпт: Ты – популяризатор науки. Объясни теорию относительности. Аудитория – школьники 10 класса. Объяснение через аналогии, без формул, с юмором, до 150 слов.

10. ICIO (Instruction – Context – Input – Output, Инструкция – Контекст – Входные данные – Формат вывода).

Один из самых структурированных фреймворков. Подходит для технических и аналитических задач.

– Промпт: Переведи текст на немецкий. Это официальное письмо от HR-директора (вводится текст письма). Используй формальный тон, без сокращений, в деловом стиле.

11. RASCEF (Role – Action – Steps – Constraints – Examples – Format, Роль – Действие – Шаги – Ограничения – Примеры – Формат).

Супердетализированный фреймворк, подходит для сложных, многоэтапных задач, где важна точность.

– Промпт: Ты – аналитик данных. Проанализируй продажи за первый квартал текущего года. Сравни с четвертым кварталом прошлого года. Выяви топ-3 по продажам среди товаров. Найди аномалии. Не используй предположения, только данные из таблицы. Если рост 20 %, укажи причины (если есть в данных). Отчет из 3 абзацев, без таблиц, на русском.

12. CRISPE (Capacity and Role – Request – Intent – Steps – Personality – Examples, Роль и способность – Запрос – Цель – Шаги – Личность/тон – Примеры).

Паттерн: Кем должен быть ИИ? Что именно нужно? Зачем это нужно? Как будет использовано? Какие действия следует выполнить? Какой стиль общения? Примеры желаемого ответа.

CRISPE – один из самых детализированных и эффективных фреймворков. Особенно хорош для сложных, креативных или многоэтапных задач, где важны стиль, контекст и примеры. Создан сообществом промпт-инженеров, популяризируется через блоги и курсы.

– Промпт: Ты – сценарист видеоконтента по финансам. Создай 45-секундный сценарий о том, как перестать делать импульсные покупки. Начни с шокирующей статистики, покажи пример из жизни, дай простой совет (например, правило 24 часов), закончи призывом «Сэкономь, чтобы жить свободно». Тон энергичный, с долей иронии. Пример: «Ты тратишь 5000 в месяц на кофе? Это 60 000 в год – новый гаджет!».

– Промпт: Ты – научный журналист. Напиши статью для блога о том, как сон очищает мозг от токсинов. Объясни механизм, покажи

последствия хронического недосыпа, дай 3 практических совета. Тон ясный и убедительный, как у National Geographic. Пример начала: «Пока вы спите, ваш мозг устраивает генеральную уборку».

Систематизируем неформальные фреймворки по назначению (табл. 2.2).

Таблица 2.2

Неформальные фреймворки для создания эффективных промптов

Фреймворк	Для каких задач подходит
TAG/APE	Простые, понятные задачи с четкой целью
STAR/PAIN/BAB	Сторителлинг, мотивация, маркетинг
CARE / RACE	Контекстно зависимые задачи с участием «роли». Креатив, обучение, сложные сценарии
RAFT	Образование, адаптация контента под аудиторию
CLEAR/ICIO/RASCEF	Технические, аналитические, структурированные задачи
CRISPE	Сложные, креативные или многофакторные запросы

Базовые правила промптинга опираются на несколько аспектов:

1. Формулировка вопроса. Четко сформулированный запрос помогает модели лучше понять намерение пользователя. Важно избегать двусмысленности и неясности. Общие и слишком расплывчатые затравки не являются препятствием для генерации ответа, но получить в результате что-то реально полезное, а не такие же общие и поверхностные комментарии, маловероятно.

2. Использование инструкций. Предоставление четких инструкций облегчает понимание задачи моделью. Это особенно важно при выполнении сложных задач, таких как написание программного кода, создание художественных произведений или проведение анализа текста. Инструкции помогают избежать ошибок и повысить точность результатов.

3. Контекст и примеры. Чем больше контекста предоставлено, тем точнее будет ответ. Примеры позволяют модели лучше понять структуру ожидаемого ответа.

4. Подсказки и ограничения. Иногда полезно заранее установить рамки или подсказать возможные варианты решений. Например, если вы хотите написать сценарий фильма, можно указать жанр, целевую аудиторию и ключевые моменты сюжета.

5. Итеративность. Промпт-инжиниринг часто требует нескольких итераций. Если первый ответ неудовлетворителен, попробуйте переформулировать запрос, добавить дополнительные инструкции или изменить подход.

Однако не в любом запросе должны присутствовать все перечисленные моменты. Разумная достаточность уточнений оставляет модели некоторое пространство для творчества и инициативы.

Практические рекомендации:

1. Проверяйте грамматику и орфографию. Ошибки в тексте могут сбить модель с толку. Убедитесь, что ваш запрос правильно сформирован и понятен.

2. Экспериментируйте с разными подходами. Попробуйте разные форматы вопросов и исследуйте, какой из них дает лучшие результаты. Возможно, один и тот же запрос, заданный иначе, даст совершенно иной ответ.

3. Оценивайте качество ответов. После получения ответа оцените его качество. Если результат вас устраивает, сохраните этот подход для будущих запросов. Если нет, проанализируйте причины неудачи и скорректируйте стратегию. Вы можете управлять экспертностью, степенью продвинутости, углом обзора, фокусом, ракурсом рассмотрения другой стороны в диалоге.

2.4. Интеллектуальные агенты, ИИ-агенты

Генеративный искусственный интеллект и LLM открыли новую грань возможностей интеллектуальных агентов и вызвали появление нового вида таких агентов – агентов с искусственным интеллектом (ИИ-агентов). И промпт-инжиниринг играет заметную роль в настройке ИИ-агентов.

ИИ-агенты, построенные на основе LLM, использующие инструменты (tools), память, способные на автономное принятие решений, требуют настройки специфичных параметров [31]:

- `temperature` (температура) – контролирует степень случайности выбора слов. При низкой температуре модель будет более предсказуемой и консервативной, выбирая наиболее вероятные слова. Высокая температура увеличивает разнообразие ответов, но может привести к менее связным текстам. Не рекомендуется использовать совместно с параметром `top_p`;

- `repetition_penalty` (количество повторений слов) – имеет значения 1.0 – ничего не менять (нейтральное значение), от 0 до 1 – повторять уже сказанные слова, от 1 и далее – стараться не использовать сказанные слова;

- `top-p` (альтернатива параметру `temperature`) – модель учитывает результаты токенов с вероятностной массой `top_p`. Таким образом, 0.1 означает, что учитываются только токены, составляющие верхнюю

10-процентную массу вероятности. Значения от 0 до 1. Не рекомендуется использовать совместно с параметром `temperature`;

– `max_tokens` (ограничение максимальной длины генерируемого текста) – полезно для контроля объема ответа и предотвращения слишком длинных текстов.

Интеллектуальный агент – это программа или система, способная взаимодействовать с окружающей средой и самостоятельно выбирать стратегию поведения, исходя из поставленной цели. Интеллектуальность агента заключается в способности обрабатывать информацию, адаптироваться к изменениям условий среды и принимать решения на основе анализа ситуации.

Интеллектуальный агент – это автономная программная или аппаратная система, способная воспринимать окружающую среду с помощью сенсоров, принимать решения на основе этих данных и выполнять действия через исполнительные механизмы для достижения поставленных целей.

Ключевые характеристики интеллектуальных агентов:

- автономность: работает без постоянного контроля человека;
- адаптивность: обучается и адаптируется к изменениям среды;
- целеполагание: действует для достижения конкретных целей;
- реактивность: реагирует на изменения в окружающей среде;
- прогнозирование: может планировать действия на основе модели мира.

Интеллектуальные агенты можно разбить на категории по уровню автономности и сложности:

1. Простые рефлексорные агенты (Simple reflex agents). Это базовая форма агентов ИИ. Действуют по принципу «стимул-реакция», реагируют непосредственно на входные сигналы, используя набор заранее определенных правил, не сохраняя информацию в памяти и не учитывая более широкий контекст.

Примеры:

- термостат, который активирует нагрев или охлаждение на основе текущих показаний температуры;
- автоматические системы управления светом (датчик движения – включение света).

2. Рефлексорные агенты на основе моделей (Model-based reflex agents). Имеют внутреннюю модель окружающей среды. Более совершенны, чем простые рефлексорные агенты, поскольку используют внутреннюю модель окружающей среды, чтобы лучше понять ситуацию. Эта модель позволяет им обрабатывать недостающую или неполную информацию и принимать более обоснованные решения.

Пример: робот-пылесос, строящий карту помещения.

Рефлекторные агенты, и простые, и основанные на моделях, обладают определенными преимуществами:

- быстрота и надежность работы в рутинных условиях. Простые рефлекторные агенты действуют по заранее определенным правилам и реагируют на текущие входные данные без учета прошлого или будущего;

- работа в динамичных или слегка непредсказуемых средах. Рефлекторные агенты на основе моделей учитывают не только текущие данные, но и то, как система функционировала в течение долгого времени;

- возможность предвидеть последствия своих действий. Рефлекторные агенты могут предсказывать, как может измениться окружающая среда на основе их действий;

- постоянное совершенствование. После выполнения действия агент обновляет свою внутреннюю модель окружающей среды на основе новой информации, что позволяет ему постоянно улучшать понимание окружающего мира и процесс принятия решений.

Однако такие агенты не лишены и определенных недостатков и ограничений.

Для простых рефлекторных агентов они следующие:

- отсутствие обучения. Простые рефлекторные агенты не могут адаптировать свое поведение на основе опыта;

- статические правила. Эффективность рефлекторных агентов ограничена predetermined правилами, что делает их непригодными для сложных или динамичных сред, где условия могут непредсказуемо меняться;

- недостаток памяти. Рефлекторные агенты не сохраняют информацию из предыдущих состояний, что приводит к реактивному, но не проактивному подходу;

- невозможность работы с частичной или неполной информацией. Это делает рефлекторных агентов подходящими только для выполнения базовых задач, не требующих принятия сложных решений.

Для рефлекторных агентов на основе моделей в целом перечень ограничений очень схож, хотя некоторые отличия имеются:

- сложность создания модели. Разработка и поддержание точной внутренней модели мира может быть сложной и ресурсоемкой;

- ограниченное обучение. Хотя такие агенты могут обновлять свои модели, они обычно не извлекают уроков из своего опыта, как это делают более продвинутые агенты;

– зависимость от точности. Эффективность принятия решений в значительной степени зависит от точности внутренней модели: если она несовершенна, производительность агента может снизиться;

– статические правила. Такие агенты действуют на основе predetermined правил «условие-действие», которые могут ограничивать их адаптивность в быстро меняющихся условиях.

3. Целеориентированные агенты, или агенты, ориентированные на достижение целей (goal-oriented agent, goal-directed agent), – это автономные программные системы, которые планируют и оценивают серию действий для достижения четко определенных целей. В отличие от традиционных программ, которые следуют жестким алгоритмам, такие агенты обладают способностью к обучению, адаптации и самостоятельному планированию действий и выбирают действия, максимизирующие вероятность достижения цели.

Функционирование целеориентированных агентов основано на цикле «восприятие – размышление – действие»:

– сенсорное восприятие – сбор данных из различных источников, включая текстовые документы, базы данных, веб-сайты, API-сервисы, датчики IoT (Internet of Things, интернет вещей) и пользовательский ввод;

– когнитивная обработка – анализ и интерпретация информации с использованием алгоритмов машинного обучения, нейронных сетей и систем логического вывода;

– планирование действий – разработка стратегии достижения целей с учетом текущего состояния среды, доступных ресурсов и ограничений. Агент может создавать как краткосрочные, так и долгосрочные планы действий;

– исполнение решений – реализация запланированных действий через взаимодействие с внешними системами, отправку уведомлений, обновление баз данных или выполнение других операций;

– обучение и адаптация – корректировка поведения на основе опыта, анализ результатов выполненных действий и обновление внутренних моделей для повышения эффективности.

Наиболее часто используются в автоматизации бизнес-процессов, так как такие агенты способны выполнять сложные задачи: анализировать документы, принимать решения по кредитным заявкам, управлять складскими запасами, обрабатывать клиентские запросы.

Примеры:

– рекомендательные системы в онлайн-кинотеатрах и музыкальных сервисах имеют цель максимизировать удовлетворенность пользователя и на основании этой цели выбирают для него контент,

который с большой вероятностью ему понравится. Система оценивает предпочтения пользователя, его прошлую активность и доступный контент, после чего выбирает рекомендации, соответствующие цели, – удерживать пользователя в сервисе, поддерживая его увлеченность контентом;

- навигация в Яндекс.Картах, обеспечивающая подбор маршрута между начальной и конечной точками, оптимальный не только по расстоянию, но и по времени, с учетом пробок, особенностей движения по дорогам;

- агент управления инвестициями, цель которого максимизировать доходность при заданном уровне риска, а предпринимаемые действия для ее достижения – анализ рынка, ребалансировка портфеля, хеджирование рисков.

Целеориентированные агенты обладают рядом преимуществ, благодаря которым способны выполнять достаточно сложные задачи:

- гибкость в принятии решений. Целеориентированные агенты способны работать с неполной или противоречивой информацией, анализировать альтернативы и выбирать оптимальные решения в условиях неопределенности;

- адаптивность к изменениям. Вместо следования жестким инструкциям целеориентированные агенты выбирают оптимальный путь к результату, подстраиваясь под изменения;

- ускорение принятия решений. Целеориентированные агенты обеспечивают быстрый анализ данных и генерацию рекомендаций.

Однако целеориентированные агенты не могут поддерживать активное общение, допускают ошибки на разных уровнях системы из-за вероятностного характера оптимизации (т.е. вероятность ошибок и неверных решений в каждый момент времени не нулевая, но в среднем они все равно дают оптимальный результат).

4. Полезностные агенты, или агенты, основанные на полезности (Utility-based agents), – это автономные системы, которые выбирают действия, максимизирующие «полезность» (удовлетворенность, эффективность, прибыль). В отличие от простых агентов, которые могут реагировать на стимулы или следовать заранее определенным целям, полезностные агенты оценивают множество потенциальных действий и выбирают то, которое максимизирует их общую полезность. В отличие от целеориентированных агентов оценивают не только достижение цели, но и ее «полезность».

Примеры:

- торговые боты, выбирающие стратегию с максимальной прибылью;

- приложение для доставки, которое находит самый быстрый маршрут с наименьшим трафиком. При принятии решения учитываются время, расход топлива и сроки;

- персональный помощник, который назначает встречи с учетом их приоритетности, доступности и потенциальных конфликтов, обеспечивая минимальные сбои и максимальную продуктивность.

Полезностные агенты сталкиваются со сложностью измерения полезности в различных контекстах, особенно если цели агента неоднозначны или противоречивы. Необходимо определить, какие показатели и метрики будут использоваться для оценки полезности, и учитывать возможные погрешности измерений. Также неопределенность и динамичность окружающей среды могут затруднить оценку полезности действий агента, так как результаты действий могут зависеть от внешних факторов, которые не поддаются контролю.

5. Обучающиеся агенты (Learning Agents) – это агенты искусственного интеллекта, которые улучшают свои способности к принятию решений, извлекая уроки из прошлого опыта, со временем адаптируют и совершенствуют свое поведение на основе обратной связи от своих действий.

Обучающиеся агенты способны развиваться в процессе взаимодействия с окружающей средой, приспосабливаться к меняющимся условиям в режиме реального времени, пошагово адаптировать свое поведение и находить новые способы решения проблем, анализировать себя в терминах поведения, ошибки и успеха.

Примеры:

- AlphaGo – программа, разработанная командой DeepMind, которая сумела победить чемпиона мира по игре Го, продемонстрировав превосходство искусственного интеллекта над человеком в сложной игровой задаче;

- Алиса/Салют – голосовые помощники, постоянно улучшающие понимание естественной речи и расширяющие функциональность на основе анализа пользовательского поведения;

- Autonomous Driving Systems (системы автономного вождения) – это технологии, которые помогают управлять транспортным средством, а наиболее продвинутые решения способны управлять без прямого вмешательства человека;

- Robotic Process Automation (RPA) – программные роботы, обученные выполнению повторяющихся задач в офисных приложениях, их эксплуатация существенно упрощает рутинные операции сотрудников.

Обучающиеся агенты обладают рядом преимуществ, наиболее ценные и полезные из которых:

- самообучение, способность автоматически приспосабливаться к изменениям в среде без вмешательства программистов;
- эффективность, связана с сокращением затрат на обслуживание и поддержку систем благодаря автоматизированному обучению;
- гибкость, легкая адаптация к различным сценариям и ситуациям, возникающим в процессе эксплуатации;
- повышенная точность, когда постоянное обновление моделей ведет к улучшению точности принимаемых решений.

Несмотря на очевидные плюсы, обучающиеся агенты имеют некоторые недостатки:

- ограниченная интерпретация данных, связана со сложностью интерпретации неопределенности и отсутствия полноты информации;
- необходимость значительных объемов данных для успешного обучения;
- проблемы этического характера, так как возможны конфликты с моральными нормами и законами при выработке определенных стратегий поведения;
- время обучения для некоторых задач очень длительное, что замедляет внедрение новой функциональности.

Остановимся подробнее на обучающихся агентах, которые основаны на LLM. ИИ-агенты – это более узкая категория, где интеллект реализуется через машинное обучение, нейросети и LLM.

ИИ-агент на основе LLM – это система, которая:

- понимает цель пользователя (через естественный язык);
- самостоятельно планирует действия;
- использует внешние инструменты (API, базы данных, браузер);
- хранит контекст и память;
- принимает решения и адаптируется;
- выполняет задачу до завершения, возможно, с итерациями и корректировками.

Архитектура ИИ-агента (особенно агента с LLM) состоит из пяти ключевых модулей:

1. LLM (Ядро агента). Принимает решения, генерирует текст, планирует действия. Роли LLM в агенте:

- интерпретация входного запроса;
- генерация плана действий;
- принятие решений на каждом шаге;

- общение с пользователем;
- обработка результатов инструментов.

LLM не просто отвечает, а рассуждает (reasoning) как человек.

Типы рассуждений:

- Chain-of-Thought (CoT) – пошаговое мышление;
- Tree-of-Thought (ToT) – перебор нескольких гипотез;
- ReAct (Reason + Act) – чередует размышления и действия.

2. Планировщик (Planner). Разбивает задачу на шаги, определяет стратегию.

Виды планирования:

- Zero-shot planning – LLM сразу предлагает план без примеров;
- Few-shot planning – использует примеры из прошлого;
- Hierarchical planning – делит задачу на подзадачи (например: «Планирование → Поиск → Анализ → Отчет»);
- Reactive planning – перепланирует при ошибках (например, сайт не отвечает – попробовать другой источник).

3. Инструменты (Tools) – внешние API, функции, базы данных, браузеры, которые агент может вызывать.

Типы инструментов:

- поиск – Google Search API;
- базы данных – SQL-запросы, векторные хранилища;
- API – прогноз погоды, курсы валют, почта (SMTP), календарь;
- файловые операции – чтение/запись файлов с расширениями .doc, .pdf, .xls и т.д.;
- код – выполнение скриптов (например, скриптов на языке Python для анализа данных);
- коммуникация – отправка email, сообщений в Telegram и т.п.

4. Память (Memory) – краткосрочная и долгосрочная память для контекста, позволяет агенту помнить контекст, учиться на своем опыте и избегать повторных ошибок.

Типы памяти:

- краткосрочная (Short-term) – контекст текущего сеанса (например, история диалога), хранится в промпте;
- долгосрочная (Long-term) – контекст сохраняется в векторной базе данных, позволяет вспоминать прошлые задачи;
- семантическая память – общие знания (например, «ставка по ипотеке зависит от первоначального взноса»);
- эпизодическая память – конкретные события («вчера ВТБ изменил ставку»).

5. Исполнитель (Executor) – выполняет действия, вызывает инструменты, управляет потоком.

Исполнитель выполняет следующую последовательность действий:

- получает команду от LLM (в виде действия);
- вызывает нужный инструмент;
- обрабатывает результат;
- возвращает данные в LLM;
- управляет циклом «размышление → действие → наблюдение».

Также можно провести классификацию интеллектуальных агентов по другим критериям. Например, по области применения:

- софтверные агенты (software agents) – это агенты поиска, помощники в электронной почте, чат-боты;
- многоагентные системы (MAS) – это группа агентов, взаимодействующих для решения общей задачи (например, управление трафиком);
- физические агенты (роботы) – роботы-пылесосы, автономные автомобили, беспилотные летательные аппараты;
- виртуальные агенты – аватары в VR, персонажи в играх, цифровые двойники.

Можно провести разбиение на классы по уровню «интеллекта»:

- детерминированные агенты работают по жестким правилам;
- вероятностные агенты используют вероятностные модели (например, байесовские сети);
- обучающиеся агенты применяют ML, глубокое обучение, подкрепление (reinforcement learning).

Рассмотрим подробнее вариант функционирования интеллектуальных агентов, когда они образуют целую систему, приобретающую новые свойства, несводимые к свойствам отдельных агентов, в ней участвующих (эмерджентность).

Многоагентная система (MAS, англ. Multi-agent system) – это система, образованная несколькими взаимодействующими интеллектуальными агентами. Многоагентные системы могут быть использованы для решения таких проблем, которые сложно или невозможно решить с помощью одного агента или монолитной системы. Агенты могут быть программами, роботами или даже интеллектуальными датчиками, работающими в общей среде.

Ключевые характеристики многоагентных систем:

- автономность агентов – каждый агент принимает решения независимо от других на основе информации, которую он собирает из окружающей среды или получает от других агентов;

- взаимодействие агентов – агенты обмениваются информацией для координации своих действий, решения совместных задач или достижения общих целей. Взаимодействие может быть как прямым, так и косвенным (например, через общие ресурсы или пространство);

- распределенность – в отличие от централизованных систем, где управление и обработка данных происходят через центральное вычислительное устройство, в многоагентных системах обработка информации происходит распределенно между всеми агентами;

- гибкость – многоагентные системы легко адаптируются к изменениям внешней среды и могут быть динамически перестроены в зависимости от изменений в задачах или в данных.

У каждого агента свои функции: один собирает данные, другой обрабатывает их, третий взаимодействует с пользователем или другими системами. При этом могут использоваться различные сценарии взаимодействия: кооперативные (агенты стремятся к общей цели и обмениваются знаниями), конкурентные (у агентов разные цели, иногда даже противоположные), гибридные (сочетаются элементы сотрудничества и соперничества).

Архитектуры многоагентных систем варьируются от иерархических, где агенты подчиняются центральному узлу, до распределенных, где отсутствует единый центр управления, а взаимодействие происходит по горизонтальной схеме.

В России разработкой интеллектуальных агентов для эксплуатации в различных отраслевых задачах занимаются и технологические гиганты, такие как Сбер и Яндекс, и компании, которые выстраивают экосистемы, такие как МТС и Т-Банк, и IT-компании, например Just AI, Cognitive Technologies, NtechLab, VisionLabs. Перспективы и риски внедрения интеллектуальных агентов в бизнес-практику отечественных компаний рассмотрены в [32].

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ Р 59276–2020. Системы искусственного интеллекта. Способы обеспечения доверия. Общие положения. М. : Стандартинформ, 2021. 11 с.
2. ГОСТ Р ИСО/МЭК 24668–2022. Информационные технологии. Искусственный интеллект. Структура управления процессами аналитики больших данных. М. : Российский институт стандартизации, 2022. 49 с.
3. ГОСТ Р 71476–2024. Искусственный интеллект. Концепции и терминология искусственного интеллекта. М. : Российский институт стандартизации, 2024. 57 с.
4. ГОСТ Р 59898–2021. Оценка качества систем искусственного интеллекта. Общие положения. М. : ФГБУ «РСТ», 2021. 19 с.
5. ГОСТ Р ИСО/МЭК 25010–2015. Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов. М. : Стандартинформ, 2015. 29 с.
6. ГОСТ Р 71484.1–2024. Искусственный интеллект. Качество данных для аналитики и машинного обучения. Ч. 1. Обзор, терминология и примеры. М. : ФГБУ «РСТ», 2024. 18 с.
7. ГОСТ Р 71484.2–2024. Искусственный интеллект. Качество данных для аналитики и машинного обучения. Ч. 2. Показатели качества данных. М. : ФГБУ «РСТ», 2024. 35 с.
8. ГОСТ Р 71484.3–2024. Искусственный интеллект. Качество данных для аналитики и машинного обучения. Ч. 3. Требования и рекомендации по управлению качеством данных. М. : ФГБУ «РСТ», 2024. 23 с.
9. ГОСТ Р 71484.4–2024. Искусственный интеллект. Качество данных для аналитики и машинного обучения. Ч. 4. Структура процесса управления качеством данных. М. : ФГБУ «РСТ», 2024. 29 с.
10. GPT-4 is OpenAI's most advanced system, producing safer and more useful responses // OpenAI. URL: <https://openai.com/index/gpt-4>
11. Gemini. URL: <https://gemini.google.com>
12. AI research and products that put safety at the frontier. URL: <https://www.anthropic.com>
13. DeepSeek. URL: <https://www.deepseek.com/en>
14. ChatQwen. URL: <https://chat.qwen.ai/>
15. Qwen. URL: <https://www.queueonline.com/>

16. GigaChat. URL: <https://giga.chat>
17. YaGPT. URL: <https://ya.ru/ai/gpt>
18. Нейроэксперт. URL: <https://expert.ya.ru/expert>
19. Большие языковые модели теперь не требуют мощных серверов: ученые Яндекса, НИУ ВШЭ, MIT, KAUST и ISTA совершили прорыв в оптимизации LLM // Яндекс. URL: <https://yandex.ru/company/news/01-11-04-2025>
20. MTS AI. URL: <https://mts.ai/?ysclid=me5hs1u6cb281412560>
21. Cotype. URL: <https://mts.ai/ru/product/generative-ai-solutions/>
22. MWS GPT. URL: <https://mws.ru/services/mws-gpt/>
23. Т-Технологии. URL: <https://t-technologies.ru/?ysclid=me5hynltsr556194595>
24. T-tech's Collections. URL: <https://huggingface.co/collections/t-tech/t-pro-20-68712f1e775d0f7b563daf52>
25. Just AI. URL: <https://just-ai.com/>
26. Jay CoPilot. URL: <https://jaycopilot.com/business>
27. Что такое малые языковые модели (SLM)? // Microsoft Azure URL: <https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-are-small-language-models>
28. Как устроены малые языковые модели: эффект масштаба // РБК. URL: <https://trends.rbc.ru/trends/industry/668510909a7947e9acd8ffc3?from=copy>
29. Нейросеть Алиса. URL: <https://alice.yandex.ru/>
30. Boonstra L. Prompt Engineering // Kaggle. URL: <https://www.kaggle.com/whitepaper-prompt-engineering>
31. Основы промпт-инжиниринга // Сбербанк. URL: <https://developers.sber.ru/docs/ru/gigachat/prompts-hub/prompt-engineering?ysclid=me5rmbilol274003129>
32. Рындина С. В., Чернова И. И. Перспективы и риски внедрения интеллектуальных агентов в бизнес-деятельность отечественных компаний // Финансовый менеджмент. 2025. № 9. С. 249–257.

Учебное издание

Рындина Светлана Валентиновна

**Интеллектуальные информационные
системы и технологии. Большие языковые модели.
Промпт-инжиниринг и интеллектуальные агенты**

Редактор *М. М. Кузнецова*
Технический редактор *Н. В. Иванова*
Компьютерная верстка *Н. В. Ивановой*
Дизайн обложки *О. И. Жаровой*

Подписано в печать 25.11.2025.
Формат 60×84¹/₁₆. Усл. печ. л. 5,35.
Тираж 21. Заказ № 672.

Издательство ПГУ
440026, г. Пенза, ул. Красная, 40.
Тел.: (8412) 66-60-49, 66-67-77; e-mail: iic@pnzgu.ru

Вниманию авторов!

Издательство ПГУ выпускает учебную, научную и художественную литературу, презентационную и акцидентную продукцию, а также полноцветные юбилейные и мемориальные издания в соответствии с ГОСТ 7.60–2003.

Издательство ПГУ принимает к изданию рукописи, подготовленные с использованием текстового редактора Microsoft Word for Windows версий **2003 и выше**. Формат – А4, основной шрифт – Times New Roman, 14–16 pt через одинарный интервал (минимальный размер шрифта в таблицах и сносках – 12,5 pt). Тип файла в электронном виде – doc, docx.

Работа должна содержать индекс УДК, аннотацию.

Аннотация (ГОСТ 7.86–2003, ГОСТ 7.9–1995) включает характеристику основной темы, проблемы объекта, цели работы и ее результаты. В аннотации указывают, что нового несет в себе данный документ в сравнении с другими, родственными по тематике и целевому назначению. Аннотация может включать сведения о достоинствах произведения. Текст аннотации начинают фразой, в которой сформулирована главная тема документа. Заканчивается аннотация читательским адресом.

Рисунки и таблицы должны быть размещены в тексте после ссылки на них (растровые рисунки предоставляются в виде отдельных файлов в формате jpg, BMP с разрешением 300 dpi, векторные рисунки в формате Corel Draw с минимальной толщиной линии 0,75 pt. Рисунки должны быть доступны для правки!). Рисунки должны сопровождаться подрисуночными подписями, на все рисунки и таблицы в тексте должны быть ссылки.

Формулы в тексте выполняются только в редакторе формул **MathType версия 5.0** и выше. Символы греческого и русского алфавита должны быть набраны прямо, нежирно; латинского – курсивом, нежирно; обозначения векторов и матриц – прямо, жирно; цифры – прямо, нежирно. Наименования химических элементов набираются прямо, нежирно. Эти же требования необходимо соблюдать и в рисунках.

В списке литературы **нумерация источников** должна соответствовать очередности ссылок на них в тексте ([1], [2], ...). Номер источника указывается в квадратных скобках. Требования к оформлению списка литературы на русские и иностранные источники (ГОСТ Р 7.0.5–2008): для книг – фамилия и инициалы автора, название, город, издательство, год издания, том, количество страниц; для журнальных статей, сборников трудов – фамилия и инициалы автора, название статьи, полное название журнала или сборника, серия, год, том, номер, страницы; для материалов конференций – фамилия и инициалы автора, название статьи, название конференции, город, издательство, год, страницы.

К материалам **должна** прилагаться следующая информация: фамилия, имя, отчество, контактные телефоны.

Контакты Издательства ПГУ: т.: (8412) 66-60-49, 66-67-77, e-mail: iic@pnzgu.ru

