1. Титульный лист

“Платформа по созданию ИИ ассистентов”

1. Введение в предметную область

Chat gpt это языковая модель для обработки и генерации текста на естественном языке. Можно запрограммировать цепочки общения с ним и использовать для частичной автоматизации процессов бизнеса. Автономные такие решения называют ИИ агентами, а решения, с которыми может взаимодействовать пользователь - ИИ ассистентами.

Проще говоря, ИИ ассистенты — это программы, которые помогают людям выполнять различные задачи с использованием искусственного интеллекта. Примеры включают голосовых помощников (например, Яндекс Алиса), чат-ботов на сайтах, отвечающих на часто задаваемые вопросы, помощников в бизнес-аналитике, предсказывающих, какие товары будут популярны, и так далее.

1. Проблемы

Чтобы модель давала хорошие ответы на запросы, надо хорошо эти запросы формулировать. Иногда бывает очень сложно объяснить модели, что от ней требуется, и приходится писать промпты (то есть запросы) на тысячи слов. К тому же существуют различные техники для улучшения запросов. С учётом всего этого задача написания хороших промптов становится очень сложной, и сами промпты представляют большую ценность в проектах, использующих ИИ. Соответственно возникают проблемы хранения и конфигурации промптов.

Создание своего ИИ ассистента путём написания кода – это не очевидная задача, а существующие преймворки по созданию ассистентов зачастую запутаны либо скрывают логику работы с моделями в деталях реализации, что ухудшает гибкость. И это тоже проблема.

1. Решение – графы

Мы будем представлять ИИ ассистентов в виде графов. Что же это значит и зачем это нужно? Представьте, что перед вами стоит следующая задача: надо сделать игру для гейм джема, то есть за короткий промежуток времени написать компьютерную игру на определённую тематику. Что бы вы сделали? Скорее всего сначала по заданной теме придумали концепцию игры, потом придумали игровые механики, сюжет, локации и так далее.

Но ассистент мог бы выполнять все эти шаги за вас. Его логика была бы следующей: [тема] → [создание концепции] → [разработка механик], [создание сюжета], [разработка локаций]. Как можно заметить, графы позволяют легко описывать поведение ассистента, и это только одно из преимуществ данного подхода.

1. Цель и задачи

Наша цель – это сделать платформу, на которой можно будет создавать, настраивать и использовать ИИ ассистентов

Задачи:

* Изучение теории и существующих решений платформ
* Проектирование собственной платформы
* Разработка платформы

1. Аналитика и проектирование

Аналитика:

Перед началом разработки мы с командой провели обширное исследование. Мы изучили методы работы с GPT, рассмотрели множество готовых агентов и проанализировали существующие платформы. На основе этого анализа мы определили, что необходимо для создания полезных ИИ ассистентов, способных решать сложные задачи, и какие особенности должна иметь наша платформа для упрощения создания таких ассистентов.

Также мы изучили, какие существуют инструменты для создания фронтэнд части и выбрали, подходящие для нас.

Проектирование:

При проектировании мы придерживались следующих аспектов:

* Гибкость (Наша платформа должна быть универсальной, чтобы решать широкий спектр задач, но также иметь возможность адаптации под конкретные задачи для их эффективного выполнения)
* Наглядность (Поведение ассистента должно быть понятным для пользователя)
* Простота (Система не должна быть слишком сложной и запутанной)

1. Структура Графа

Граф состоит из узлов и связей. Узлы представляют функциональные элементы агента, а связи передают информацию между узлами.

Основные типы узлов:

* Model (представляет собой искусственный интеллект, а именно языковую модель gpt)
* Prompt (это запрос от пользователя)
* Chain (это связка между моделью и промптом)
* Visual (это узел нужен для отображения результата генерации. Он может работать с такими стилями как Markdown, Mermaid и KaTex, что позволяет отображать не только текст, но и различные графики, диаграммы, графы, формулы и т.д.)
* Var (то есть переменная. Она может подключаться к промпту, для удобного управления изменяемыми значениями в промпте)
* Web Search (это один из инструментов, которые подключаются к чейну. Представляет собой дополнительную функциональность, которую может использовать gpt при генерации ответа)

Объяснение примера с генерацией пословиц…

1. Пример посложнее

Это пример ассистента – репетитора. Он придумывает задачку по математике, а затем решает и объясняет её. Здесь присутствует 2 узла prompt и 2 узла chain. Первый чейн отвечает за придумывание задачки, а второй – за решение. И благодаря узлам visual можно посмотреть, что получилось на выходе из этих чейнов. Также для удобства здесь есть узел переменной, в которой можно указывать, для какого класса надо придумывать задачку.

1. Объяснение примера
2. Демонстрация интерфейса (видео)
3. Генерация

Немаловажной особенностью нашей платформы является возможность генерации графа. Всё что для этого нужно, это описание ассистента.

Как это работает:

* Изначально стоится описание всей системы.
* Оно вместе с описанием желаемого ассистента отправляется к gpt, который создаёт первую версию графа.
* Этот граф тестируется на валидность.
* Если граф не валиден, к gpt отправляется код ошибки и просьба переделать граф
* Так повторяется пока граф не станет валидным.