

Виртуальный помощник с личностными характеристиками.
Черногор Е. А., Быков А. Е., Захарова Е. А., Куставинова Д. А., Воловикова З. А.
Иркутский государственный университет
Лаборатория робототехники и программирования

В рамках этой работы у нас стояла цель упростить взаимодействия с цифровым устройством за счет человекоподобной модели для разговора. Отличительная особенность нашего подхода - визуализация эмоций, основанная на анализе речи пользователя, имитация взаимодействия с живым человеком за счет биографии виртуального помощника.

Для достижения этой цели выделены следующие задачи: анализ поведения целевой аудитории, подбор характера и внешности ВП, вызывающих положительный отклик у большинства пользователей, разработка поведенческой модели ВП, распознавание речи, ответ на реплику пользователя, визуализация эмоций.

Распознавание речи было реализовано с помощью библиотек SpeechRecognition и PyAudio. Ответ на реплику пользователя происходил в зависимости от типа запроса: предзаданный или нет, для этого находился коэффициент корреляции между запросом пользователя и вопросами из предзаданной базы. Если запрос предзадан возвращается соответствующий ответ из базы, иначе возвращается ответ диалогового модуля. Для разработки диалогового модуля использовалась библиотека DeepPavlov, позволяющая использовать встроенную QuestionAnsweringModel, обученную на датасете SDSJ Task B для ответа на вопрос по заданному контексту. Для взаимодействия с моделью был создан сервер на фреймворке Flask. Чтобы получить ответ сервера отправляется POST запрос, содержащий вопрос. В полученном ответе содержится ответ на заданный вопрос.

Графический интерфейс выполнен с помощью pygame. Интерфейс состоит из окна с gif изображением персонажа, которое меняется в зависимости от тональности ответа на заданный пользователем вопрос, затем воспроизводится ответ бота с помощью библиотеки pyttsx3.

В дальнейшем планируется улучшить работу диалогового модуля для не пред заданных вопросов, перейти с 2D на 3D поведение бота-помощника, построение и переход на более сложную модель поведения бота, перевод приложение в дополненную реальность. Добавление нашего собственного модуля анализа на сантименты, расширение возможностей распознавания, добавив анализ тембра голоса, совместить мобильную часть проекта, написанную в Unity на C# с текущей.

Виртуальный помощник с личностными характеристиками.
Черногор Е. А., Быков А. Е., Захарова Е. А., Куставинова Д. А., Воловикова З. А.
Иркутский государственный университет
Лаборатория робототехники и программирования

Сейчас компьютерная отрасль переживает становление новых типов интерфейсов. Помощники с элементами искусственного интеллекта, уже сегодня способны серьезно упрощать работу с мобильными и стационарными компьютерами, которым для выполнения простых задач достаточно дать короткий запрос, даже не подходя к гаджету; компьютер становится намного проще для общения, если оба интерфейса работают сообща. Тем не менее, если с системами нет контакта на эмоциональном уровне, человек испытывает трудности при взаимодействии с «машиной». ^[1]

Поэтому мы поставили перед собой цель упростить взаимодействия с цифровым устройством за счет человекоподобной модели для разговора. Отличительная особенность нашего подхода - визуализация эмоций, основанная на анализе речи пользователя, имитация взаимодействия с живым человеком за счет биографии виртуального помощника.

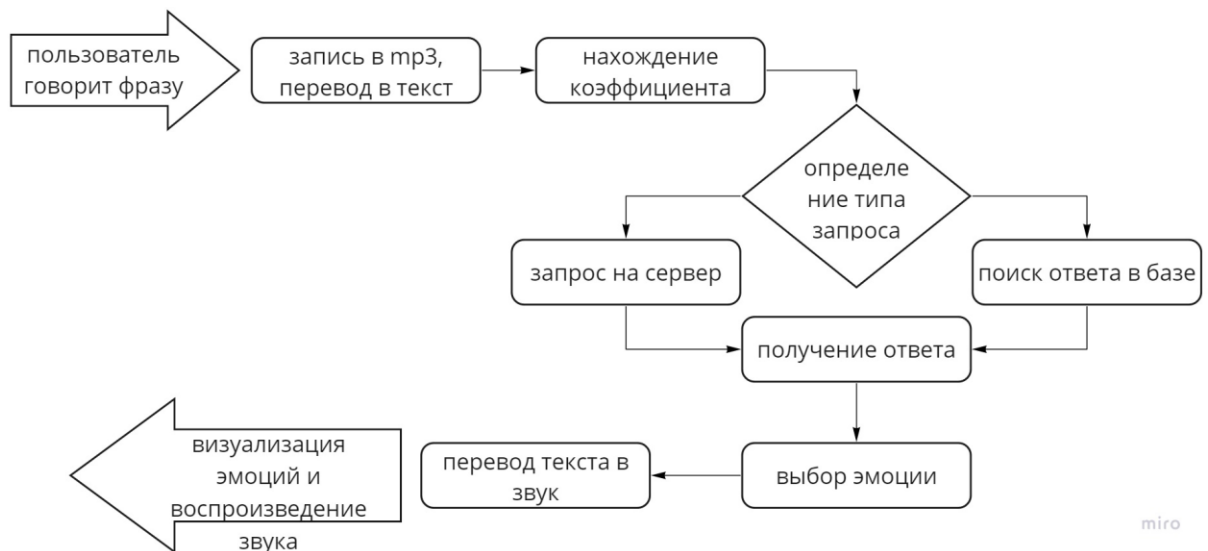
Для достижения этой цели выделены следующие задачи:

- анализ поведения целевой аудитории;
- подбор характера и внешности ВП¹, вызывающих положительный отклик у большинства пользователей;
- разработка поведенческой модели ВП;
- распознавание речи;
- ответ на реплику пользователя;
- визуализация эмоций.

Как правило, бот-помощник решает конкретную задачу, поэтому он выполняет ограниченный набор функций, прописанных в программе. Задача бота - консультировать абитуриентов по вопросам, связанным с поступлением в вуз, следовательно, целевая аудитория - абитуриенты, молодые люди от 14 до 18 лет. Бот должен поддерживать небольшой диалог на отвлеченные тематики (погода, настроение, последние события в мире).

На основании сформулированных задач сначала разработана человекоподобная модель, а потом реализован алгоритм реакции помощника на действия человека. Общая схема работы программы следующая: сначала происходит распознавание речи, далее определение принадлежности вопроса к предзаданной базе и в соответствии с результатом происходит решение какой дать ответ, также в зависимости от этого происходит визуализация эмоции.

¹ Виртуальный помощник - это программное обеспечение, созданное для того чтобы выполнять несложные команды пользователя; с развитием технологий сложность и количество выполняемых задач увеличивалось, таким образом сейчас виртуальный помощник рассматривается как консультант, собеседник для разговора или секретарь.



Для выбора характера и внешности бота производилась оценка путем социологического опроса среди школьников и студентов младших курсов. Проработана схема эмоциональной реакции виртуального помощника, в зависимости от контекста, произнесенного пользователем, и предзаданная база вопросов и ответов. Для создания целостной модели бота-помощника написана его биография, подобраны внешность и стиль, максимально соответствующие персонажу.

Распознавание речи было реализовано с помощью библиотек SpeechRecognition и PyAudio. PyAudio отвечает за доступ и управление микрофоном, а SpeechRecognition непосредственно за распознавание речи. Для распознавания используется класс Recognizer, возвращающий звуковую дорожку, очищенную от посторонних шумов с помощью метода `adjust_for_ambient_noise`. Далее звук переводится в строку с помощью метода `recognize_google`.^[2]

Получив текущую реплику пользователя, определяем тип: предзаданный или нет. Определение происходит с использованием коэффициента корреляции по приведенной ниже формуле:

$$\min \sum \left(\frac{\sum distance(current\ word, word_i)}{length(question_i)} : length(question_i) \right)$$

Где *current word*- это слово из текущего вопроса, *question_i*- это вопрос из базы, *word_i*- слово из вопроса, *distance()* - это расстояние Левенштейна между двумя строками, а *length()* соответственно длина строки.

По этой формуле сначала находим сумму расстояний между каждым словом из текущего вопроса и словами из базы вопросов для *i*-го вопроса и нормируем это поделив на длину *i*-го вопроса, после этого собираем вместе значения слов текущего вопроса по отношению к *i*-му вопросу, и снова это нормируем поделив на длину *i*-го вопроса, а в самом конце просто находим индекс минимального коэффициента, который должен соответствовать заданному вопросу. Таким образом мы избегаем орфографические ошибки и неточности формулировки вопроса. После этого находится расстояние для текущего вопроса и предсказанного формулой вопроса. И если минимальный коэффициент из формулы меньше 3 или расстояние по двум вопросам меньше 0.6 тогда

считается что запрос предзадан а значит возвращается соответствующий ответ иначе возвращается ответ диалогового модуля, который как раз и отвечает за поддержание диалога.

Для разработки диалогового модуля использовалась библиотека DeepPavlov, позволяющая использовать встроенную QuestionAnsweringModel, обученную на датасете SDSJ Task B для ответа на вопрос по заданному контексту. Так как заранее тренированная модель занимает около 5 GB памяти и время ответа модели сильно зависит от мощности процессора, был создан сервер на фреймворке Flask для взаимодействия с ней. Чтобы получить ответ сервера отправляется POST запрос, содержащий вопрос. В полученном ответе содержится ответ на заданный вопрос.

Графический интерфейс выполнен с помощью pygame. Интерфейс состоит из окна с gif изображением персонажа, которое меняется в зависимости от тональности ответа на заданный пользователем вопрос, затем воспроизводится синтезированный ответ бота с помощью класса engine и метода say() входящих в состав библиотеки pyttsx3. [3]

Сейчас интерфейс находится в процессе тестирования, наша команда работает над расширением и улучшением базы эмоций.

Актуальность использования эмоций в связке с искусственным интеллектом доказывает то, что компании Huawei, Magic Leap, Mit Media Lab в данный момент так или иначе реализуют своих помощников или ПО направленное на эмоциональное вовлечение человека.

В дальнейшем планируется улучшить работу диалогового модуля для не пред заданных вопросов, перейти с 2D на 3D поведение бота-помощника, построение и переход на более сложную модель поведения бота, перевод приложение в дополненную реальность. Добавление нашего собственного модуля анализа на сантименты (он уже в разработке), расширение возможностей распознавания, добавив анализ тембра голоса, совместить мобильную часть проекта, написанную в Unity на C# с текущей.

Список используемых источников:

1. Виртуальный помощник или как автоответчики распугивают клиентов URL: <https://www.demyanoff.ru/blog/golosovoe-menju-virtualnyj-pomoshhnik-ili-chelovek>
2. ИИ на Python для распознавания голоса и выполнения команд URL: <https://itproger.com/news/157>
3. Pyttsx text to speech URL: <https://pythonprogramminglanguage.com/text-to-speech/>