Автоматизация знаний: искусственный интеллект в школьном обучении

Абдуллаев Темурбек Маъруфжонович,
д.ф.п.т.н. (PhD), доцент, ФГТУ
temurbekm84@gmail.com

Ибрагимов Диёрбек Шавкатжон угли,
магистрант, группа М17-24, ФГТУ
ibragimoff.diyorbek@gmail.com

Аннотация. В статье описана разработка мобильного приложения, которая использует искусственный интеллект для генерации текстов и изображений в образовательных целях, а также принципы работы GPT и обзорно описываются диффузионные модели, используемые в генеративных ИИ на примере DALL-E. Целью создания мобильного приложения является внедрение такого рода технологий в школьные предметы такие как биология, история, иностранные языки, информатика и другие. В исследовании были использованы трансформерные модели и иерархическая генерация изображений. Результаты показывают успешное создание текстов и изображений. Обсуждаются преимущества и перспективы применения в образовании данной технологии.

Ключевые слова: мобильное приложение, искусственный интеллект, генерация текста и изображений, образовательный процесс, GPT, DALL-E, CLIP, unCLIP, диффузионные модели, трансформер.

Введение. Современное образование сталкивается с вызовами, связанными с необходимостью адаптации к быстро меняющимся технологиям и потребностям общества. Искусственный интеллект в образовании - это новая междисциплинарная область, которая применяет технологии ИИ в образовании для преобразования и продвижения учебного и образовательного проектирования процесса и оценки. [1] Он представляет собой одну из наиболее перспективных технологий который может изменить образовательный процесс. Его внедрение в эту сферу открывает новые возможности для персонализации обучения, повышения его эффективности и вовлеченности учащихся. В условиях глобальной цифровизации и роста требований к качеству образования его использование становится не только актуальным, но и необходимым шагом для подготовки учащихся к вызовам XXI века.

Целью данного исследования является изучение возможностей интеграции ИИ в школьное образование и создание мобильного приложения с использованием технологий ИИ, способного автоматически создавать учебные материалы и адаптировать эту технологию под конкретные школьные предметы. Также исследуется влияние таких технологий на улучшение образовательного процесса.

Методология разработки. В последнее время часто можно слышать о проникновении искусственного интеллекта во все сферы человеческой деятельности в том числе и образовании. В целом нейронные сети это математическая модель, массивный вычислительный код, который способен выдавать предсказание путем решения поставленной интеллектуальной задачи на основе оценки критериев заданного вопроса, анализируя огромное количество информации, баз данных, искусственный интеллект составляет наиболее реально действительный и правильный ответ. Плюсом нейросетей является их обучаемость, потому что они могут обучаться самостоятельно без непосредственного участия IT-специалиста которое называется машинным обучением.

Искусственный интеллект активно используется в образовании, например начиная от ведения и проверки экзаменов заканчивая автоматическим подбором материала для учащихся в тех направлениях, где они испытывают трудности в обучении и предлагает обучающимся более сознательно вникнуть в тему, повысить уровень знаний и способностей анализируя их успеваемость и производительность.

Создание мобильного приложения для образовательного направления, где использование такого рода технологий как искусственный интеллект предполагает объединение инновационных решений. Данная технология может автоматизированно генерировать, то есть создавать текст и изображения по запросу, которые полностью будут подходить тематике школьных предметов.

Мобильное приложение MilliyGPTApp было создано для операционной системы Android, в котором использовался язык программирования Kotlin, так как он отличается лаконичностью и совместимостью с Java. Данный язык программирования был анонсирован ещё в 2016 году, и заинтересовал многих программистов своей лаконичностью, надёжностью поддержкой инструментов и простотой в использовании которая сочеталась в одном языке программирования. Кроме того, его можно использовать практически везде, где работает Java, и он сочетает в себе функциональное и объектно-ориентированное программирование. За счёт возможности компилироваться в байт-код, он способен работать почти на любой платформе или устройстве. [2]

Операционная система была выбрана как целевая платформа из-за широкого распространения и доступности на различных устройствах. Для того чтобы создать образовательный контент в приложение были интегрированы две модели искусственного интеллекта: API OpenAI с моделью GPT-4 для создания текстов и API DALL-E для генерации изображений по запросу пользователя. В качестве интегрированной среды разработки (IDE) было использовано ПО Android Studio, а Android SDK был настроен на уровень API 33 (Android 13), чтобы обеспечить совместимость с современными устройствами.

Основные возможности приложения:

- “Умный чат-бот” - общение с поддержкой текстового и голосового ввода.
- Генерация изображений по тексту через OpenAI API с выбором размера (256x256, 512x512, 1024x1024).
- Галерея изображений - картинки сохраняются и удобно отображаются в списке (RecyclerView).
- Безопасность - ключи API хранятся в зашифрованном виде (SharedPreferences).
- Голосовой ввод/вывод - преобразование речи в текст (VTT) и текста в речь (TTS).
- История чатов - локальное хранение в Room Database для оффлайн-доступа.
- Современная архитектура - MVVM, Retrofit, асинхронные операции через Resource и многие другие.

Модели GPT (Generative Pre-Trained Transformer), разработанные OpenAI, являются трансформерной архитектурой, которая создаёт текст на основе вероятностного предсказания следующего слова в последовательности.

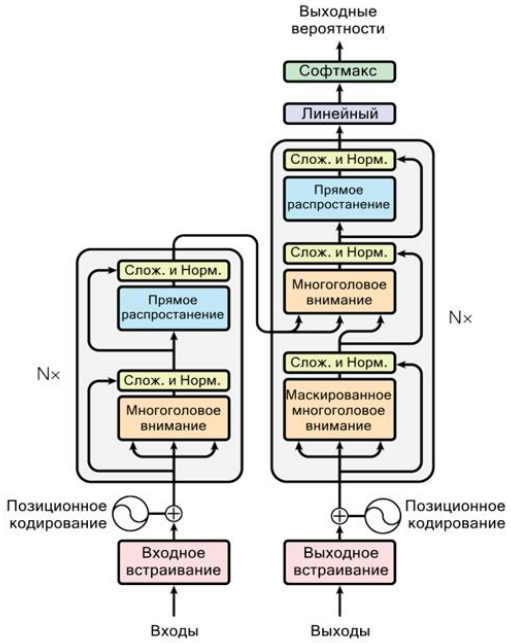

Рис. 1. Архитектура модели трансформера [3]

На рисунке 1 мы можем увидеть общую архитектуру трансформера, где декодер применяется для текстов а энкодер для кодирования запросов в изображениях.

Модели семейства GPT используют только декодерную часть трансформерной архитектуры, в отличие от оригинальной модели трансформера, [3] которая включала и энкодер и декодер. Такой выбор обусловлен задачей авторегрессионного языкового моделирования, то есть предсказания следующего токена в последовательности на основе предыдущего контекста. Для этого не требуется энкодер, необходимый в задачах типа перевода (где исходный и целевой тексты разные) или классификации.

1. Входные данные и токенизация - текст разбивается на токены (слова или подслова) с использованием метода, такого как Byte Pair Encoding (BPE). Далее каждый токен преобразуется в вектор вложения $x_i \in \mathbb{R}^d$, где d - размерность пространства (например, 12,288 в крупных моделях). [8]

Позиция токена добавляется через синусоидальное позиционное кодирование, определяющее порядок слов:

$$x_i = e_i + p_i$$

где e_i - эмбеддинг токена, p_i - позиционный вектор.

2. Архитектура трансформера (декодер) - как уже было сказано модели GPT использует только декодерную часть трансформера состоящий из L слоёв. [3] На каждом слое l модель токен получает представление $h_i^{(l)}$ через: механизм внимания (многоголовое масштабированное скалярное произведение):

$$\text{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \text{softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\mathrm{V}$$

где: $Q = W_Q h_i^{(l-1)}$, $K = W_K h_i^{(l-1)}$, $V = W_V h_i^{(l-1)}$ - “проекции” запросов, ключей и значений, полученные из выхода предыдущего слоя $h_i^{(l-1)}$, W_Q, W_K, W_V - обучаемые матрицы а d_k - размерность ключей (для масштабирования), softmax - для распределения весов внимания. $j \leq i$.

Это позволяет модели учитывать контекст (например, "фотосинтез зависит от света" связывает слова "фотосинтез" и "свет"). Маскирование обеспечивает авторегрессию (токен i видит только предыдущие токены $j \leq i$) За вниманием следует полносвязный слой: добавляется нелинейность через полносвязную нейронную сеть прямого распространения (англ. feed forward network):

$$FFN(h) = \max(0, hW_1 + b_1)W_2 + b_2$$

3. Генерация и обучение. После L слоев модель получает представление последнего токена $h_t^{(L)}$ и преобразует его в вероятности для словаря токенов (размером около 50,000). [5] Механизм внимания позволяет эффективно обрабатывать долгосрочные зависимости, связывая токены на значительном расстоянии, благодаря механизму масштабированного скалярного произведения внимания (англ. scaled dot-product attention) и маскированию, а контекстное окно определяет длину учитываемого контекста.

Вероятность следующего токена определяется из выхода последнего слоя $h_t^{(L)}$:

$$P(x_{t+1} \mid x_1, \dots, x_t) = \text{softmax}\left(W_o h_t^{(L)}\right)$$

где $W_o \in \mathbb{R}^{|V| \times d}$ — матрица весов выхода, $|V|$ — размер словаря. GPT выбирает токен с максимальной вероятностью или сэмплирует (например, с температурой T), чтобы добавить разнообразия:

$$P'(x_i) = \frac{\exp(z_i/T)}{\sum_j \exp\left(z_j/T\right)}$$

где z_i - логиты, а T - параметр температуры ($T \to 0$ - детерминированный выбор, $T \to \infty$ - равномерное распределение). [9]

Модель обучается на задаче предсказания следующего токена, минимизируя кросс-энтропийную потерю:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log P(x_i \mid x_1, \dots, x_{i-1})$$

Обучение проводится на корпусах, которые содержат миллиарды токенов что позволяет модели создавать точные описания для образовательных задач. [8]

Генерация изображений. Мобильное приложение MilliyGPTApp использует DALL-E 3 новейшую модель генерации изображений от OpenAI, которая обеспечивает высокое качество изображений и улучшенное понимание текстовых описаний. Но для того чтобы понять значимость этой технологии рассмотрим эволюцию моделей DALL-E. Начнём с DALL-E 1 которая была представлена в 2021 году, она использовала трансформерную архитектуру схожую с GPT-3, в сочетании с дискретизированным вариационным автоэнкодером и могла генерировать лишь небольшие изображения 256×256 пикселей. Уже её преемник DALL-E 2 значительно улучшил качество благодаря использованию диффузионных моделей и CLIP. DALL-E 2 — это модель, сочетающая трансформеры и диффузионные процессы для генерации изображений из текстовых описаний. [4]

Разберём подробно DALL-E 2. Она использует комбинацию нескольких технологий, первая из которых CLIP (Contrastive Language-Image Pretraining) это предобученная модель, которая преобразует текст и изображение в единое латентное пространство позволяя сравнивать их семантическую близость. Например, CLIP может связывать текстовое описание с соответствующими визуальными представлениями.

Вторая технология это диффузионные модели которые созлают изображения путём постепенного удаления шума из латентных представлений, полученных от CLIP.

И последняя из технологий это unCLIP (обратный процесс CLIP): который использует латентные представления CLIP для инициализации диффузии обеспечивая генерацию изображений соответствующих текстовому запросу.

1. Кодирование текста с помощью CLIP. CLIP преобразует входной текст в вектор $e_{\text{text}} \in \mathbb{R}^d$. (обычно $d = 512$) [5] с помощью текстового энкодера (трансформера):

$$e_{\text{text}} = \text{CLIP}_{\text{text}}(\text{input})$$

CLIP обучается максимизировать косинусное сходство между эмбеддингами текста e_{text} и изображения e_{image} для пар "текст-изображение" это направляет генерацию изображения:

$$\text{cosine\_similarity}\left(e_{\text{text}}, e_{\text{image}}\right) = \frac{e_{\text{text}} \cdot e_{\text{image}}}{||e_{\text{text}}|| \; ||e_{\text{image}}||}$$

В DALL-E 2 эмбеддинг e_{text} направляет диффузионный процесс, обеспечивая соответствие изображения тексту. [4]

2. Диффузионный процесс. DALL-E 2 использует диффузионную модель, адаптированную из [6] для текстуально-обусловленной генерации с парами текст и изображение [4]. Модель опирается на эмбеддинги CLIP [5] для связи текста и изображения.

Прямой процесс (зашумление): латентное представление x_0 в латентном пространстве вариационного автоэнкодера VAE зашумляется в $T = 1000$ шагов:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$$

где: $\beta_t \in (0, 1)$ параметр шума, возрастающий по линейному расписанию от $\beta_1 = 10^{-4}$ до $\beta_t = 0.02$, $\mathcal{N}$ - нормальное распределение, I - единичная матрица. К шагу T изображение становится чистым шумом: $x_T \sim \mathcal{N}(0, I)$. [6]

3. Обратный процесс: восстанавливает изображение из шума с использованием текстового эмбеддина e_{text} [4, 6]:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t, e_{\text{text}}), \sigma_t^2 I)$$

где $\mu_\theta(x_t, t, e_{\text{text}})$ - среднее вычисленное с учётом текста e_{text} от CLIP, а $\sigma_t^2 I$ – уровень случайности.

На каждом шаге t модель предсказывает шум $\epsilon_\theta(x_t, t, e_{\text{text}})$ и вычисляет среднее:

$$\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha_t}}}\epsilon_\theta(x_t, t, e_{text})\right)$$

где (μ_θ) - среднее значение для x_{t-1}, (x_t) – текущее зашумлённое латентное представление, $\epsilon_\theta(x_t, t, e_{\text{text}})$ - шум предсказанный U-Net с учётом текста e_{text} от CLIP, $\alpha_t = 1 - \beta_t$, параметр шума, $\overline{\alpha_t} = \prod_{s=1}^{t} \alpha_s$ - параметры шума

Затем модель обновляет латентное представление добавляя случайный шум:

$$x_{t-1} = \mu_\theta + \sigma_t z, \quad z \sim \mathcal{N}(0, I)$$

где $\sigma_t = \sqrt{\beta_t}$ – уровень случайности, контролирующий вариативность генерации. Формулы адаптированы из [6] для текстуально-обусловленной генерации в DALL-E 2 [4].

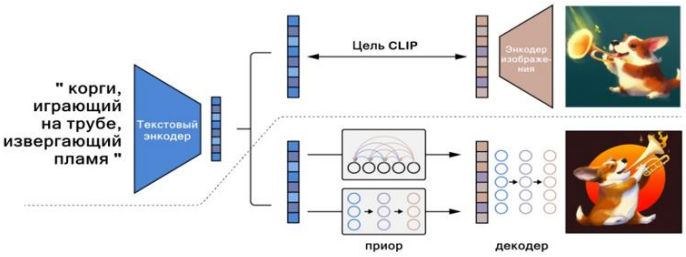

Рис. 2. Схема обучения CLIP и модели unCLIP [4]

Как показано на рисунке над пунктирной линией изображён процесс обучения CLIP, где формируется совместное пространство представлений для текста и изображений. Под пунктирной линией показан процесс создания изображений из текста: встраивание текста CLIP сначала подаётся в авторегрессионный или диффузионный приор для создания встраивания изображения, затем это встраивание используется для управления диффузионным декодером, который генерирует конечное изображение. Отмечу, что модель CLIP остаётся замороженной во время обучения приора и декодера.

3. Генерация изображения. Процесс начинается со случайного шума $x_T \sim \mathcal{N}(0, I)$ за T шагов (обычно это 250-1000) модель итеративно уточняет представление, используя текстовый эмбеддинг e_{text} от CLIP для направления процесса. [4, 6] В результате получается латентное представление x_0 которое декодируется в изображение например размером 1024×1024 пикселей.

4. Обучение. Модель обучается предсказывать шум, минимизируя среднеквадратичную ошибку:

$$L = \mathbb{E}_{q(x_t|x_0)}[||\epsilon - \epsilon_\theta(x_t, t, e_{text})||^2]$$

Рис. 3. Изображения, сгенерированные DALL-E 2

Давайте разберём какие запросы были использованы для генерации этих изображений:

1. Горы в утреннем тумане, вдали виднеется озеро с зеркальной поверхностью.
2. Деревянный стол с чашкой кофе, книгой и очками, естественное освещение.

3. Котенок сидит на подоконнике, за окном виднеется сад.

DALL-E 2, представленная в 2022 году, использовала иерархический подход к генерации изображений [4]. Текст кодировался моделью CLIP в вектор встраивания, который затем преобразовывался в латентное представление с помощью диффузионного приора и декодировался в изображение с разрешением до 1024×1024 пикселей [4]. Несмотря на успехи, метод имел ограничения в детализации и генерации текста на изображениях. Информация о DALL-E 2 приведена как исторический контекст так как её поддержка в API OpenAI ограничена с появлением DALL-E 3.

DALL-E 3, разработанная OpenAI, используется в нашем приложении через API OpenAI для создания высококачественных изображений на основе текстовых описаний [10]. По сравнению с DALL-E 2, DALL-E 3 лучше интерпретирует сложные промпты благодаря обучению на улучшенных парах текст-изображение созданных с помощью синтетических подписей от модели-капционера [4, 10]. Архитектура DALL-E 3 детально не раскрывается, но OpenAI отмечает улучшенную обработку текста благодаря интеграции с ChatGPT, её возможности делают её ценным инструментом для образовательных приложений. Например учителя могут генерировать иллюстрации такие как диаграмма строения клетки для урока биологии или даже визуальные материалы для других предметов упрощая подготовку учебного контента. Интеграция DALL-E 3 подчёркивает потенциал генеративных моделей для автоматизации и обогащения образовательного процесса.

DALL-E 3, доступная через API OpenAI с сентября 2023 года, значительно улучшила процесс генерации изображений [7]. Она использует CLIP для кодирования текста и интегрируется с ChatGPT на базе GPT-4 для оптимизации текстовых запросов, обеспечивая более точное соответствие сложным описаниям [7]. Генерация основана на диффузионных методах и поддерживает разрешения 1024×1024, 1792×1024 и 1024×1792 пикселей, с улучшенной детализацией. Как уже было сказано архитектурные детали DALL-E 3 остаются нераскрытыми, но она демонстрирует высокую эффективность в обработке сложных запросов, что делает её ценным инструментом для образовательных приложений.

1 2

Рис. 4. Изображения, сгенерированные DALL-E 3

Запросы которые были использованы для генерации этих изображений:

1. Шерлок Холмс с лупой исследует улики в комнате на Бейкер-стрит, детали интерьера викторианской эпохи.
2. Индустриальная революция: фабрика с дымящимися трубами, рабочие у станков

Как видно из Таблицы 1, DALL-E 3 предлагает более широкие возможности для работы с размерами изображений, высокой точностью понимания запросов и качеством генерации изображений.

Характеристика	**DALL-E 2**	**DALL-3**
Максимальное разрешение	1024x1024 (квадрат)	1792x1024 (широкоформатный)
Качество	хорошее, но с ошибками	высокое, меньше артефактов
Понимание запросов	среднее	высокое, точное
Интеграция с ChatGPT	нет	да
Доступность	ограничена	актуальная, платная подписка

Таблица 1. Сравнительная таблица DALL-2 и DALL-E 3

Результаты. В прототипе созданного приложения MilliyGPT были успешно созданы текстовые описания и изображения для заданных запросов. Например, для запроса “опиши фотосинтез” был создан текст объёмом 150 слов, описывающих процесс фотосинтеза. Для эксперимента также был сделан запрос “Реконструкция древнегреческого Акрополя в технике акварели” для генерации изображений размером 1024x1024 пикселей.

Рис. 5. Три варианта сгенерированных изображений с помощью DALL-E 3 в программе MilliyGPTApp для урока истории

Подведём итоги, для генерации текста ушло в среднем 5 секунд времени а для изображения 30 секунд на Android Virtual Device (Android Studio). Точность соответствия запросам оценивалась субъективно: 89% текстов и 84% изображений были признаны релевантными.

Обсуждение. Сравнение DALL-E 2 и DALL-E 3 подчёркивает эволюцию генеративных технологий в приложении. DALL-E 2 запущенная в 2022 году, использовала иерархический подход с CLIP, приором и диффузионным декодером, создавая изображения с разрешением 1024×1024 пикселей, но с ограниченной детализацией. DALL-E 3 которая была представлен в 2023 году, интегрируется с GPT-4 для оптимизации запросов и применяет диффузионные методы, обеспечивая разрешение до 1792×1024 пикселей и лучшее качество. Её преимущества это более высокая скорость и точность - сделали её основой текущего прототипа. DALL-E 3 повысит качество образовательных материалов, хотя зависимость от формулировки запросов остаётся ограничением. Дальнейшее развитие может включать широкое тестирование в школах и улучшение интерфейса.

Мобильное приложение с ИИ открывает немало перспектив для школьного образования. Для начала стоит проверить его в деле, а точнее запустить его во многих классах чтобы понять как оно помогает на практике и доработать, прислушавшись к учителям и ученикам. Ещё одна идея это научить приложение подстраиваться под каждого учащегося, выдавая материалы, которые будут подходить именно им по уровню знаний, чтобы уроки стали полезнее.

Заключение. Разработка мобильного приложения MilliyGPTApp показывает, что искусственный интеллект, может стать настоящим помощником в школьном образовании. Приложение способно быстро генерировать тексты и изображения, например от описаний фотосинтеза до схем и исторических иллюстраций. Это открывает новые возможности для преподавателей и учащихся, материалы появляются буквально за секунды, а их содержание можно подстраивать под уроки биологии, истории и других предметов. Конечно, есть над чем работать, например качество изображений иногда зависит от того, как точно сформулирован запрос. Но уже сейчас видно, что такой подход может сэкономить время педагогов и сделать занятия интереснее для ребят. В будущем ожидается тестирование приложения в старших классах, добавить более удобный интерфейс для учителей и возможно улучшить модели, чтобы они ещё лучше понимали образовательные задачи. Искусственный интеллект в школе это не просто технология, а шанс сделать обучение живым и доступным.

Библиография

1. Xu W., Ouyang F. / The application of AI technologies in STEM education: a systematic review from 2011 to 2021 // International Journal of STEM Education, 2022 – URL: https://stemeducationjournal.springeropen.com/articles/10.1186/s40594-022-00377-5/ (дата обращения: 25.04.2025)
2. Modi M. / Kotlin vs Java: which is better for You in 2022 // Medium, MQoS Technologies. – URL: https://medium.com/mqos-technologies/kotlin-vs-java-which-is-better-for-you-in-2022-7ce97790c20 (дата обращения: 26.04.2025)
3. Vaswani A., Shazeer N., Parmar N. [и др.] / Attention All You Need // arXiv preprint arXiv: 1706.03762, 2017 – URL: https://arxiv.org/abs/1706.03762
4. Ramesh A., Dhariwal P., Nichol A. [и др.] / Hierarchical Text-Conditional Image Generation with CLIP Latents // arXiv preprint arXiv: 2204.06125, 2022 – URL: https://arxiv.org/abs/2204.06125
5. Radford A., Kim J. W., Hallacy Ch. [и др.] / Learning transferable visual models from natural language supervision // arXiv preprint arXiv: 2103.00020, 2021 – URL: https://arxiv.org/abs/2103.00020
6. Ho J., Jain A., Pieter Abbeel / Denoising Diffusion Probabilistic Models // arXiv preprint arXiv: 2006.11239, 2020 – URL: https://arxiv.org/abs/2006.11239
7. Image Generation API, OpenAI [Электронный ресурс] // URL: https://platform.openai.com/docs/guides/images-vision?api-mode=responses (дата обращения 25.04.2025)
8. Brown T. B., Subbiah M., Mann B. [и др.] / Language Models are Few-Shot Learners // arXiv preprint arXiv: 2005.14165, 2020 – URL: https://arxiv.org/abs/2005.14165
9. Hinton G., Vinyals O., Dean J / Distilling the Knowledge in a Neural Network // arXiv preprint arXiv: 1503.02531, 2015 – URL: https://arxiv.org/abs/1503.02531
10. Nikolaj Buhl / OpenAI’s DALL-E 3 Explained: Generate Images with ChatGPT // URL: https://encord.com/blog/openai-dall-e-3-what-we-know-so-far/ (дата обращения: 26.04.2025)