МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

ИНТЕГРАЦИЯ БОЛЬШОЙ ЯЗЫКОВОЙ МОДЕЛИ В ИГРОВОЕ ПРИЛОЖЕНИЕ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Мазанова Максима Александровича

Научный руководитель
доцент, к. ф.-м. н. _____ А. С. Иванова

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2024

ВВЕДЕНИЕ

Актуальность темы. В современном мире большие языковые модели (LLM) становятся все более распространенными и широко применяемыми. Они представляют собой инновационные технологии, основанные на принципах машинного обучения и нейронных сетей, способные анализировать и генерировать естественный язык с высокой точностью и контекстуальной гибкостью. От чат-ботов для обслуживания клиентов до помощников врачей и виртуальных репетиторов, LLM активно применяются в самых разных областях, трансформируя способы взаимодействия человека с технологиями.

В этом контексте внедрение больших языковых моделей в компьютерные игры представляет собой уникальную и перспективную область исследований. Игры, как важный сегмент развлекательной индустрии, постоянно ищут новые способы обогащения игрового опыта и повышения уровня вовлеченности игроков. Внедрение LLM в игровой процесс может открыть широкие возможности для создания более реалистичных и интерактивных игровых миров, где игроки могут взаимодействовать с виртуальными персонажами посредством прямых разговоров. Исследование данной темы расширит понимание о возможностях и перспективах использования LLM в компьютерных играх, а также поспособствует развитию инновационных подходов к улучшению пользовательского опыта в игровых проектах.

Цель бакалаврской работы — исследовать и оценить возможность внедрения больших языковых моделей в компьютерные игры для улучшения интерактивного взаимодействия и погружения игроков.

Поставленная цель определила **следующие задачи**:

1. Исследовать теоретические основы работы больших языковых моделей.
2. Разработать прототип компьютерной игры на базе движка Godot с интеграцией большой языковой модели.
3. Проанализировать влияние интеграции большой языковой модели в игровое приложение на качество взаимодействия и погружение игрока в игровой процесс.
4. Сформулировать выводы и рекомендации для дальнейшего развития и эффективного применения больших языковых моделей в игровых приложениях.

Методологические основы работы с ключевыми фразами представлены в работах D. A. Roberts, S. Yaida и B. Hanin, S. Amari, L. Tunstall, L. Von Werra и T. Wolf, A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser и I. Polosukhin, R. Gallotta, G. Todd, M. Zammit, S. Earle, A. Liapis, J. Togelius и G. N. Yannakakis.

Теоретическая значимость бакалаврской работы. В работе проведен обзор теоретических основ нейронных сетей и больших языковых моделей, построенных на архитектуре трансформеров. Рассмотрены различные методики применения данных моделей в контексте игровых приложений, включая потенциальные подходы. Значимость исследования обусловлена необходимостью глубокого понимания принципов работы больших языковых моделей. Детальное изучение архитектуры трансформеров способствует пониманию их функционирования и предоставляет базу для разработки инновационных решений в игровых приложениях.

Практическая значимость бакалаврской работы. Разработано игровое приложение на движке Godot Engine с интеграцией большой языковой модели через text-generation-webui. Созданный прототип может послужить основой для разработки игр в жанре Action-RPG и представляет собой экспериментальную платформу для проверки концепции интеграции больших языковых моделей в качестве диалоговых партнеров для игроков через неигровых персонажей. Это позволяет на практике оценить эффективность использования языковых моделей для создания динамичных и интерактивных игровых сценариев.

Структура и объем работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы — 83 страницы, из них 52 страницы — основное содержание, включая введение, основные разделы и заключение. Работа содержит 27 рисунков, цифровой носитель в качестве приложения, список использованных источников из 21 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретические основы и применение больших языковых моделей» посвящен обзору глубокого обучения и обработки естественного языка, трансформерам, а также методам применения больших языковых моделей в игровых приложениях.

Глубокое обучение. Нейронные сети — «рецепт» для вычисления функции, состоящий из множества вычислительных единиц, называемых нейронами. Каждый нейрон сам по себе является очень простой функцией, которая рассматривает взвешенную сумму входящих сигналов и затем срабатывает характерным образом, сравнивая значение этой суммы с некоторым порогом. Глубокие нейронные сети имеют несколько последовательных слоев. Градиентный спуск используется для минимизации функции ошибки, направляя модель к минимальной ошибке. Алгоритм обратного распространения ошибки дополняет градиентный спуск, предоставляя метод для вычисления градиентов параметров сети.

Обработка естественного языка. Чтобы модели могли понимать язык, слова представляются в виде чисел. Токенизация разбивает текст на токены, которые преобразуются в числовые векторы. Эти векторы инициализируются случайным образом и адаптируются в процессе обучения, отражая значения токенов и их связи с другими токенами в словаре.

Большие языковые модели и трансформер. Большие языковые модели обучаются на огромных объемах текстовых данных и способны генерировать связные тексты. Основная идея языковой модели заключается в способности предсказывать следующее слово или токен на основе текста, который она уже видела. Модель предсказывает токены по одному, используя вероятности, присвоенные каждому токену на основе обучения. Этот процесс повторяется до появления специального токена <stop>.

В основе больших языковых моделей лежит архитектура трансформера, одним из основных механизмов которого является механизм самовнимания. Когда модель обрабатывает каждое слово в предложении, самовнимание позволяет ей учитывать другие позиции в этом же предложении для улучшения кодирования текущего слова. Самовнимание позволяет трансформеру учитывать все слова одновременно, что обеспечивает более глубокое понимание контекста.

Работа оригинального трансформера определяется следующим образом:

— Энкодер обрабатывает входное предложение и генерирует набор векторов ключей K и значений V. Декодеры в слое внимания «энкодер-декодер» используют эти векторы для акцентирования внимания на соответствующих частях входного предложения. На каждом этапе фазы декодирования генерируется элемент выходной последовательности.
— Генерация выходной последовательности продолжается до появления специального символа, сигнализирующего о завершении процесса. Выход каждого этапа передается на нижний уровень декодера в следующем временном отрезке, и декодеры генерируют свои результаты аналогично энкодерам. Кроме того, как и с входами энкодера, позиционное кодирование добавляется к входам декодера, указывающим на позицию каждого слова.
— В слое внутреннего внимания декодера акцент сосредотачивается только на предыдущих позициях в выходной последовательности. Для этого перед этапом вычисления внутреннего внимания все позиции после текущей маскируются.
— Слой внимания «энкодер-декодер» действует аналогично механизму многоголового внимания, за исключением того, что он формирует матрицу запросов из слоя, расположенного ниже, и использует матрицы ключей и значений из выхода стека энкодеров.
— Декодеры на выходе возвращают вектор чисел с плавающей точкой, требующие дополнительной обработки полносвязной нейронной сетью, переводящей вектор в логиты (линейный слой), и далее слоем Softmax, переводящем логиты в нормализованные положительные последовательности, из которых выбирается ячейка с наибольшей вероятностью.

Использование больших языковых моделей в игровых проектах. Большие языковые модели в играх выполняют различные роли: они могут действовать как игроки, управлять игровыми диалогами и даже помогать разработчикам.

В роли игрока модели могут использоваться в тех приложениях, где доступно текущее состояние игры и возможно для описания модели, ввод/вывод осуществляется на естественном языке или имеется API, позволяющее управлять игровым процессом в реальном времени.

Модели также применяются для генерации диалогов неигровых персонажей, предлагая динамичные и контекстуальные ответы на ввод от игрока. Это

повышает глубину взаимодействия с игровым миром и уменьшает монотонность повторяющихся разговоров.

Третьим способом применения больших языковых моделей является использование в качестве игрового мастера. Модели могут стать инструментом для того, чтобы привести цифровые ролевые игры на уровень обычных настольных.

Другими возможными способами применения больших языковых моделей являются: роль игрового ассистента, дающего игроку необходимые игровые подсказки, роль комментатора, описывающего события игры в реальном времени, а также роль ассистента разработчика, используемого как инструмент при разработке игрового приложения.

Большие языковые модели представляют собой потенциально мощный инструмент при разработке игр, позволяющий их использование при разных требованиях и вариациях. Применение больших языковых моделей в игровых приложениях расширяет возможности и повышает взаимодействие игрока с виртуальным миром.

Второй раздел «Реализация игрового приложения и внедрение модели» посвящен реализации игрового приложения на движке Godot Engine с интеграцией функциональности большой языковой модели, а также оценке полученных результатов.

Используемые технологии. В качестве игрового движка для разработки приложения будет использован движок с открытым исходным кодом Godot. Этот выбор обусловлен простотой освоения и интуитивно понятным интерфейсом, что сокращает время разработки и упрощает создание игр. Godot обладает большой встроенной функциональностью для работы с 2D и 3D графикой, анимациями, физикой и сетевыми взаимодействиями, что делает его подходящим для реализации основных игровых механик. Также движок поддерживается большим и разнообразным сообществом, предоставляющим множество ресурсов, таких как форумы и руководства, что помогает в решении возникающих проблем.

Для взаимодействия с большими языковыми моделями используется приложение text-generation-webui от oobabooga. Этот инструмент был выбран за удобный совместимый с OpenAI API, что упрощает интеграцию и позволяет сосредоточиться на логике игры и сценариях взаимодействия. Text-generation-

webui поддерживает различные backend-библиотеки, обеспечивая гибкость в выборе языковых моделей. Кроме того, он имеет функцию загрузки описаний персонажей в YAML-формате, что позволяет быстро настраивать модель для генерации диалогов персонажей в игре.

Для данного проекта был выбран локальный подход к использованию больших языковых моделей, поскольку он обеспечивает автономность и контроль над процессом, избегая проблем с доступностью серверов и необходимостью постоянного интернет-соединения. Это также позволяет проводить эксперименты без дополнительных затрат на серверную инфраструктуру. Однако, ограниченность ресурсов компьютера игрока требует использования моделей с оптимизированной производительностью и низкими требованиями.

Для проекта выбрана модель Mistral 7B, благодаря её эффективной архитектуре и продвинутому датасету. В данном проекте используется квантизированная fine-tuned модель OpenHermes 2.5 Mistral 7B, версия Q5_K_M GGUF от TheBloke с 5-битной квантизацией, что обеспечивает оптимальное использование ресурсов.

Также рассматривается модель TinyLlama, обладающая 1.1 миллиарда параметров и обученная на смеси данных SlimPajama и Starcoder. TinyLlama использует ту же архитектуру и токенизатор, что и Llama 2, что облегчает интеграцию с open-source проектами. Модель подходит для генерации диалогов для NPC в реальном времени, что делает её полезной для эксперимента.

Системные инструкции. Для описания персонажей для модели используются системные инструкции, которые удобно записываются в соответствующий YAML-файл особого формата, понимаемого text-generation-webui. Таким образом реализован персонаж Andrew, торговец, у которого есть задание для игрока. Подготовлено несколько системных инструкций для персонажа: до получения задания и после.

Системные инструкции позволяют модели понять, что от нее требует пользователь. Обычно чат-бот с большой языковой моделью изначально содержит инструкцию, которая предписывает модели действовать как помощник искусственного интеллекта для пользователя, быть креативной и полезной. Однако их можно использовать и для описания других персонажей со своими особенностями и заданиями.

Интеграция взаимодействия с большой языковой моделью. Интегра-

ция реализована с помощью запросов от приложения к API text-generation-webui. В качестве запроса посылается POST-запрос /v1/chat/completions с текущим контекстом, температурой, максимальным количеством токенов, необходимой моделью и другими параметрами. После генерации ответа в диалоговой системе игры отображается принятое сообщение.

Игровое приложение. Для реализации игрового приложения на движке Godot создаются следующие сцены и системы:

- Сцена Player. Разработана для представления игрока с функциями передвижения, атаки врагов, получения урона и взаимодействия с неигровым персонажем.
- Сцена Slime. Отображает враждебное существо, способное перемещаться между двумя точками в игровом мире и атаковать игрока.
- Сцена NPC. Обеспечивает визуальное представление и интерактивность с неигровым персонажем, включая параметры: имя, состояния персонажа как системные инструкции, задания и другие.
- Базовая сцена игры. Интегрирует все описанные выше элементы, а также обеспечивает взаимодействие между различными элементами с помощью системы сигналов.
- Сцена Dialogue. Реализует диалоговую систему UI для взаимодействия с неигровым персонажем путем разговоров. В данной сцене в качестве одного из элементов используется узел HTTPRequests, посылающий запросы модели.
- Квестовая система. Игровые сцены Quest, QuestHandler и ресурсы с игровыми заданиями разработаны в качестве системы заданий для использования вместе с неигровым персонажем.

Достоинства и недостатки интеграции. Положительные моменты включают возможность «разговора» с неигровым персонажем в реальном времени, что повышает вовлеченность игроков и улучшает их понимание игрового мира. Модель позволяет создавать сложные механики NPC, такие, как динамические системы кармы, без значительного увеличения объема данных. Многоязычность модели позволяет избежать необходимости локализации диалогов, улучшая игровой опыт для международной аудитории. Также игра может использовать любую модель генерации текста, что делает процесс разработки более гибким и эффективным.

Недостатки использования больших языковых моделей также имеются. Непостоянство ответов и галлюцинации моделей могут нарушить игровой процесс. Важные моменты должны быть зафиксированы напрямую в игре, и для этого потребуется мощный фильтр для отсева нежелательных реплик. Запросы, которые могут нарушить игровой процесс, должны быть запрещены через умный фильтр, блокирующий вопросы о запрещённых темах.

Кроме того, модель может генерировать ложную информацию, разрушая погружение в игру. Важные реплики должны быть заранее подготовлены и фиксированы в игре, а не генерироваться моделью. Это поможет сохранить целостность игрового опыта и избежать ошибок.

Использование больших языковых моделей в играх имеет свои плюсы и минусы. Модель может обогатить игровой мир, если NPC не являются ключевыми для сюжета. Для важных сюжетных персонажей необходимо ограничить взаимодействие с моделью, чтобы обеспечить стабильное прохождение игры.

ЗАКЛЮЧЕНИЕ

В ходе работы были изучены теоретические основы больших языковых моделей, включая архитектуру трансформеров и их применение в различных областях. На базе движка Godot был разработан прототип компьютерной игры с интеграцией большой языковой модели, которая использовалась для взаимодействия игрока с неигровым персонажем.

В процессе исследования было проанализировано влияние интеграции LLM на качество взаимодействия и степень погружения игрока в игровой процесс. Оценка показала, что большие языковые модели могут значительно улучшить интерактивность и реализм NPC в ролевых и приключенческих играх. Однако выявлены и проблемы, такие как галлюцинации модели, которые остаются серьезной проблемой и требуют разработки дополнительных механизмов фильтрации и блокировки для предотвращения неэтичных или разрушающих игровой процесс действий.

По итогам работы сформулированы выводы и рекомендации для дальнейшего развития и эффективного применения больших языковых моделей в игровой индустрии. В частности, внедрение LLM в игры открывает новые возможности для создания интерактивных и иммерсивных игровых опытов, хотя требует специфических знаний в области глубокого обучения, обработки данных и prompt-инжиниринга.

Таким образом, данное исследование подтверждает, что большие языковые модели обладают значительным потенциалом для использования в игровой индустрии, способствуя созданию новых игровых жанров и улучшению существующих.

Основные источники информации:

1. *Roberts, D. A.* The principles of deep learning theory / D. A. Roberts, S. Yaida, B. Hanin. — Massachusetts, USA: Cambridge University Press, 2022. — Vol. 46.
2. *Amari, S.* Backpropagation and stochastic gradient descent method / S. Amari // Neurocomputing. — 1993. — Vol. 5, no. 4-5. — Pp. 185–196.
3. *Tunstall, L.* Natural language processing with transformers / L. Tunstall, L. Von Werra, T. Wolf. — California, USA: O'Reilly Media, Inc., 2022.
4. Attention is all you need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin // *Advances in neural information*

processing systems. — 2017. — Vol. 30.

5. *Gallotta, R.* Large language models and games: A survey and roadmap / R. Gallotta, G. Todd, M. Zammit, S. Earle, A. Liapis, J. Togelius, G. N. Yannakakis // *arXiv preprint arXiv:2402.18659.* — 2024