

Введение. Виртуальный персонаж с ИИ (искусственным интеллектом) – это компьютерная программа, обладающая способностью взаимодействовать с людьми, имитировать различные аспекты человеческого поведения, такие как речь, эмоции, ответы на вопросы, то есть выступать в качестве базы знаний. Также виртуальные персонажи выполняют роль виртуальных ассистентов, выполняя простые поставленные команды. При этом существует большое количество сервисов, предоставляющих свои решения в сфере ИИ ассистентов или ИИ чат-ботов. Обычно такие решения не позволяют взаимодействовать с другими сервисами, или позволяют, но с ограниченным выбором. В данной работе рассматривается архитектура решения, позволяющего выбирать и настраивать работу со сторонними сервисами при взаимодействии с ИИ персонажами.

Основная часть.

Ключевая идея сервиса — использование классификатора интенгов перед отправкой запросов пользователей к ИИ-персонажу и реагирование на эти интенги. То есть выполнить роль — виртуального ассистента [1 - 2], но особенностью является то, что и интенги и реакцию сервиса на них задает пользователь в виде JSON или YAML файла — плагина. Реакцией является отправка REST API запроса на указанный ресурс и анализ результата.

Данный подход позволит реализовать сервис, который сохранит возможности уже имеющихся сервисов с ИИ-персонажами, а также добавит возможность взаимодействовать с новыми сервисами, или даст возможность унифицировать использование конкретных сервисов при взаимодействии с разными ИИ-персонажами.

Использованию подхода плагинов обусловлено возможностью создания библиотеки плагинов, в которой сообщество сервиса сможет предложить свои плагины другим пользователям или получить уже готовые плагины для взаимодействия с сервисами. Такой подход позволит расширять функционал сервиса независимо от команды разработки сервиса.

Такой сервис будет распространяться как веб-сервис, и основой архитектуры является микросервисная архитектура [3 - 4], что позволит использовать сервис на любом устройстве с выходом в Интернет, а также позволит распространять сервис в виде PWA — приложений, при этом обеспечивая высокую масштабируемость, надежность и доступность сервиса.

Выводы. Архитектура сервиса виртуального персонажа с ИИ с системой плагинов позволит взаимодействовать со сторонними сервисами и также взаимодействовать с разными ИИ, что увеличит функционал ИИ-ассистентов и чат-ботов.

Список использованных источников:

1. YandexGPT в Алисе: как мы создаём языковую модель нового поколения [Электронный ресурс] URL: <https://habr.com/ru/companies/yandex/articles/739626/> (дата обращения 24.12.2024)
2. How Does Amazon Alexa Algorithm Work [Электронный ресурс] – URL: <https://codecondo.com/how-does-amazon-alexa-algorithm-work/> (дата обращения 24.12.2024)
3. Microsoft Application Guide, 2nd Edition [Электронный ресурс] – URL: [https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff650706\(v=pandp.10\)](https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff650706(v=pandp.10)) (дата обращения 24.12.2024)
4. Руководство по архитектуре облачных приложений. Microsoft Press. – 2017. – 329с