



Содержание

Об авторе	12
О рецензенте	13
Предисловие	13
О чем эта книга	14
Что вам потребуется при чтении этой книги	15
Для кого эта книга.....	16
Условные обозначения	16
Загрузка кода примеров	17
Глава 1. Введение.....	18
Диалоговые пользовательские интерфейсы	19
Краткая история чат-ботов.....	20
Последние разработки	21
Архитектура диалогового пользовательского интерфейса.....	23
Классификация	25
Приложения	25
Инструменты разработчика	28
Инструменты прототипирования	28
Каналы	28
Средства разработки чат-ботов	29
Аналитика	29



Распознавание естественного языка	30
Службы каталогов	30
Монетизация.....	31
Выгоды	31
Чат-боты никуда не денутся	32
Приступим!	34
Глава 2. Путеводитель по городу	35
Начало работы	36
Создание вашего первого бота.....	37
Базовые элементы.....	40
Блоки	40
Карты	41
Кнопки	41
Плагины	41
Атрибуты.....	41
Блоки по умолчанию	41
Настройка бота.....	42
Дополнительные карты.....	45
Графика	45
Аудио	46
Видео.....	47
Быстрые ответы	47
Галерея	48
Списки	48
Навигация.....	49
Кнопки	50
Переход к блоку	51
Учет контекста	52
Пользовательский ввод.....	53
Настраиваемые пользовательские атрибуты.....	55
Контекстные переменные по умолчанию	56
Распознавание естественного языка	57
Блок по умолчанию	57

Обработка на стороне сервера	60
Широковещание	66
Шаблоны	67
Аналитика	68
Заключение	68
Ссылки	69
Глава 3. Поговорим о погоде.....	70
Задачи чат-бота.....	71
Конструирование диалога	71
Задачи на стороне сервера.....	72
Получение данных о погоде.....	72
Построение интерфейса на стороне сервера.....	76
Реализация чат-бота.....	81
Создание веб-службы чат-бота.....	88
Публикация в Facebook Messenger	93
Создание страницы Facebook	94
Создание имени пользователя.....	95
Создание приложения Facebook	96
Создание интерфейса для веб-приложения Facebook..	100
Связывание приложения Facebook и приложения интерфейса.....	104
Тестирование чат-бота.....	105
Добавление других тестировщиков.....	106
Публикация для общего доступа	106
Дополнительные форматы сообщений Facebook	108
Типы содержимого	108
Шаблоны.....	109
Заключение	115
Ссылки	115



Глава 4. Создание бота-персонажа	116
Знакомство с Dialogflow	116
Настройка Dialogflow	117
Создание базового агента	118
Развертывание чат-бота	121
Контекст	127
Сущности	131
Развернутые ответы	141
Импорт и экспорт агентов	143
Осуществления	149
Заключение	154
Ссылки	156
Глава 5. Общественный транспорт	157
Обзор Transport API	157
Создание учетной записи разработчика	158
Изучение набора данных	159
Структура диалога	163
Ближайшая станция	163
Следующий поезд	164
Время прибытия	164
Создание простого SMS-бота	165
Начало работы	165
Настройка панели управления	166
Отправка простых сообщений	167
Уведомления о поездах	170
Планирование задач	173
Создание двухстороннего чат-бота	175
Обработка произношения пользователей	181
Заключение	193
Ссылки	193

Глава 6. Поиск ресторана	194
Платформа MS Bot.....	195
Эмулятор канала	195
Создание бота	195
Развертывание бота.....	198
Другие типы сообщений.....	200
Поток диалога	209
Состояние диалога.....	221
Начало работы с Zomato	222
Получение данных из Zomato	223
Служба для поиска ресторанов.....	225
Подключение к Skype.....	233
Заключение	238
Ссылки	238
Глава 7. Новостной бот	239
Начало работы с приложением Twitter.....	240
Создание Twitter-бота	242
Обзор SDK Twitter	246
Обновление статуса	246
Ретвиты сообщений	247
Поиск твитов	247
Обзор новостной службы.....	248
Настройка NewsAPI	248
Создание новостного Twitter-бота.....	252
Персонализация новостного бота	255
Создание базы данных интересов пользователей	259
Доступ к базе интересов пользователей	264
Информирование бота об интересах пользователей....	267
Заключение	272
Ссылки	273



Глава 8. Мой телегид	274
Amazon Alexa	275
Разработка бота цитат	276
Настройка навыков	277
Модель взаимодействия	278
Менеджер диалогов – цитаты	285
Тестирование на Echo	291
Телегид	291
Построение модели взаимодействия	292
Настройка сервера для навыка телегида	296
Тестирование навыка телегида	303
Встроенные цели и слоты	305
Заключение	307
Ссылки	307
Глава 9. Мой верный слуга	308
Служба Todoist	309
Получение ключа	309
Добавление задачи	311
Получение всех задач на день	312
Создание действия	313
Создание агента Dialogflow	313
Создание веб-приложения	318
Тестирование агента на Dialogflow	322
Интеграция в Ассистента Google	325
Проверка на устройстве Google Home	330
Заключение	331
Выводы	332
Ссылки	332

Приложение. Дополнительные ресурсы	333
Статьи	333
Список конференций	333
Список журналов	334
Группы в социальных сетях	334
Предметный указатель	335



Об авторе

Срини Джанарсанам – эксперт в области диалоговых систем, более 15 лет работающий в этой сфере. Он руководил и работал над несколькими проектами, создавая диалоговые системы для различных областей, включая туризм, здравоохранение и образование. Он получил докторскую степень по философии (PhD) в Эдинбургском университете за работу в области искусственного интеллекта и обработки естественного языка.

В настоящее время он является соучредителем и директором Chatomate, британского бренда, специализирующегося на чатах, искусственном интеллекте и решениях для автоматизации бизнеса. Ранее он работал научным сотрудником в университете Хериот-Ватт и университете Эдинбурга. Он опубликовал более 50 статей и тезисов в онлайн-журналах по чат-ботам, других журналах и на международных научных конференциях.

Я бесконечно благодарен всем, кто поддерживал меня в этом невероятном путешествии. Я благодарю мою жену Джайантхи, моего сына Адвайта, моих родителей и всех в семье за их ободрение, поддержку и помощь в этом проекте. Я хочу поблагодарить моих друзей Приянку, Анупа, Тиру, Суреша, Джайю, Нару, Дивакара и многих других за то, что они побудили меня заняться этим проектом и следить за его прогрессом на всем протяжении.

Я хочу поблагодарить рецензентов книги за то, что они протянули мне руку, чтобы сделать книгу такой, какой она есть сейчас. Я также хочу поблагодарить всю команду издательства Packt Publishing, которая внесла огромный вклад, чтобы вы получили эту книгу.

Наконец, я хочу поблагодарить Бога за то, что он дал мне возможность узнать мои сильные стороны, и за его благодать и поддержку при написании этой книги.

О рецензенте

Вамси Венигалла – лидер в технологии с более чем 12-летним опытом работы в строительных организациях, стратегическом планировании, развертывании множества трансформационных платформ/продуктов, ИТ-программ, в управлении проектами, стратегии и трансформации (бизнес, люди, процессы и технологии) и в использовании результатов для различных бизнес-функций в мобильных, финансовых системах, снабжении, HR, розничной торговле и цепочке поставок в областях изготовления и производства продуктов питания.

Вамси закончил магистратуру по информатике в Университете штата Северная Каролина, Северная Каролина.

Предисловие

Создание чат-ботов – это весело. Хотя чат-боты – это лишь еще один вид программного обеспечения, они очень разные с точки зрения ожиданий пользователей. Чат-боты направлены на поддержание диалога. Эта способность обрабатывать язык заставляет проектировать их своего рода человеческую личность и интеллект, независимо от того, хотим мы этого или нет. Разработать программное обеспечение с личностью и интеллектом достаточно сложно и, следовательно, интересно.

Это книга для программистов, которые заинтересованы в изучении мира диалоговых пользовательских интерфейсов. Эта книга организована в виде восьми проектов чат-ботов, которые представляют экосистему инструментов, методов, концепций и даже гаджетов, связанных с диалоговыми интерфейсами. Мы начнем изучать простые кнопочные чат-боты, перейдем к использованию текстовых сообщений и, наконец, голоса. Мы начнем с платформы без кода для создания нашего первого чат-бота и перейдем к изучению библиотек и служб, чтобы самостоятельно программировать модули. Каждая глава представляет собой уникальный проект диалогового интерфейса с источником данных. Вы будете учиться понимать данные, идентифицировать диалоговые задачи, проектировать диалог и, наконец, следовать пошаговым инструкциям для реализации интерфейса.



О чём эта книга

Глава 1 начинается с изучения того, что такое диалоговые пользовательские интерфейсы, их истории и последних разработок. Затем мы рассмотрим их базовую архитектуру, приложения и преимущества, а также некоторые факторы, в силу которых диалоговые интерфейсы так популярны.

В главе 2 вы узнаете, как создавать чат-боты без написания кода. Мы изучим платформу для разработки и хостинга под названием Chatfuel, на которой разработаем путеводитель по вашему городу. Мы разработаем и построим простого чат-бота с кнопками и научимся разворачивать чат-боты в Facebook Messenger, одном из самых популярных каналов для чатов.

Глава 3 рассказывает о построении чат-бота для того же канала, но на этот раз мы создадим его с нуля, используя язык Java и платформу Node.js. Мы создадим чат-бота, который расскажет нам о погоде, используя открытый источник данных под названием OpenWeatherAPI, разместим его в виде облачного приложения и свяжем с Facebook-приложением, которое подключается к странице Facebook. Мы изучим технологии Facebook Pages и SDK Messenger с точки зрения создания и развертывания чатов.

Глава 4 переходит от кнопочных чат-ботов к тем, которые распознают естественный язык. Мы рассмотрим популярный инструмент под названием Dialogflow (ранее известный как API.AI), который можно использовать для создания чатов, распознающих естественный язык. Мы построим бота – персонажа, который эмулирует популярную личность Альберта Эйнштейна. Затем мы рассмотрим, как интегрировать чат-бота в веб-сайт и Facebook Messenger.

В главе 5 переходим к более традиционному каналу связи – «службе коротких сообщений» (SMS). Мы проанализируем, как создать чат-бота для канала SMS. Используя службу данных, называемую Transport API, мы создадим чат-бота, информирующего о расписании поездов посредством SMS. Мы также интегрируем модуль Dialogflow в чат-бот, чтобы тот мог распознать, что говорит пользователь.

В главе 6 вы создадите чат-бота с помощью Microsoft Bot Builder. Это быстро набирающий популярность инструмент для создания ботов, который применяется для проектирования эффективного менеджера диалогов, а не создания его с нуля, как это было ранее. Мы создадим чат-бота для поиска ресторанов, использующего службу данных под названием Zomato. Затем встроим его в канал Skype.

Глава 7 объясняет совершенно другой канал взаимодействия – Twitter. Мы разработаем Twitter-бота, который сканирует твиты пользователей, анализирует их интересы и отправляет им персонализированные новости ежечасно, подключаясь к источнику данных, называемому News API.

В главе 8 мы рассмотрим развитие голосовых ботов, таких как Alexa. Затем создадим бота Alexa, чтобы поговорить о телепрограмме. Мы создадим инструмент, который получает доступ к данным телевизионной программы из службы данных TVMaze и отвечает на запросы пользователей в Amazon Echo.

Глава 9 посвящена продолжению работы над голосовыми ботами и направлена на создание действий для Ассистента Google. Этот инструмент позволит пользователям управлять своим списком дел с помощью устройства Google Home. Мы создадим облачное приложение, в которое будет интегрирована онлайн-служба Todoist для записи новых задач и выполнения текущих.

Приложение перечисляет различные статьи, список журналов, список конференций и т. д., где вы можете получить больше знаний о чат-ботах.

Что вам потребуется при чтении этой книги

Ниже перечислены некоторые инструменты, которые мы широко используем во всей книге. Я рекомендую вам установить их на свой компьютер и ознакомиться с ними, так как это поможет вам при выполнении проектов.

- ❑ **Heroku** – это облачная платформа, которая позволяет размещать веб-приложения в облаке. Чтобы начать работу, зарегистрируйтесь с бесплатной учетной записью на сайте www.heroku.com и установите консольный инструмент Heroku на странице devcenter.heroku.com/articles/heroku-cli. Вы также можете узнать, как развернуть приложения на платформе Heroku. Я рекомендую вам ознакомиться со справочными материалами по адресу devcenter.heroku.com/start.
- ❑ **Git** – это инструмент управления версиями, с которым вы, возможно, знакомы. Мы будем интенсивно использовать Git, поскольку это часть процесса Heroku. Вам нужно установить консольный инструмент Git в вашей системе. Вы найдете его на странице git-scm.com/downloads.



- **Node.js** – это среда выполнения JavaScript, в которой понадобится создавать веб-приложения в большинстве наших проектов. Вы найдете ее дистрибутив на странице nodejs.org/en/download/.
- **Java SDK и Eclipse:** Java 1.8 используется для создания чат-бота в главе 3. Вам нужно установить Java SDK и среду разработки Eclipse, чтобы выполнять инструкции в этой главе. Кроме того, вы можете запрограммировать чат-бота, используя Node.js или другой язык программирования, поддерживающий Heroku. Вы найдете дистрибутив Java 1.8 на сайте www.oracle.com/technetwork/java/javase/downloads/index.html, а Eclipse – на сайте www.eclipse.org/downloads/.

Для кого эта книга

Эта книга предназначена для разработчиков, которые заинтересованы в создании интерактивных пользовательских интерфейсов/чат-ботов. Вы должны уметь работать с платформой Node.js и программировать на языке Java. Опыт использования библиотек Node.js, таких как `request` и `express`, и базовое понимание инструментов типа Git и Heroku CLI также будут полезны.

Условные обозначения

В этой книге вы увидите текст с разным форматированием, которым различаются разные виды информации. Ниже показано несколько примеров этих стилей и дано объяснение их смысла.

Фрагменты кода в тексте, имена таблиц базы данных, имена папок, имена файлов, расширения файлов, имена путей, фиктивные URL-адреса, ввод пользователя и твиттер-дескрипторы показаны следующим образом: «Создайте файл `index.js` и добавьте в него инструкцию `request`».

Блок кода оформляется так:

```
// Комментарий EddieBot
const express = require('express')
const bodyParser = require('body-parser')
const app = express()
app.set('port', (process.env.PORT || 5000))
```

Любой ввод или вывод в командной строке записан следующим образом:

```
npm init
```

Новые термины и важные слова выделены полужирным шрифтом. Слова, которые вы видите на экране, например в меню или диалоговых окнах, отображаются в тексте следующим образом: «Щелкните мышью по ссылке **Get Started** (Начать) в нижней части окна чата».

[зн] Примечания и важные заметки выглядят так.

[зн] Советы и хитрости выглядят так.

Загрузка кода примеров

Вы можете загрузить файлы с примерами кода для этой книги по адресу github.com/PacktPublishing/Hands-On-Chatbots-and-Conversational-UI-Development.



Глава 1. Введение

Мы вошли в эру интеллектуальных машин, в которой лидируют диалоговые интерфейсы. За последние пару лет мы были охвачены множеством новых видов машин и программного обеспечения, общеизвестных как боты. Боты – это автоматизированные аппаратные или программные средства, которые основаны на технологиях искусственного интеллекта (ИИ). Недавние разработки в алгоритмах машинного обучения, такие как глубокое обучение и углубленное закрепление, улучшили производительность задач ИИ, таких как автоматическое распознавание речи (ASR), распознавание естественного языка (NLU), синтез текста в речь (TTS) и распознавание изображений. Это сократило время пути человечества к технологической сингулярности, момент, когда искусственный интеллект пре- восходит естественный человеческий разум не по дням, а по часам.

Одной из долгосрочных целей в области ИИ является создание компьютерных систем, которые смогут проводить человекоподобные беседы с пользователями. Благодаря недавним достижениям в области технологий ИИ мы сейчас на один шаг ближе к достижению этой цели. Теперь уже не за горами, что мы можем взаимодействовать с устройствами и гаджетами в наших домах и офисах, используя только голос. Нам еще предстоит пройти долгий путь по созданию стандартов и цифровых сущностей, способных беспрепятственно разговаривать на естественном языке. Однако недавний всплеск интересов и огромные инвестиции в продвижение этих идей свидетельствуют о том, что мы находимся на пути к разработке такого глобального стандарта. Если вы в восторге от последних разработок в области ИИ и технологий автоматизации, эта книга для вас. Мы отправимся в путь к будущему, что гуру-дизайнер Марк Кертис называет разговорной сингулярностью, когда диалоговые устройства исчезают, а разговор между человеком и машиной беспрепятствен и естествен.

Это книга для программистов, которые начинают создавать диалоговые интерфейсы. Сегодня простые кнопочные чат-боты могут

быть созданы без необходимости написать хоть одну строку кода. С них мы и начнем. Мы постепенно перейдем к более сложным и гибким архитектурам и изучим каналы для использования, такие как Facebook Messenger, SMS и Twitter. Мы также рассмотрим инструменты для распознавания естественного языка и управления разговорами по мере продолжения. Наконец, мы закончим наш эксперимент, создав ботов с голосовой связью на таких платформах, как Amazon Alexa и Google Assistant.

Диалоговые пользовательские интерфейсы

Диалоговые пользовательские интерфейсы так же стены, как и современные компьютеры. ENIAC, первый программируемый универсальный компьютер, был построен в 1946 году. В 1950 году Аллан Тьюринг, британский компьютерный ученый, предложил измерить уровень интеллекта в машинах с использованием диалогового теста, называемого тестом Тьюринга. Тест состоял в том, что машина конкурировала с человеком как партнером по диалогу и судей-людей (еще одного человека). Судьи должны были взаимодействовать с каждым из двух участников (человеком и машиной), используя интерфейс текстового типа, который не похож на большинство современных приложений чата обмена сообщениями. В ходе беседы судьи должны были определить, кто из двух участников был машиной. Если, по меньшей мере, 30% судей не смогли провести различие между двумя участниками, считалось, что машина прошла тест. Это была одна из самых ранних концепций по диалоговым интерфейсам и их отношению к уровню интеллекта машин, которые обладают такими возможностями. Однако попытки создания подобных интерфейсов не были успешными в течение нескольких последующих десятилетий.

В течение примерно 35 лет, с 1980-х годов, графические пользовательские интерфейсы (GUI) были доминирующим способом взаимодействия с машинами. С недавним развитием в области ИИ и растущими ограничениями, такими как сокращение размеров гаджетов (от ноутбуков до мобильных телефонов), сокращение экранных объектов (смарт-часы), и необходимостью того, чтобы интерфейсы стали невидимыми (умный дом и роботы), диалоговые пользовательские интерфейсы снова становятся реальностью.



Например, лучшим способом взаимодействия с мобильными роботами, используемыми в умных домах, будет голос. Таким образом, система должна распознавать запросы пользователей и отвечать на естественном человеческом языке. Такие возможности систем помогут уменьшить наши усилия в изучении и понимании современных сложных интерфейсов.

Диалоговые пользовательские интерфейсы известны под несколькими названиями: интерфейсы естественного языка, разговорные диалоговые системы, разговорные чат-боты, интеллектуальные виртуальные агенты, виртуальные помощники и т. д. Фактическая разница между этими системами связана с интеграцией в них серверных элементов (например, баз данных или модулей задач/управления), модальностями (например, текстовыми, голосовыми и визуальными аватарами) и каналами, на которых они развернуты. Однако одним из общих моментов для этих систем является их способность взаимодействовать с пользователями в разговорной манере с использованием естественного языка.

Краткая история чат-ботов

Истоки современных чат-ботов можно проследить до 1964 года, когда Джозеф Вайзенбаум в Массачусетском технологическом институте (МТИ) разработал чат-бота под названием Eliza. Он использовал простые правила разговора и перефразировал большую часть того, что пользователи сказали, чтобы имитировать роджерянского терапевта. Хотя это показало, что наивных пользователей можно обмануть, если они думают, что разговаривают с фактическим терапевтом, сама система не понимала проблемы пользователя. После этого в 1991 году была присуждена премия Лобнера, которая призвала исследователей ИИ создавать чат-ботов, которые смогут пройти испытание Тьюринга и помочь в продвижении ИИ. Несмотря на то что они не прошли тест вплоть до 2014 года, многие известные чат-боты выиграли призы за победу в других ограниченных задачах. К ним относятся ALICE, JabberWacky, Rose и Mitsuku. Однако в 2014 году на конкурсе Тьюринга, посвященном 60-летию смерти Алана Тьюринга, чат-бот под названием Eugene Goostman, имитирующий 13-летнего ребенка, сумел обмануть 33% судей, тем самым выполнив тест. Язык разметки искусственного интеллекта (Artificial Intelligence Markup Language – AIML) и ChatScript были разработаны как способ для обозначения знаний и диалогового

контента для большинства этих чат-ботов. Скрипты, разработанные с использованием этих сценарных языков, могут затем передаваться в интерпретаторы для создания диалогового поведения. Чат-боты, разработанные для прохождения теста Тьюринга, были в основном болтливыми с одной целью – победить тест Тьюринга. В большинстве они не рассматривались как какое-либо продвижение в ИИ или шаг к созданию полезных диалоговых помощников.

С другой стороны, исследования в области искусственного интеллекта, в частности в области машинного обучения и обработки естественного языка, привели к появлению различных диалоговых интерфейсов, таких как системы ответов на вопросы, интерфейсы естественного языка для баз данных и разговорные диалоговые системы. В отличие от чат-ботов, построенных для прохождения теста Тьюринга, эти системы имели очень четкие цели. Системы ответа на вопросы обрабатывали естественные языковые вопросы и находили ответы в наборах неструктурированных текстовых данных. Интерфейсы естественного языка для баз данных (Natural Language Interfaces to Database Systems – NLIDBS) были интерфейсами для больших баз данных SQL, которые интерпретировали запросы к базе данных, создаваемые на естественном языке, таком как английский, преобразовывали их в SQL и возвращали ответ. Разговорные системы (Spoken Dialogue Systems – SDS) – это системы, которые могли бы поддерживать контекстные разговоры с пользователями для обработки диалоговых задач, таких как бронирование билетов, контроль других систем и обучение учащихся. Это были предшественники современных чат-ботов и диалоговых интерфейсов.

Последние разработки

В 2011 году компания Apple выпустила интеллектуального помощника под названием Siri в составе своих устройств iPhone/iPad. Siri была смоделирована как персональный помощник пользователя, выполняя такие задачи, как вызов, чтение сообщений и настройка аварийных сигналов и напоминаний. Это одно из самых значительных событий в недавнем прошлом, которые перезагрузили историю диалоговых интерфейсов. В первые дни работы пользователи Siri использовали ее всего несколько раз в месяц для выполнения таких задач, как поиск в Интернете, отправка SMS и совершение телефонных звонков. Несмотря на новизну, Siri предназначалась для работы с гораздо большим количеством функций, которые были добавлены

в последующие годы. В первые дни у Siri было много клонов и последователей на Android и других платформах для смартфонов. Большинство из них было смоделировано в качестве помощников и доступно в виде мобильных приложений.

В том же году (2011 г.) IBM представила Watson, систему ответов на вопрос, которая участвовала в игровом шоу под названием Jeopardy и выиграла его у предыдущих победителей, Брэда Раттера и Кена Дженнингса. Это ознаменовало собой веху в истории ИИ, поскольку Watson смог обрабатывать вопросы естественного языка и отвечать на них в режиме реального времени. С тех пор Watson был переработан в набор инструментальных средств с набором инструментов когнитивного обслуживания для распознавания естественного языка, анализа настроений, управления диалогами и т. д.

Следуя Siri и Watson, следующее крупное событие появилось в Microsoft в 2013 году, когда они представили Cortana как стандартную функцию на смартфонах под управлением операционной системы Windows, а затем в 2015 году и на платформе Windows 10. Как и Siri, Cortana была персональным помощником, который управлял такими задачами, как настройка напоминаний, ответы на вопросы и т. д.

В ноябре 2014 года Amazon пригласил своих премьер-членов опробовать своего персонального помощника по имени Alexa. Alexa была доступна на собственном продукте Amazon под названием Echo. Echo был первым в своем роде умным динамиком, который размещал в нем помощника, как «призрак» в машине. Хотя он назывался динамиком, на самом деле это был крошечный компьютер с голосом в качестве единственного интерфейса, в отличие от смартфонов, планшетов и персональных компьютеров. Пользователи могли говорить с Alexa, используя голос, просить ее выполнять такие задачи, как установка напоминаний, воспроизведение музыки и т. д.

В апреле 2016 года социальная сеть Facebook объявила, что открывает свою популярную платформу Messenger для чат-ботов. Это был радикально иной подход к диалоговым интерфейсам, по сравнению с Siri, Alexa и Cortana. В отличие от этих персональных помощников, объявление Facebook привело к созданию пользовательских и фирменных чат-ботов. Эти боты очень похожи на Siri, Cortana и Alexa, но могут быть настроены на потребности их создающего бизнеса. Теперь чат-боты могли нарушить работу нескольких рынков, включая обслуживание клиентов, продажи, маркетинг, техническую поддержку и т. д. Многие платформы обмена сообщениями, такие

как Skype, Telegram и другие, также стали доступны для интеграции в чат-боты в то же время.

В мае 2016 года Google анонсировала Assistant, свою версию персонального чат-бота, доступную на нескольких платформах, таких как приложение Allo и Google Home (умный динамик, такой как Echo). Все помощники, такие как Siri, Cortana, Alexa и Google Assistant, также открылись в качестве каналов для диалоговых возможностей сторонних разработчиков. Таким образом, теперь можно персонализировать программное обеспечение Alexa и Google Assistant, добавив диалоговые возможности (называемые навыками или действиями) из библиотеки решений сторонних разработчиков. Так же, как бренды могут создавать своих собственных чат-ботов для различных служб обмена сообщениями (например, Skype и Facebook Messenger), они могут развивать навыки для Alexa или действия для Ассистента Google. В 2018 году был выпущен смарт-динамик компании Apple, Homepod, работающий под управлением голосовой платформы Siri.

Параллельно с этими событиями также наблюдается значительный рост с точки зрения инструментов, доступных для создания и размещения чат-ботов. В последние два года отмечен экспоненциальный рост инструментов для разработки, моделирования, сборки, развертывания, управления и монетизации чатов. Это привело к созданию экосистемы, которая разрабатывает и создает пользовательские диалоговые интерфейсы для предприятий, благотворительных организаций, правительственные и других организаций по всему миру.

Архитектура диалогового пользовательского интерфейса

В этом разделе мы рассмотрим базовую архитектуру диалогового интерфейса (рис. 1.1.).

Основным модулем диалогового интерфейса является менеджер диалогов. Этот модуль контролирует поток разговора. Он берет семантическое представление того, что пользователь говорит, как ввод и решает, каков должен быть ответ системы. Он поддерживает представление диалогового контекста в той или иной форме, например набор пар значений ключа для выполнения содержательного разговора в течение нескольких обменов между пользователем и системой.

Семантическое представление пользовательского ввода возможно напрямую по нажатии кнопки. В системах, которые распознают язык,

**Рис. 1.1**

пользовательские высказывания переводятся в семантическое представление, состоящее из пользовательских целей и параметров (слов и сущностей), с помощью модуля распознавания естественного языка. Возможно, этот модуль необходимо предварительно обучить, чтобы распознать набор пользовательских целей, определенных разработчиком, относящихся к доступным диалоговым задачам.

Интерфейсы с поддержкой голоса, которые принимают голосовые инструкции пользователя, также нуждаются в модуле распознавания речи, который может транслировать речь в текст, прежде чем передавать его в модуль распознавания естественного языка. Симметрично, с другой стороны, необходим модуль синтеза речи (или модуль преобразования текста в речь), который преобразует текстовый ответ системы в речь.

Менеджер диалогов должен взаимодействовать с серверными модулями. Это может быть база данных или онлайновый источник данных, который запрашивается для ответа на вопрос пользователя (например, телевизионное расписание), или онлайн-служба для выполнения инструкций пользователя (например, бронирование билета).

Канал – это место, где чат-бот ожидает пользователя. В зависимости от канала может быть один или несколько модулей, из которых состоит этот слой. Например, если чат-бот находится в Facebook Messenger, этот слой состоит из Facebook-страницы и Facebook-приложения, которое соединяется с остальными модулями чат-бота, оформленными в виде веб-приложения.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине «Электронный универс»
[\(e-Univers.ru\)](http://e-Univers.ru)