

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАЛТИЙСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. И. КАНТА»
ИНСТИТУТ ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Рекомендована к защите:
методический руководитель
направления подготовки
к.т.н., доцент ИФМНиИТ

_____ С.Н. Ткаченко

" ____ " _____ 2020 г.

Допущена к защите:
первый заместитель директора
ИФМНиИТ
к. ф.-м. н., доцент

_____ А.А. Шпилевой

" ____ " _____ 2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема: «РАЗРАБОТКА ЧАТ-БОТА ДЛЯ БАНКОВСКИХ СИСТЕМ»

Направление подготовки: 02.04.03 «Математическое обеспечение и администрирование информационных систем»

Магистерская программа: «Банковские информационные технологии»

Квалификация (степень): **магистр**

ВКР защищена на оценку:

Выполнил: студент 2 курса

Белова Е.Е. Белова

Руководитель: к.т.н., доцент ИФМНиИТ

Толстель О.В. Толстель

Рецензент: главный инженер отдела информатизации Отделения по Калининградской области Северо-Западного Главного управления ЦБ РФ

Ложкин П.В. Ложкин

Калининград, 2020

Реферат

Сведения об объеме работы: работа содержит 167 страниц, 29 рисунков, 1 таблицу и 5 приложений.

Ключевые слова: чат-бот, виртуальный помощник, оптимизация бизнес-процессов, сценарии диалогов, банковская сфера, клиент банка, анализ данных, нейронное обучение, искусственный интеллект, язык Python, библиотека deep-ravlov, социальная сеть «ВКонтакте».

Результатом работы является разработанный чат-бот, который может использоваться в банковской сфере для упрощения взаимодействия клиента с банком.

Созданный чат-бот может отследить баланс клиента, указать курсы валют, предоставить информацию о ближайших отделениях банка и часах их работы.

Разработанный чат-бот относится к типу «Разговорный на основе языковой модели и правил». Созданный чат-бот будет полезен клиентам банка, которые зарегистрированы в социальной сети «ВКонтакте».

Бот может вывести данные по конкретным валютам, которые чаще всего используют жители Калининградской области.

В ходе работы над магистерской диссертацией автором были опубликованы две статьи.

Оглавление

Введение.....	4
Глава 1. Анализ возможностей применения чат-ботов в банковской сфере	8
1.1. История развития чат-ботов.....	8
1.2. Преимущества чат-ботов	10
1.3. Использование чат-ботов	12
1.4. Чат-боты и банки.....	15
1.4.1. Категории обращения клиентов	19
1.4.2. Типы чат-ботов	19
1.4.3. Трудности.....	20
1.5. Использование чат-ботов в зарубежных банках.....	21
1.6. Использование чат-ботов российскими банками	23
Глава 2. Прототип чат-бота для банка	28
2.1. Функционирование прототипа чат-бота	28
2.2. Прикладная функциональность чат-бота.....	30
2.3. Пример работы чат-бота в «ВКонтакте»	31
Глава 3. Чат-бот для банка	33
3.1. Создание сообщества.....	33
3.2. Разработка чат-бота.....	38
3.3. Размещение бота на хосте	43
3.4. Функционирование чат-бота.....	45
Заключение	47
Библиографический список.....	50
Приложение 1. Сценарии диалогов для прототипа чат-бота.....	53
Приложение 2. Расширенная база сценариев диалогов	61
Приложение 3. Обучение прототипа чат-бота	102
Приложение 4. Сценарии диалогов для чат-бота	104
Приложение 5. Код чат-бота	124

Введение

Актуальность данной работы обусловлена современным всплеском интереса к чат-ботам, что объясняется двумя ключевыми факторами: развитием мессенджеров и успехами прикладных разработок в области искусственного интеллекта [1].

Чат-боты являются виртуальными собеседниками, они используются в диалоговых системах при обслуживании клиентов банка. Чат-боты нужны для того, чтобы автоматизировать рутинные действия сотрудников банка, оптимизировать внутренние и внешние процессы. Использование чат-ботов позволяет снизить издержки и удовлетворить потребности прогрессивных клиентов.

Использование чат-ботов в банковской сфере становится еще более востребовано в условиях карантина, объявленного из-за существующей пандемии COVID-19. Чтобы лишний раз не контактировать с возможными носителями инфекции, можно совершались банковские операции, используя чат-бот.

Чат-боты являются эффективным инструментом построения коммуникаций с клиентами, являясь новым каналом таких коммуникаций [24]. По прогнозам в 2022 году за счет использования чат-ботов взаимодействие с клиентами банков будет на 90% автоматизировано [11].

Согласно статистике, 2 млрд. человек используют мессенджеры, а за период 2016–2021 гг. число пользователей мессенджеров вырастет почти на миллиард человек. Кроме того, активность пользователей очень высока: средний пользователь WhatsApp проводит в мессенджере 195 минут [17]. «ВКонтакте» является одной из самых популярных социальных сетей в России. 38,2 млн пользователей из РФ заходят в данную сеть ежемесячно, а 23 млн — ежедневно [12].

Аудитория мессенджеров в России составляет более 50 млн человек. Более половины из них являются пользователями WhatsApp и Viber. Далее идет Skype и приложения социальных сетей «ВКонтакте», Facebook Messenger и Одноклассники. Аудитория Telegram составляет всего 1–1,5 млн пользователей.

При этом именно Telegram обладает наибольшим потенциалом в сфере чат-бот разработок [24]. Но, так как в настоящее время использование Telegram нежелательно из-за претензий Роскомнадзора [7], то для своего чат-бота я буду использовать приложение социальной сети «ВКонтакте».

Важно, чтобы чат-боты были не только информационными, а чтобы с их помощью совершались банковские операции, которые описывались бы текстом или с помощью голоса [24]. Этот новый канал коммуникации банка с клиентом должен стать очень персонифицированным, то есть помнить всю историю и предлагать только те продукты и услуги, которые нужны клиенту [24], не навязывая дополнительные ненужные услуги.

Технология чат-ботов рассматривается многими банками как очень перспективная.

Чат-боты нужны для того, чтобы автоматизировать рутинные действия, вследствие чего они будут полезны в любых отраслях бизнеса, где хотелось бы оптимизировать внутренние и внешние процессы. Освобождая сотрудников от выполнения простых скриптов, предприятие может направить ресурсы людей на выполнение более сложных функций, в которых требуется участие человека.

Для того, чтобы ускорить и облегчить связь между клиентом и сотрудником банка, а также минимизировать использование телефона или электронной почты, как раз и используются чат-боты [26].

По функциональным обязанностям (см. [6]) можно различать чат-боты, которые:

- 1) автоматизируют;
- 2) помогают;
- 3) быстро ищут;
- 4) оформляют заказы;
- 5) собирают лиды.

СМС-сообщения также являются важнейшим способом взаимодействия с большим количеством людей, безопасной верификации пользователей и прове-

дения мгновенных транзакций. Аналогично продавцам, научившимся реагировать на предпочтения клиентов, чат-боты предоставят клиенту возможность самому выбирать удобную для него платформу, обеспечивая таким образом гибкость, сравнимую с сервисами для личных контактов [8].

К 2020 году 80% компаний планируют использовать чат-боты. Однако типы систем, с которыми сотрудники будут взаимодействовать, будут очень сильно варьироваться. Некоторые чат-боты будут отвечать на популярные вопросы шаблонными ответами, другие смогут переключаться между мессенджерами, автоматизировать сервисы и даже обеспечивать выполнение сложных транзакций [8].

Некоторые из этих чат-ботов будут подключены к внешним системам, которые позволят им либо получать доступ к большему количеству информации (через клиентские базы данных), либо совершать больше действий (через интеграцию с другими системами).

Цель ВКР: создать чат-бот для упрощения взаимодействия клиента с банком.

Задачи:

- 1) проанализировать возможности применения чат-ботов в банковской сфере и подобрать технологии реализации;
- 2) подготовить данные и разработать сценарии диалогов для чат-бота;
- 3) настроить и подключить чат-бот к социальной сети «ВКонтакте».

Термины, встречающиеся в работе

Лид-менеджмент (lead management) — это термин, который используется в теории бизнеса для описания методов, систем и практик, разрабатываемых для привлечения новых потенциальных клиентов, как правило с помощью использования различных маркетинговых технологий.

Генерация лидов — маркетинговый термин, который обозначает создание или генерацию интереса/запроса со стороны потенциального потребителя по отношению к определенным продуктам или услугам.

Фишинг — вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей — логинам и паролям.

Чат-бот — это автоматизированный и персонализированный чат между компьютерной системой и пользователем, т. е. программа, разрабатываемая на основе технологий машинного обучения и нейросетей под определенный набор целей.

Эквайринг — технология, позволяющая принимать к оплате банковские карты через Интернет.

Глава 1. Анализ возможностей применения чат-ботов в банковской сфере

1.1. История развития чат-ботов

Определение. *Чат-бот* (от англ. — *chatterbot*) — это компьютерная программа, которая ведет разговор с помощью слуховых или текстовых методов.

Термин «чат-бот» придумал Майкл Молдинг в 1994 г. для описания разговорных программ [15].

Само понятие «чат-бот» вошло в массовый обиход в начале 1990-х., однако концепция и технологии для его реализации восходят к 1950-м, когда о подобного рода решениях писали Алан Тьюринг и Джозеф Визенбаум [12]. История развития принципа коммуникаций от первого чат-бота до новых форм представлена в таблице 1.1.

Таблица 1.1. История развития чат-ботов.

Год	Событие
1950	Тест Тьюринга, определяющий способность машины мыслить.
1961	Shoebox от IBM — первый виртуальный голосовой ассистент.
1966	ЭЛИЗА (ELIZA) — первый виртуальный собеседник.
1972	ПАРРИ (PARRY) — чат-бот, имитирующий собеседника с психическим расстройством.
1984	Бот «Марк В. Шэйни» (Mark V. Shaney) по технологии «цепь Маркова».
1988	Чат-бот Джаббервоки (Jabberwacky) по кличке «бот-болтун».
1990	Премия Лёбнера (Loebner prize) для чат-ботов.
1995	A.L.I.C.E. — первый чат-бот на основе обработки естественного языка.

1996	КЛИППИ (Clippy, Clippit) — виртуальный ассистент-помощник от Microsoft.
2010	СИРИ (SIRI) для системы IOS от компании Apple.
2014	АЛЕКСА (ALEXA) и КОРТАНА (CORTANA) — виртуальные помощники от Amazon и Microsoft.
2015	API для сторонних разработчиков ботов в Telegram.
2016	API для разработчиков ботов в Facebook Messenger.
2017	БИКСБИ (Bixby) от Samsung и АЛИСА от Яндекса.
май 2018	Технология GoogleDuplex (добавление человеческого голоса виртуальному ассистенту Google Assistant).

Из данной таблицы мы видим, как совершенствовались боты в своем развитии [27].

Чат-боты работают в ставшей уже привычной для многих людей среде виртуального общения — это мессенджеры и чаты социальных сетей [19] (см. рис. 1.1). Чат-бот-платформы представлены на рис. 1.2.



Рис. 1.1. Мессенджеры и чаты социальных сетей.

Пользователям удобнее общаться в мессенджерах, которые они открывают десятки раз в день, поэтому многие компании запускают свои чат-боты именно

в мессенджерах [9]. Можно сказать, что чат-бот является программой, которая имитирует диалог в мессенджере [16].

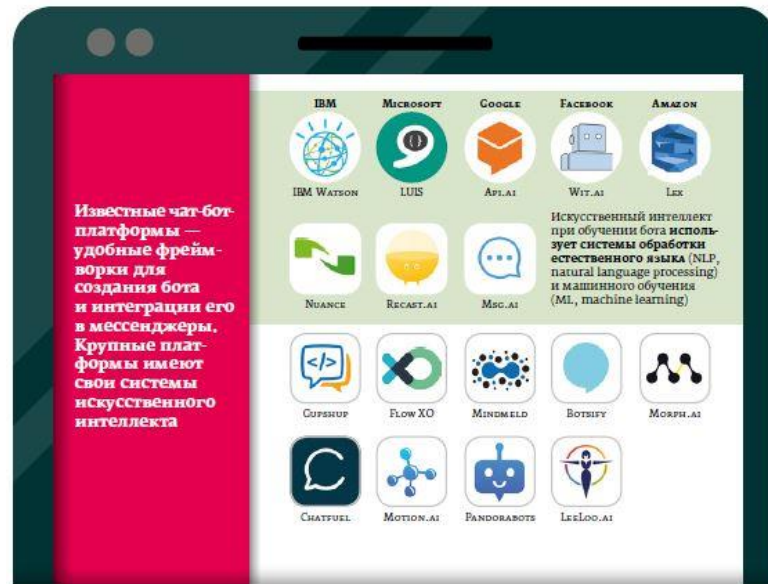


Рис. 1.2. Известные чат-бот платформы.

1.2. Преимущества чат-ботов

Чат-боты — это не столько модный тренд, сколько ресурсосберегающая технология, позволяющая экономить бюджет компаний и драгоценное время сотрудников [28].

В чем же состоят плюсы чат-ботов? Можно перечислить следующие основные моменты [23]:

1. Чат-боты предоставляют нужную информацию там, где пользователь общается с друзьями, при этом ему не нужно переключаться между разными приложениями и ждать, когда появится возможность сесть за компьютер или ноутбук.

2. В отличие от сайтов, которые больше ориентированы для общения путем нажатия кнопок на клавиатуре, чат-боты ориентированы для набора текста пальцами на экранах смартфонов.

3. Сообщения от чат-бота легко прочитать на экране смартфона, в отличие от информации на сайтах, которые часто просто не оптимизированы для мобильных устройств.

4. Если пользователь хотя бы раз запустил чат-бота, то чат-бот сохраняется в контактах мессенджера, и можно к нему вернуться в любое время.

5. Чат-бот позволяет отказаться от расходов на содержание сервисов рассылки писем, смс-сообщений, ведь функция рассылки уже встроена в него.

6. Открываемость рассылок в ботах составляет 80–95%, при этом сервисы e-mail рассылок, как правило, могут «похвастаться» не более чем 20% открываемости писем.

7. Чат-бот создать быстрее и дешевле, чем мобильное приложение. Чат-бот универсален для IOS и Android.

8. Чат-бот не занимает виртуальной памяти смартфона (в отличие от мобильного приложения, его не нужно скачивать, обновлять и устанавливать на телефон).

9. Чат-бот в отличие от сайтов и мобильных приложений может распределять своих пользователей на группы, в зависимости от их действий в боте и делать адресную рассылку для каждой группы индивидуально.

10. Чат-боты позволяют сократить издержки на содержание он-лайн консультантов, или ускорить обслуживание клиентов, что повышает их лояльность.

11. Чат-боты проводят опросы для выяснения потребностей клиента без интеграции с дополнительными сервисами. Владелец чат-бота может делать адресные предложения каждому клиенту индивидуально с учетом его предпочтений, или сформировать новое предложение, интересное большинству пользователей, или улучшить работу, получив обратную связь.

12. Чат-бот легко распространяется по «сарафанному» радио, ведь пользователь чат-бота может одним движением пальца поделиться им с любым человеком из своей записной книжки.

13. Чат-бот сейчас вызывает эффект «WOW!». Даже если пользователь бота сейчас не нуждается в предлагаемом товаре или услуге, то, когда придет время, он вспомнит не сайт, а удививший его бот.

1.3. Использование чат-ботов

Основные направления, которые можно выделить в использовании чат-ботов:

- 1) информационное;
- 2) коммуникационное;
- 3) коммерческое.

Возможности использования чат-ботов безграничны. Среди традиционных направлений бизнеса, в которых активно используют чат-ботов, можно выделить банки, страховые компании и службы поддержки. Одним из первых такой сервис запустил банк «Точка». Сервис стал помощником как для клиентов (через мессенджер Facebook можно совершить платеж и проверить предыдущие операции), так и для самих сотрудников. Данный чат-бот введен для экономии времени работника на поиск информации. Он анализирует текст обращения клиента и предлагает сотруднику подсказки. Отталкиваясь от шаблона, который ему предлагает помощник, сотрудник готовит ответ клиенту [21].

За последний год значительно увеличилось количество обращений от сферы торговли: она активно интересуется переводом программ лояльности в мессенджеры и рекламными рассылками. Их стоимость намного ниже, чем отправка смс-сообщений, а открываемость выше, чем через электронную почту [21].

Общение с помощью электронной почты становится не таким популярным, так как пользователь становится все более мобильным [27]. Основное же преимущество чат-ботов — в скорости доступа к нужной информации. Чат-боты

встраиваются в социальные сети и мессенджеры, являющиеся наиболее популярными средствами коммуникации, особенно у молодежи. Данный канал устанавливает новые возможности общения и круглосуточный доступ к сервису из любого места, где есть выход в интернет.

Благодаря использованию чат-ботов участие в процессе сотрудников организаций сводится к минимуму, за счет чего вероятность совершения ошибок и влияние человеческого фактора снижаются. При этом лояльность клиентов и их привязанность к организации повышаются. Также чат-боты позволяют сократить количество посещений людьми офисов, магазинов и филиалов организаций, значительно уменьшить взаимодействие физических лиц с call-центрами поставщиков услуг [27].

Аналитики подчеркивают, что в последнее время набирают популярность корпоративные боты, рассчитанные исключительно на сотрудников компании. Примером такого бота является помощник IT-компании «Крок», с помощью которого можно заказать пропуск, оформить командировку или забронировать переговорную комнату. Не менее перспективен для ботов отельный и туристический бизнес — сервис HotelChat выполняет роль консьержа, с которым можно пообщаться, не выходя из номера [21].

Некоторые эксперты полагают, что вскоре чат-ботов вытеснят с рынка голосовые помощники (вроде Алисы от Яндекса).

По данным агентства Juniper Research, к 2023 году число голосовых цифровых помощников во всем мире возрастет до 8 млрд (к концу 2018 года их было около 2,5 млрд).

На текущий момент разработчики решают проблемы с возможностью перебивать бота, с определением настроения собеседника, чтобы боты стали еще ближе к живому человеку. На смену узкоспециализированным чат-ботам придут некие универсальные конструкторы.

Технологически индустрия чат-ботов и голосовых ассистентов в России ничем не уступает зарубежным рынкам — мы просто находимся на более раннем этапе распространения технологии. По нашим оценкам, горизонт массового

внедрения виртуальных ассистентов — это еще год для США, 2 года для России и 2–4 года для Европы и Китая. В этом направлении уже очень развит рынок Китая. Технологические гиганты Alibaba, Tencent и Baidu создали свою экосистему виртуальных ассистентов и устройств. Китайский бизнес активно встраивается в эту экосистему: помимо месячной аудитории в 1 млрд пользователей в мессенджере WeChat зарегистрировано более 14 млн бизнес-аккаунтов. Чат-боты в мессенджере позволяют учитывать в интерактивных коммуникациях контекст разговора и доводить клиента до стадии покупки товара или услуги. Самые востребованные сферы применения чат-ботов — клиентский сервис, автоматизация службы поддержки и определенные функции кадровых компаний [21].

Повышенный интерес к чат-ботам связан с развитием RPA-решений (роботизации операционных и бизнес-процессов), которые расширяют возможности цифровых помощников. Обычно чат-боты воспринимаются как инструмент общения с покупателями, но крупные компании давно применяют их для связи пользователей с технической поддержкой [21].

Стоимость разработки или готовых решений варьирует от вида задачи, которую будет решать чат-бот, и может составлять от 0 руб. при использовании бесплатных онлайн-сервисов, которые предлагают чат-ботов со стандартным набором функций, до 500 тыс. руб. за разработку чат-бота со сложным функционалом. Разработчики уверены: эти средства могут окупиться, если грамотно продумать, какие бизнес-процессы будут автоматизированы и закрыты [19].

Благодаря интегрированию собственных приложений и чат-ботов, формируется система сервисов для эффективного решения ежедневных рабочих задач. Смартфон уже может понимать голосовые команды, системы справок и поиска распознают речь и дают разумные ответы. В данное время чаще встречаются боты, которые построены на наборе заранее написанных сценариев. Но по мере развития технологий машинного обучения боты выйдут за пределы стандартных сценариев [27].

Благодаря ботам отпадает необходимость установки на смартфон приложений, занимающие большие объемы памяти, которые часто не оптимизированы под мобильные устройства и часто дублируют сайты компаний [27].

Надо отметить, что самым трудоемким в чат-ботах является их интеллектуальность. Кроме того, в идеальном случае, чат-бот должен владеть определенным «сленгом», который характеризует компанию [27].

1.4. Чат-боты и банки

Задача любых банковских чат-ботов — полностью заменить приложения банков, создав новый подход взаимодействия клиентов с банками.

В банковском бизнесе чат-боты работают следующим образом: с разрешения клиента и с соблюдением достаточных мер безопасности они подключаются к базе финансовых транзакций клиентов, анализируют доходы, расходы и предлагают клиенту совершить различные операции — накопить, сэкономить или пополнить сберегательный счет. Также чат-боты решают задачу финансового консультирования клиентов [14].

Преимущества чат-ботов по сравнению с «живым» общением клиента с технической поддержкой?

- 1) автоматически выполняющаяся персонализация клиента;
- 2) встраивание в естественное поведение клиента при его обращении в Интернет;
- 3) борьба с мошенничеством, так как боты в значительной степени помогают автоматизации, пользователи уведомляются о каждой транзакции;
- 4) за счет использования чат-бота автоматизируются рутинные действия сотрудников.

На скромных серверных мощностях чат-бот может за 1 минуту обрабатывать 2500 текстовых и голосовых обращений клиентов с возможностью увеличения обработки обращений до 25 тыс. в минуту.

Можно говорить об активном проникновении искусственного интеллекта в жизнь людей и неизбежности использования чат-ботов в любой индустрии, предполагающей коммуникацию с большим числом клиентов, каковой и являются банки [19] (см. рис. 1.3).



Рис. 1.3. Основные сферы применения технологий искусственного интеллекта в банках.

Ожидается, что к концу 2020 года 85% банков и предприятий будут взаимодействовать с клиентами с помощью чат-ботов, причем, экономия затрат от использования чат-ботов составит около 8 миллиардов долларов до 2022 года (см. [4]).

Очевидно, что скорость внедрения чат-ботов в банковский бизнес будет увеличиваться.

В настоящее время [14] уровень проникновения роботов в чатах российских банков эквивалентен доле роботизации западных контакт-центров и составляет около 13%.

Искусственный интеллект является чрезвычайно мощным инструментом, который может помочь в улучшении взаимодействий и коммуникации с клиентами банка [4].

В банковской системе современные чат-боты умеют:

- 1) информировать об особенностях продуктов и сервисов;
- 2) предоставлять контактные данные;
- 3) проводить платежные операции;
- 4) давать рекомендации клиенту по финансовым вопросам;
- 5) показывать курсы валют;
- 6) учитывать личные финансы;
- 7) переводить денежные средства с карты на карту;
- 8) отправлять заявки на интернет эквайринг и проверять контрагента по ИНН/ОГРН (ИП);
- 9) отвечать на вопросы, касающиеся банковской сферы [10].

Способы улучшения обслуживания клиентов банков при использовании чат-ботов

1. Прогресс Персонального банкинга.
2. Мощное автоматизированное обслуживание клиентов.
3. Качество обратной связи.
4. Персонализированный маркетинговый опыт клиентов.
5. Повышение производительности труда сотрудников [4].

Как консультанты, способные обслуживать неограниченное количество потребителей [27], чат-боты способствуют повышению качества обслуживания.

Направления использования

- 1) связь с клиентами;

2) корпоративные чат-боты, решающие задачи оказания услуг внутри компании [29].

Чат-бот в банке способен напоминать о сроках погашения кредита, конвертировать валюту и выполнять другие полезные функции. Особенно успешна работа чат-ботов в банках, у которых нет физических точек отделений. В таких банках все взаимодействие с клиентами осуществляется дистанционно [22].

Банковские чат-боты могут быть использованы в качестве альтернативы бесплатным смс-подтверждениям [27]. Чат-боты будут наиболее востребованы как канал по предоставлению популярных услуг: оплата счетов и денежные переводы.

Банки могут более качественно взаимодействовать с большим количеством клиентов, оказывая им мгновенную поддержку, что является большим плюсом такой двусторонне коммуникации. В контакт-центре крупного банка для того, чтобы оперативно отвечать на вопросы многомиллионных клиентов, требуется армия операторов и колоссальные затраты. Передавая чат-ботам простые задачи больших объемов, сотрудники банков могут сконцентрироваться на других важных и более сложных запросах пользователей [27].

Благодаря ботам меняется само офисное пространство, ведь с их помощью можно реализовать систему вознаграждений и поощрений в банке, улучшить физическое и эмоциональное состояние сотрудников, давая рекомендации по несложным физическим упражнениям, которые помогают снять усталость. С помощью бота можно усовершенствовать процесс управления аналитикой и таким образом избавить сотрудников от необходимости постоянно переключаться между несколькими аналитическими приложениями. Боты могут отслеживать упоминания бренда в различных социальных медиа и отраслевые новости, интересующие сотрудника [27]. Бот может выполнять роль толкового интерактивного справочника, интерактивного неструктурированного рубрикатора и каталога услуг.

Уже сейчас боты способны проконсультировать и дать ссылки на материалы не хуже, чем вы получили бы их при общении с живыми людьми. Конечно,

иногда возникают проблемы, ведь чат-боты не обладают интеллектом людей. Именно в этом и заключается задача дальнейшего совершенствования ботов и перспектива развития технологий [27].

Заметим, что уже сейчас любой достаточно крупный банк имеет чат-бота в контактном центре, но, конечно, не все банки используют ботов одинаково [29].

1.4.1. Категории обращения клиентов

1. Обращения общего характера.

Большинство клиентов банков интересуется справочная информация. 90% всех таких обращений составляет около десятка типичных вопросов. Ответы на такие вопросы отлично автоматизируются.

2. Юридические запросы и жалобы.

Это более сложный вариант обращений, так как в основном они связаны с негативом, их поток зависит не только от роста клиентской базы, но и от активности «антиколлекторских помощников».

3. Нестандартные запросы.

Данные обращения связаны с человеческим фактором со стороны клиента. Например, клиент забыл пароль, чтобы войти в личный кабинет. У него может смениться номер телефона и адрес электронной почты [18].

1.4.2. Типы чат-ботов

Чат-боты разделяют на два типа: кнопочные («интерфейсные») и разговорные [5].

Кнопочные чат-боты достаточно просты в освоении, но малофункциональны. Если сделать меню с более сложным и разветвленным, то получится не очень удобный интерфейс, разобраться в котором будет совсем непросто.

Разговорные боты, которые реагируют на реплики собеседника и могут вести с ним диалог, подразделяются на:

- 1) боты, которые используют искусственный интеллект;
- 2) боты, основанные на языковой модели и правилах.

Для обучения разговорных чат-ботов с искусственным интеллектом требуются огромные объемы специально подготовленных данных, собрать которые по силам далеко не каждому банку. Поэтому из-за недостаточности обучения такие чат-боты могут вести себя непредсказуемо, неправильно отвечать на вопросы, неверно оценивать намерения клиента и т.п. [5].

Стоит отметить, что оптимальным вариантом бота является «гибрид», в котором берется все лучшее, что представлено в ботах всех типов. В таких «гибридных» чат-ботах некоторые функции могут быть вынесены на кнопки, но при этом они имеют возможность параллельно вести диалог с собеседником и исполнять текстовые «команды» [5].

1.4.3. Трудности

1. Самое сложное наладить протокол взаимодействия API между чат-ботом и серверами банка. При запрашивании клиентом информации из своего личного кабинета (например, о размере задолженности) необходима связь с внутренней базой данных.

2. Человеческий язык слишком тяжел для машинного понимания. Надо учитывать и тот факт, что клиенты часто используют сокращения, сленг, могут допустить опечатки.

3. Необходимо использовать технологии машинного обучения и нейронных сетей при нестандартных запросах [18].

1.5. Использование чат-ботов в зарубежных банках

В мире насчитывается огромное количество чат-ботов. США является одним из лидеров по использованию ботов, где создано более 150 тыс. чат-ботов. Данные чат-боты позволяют клиентам переписываться с банком и оплачивать счета, получать информацию о счете и остатках, используя обычный язык. Они также создают финансовый профиль, в котором объединены рекомендации по финансовым операциям и подбору инструментов для торговли [24].

Лидер рынка США по использованию мобильного банкинга и внедрению искусственного интеллекта Bank of America запустил чат-бота по имени Erica (от слова AmErica), который используется для предоставления информации об остатке на счете, отправки различных уведомлений клиентам, предоставления обновлений в отношении кредитных отчетов, предоставления предложений о том, как можно сэкономить деньги, оплачивать счета и помогает клиентам с транзакциями. Это виртуальный помощник, который помогает клиентам сделать правильный выбор [4].

Erica доступна в мобильном приложении Bank of America. Клиенты могут сообщать ей о своих банковских проблемах голосом, либо же текстовыми командами. Этот банковский чат-бот также дает персональные рекомендации, предложения и советы после анализа данных клиента, и даже может отправлять образовательные видеоролики по финансам [26].

Банк JPMorgan Chase использует чат-ботов для оптимизации операций бэк-офиса. Банк запустил COIN, который анализирует сложные контракты быстрее и эффективнее, чем это делают обычные юристы. Чат-бот также используется для анализа сообщений для сотрудников, предоставления доступа к программ-

ным системам и выполнения базовых IT-запросов, вроде сброса паролей. В будущем банк намерен продолжать использовать чат-ботов для поиска источников дохода, способов снижения расходов и рисков [26].

Американская банковская холдинговая компания Capital One представила текстового чат-бота по имени Eno, чтобы помочь клиентам управлять своими финансами через мобильные телефоны. Этот бот также может мгновенно давать информацию о балансе, истории транзакций и кредитном лимите, оплачивать счета. Кроме того, бот понимает emoji, отправляемые клиентами. У Capital One есть еще один виртуальный помощник, работающий через приложение для Amazon Alexa. Он понимает голосовые команды и предоставляет информацию о предстоящих платежах, проверяет баланс и может совершать платежи [26].

Ally Bank одним из первых среди банков запустил чат-бот. Его Ally Assist появился в мобильном приложении банка еще в 2015 году. Он понимает голосовые и текстовые команды, позволяет осуществлять платежи, переводы, P2P-транзакции и открывать депозиты. Клиент может запрашивать сводку по своему счету или историю транзакций, а также отслеживать свои накопления и расходы. Ally Assist использует машинное обучение, вследствие чего может прогнозировать потребности клиентов, анализируя счета и транзакции для предоставления справок и сообщений на соответствующие темы. Этот помощник использует и естественный язык для удовлетворения общих запросов клиентов [26].

Банк USAA вышел за рамки традиционных банковских цифровых решений, предложив не бота в традиционном понимании, а нечто совершенно иное. В проекте Clinc для Amazon Alexa используются сложные системы обработки естественного языка, которые обучены знаниям финансовой и банковской индустрии. При помощи машинного обучения расширяется база знаний и улучшаются ответы при каждом запросе. Clincуже способен понимать, запоминать и реагировать на неограниченный, контекстуальный и беспорядочный человеческий язык [26].

Клиенты USAA получают опыт, аналогичный привычному человеческому разговору. Через API решение Clinc можно легко подключить к мобильному приложению и другим цифровым продуктам [26].

Аму — платформа обслуживания клиентов, которая принимает форму виртуального чат-бот помощника для корпоративного банковского обслуживания в HSBC (Гонконг). Аму отвечает на запросы клиентов в режиме 24/7. Бот доступен на ПК и мобильных устройствах на английском и упрощенном китайском языках. Аму доступна на страницах нескольких продуктов, но ее охват продолжает расширяться.

Наличие встроенного механизма обратной связи с клиентами позволяет боту учиться с течением времени. Таким образом он может справляться с более сложными запросами.

Банк SEB (Швеция) запустил два чат-бота — Aida для клиентов и Amelia для сотрудников [26].

В банковской холдинговой компании Wells Fargo & Company используют мессенджер Facebook в связке с чат-ботом. Бот предоставляет информацию о том, где находится ближайший банкомат, и об остатке на счете клиента.

Банк Commonwealth Bank (Австралия) запустил Seba, чат-бота, который используется для выполнения таких задач, как активация карты, осуществление платежей, проверка баланса счета или получение наличных денег без карты [4].

1.6. Использование чат-ботов российскими банками

Российские банки не отстают от мировых тенденций. Свои чат-боты имеют «Сбербанк», ВТБ, «Тинькофф банк», «Альфа-Банк» [18].

Программа «Добро пожаловать в Сбербанк!» была разработана специально для руководителей, пришедших в банк по внешнему найму. Чтобы сотрудники

охотнее знакомилась с этой программой, было принято решение создать чат-бот, который возьмет организационные задачи на себя [28].

Данный бот получил имя Олег (сокращение от «он-лайн-коллега») и взял на себя роль teacher assistant. Общение с ботом происходит нативно и встроено в рабочую коммуникацию. Принцип использования этого бота представлен на рис. 1.4.

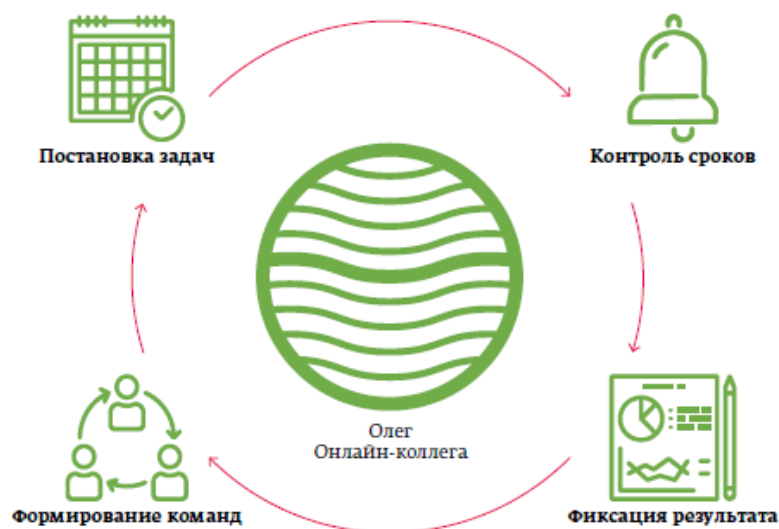


Рис. 1.4. Чат-бот Олег в программе «Добро пожаловать в Сбербанк».

Программа «Сбербанк Мини-МВА» — это уровневая программа для линейных руководителей. По итогам испытаний из тысячи участников отбирают 300 лучших. Чат-боты помогают сделать общение со слушателями программы «Сбербанк Мини-МВА» персональным, не тратить время на организационные вопросы и сфокусироваться на обсуждении бизнес-проектов [28].

В Промсвязьбанке планируется разработка самообучающегося чат-бота, который поможет по запросу клиента регистрировать бизнес и в зависимости от специфики бизнеса клиента подбирать ему наиболее подходящий тариф обслуживания. Кроме того, бот поможет предпринимателям зарезервировать расчетный счет и подскажет список необходимых документов для предоставления в офис банка, либо самостоятельно направит менеджера к клиенту в офис для за-

вершения всех формальностей [20]. Такой виртуальный помощник позволит снизить нагрузку на контакт-центр для малого и среднего бизнеса до 30%, уменьшить время получения информации.

Данный формат «единого окна» позволит сэкономить время клиентов банка, а специалисты контакт-центра уделят больше внимания персональным вопросам клиентов.

Банк «Точка» и Райффайзенбанк внедрили обучающегося чат-бота в режиме помощника оператора. Бот предоставляет подсказки оператору для ответов на распространенные вопросы [29].

Если клиент нервничает, то программа не в состоянии составить адекватный диалог. Зато сотрудник, обладая необходимым навыком коммуникаций, может успокоить собеседника.

Среди ключевых функций виртуальных ассистентов также следует выделить распознавание эмоций (sentiment analysis). Настроение человека на протяжении разговора может меняться, и хороший бот замечает перемены в эмоциях и отвечает соответственно. Это делает общение более похожим на обычное общение между людьми. Кроме того, такой сентимент-анализ позволяет уловить момент, когда напряженный диалог следует передать оператору [29].

В некоторых банках чат-боты работают без участия оператора. Примером является банк Хоум Кредит, в котором бот, без участия оператора, отвечает на 20% вопросов клиентов [29]. Использование данного бота позволило банку сэкономить на закупке специализированного ПО и не потребовало создания дополнительных рабочих мест [30].

Хорошо написанный чат-бот в некоторых случаях способен отлично работать на первой линии техподдержки. В Альфа-Банке обрабатываются реальные диалоги с клиентами банка: с помощью специальных алгоритмов переписка «раскладывается» на смысловые кластеры, на базе которых впоследствии формируется пул текстовых подсказок. Эти подсказки автоматически предлагаются сотруднику техподдержки, значительно сокращая время, затрачиваемое им на написание ответов клиентам [30].

Даже внутри одной организации чат-бот может использоваться в разных точках взаимодействия. К примеру, Ак Барс Банк внедрил функционал в контакт-центр в зону работы как с уже авторизованными клиентами, так и с новыми пользователями. Данный бот используется и для внутренней коммуникации сотрудников [29].

RS-Bot, разработанный компанией R-Style Softlab, — чат-бот для банков. Он может давать информацию об офисах и филиалах банка, режиме их работы, продуктах и услугах, курсах валют, предоставлять выписки или информацию об остатках на счетах и даже выполнять платежи по предварительно настроенным шаблонам [29].

Преимущества для клиентов банков

Прежде всего клиента должна привлекать скорость взаимодействия с банком. В отличие от «общения» с банком через приложения для смартфона, в мессенджер не нужно лишний раз заходить, вводить какие-либо коды или ждать их прихода по СМС. При этом во время использования такого интерфейса нет необходимости придерживаться каких-либо правил: вопрос может быть задан в вольной форме. «Автоответчик» либо сам попытается распознать его, либо, в случае затруднений, переложит эту задачу на плечи настоящего оператора. Еще одна немаловажная деталь — персональный сервис. Клиент получает возможность задать вопрос напрямую банку и тут же получить персонализированный ответ [30].

Минусы

Однако в использовании мессенджеров как средства взаимодействия между банком и вкладчиком есть свои ограничения и минусы. К примеру, за идентификацию и аутентификацию пользователя отвечает не банк, а сам мессенджер. В случае с Telegram возможна подмена идентификатора пользователя и получение данных по чужим договорам и движений средств по счетам.

Нельзя забывать о том, что в Telegram сообщения сохраняются на сервере до тех пор, пока пользователь сам их не удалит. А как показывает практика, получить доступ к переписке хоть и сложно, но все-таки можно: уязвимость SS7 никуда не делась [30].

Чат-боты могут стать оружием для фишинга, что гораздо опаснее, чем просто электронное письмо [25]. Человек, который общается с чат-ботом, не имеет возможности проверить с кем он говорит, а злоумышленнику в ходе непринужденного разговора легко заставить жертву нажать на ссылку, при этом хакерам не придется направлять собеседника на поддельный веб-сайт. Если хакеры захотят получить некоторую персональную информацию от пользователя, они могут просто спросить его об этом [25].

Безопасность самого канала — это еще один фактор, который необходимо принимать во внимание при использовании чат-бота [25].

Необходимо обращать внимание на то, какая информация предоставляется чат-ботам. При общении с ботом, как с человеком, можно случайно раскрыть лишнюю конфиденциальную информацию [25].

С одной стороны, мы видим, что чат-боты улучшают работу банков и способ взаимодействия с пользователями, а с другой стороны, их популяризация принесет дополнительные проблемы для информационной безопасности [25].

Глава 2. Прототип чат-бота для банка

2.1. Функционирование прототипа чат-бота

Открытая библиотека DeepPavlov для создания диалоговых систем содержит набор претренированных компонент для анализа языка, включая компоненты текстовой классификации.

В обучении чат-бота участвуют три файла: данные обучения (`assistant-trn.json`), проверки (`assistant-val.json`) и тестирования (`assistant-tst.json`). Пример сценария «вопрос–ответ» приведен в приложении 1, где `speaker 1` — это клиент банка, а `speaker 2` — чат-бот. При данном (небольшом) наборе данных время обучения чат-бота составляет 20 мин.

В первоначальной базе имелась информация об адресе банка, его режиме работы. Результат работы прототипа чат-бота после обучения на малой базе данных (10 диалогов) представлен на рис. 2.1.

```
bot(['Здравствуйте'])
```

```
['Здравствуйте, чем я могу вам помочь?']
```

```
bot(['Когда работает банк?'])
```

```
['Банк работает до 19:00 в буднии дни']
```

```
bot(['Спасибо'])
```

```
['Рад был помочь! До свидания!']
```

```
bot(['Привет'])
```

```
['Здравствуйте, чем я могу вам помочь?']
```

```
bot(['Адрес главного офиса'])
```

```
['Ул. А. Невского 14']
```

```
bot(['Очень далеко'])
```

```
['Какая жалость! В следующий раз, может быть. Хорошего дня!']
```

```
bot(['Я не успею'])
```

```
['Банк работает до 19:00 в буднии дни']
```

Рис. 2.1. Диалог чат-бота и клиента.

Из диалога (см. приложение 1) видно, что чат-бот отвечает корректно лишь на узкий круг вопросов, так как было прописано мало сценариев диалогов.

Расширим базу вопросов. Добавим возможность для пользователя узнать курсы валют, включим дополнительную информацию о банкоматах и расширим информацию о режимах работы банка.

После обучения на расширенной базе данных, состоящей из 49 диалогов (см. приложение 2), прототип чат-бота стал отвечать более гибко и конкретно (см. рис. 2.2).

```

bot(['Привет, бот'])

['Здравствуйте, чем я могу вам помочь?']

bot(['Время работы банка'])

['Вас интересует полное расписание или конкретный день?']

bot(['Полное'])

['ПН: с 09:00 до 18:00 ВТ: с 09:00 до 18:00 СР: с 09:00 до 18:00 ЧТ: с 09:00 до 18:00 ПТ: с 09:00 до 18:00 СБ: с 10:00 до 17:00 ВС: выходной день']

bot(['Спасибо'])

['Рад был помочь! До свидания!']

bot(['Привет'])

['Здравствуйте, чем я могу вам помочь?']

bot(['До сколько работает банк?'])

['Вас интересует полное расписание или конкретный день?']

bot(['Суббота'])

['СБ: с 10:00 до 17:00']

bot(['Не успею'])

['Какая жалость! Я могу Вам еще чем-то помочь?']

```

Рис. 2.2. Диалог с использованием расширенной базы вопросов.

Код программы проиллюстрирован в приложении 3.

2.2. Прикладная функциональность чат-бота

Данная версия чат-бота предназначена для использования в банковской сфере. С его помощью можно:

- получить контактную информацию о филиалах и офисах Сбербанка, режиме их работы;
- узнать курсы валют, установленные ЦБ.

На вопрос о курсе валют чат-бот запрашивает какой тип валюты интересует клиента и предлагает выбрать конкретную валюту из трех видов валют. После чего дается информация о конкретной валюте.

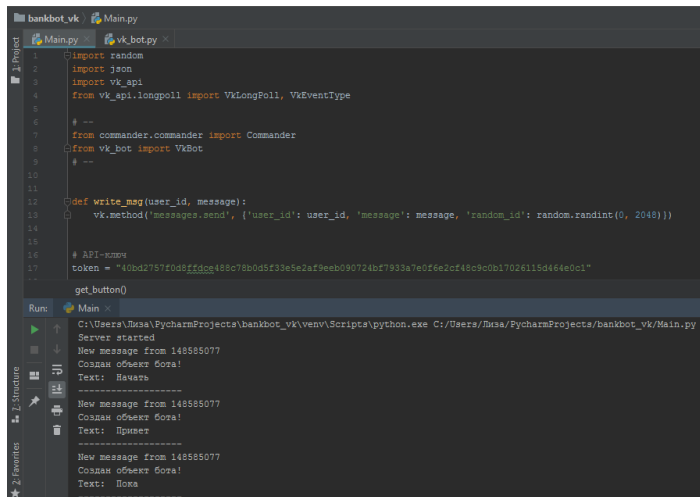
Во время диалога с прототипом чат-бота все слова клиента записываются и упорядочиваются по частоте употребления (см. рис. 2.3).

1	Привет	14
2	Спасибо	13
3	работы	11
4	банка	9
5	Здравствуйте	8
6	банкомата	8
7	Адрес	6
8	режим	6
9	Курс	6
10	валют	6
11	Время	5
12	Не	4
13	Где	3
14	выгодно	3
15	Привет,	2
16	бот	2
17	работает	2
18	главный	2
19	офис	2
20	банка?	2
21	находится	2
22	Отлично	2
23	Суббота	2
24	Далеко	2

Рис. 2.3. Используемые слова при диалоге с чат-ботом.

2.3. Пример работы чат-бота в «ВКонтакте»

Для того, чтобы понять, возможно ли подключить чат-бот, написанный на языке Python, к социальной сети «ВКонтакте», мною был создан простой бот (необученный), который реагирует на заранее прописанный вопросы пользователя (см. рис. 2.4). По userID он определяет имя пользователя, зарегистрированного «ВКонтакте». Таким образом, чат-бот знает, как обратиться к клиенту.



```

1  import random
2  import json
3  import vk_api
4  from vk_api.longpoll import VkLongPoll, VkEventType
5
6  # --
7  from commander.commander import Commander
8  from vk_bot import VKBot
9  # --
10
11
12  def write_msg(user_id, message):
13      vk.method('messages.send', {'user_id': user_id, 'message': message, 'random_id': random.randint(0, 2048)})
14
15
16  # API-ключ
17  token = "40bd2757f0d8ffdcg488c78bd05f33e5e2af9eeb090724bcf7933a7e0f6e2cf48c9c0b17026115d664e0c1"
18
19  get_button()

```

Run: Main

C:\Users\Tara\PycharmProjects\bankbot_vk\venv\Scripts\python.exe C:/Users/Tara/PycharmProjects/bankbot_vk/Main.py

Server started

New message from 148585077

Создан объект бота!

Text: Начать

New message from 148585077

Создан объект бота!

Text: Привет

New message from 148585077

Создан объект бота!

Text: Пока

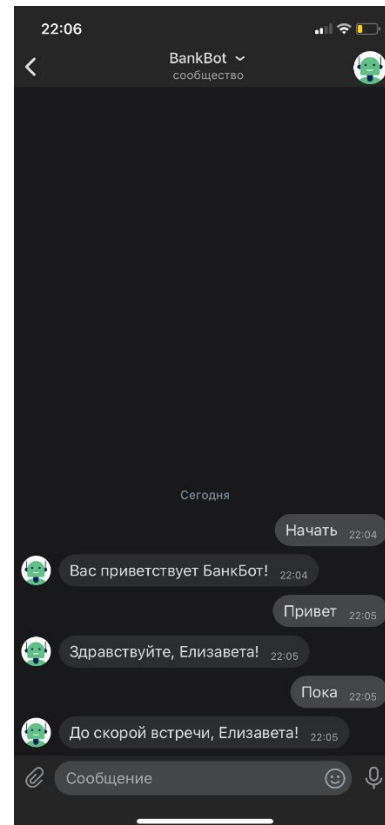


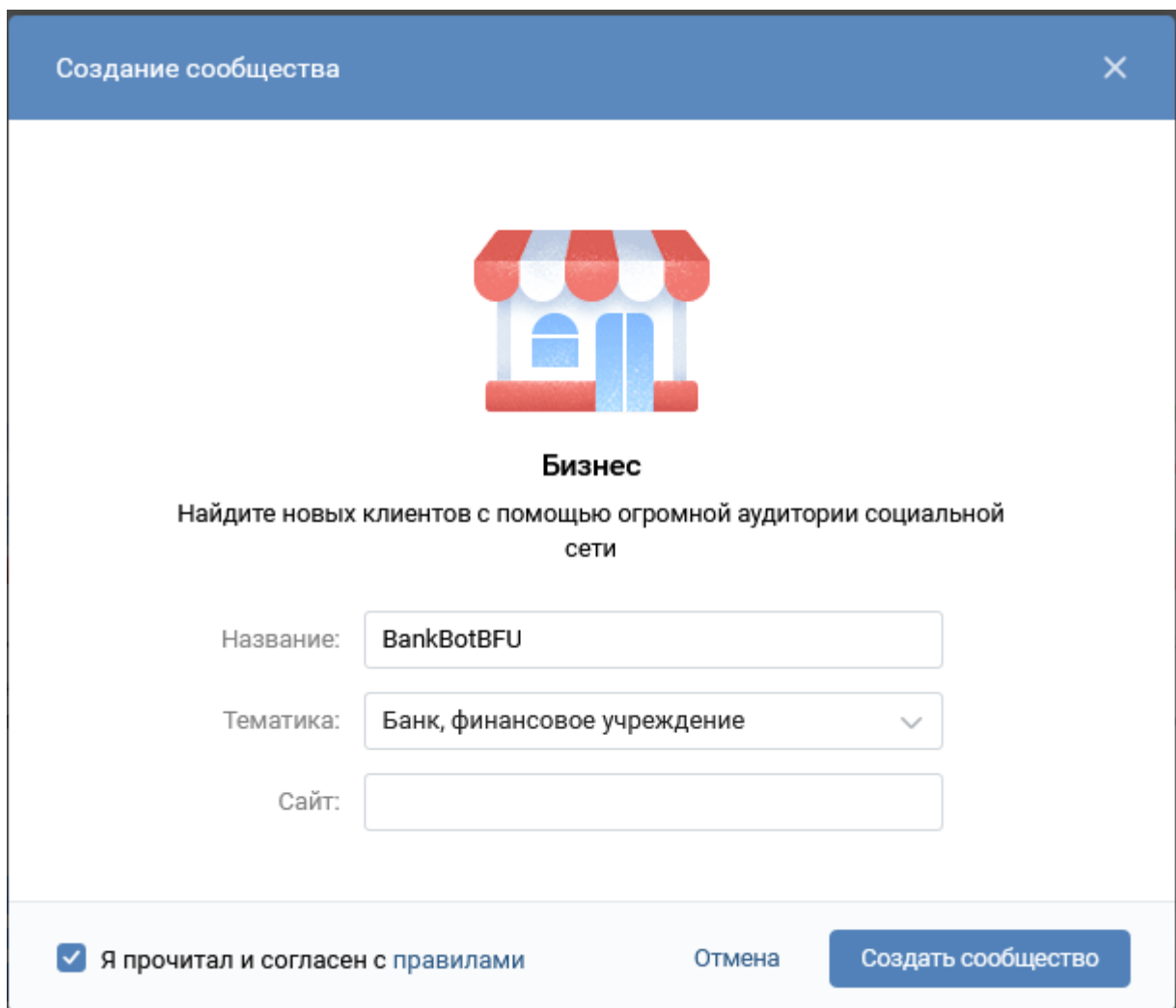
Рис. 2.4. Простой чат-бот в социальной сети «ВКонтакте».

Глава 3. Чат-бот для банка

3.1. Создание сообщества

В социальной сети «ВКонтакте» было создано сообщество, в котором клиенты общались с ботом.

Первоначально выбираются название бота и тематика (профиль). В данном случае нас интересует банковская сфера (см. рис. 3.1).



Создание сообщества

Бизнес

Найдите новых клиентов с помощью огромной аудитории социальной сети

Название: BankBotBFU

Тематика: Банк, финансовое учреждение

Сайт:

☒ Я прочитал и согласен с правилами

Отмена

Создать сообщество

Рис. 3.1. Регистрация сообщества.

Вводим основную информацию сообщества с указанием адреса (рис. 3.2).

Основная информация

✓ **Изменения сохранены.**
Основная информация и настройки сообщества сохранены.

Название:

Описание сообщества:

Обложка сообщества: [Добавить](#)

Адрес страницы:
Вы можете создать наклейки для Вашего сообщества [здесь](#).

Верификация: [Подать заявку](#)

[Сохранить](#)

BankBotBFU
[вернуться к странице](#)

Настройки

- [Разделы](#)
- [Комментарии](#)
- [Ссылки](#)
- [Адреса](#)
- [Меню](#)
- [Работа с API](#)
- [Истории](#)
- [Участники](#)
- [Сообщения](#)
- [Беседы](#)
- [Приложения](#)
- [Денежные переводы](#)
- [Рекомендации](#)

Рис. 3.2. Основная информация.

Для связи обученного прототипа чат-бота с сообществом нам потребуется ключ доступа. Для этого переходим в раздел «Работа с API» (см. рис. 3.3).

Ключи доступа Callback API Long Poll API [Создать ключ](#)

Здесь будут выводиться полученные ключи.

BankBotBFU
[вернуться к странице](#)

Работа с API

- [Настройки](#)
- [Разделы](#)
- [Комментарии](#)
- [Ссылки](#)
- [Адреса](#)
- [Меню](#)

Рис. 3.3. Работа с API.

Нажимаем на кнопку «Создать ключ», после этого появляется окно с выбором необходимых прав для создаваемого ключа (см. рис. 3.4). Выбираем все предлагаемые категории.

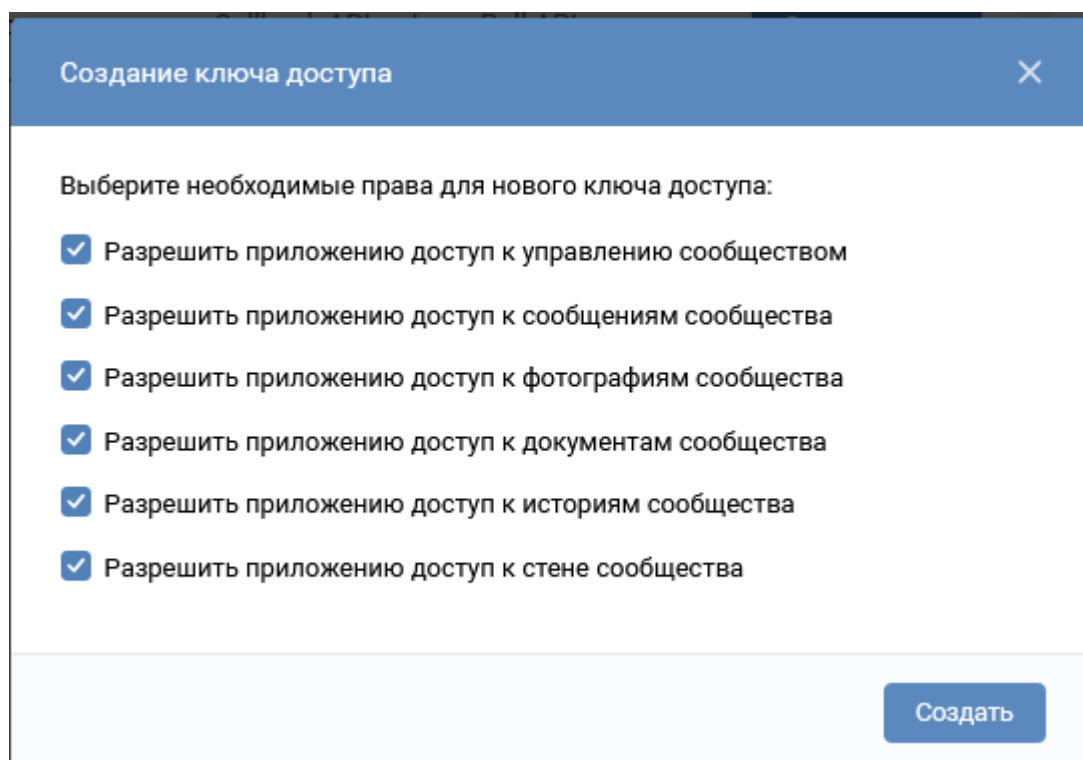





Рис. 3.4. Создание ключа доступа.

После нажатия на кнопку «Создать» требуется подтверждения действий. Нажимаем кнопку «Позвонить» (см. рис 3.5).

 Подтверждение действия 



Для подтверждения действия мы совершим звонок-сброс с уникального номера на Ваш мобильный телефон. Это бесплатно.

Мобильный телефон


[Изменить номер](#)

[Позвонить](#)

Рис. 3.5. Подтверждение действий.

При этом на указанный вами ранее при регистрации в социальной сети мобильный номер поступит звонок-сброс. Необходимо запомнить последние четыре цифры номера, с которого совершается звонок, и ввести их в соответствующее поле (см. рис. 3.6).

Подтверждение действия



Для подтверждения действия мы совершим звонок-сброс с уникального номера на Ваш мобильный телефон. Это бесплатно.

Мобильный телефон

+7*****05

[Изменить номер](#)

Последние 4 цифры номера

Отправить код

Отправить код в SMS через 1:38

В течение нескольких секунд на Ваш телефон поступит звонок-сброс с уникального номера. Введите **последние 4 цифры** этого номера.

Пример: **3487**

BankBotB

вернутьсяс

астройки

Разделы

Комментарии

Ссылки

Адреса

Меню

Работа с API

Истории

частники

екомендации

урнал действ

Рис. 3.6. Введение кода.

Если код введен правильно, мы получаем ключ доступа (рис. 3.7).

Ключи доступа 1

Callback API

Long Poll API

Создать ключ

bb8333803012ca2a7e41b607dd2a5aed569a32b420788dd750c1bf5cee68dd085030

Права доступа: управление сообществом, сообщения сообщества, фотографии, документы, истории, стена

Создан Е. Беловой, сегодня в 13:38

Удалить ключ

BankBotBFU

вернуться к странице

Настройки

Разделы

Комментарии

Ссылки

Адреса

Меню

Работа с API

Рис. 3.7. Созданный ключ доступа.

Ключ доступа нужно держать в секрете, чтобы злоумышленник не смог управлять сообществом.

Далее настраиваем работу с сообщениями (см. рис. 3.8).

Рис. 3.8. Настройка работы с сообщениями.

3.2. Разработка чат-бота

Обученного бота, написанного на языке Python с использованием библиотеки Deerpavlov, внедряем в социальную сеть «ВКонтакте», при этом используем сценарии диалогов (см. рис. 3.9).

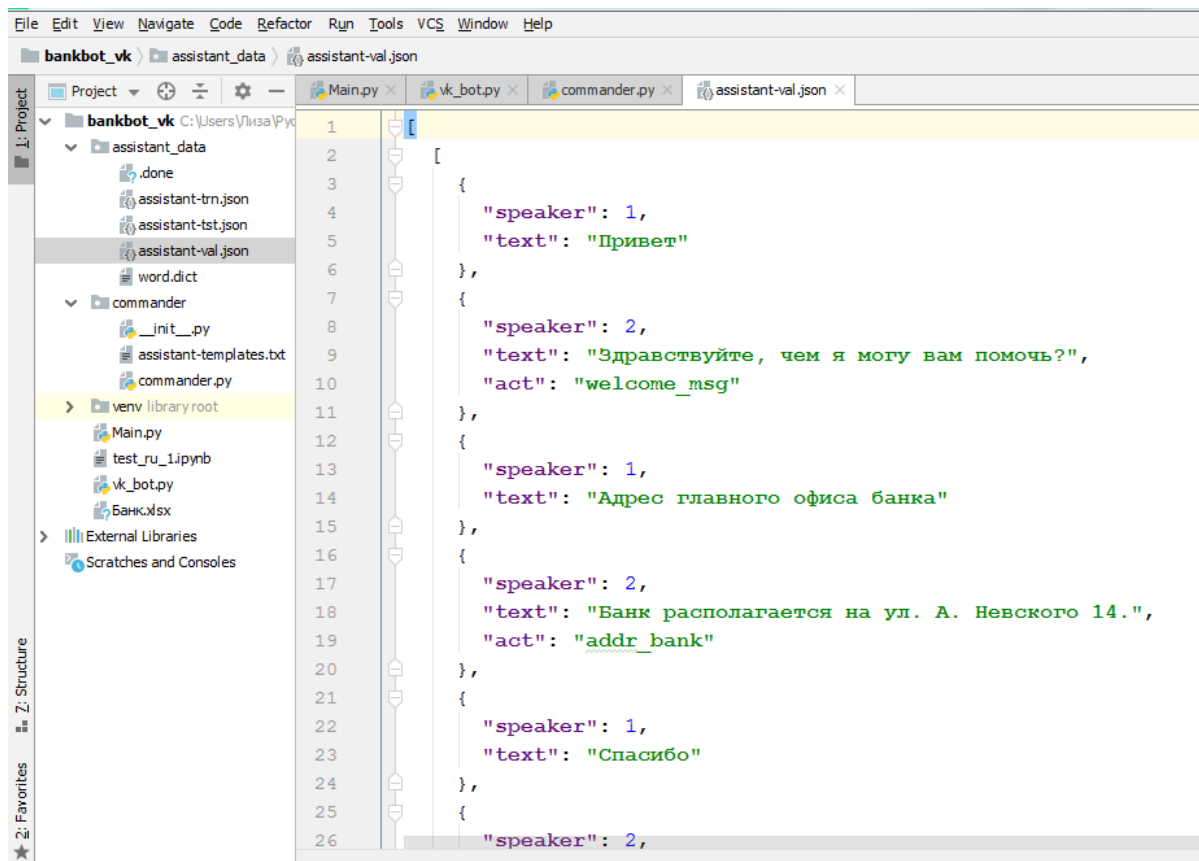


Рис. 3.9. Сценарии диалогов.

Добавим в проект библиотеки для взаимодействия обученного бота с социальной сетью «ВКонтакте»: vk, vk-api, vk-api-helper, vk-bot, vk-client, jupyter, jupyter-client, jupyter-console, jupyter-core, deeppavlov, deeppavlov-agent (см. рис. 3.10).

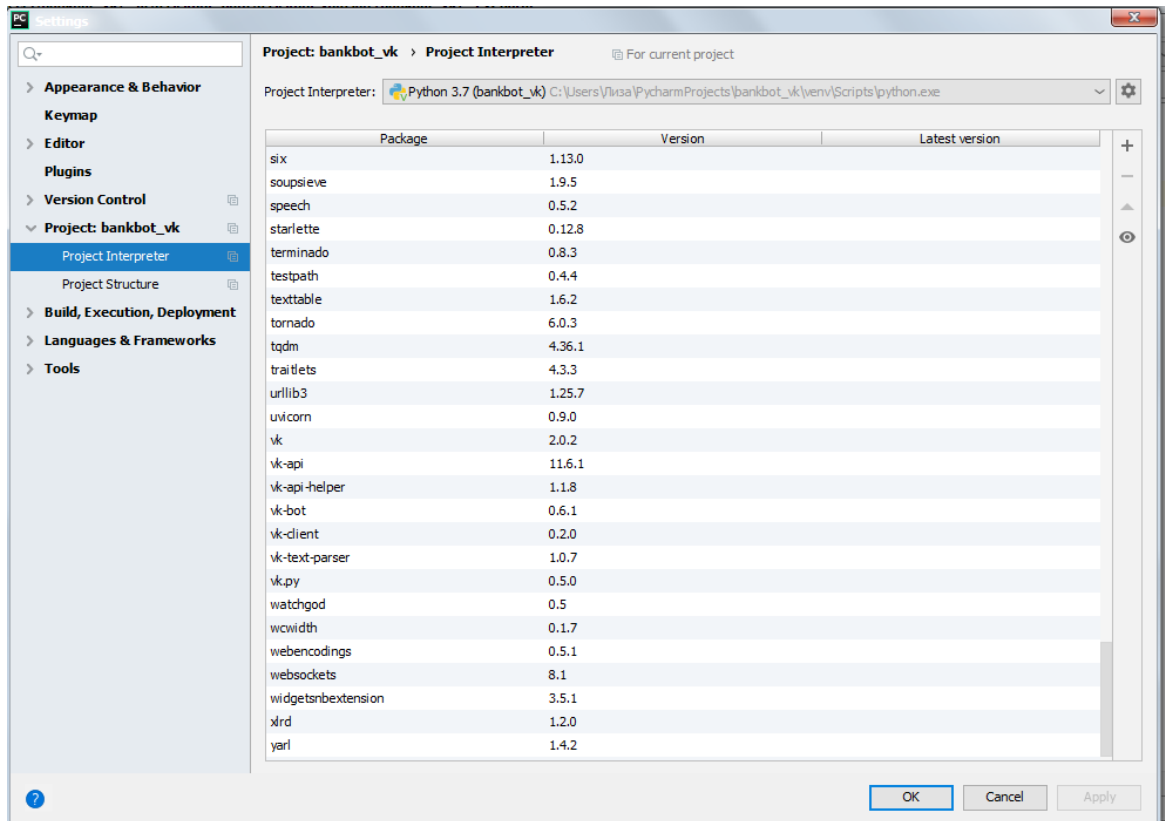


Рис. 3.10. Установленные библиотеки.

На рис. 3.11 представлен код для запуска бота в ранее созданное сообщество.

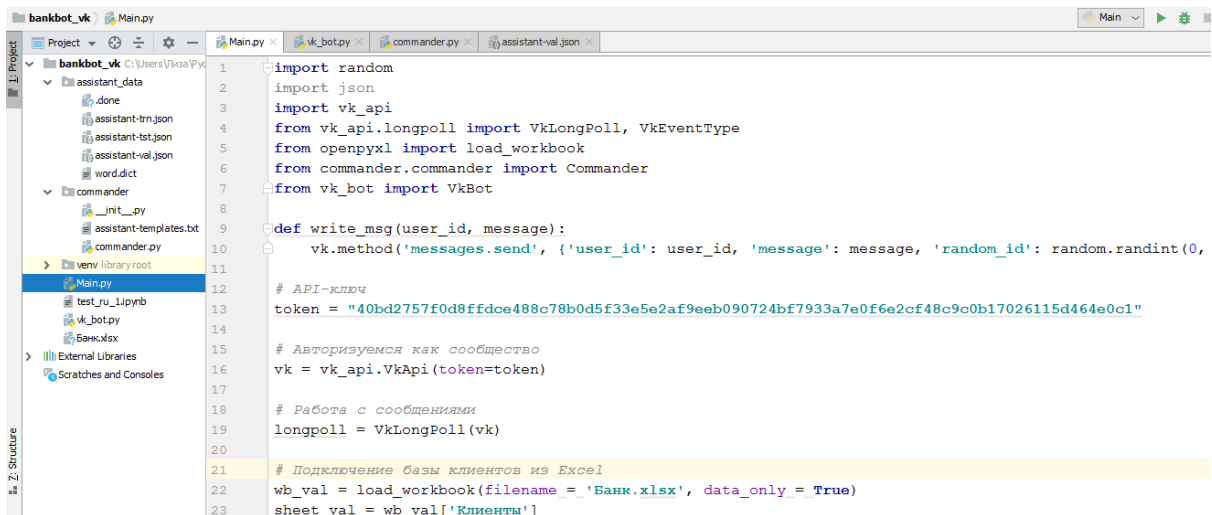


Рис. 3.11. Запуск бота.

Связываем код с файлами обученного бота (см. рис. 3.12).

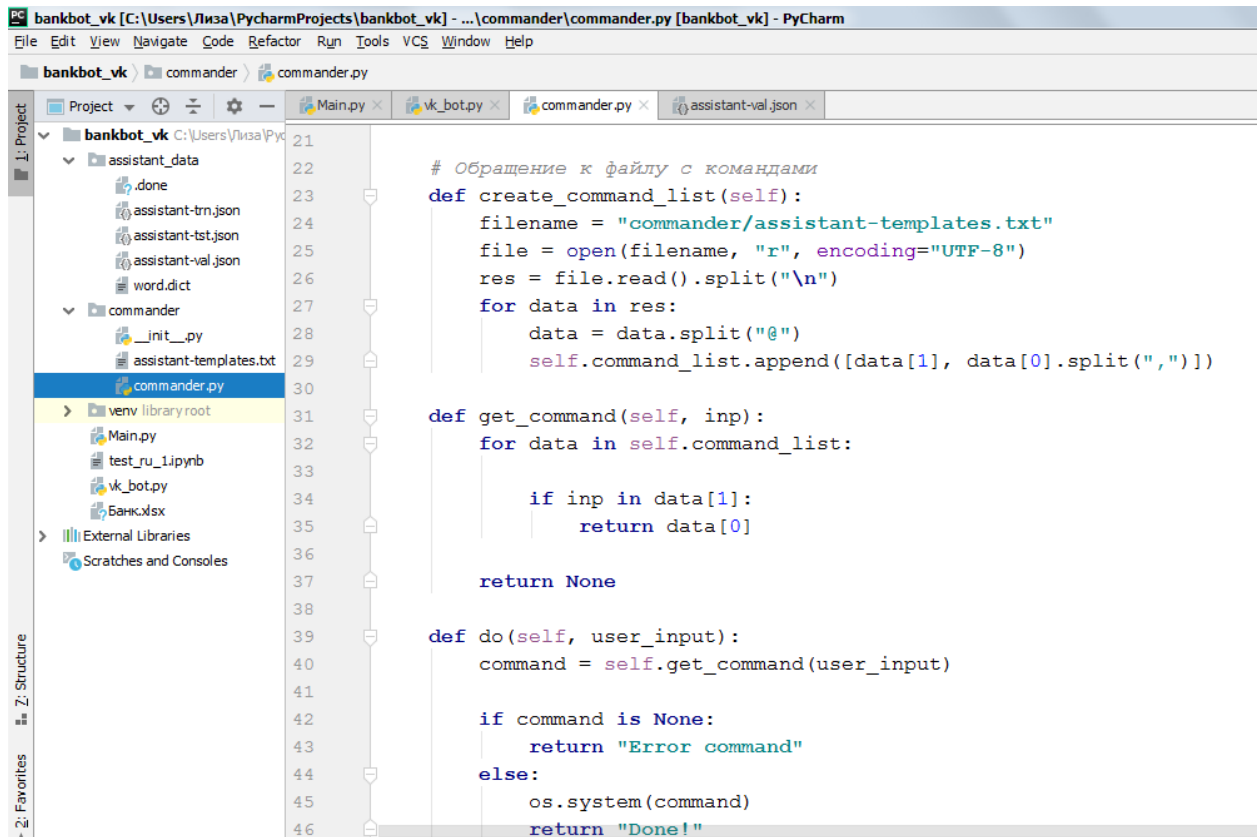


Рис. 3.12. Обращение к файлу assistant-templates с командами.

Если пользователь ввел слово некорректно (с опечатками), то бот поймет сообщение, так как имеется проверка (см. рис. 3.13).

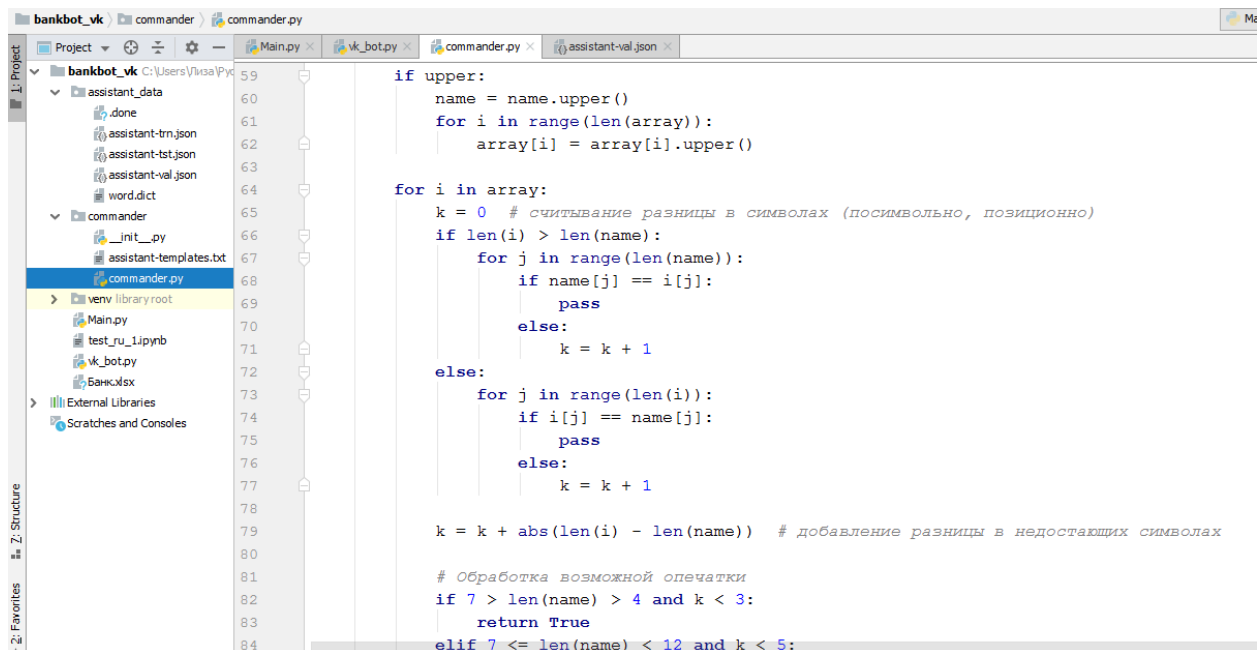


Рис. 3.13. Обработка разницы в символах и возможных опечаток.

Для получения данных по курсу валют и стоимости драгоценных металлов воспользуемся двумя библиотеками: `requests` и `beautifulsoup4`. Первый из перечисленных пакетов обеспечивает работу с `http`-запросами, т.е. возможность отправки `http`-запросов и получения контента в ответ на данные запросы, второй помогает распарсить этот контент (получить `http`-страницу). Библиотека `beautifulsoup4` позволяет из `DOM`-дерева получить объекты Python, с которыми можно дальше работать. Заметим, что каждый узел может быть представлен в виде объекта Python.

Импортируем вышеперечисленные библиотеки. В код добавляем константы (ссылку и заголовки). Заголовки нам будут нужны для того, чтобы в процессе парсинга сервер не посчитал нас за ботов и не заблокировал. Таким образом, с помощью заголовков мы будем имитировать работу браузера.

Опишем следующие две функции:

1) `get_html` (см. Рис. 3.14) содержит два аргумента (`url` — ссылку на страницу и `params` — параметр, который равен `None`). Аргумент `params` нужен для того, чтобы была возможность передавать дополнительный параметр к этому адресу;

```
def get_html(url, params = None):
    r = requests.get(url, headers = HEADERS, params = params)
    return r
```

Рис. 3.14. Функция `get_html`.

2) `parse` (см. Рис. 3.15).

```
def parse():
    html = get_html(URL)
    print(html.status_code)
```

Рис. 3.15. Функция `parse`.

Запускаем `parse`. Если `status_code=200`, это значит, что мы достучались до страницы.

Теперь можем получить текст страницы.

Модифицируем ранее прописанные функции, получим информацию о курсах валют и стоимости драгоценных металлов (см. Рис. 3.16).

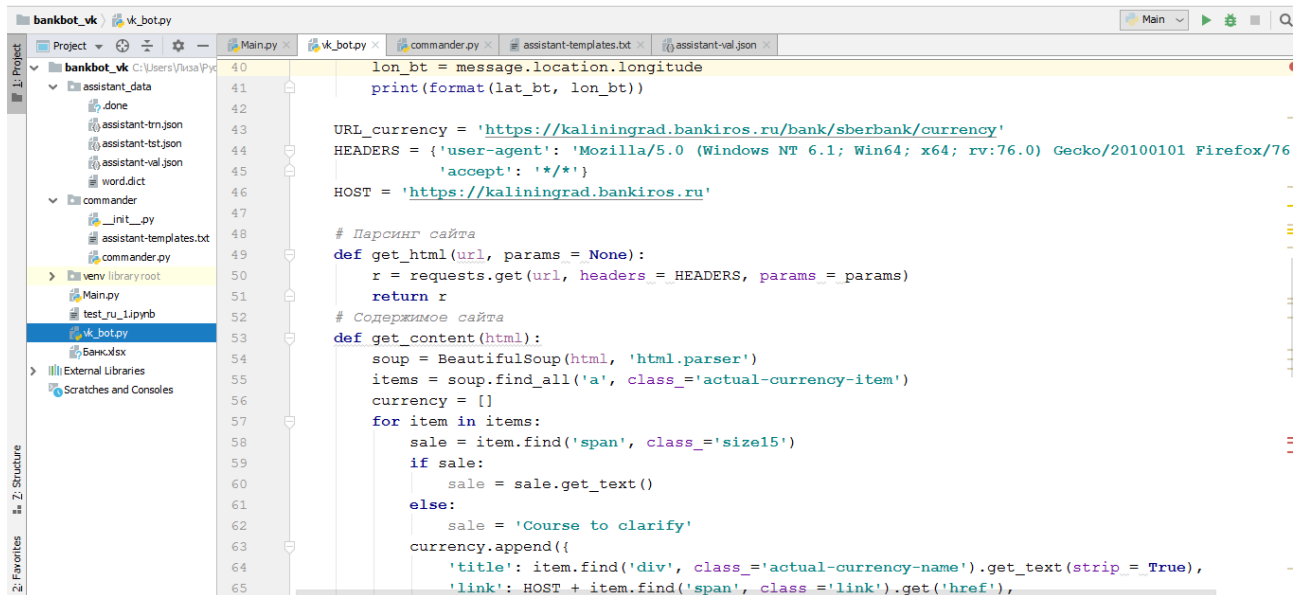


Рис. 3.16. Получение данных с сайта.

Считается, что у пользователя имеется счет в банке. База клиентов содержится в Excel файле. В данном файле указана следующая информация о клиентах (фамилия, имя, отчество, счет и номер телефона). Код обращения к указанному файлу имеет вид (см. Рис. 3.17).

```

from openpyxl import load_workbook
wb_val = load_workbook(filename = 'Банк.xlsx', data_only = True)
sheet_val = wb_val['Клиенты']

```

Рис. 3.17. Код обращения к базе данных клиентов.

В данном фрагменте кода в переменной `wb_val` указывается название Excel файла, а в переменной `sheet_val` — наименование листа Excel файла. Таким образом, мы можем обратиться к нужной ячейке файла.

3.3. Размещение бота на хосте

Для того чтобы чат-бот был доступен для пользователей, размещаем его на бесплатном хосте Pythonanywhere (см. рис. 3.18).



Create your account

Username:

Email:

Password:

Password (again):

☒ I agree to the [Terms and Conditions](#) and the [Privacy and Cookies Policy](#), and confirm that I am at least 13 years old.

[Register](#)

We promise not to spam or pass your details on to anyone else.

Рис.3.18. Регистрация на сайте.

После регистрации на сайте необходимо установить нужные библиотеки для «ВКонтакте» (рис. 3.19).

```

17:35 ~$ pip3.6 install --user vk_api
Looking in links: /usr/share/pip-wheels
Collecting vk_api
  Downloading https://files.pythonhosted.org/packages/74/7c/435c5e1d7ac7b688b967f0bece58687671dcad6b4260eadelb049dc16403/vk_api-11.8.0.tar.gz
Requirement already satisfied: requests in /usr/lib/python3.6/site-packages (from vk_api) (2.22.0)
Requirement already satisfied: six in /usr/lib/python3.6/site-packages (from vk_api) (1.12.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/lib/python3.6/site-packages (from requests->vk_api) (3.0.4)
Requirement already satisfied: certifi<2017.4.17 in /usr/lib/python3.6/site-packages (from requests->vk_api) (2019.9.11)
Requirement already satisfied: idna<2.9,>=2.5 in /usr/lib/python3.6/site-packages (from requests->vk_api) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/lib/python3.6/site-packages (from requests->vk_api) (1.24.3)
Building wheels for collected packages: vk_api
  Building wheel for vk_api (setup.py) ... done
  Created wheel for vk_api: filename=vk_api-11.8.0-cp36-none-any.whl size=42251 sha256=ff681ff52fa8ff8b52bd21395f5dd8839750a8a504edf8a24f401144c3434
0fb
  Stored in directory: /home/bankbottest/.cache/pip/wheels/39/bb/05/c3d370a1097f993c1089e0c6265f7cb782b4a583110f747805
Successfully built vk_api
Installing collected packages: vk_api
Successfully installed vk_api-11.8.0

17:40 ~$ pip install --user gtts
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that
date. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/
latest/development/release-process/#python-2-support
Looking in links: /usr/share/pip-wheels
Collecting gtts
  Using cached https://files.pythonhosted.org/packages/66/4f/e10fe72d83ab5dd9e33eb5eb3b3f59d2725c0e2e7a8b56b93e7d73ddd45/gTTS-2.1.1-py2-none-any.wh
l
Processing ./cache/pip/wheels/dd/11/61/33f7e51bf545e910552b2255ead2a7cd8ef54064b46dceb34/gTTS_token-1.1.3-cp27-none-any.whl
Requirement already satisfied: click in /usr/local/lib/python2.7/dist-packages (from gtts) (7.0)
Requirement already satisfied: requests in /usr/local/lib/python2.7/dist-packages (from gtts) (2.22.0)
Requirement already satisfied: six in /usr/local/lib/python2.7/dist-packages (from gtts) (1.13.0)
Requirement already satisfied: BeautifulSoup4 in /usr/local/lib/python2.7/dist-packages (from gtts) (4.8.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python2.7/dist-packages (from requests->gtts) (1.24.3)
Requirement already satisfied: certifi<2017.4.17 in /usr/local/lib/python2.7/dist-packages (from requests->gtts) (2019.9.11)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python2.7/dist-packages (from requests->gtts) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python2.7/dist-packages (from requests->gtts) (2.8)
Requirement already satisfied: soupsieve>=1.2 in /usr/local/lib/python2.7/dist-packages (from BeautifulSoup4->gtts) (1.9.4)
Requirement already satisfied: backports.functools-lru-cache; python_version < "3" in /usr/local/lib/python2.7/dist-packages (from soupsieve>=1.2->B
eautifulSoup4->gtts) (1.6.1)
Installing collected packages: gtts-token, gtts
Successfully installed gtts-2.1.1 gtts-token-1.1.3

```

Рис. 3.19. Установка библиотек.

Заливаем скрипты файлов (см. рис. 3.20)

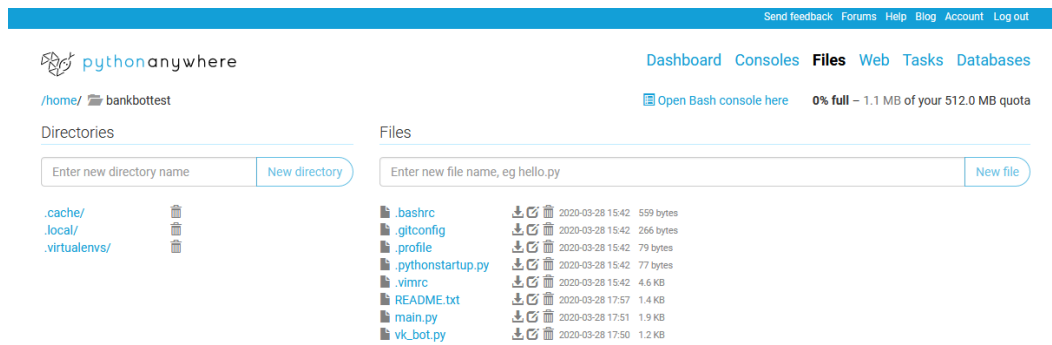


Рис. 3.20. Заливка скриптов.

Наш бот готов для работы.

3.4. Функционирование чат-бота

Разработанный чат-бот относится к типу «Разговорный на основе языковой модели и правил» [5].

На рисунке 3.21 представлены скрины работы чат-бота. Код программы приведен в приложениях 4 и 5.

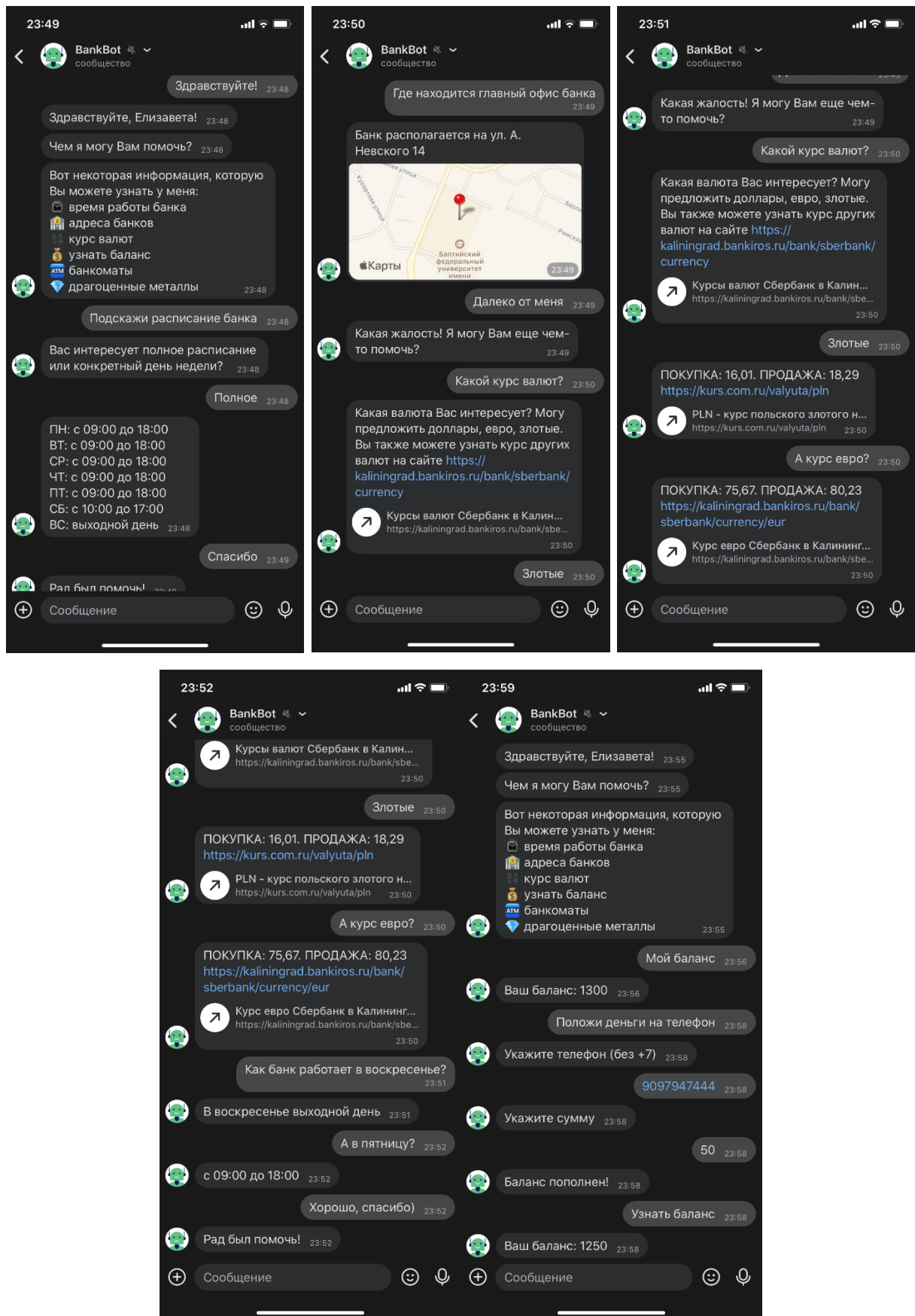


Рис. 3.21. Скрины работы чат-бота.

Заключение

Мы видим, что из-за роста числа российских интернет-пользователей, доступности смартфонов и дальнейшего развития мобильного интернета формируются новые привычки и модели поведения людей [10]. Искусственный интеллект уже становится частью нашей повседневной жизни.

Чат-боты неизбежно будут играть большую роль в функционировании банков в ближайшем будущем, но это будут не те чат-боты, что мы видим сегодня. Как и люди, боты лучше всего работают тогда, когда у них есть доступ к множеству различных инструментов и типов информации. Чем их больше, тем более впечатляющим получается конечный результат [8].

Абсолютное большинство банков считают использование чат-ботов перспективным направлением. По данным опросов уже половина банков готова перенести платежные операции и информационные сервисы в мессенджеры, каждый третий банк готов доверить чат-ботам функции блокировки платежных карт, а каждый пятый — подтверждение операций [10].

Технология чат-ботов позволяет оптимизировать бизнес-процессы и найти разумный компромисс в решении разноплановых задач, что значительно улучшит дистанционное банковское обслуживание.

Но на данный момент остаются еще некоторые трудности с полноценным распознаванием текста и обработкой произвольных запросов собеседника, но чат-боты на искусственном интеллекте развиваются стремительными темпами [12].

Забирать на себя рутину, позволяя человеку развиваться, а не стоять на месте — в этом и есть будущее ботов. Использование чат-бота в банке поможет автоматизировать работу, сократить рекламный бюджет, получить лояльных клиентов, повысить престиж, увеличить продажи. Таким образом, использование ботов может ярко выделить банк на фоне конкурентов.

Денежные манипуляции заслуживают особого внимания. Для оплаты или денежных переводов с помощью чат-ботов требуется особый подход к обеспечению безопасности передачи данных, и это — тоже будущее чат-ботов.

Впереди еще много задач, в их числе — сложности с идентификацией пользователя и безопасностью обмена информацией. Особенно острым становится вопрос на фоне новой функциональности мессенджеров, связанной с дистанционными платежами и переводом денежных средств. Встроенная идентификация пользователя поможет избежать мошенничества и повысит интерес бизнеса к чат-ботам [29].

По мере того, как боты начнут совершенствоваться и использовать все больше элементов искусственного интеллекта, произойдет значительная эволюция их возможностей [27]. Внедрение искусственного интеллекта напрямую в существующие каналы связи — это быстрый способ моментально сделать чат-боты более продвинутыми [8].

Создавая чат-боты, современные разработчики должны стремиться не просто обеспечить функционал по запросу заказчика, но и приблизить общение с ботом к опыту разговора с реальным человеком. Поэтому сегодня в практике многих компаний все чаще возникает термин «виртуальные ассистенты», тогда как «чат-бот» понемногу уступает место — именно вытеснение «бот»-составляющей из этих решений является одной из актуальных задач направления [12].

Объем российского рынка чат-ботов за последний год увеличился вдвое и достиг одного миллиарда рублей. Пока экономический смысл использования чат-ботов ограничивается снижением трудозатрат человеческого персонала, но с развитием технологий умные программы могут получить более широкие полномочия [13].

В ходе выполнения работы были выполнены все поставленные задачи:

- 1) проанализированы возможности применения чат-ботов в банковской сфере и подобраны технологии реализации;
- 2) подготовлены данные и разработаны сценарии диалогов для чат-бота;
- 3) чат-бот настроен и подключен к социальной сети «ВКонтакте».

Разработанный чат-бот относится к типу «Разговорный на основе языковой модели и правил». Созданный чат-бот будет полезен клиентам банка, которые зарегистрированы в социальной сети «ВКонтакте».

Чат-бот предоставляет следующие возможности:

- получение справочной информации;
- предоставление контактных данных;
- ответы на вопросы пользователя.

Бот может вывести данные по конкретным валютам, которые чаще всего используют жители Калининградской области.

Каждая существующая тема, по которой происходит взаимодействие с чат-ботом, может настраиваться, изменяться и расширяться.

В ходе работы над магистерской диссертацией автором были опубликованы две статьи [2, 3].

Библиографический список

1. Аудитория социальных сетей в России 2019. [Электронный ресурс]. URL: <https://clck.ru/K2ogh> (дата обращения 05.12.2019).
2. Белова Е.Е., Толстель О.В. Использование библиотек языка программирования Python для анализа оттока клиентов банка // Вестник БФУ им. И. Канта. Сер. физ.-мат. и техн. науки. — 2019. — № 4. — С. 21-30.
3. Белова Е.Е., Толстель О.В. Создание прототипа чат-бота // Материалы 5-ой Всероссийской Поспеловской конференции с международным участием «Гибридные и синергетические интеллектуальные системы ГИСИС' 2020».
4. Богдан В. Пять невероятных способов, с помощью которых чат-боты могут улучшить качество обслуживания клиентов банков. — 2019. [Электронный ресурс]. URL: <https://clck.ru/GiQzj> (дата обращения 16.06.2019).
5. Большев М. Siri, Алиса и финансовый чат-бот. Как робот-собеседник определит эмоциональное состояние вашего клиента? [Электронный ресурс]. URL: <https://clck.ru/LbEWU> (дата обращения: 06.12.2019).
6. В чем главные преимущества чат-ботов? — 2017. [Электронный ресурс]. URL: <https://clck.ru/GUvSQ> (дата обращения 08.06.2019).
7. Газета.ru «Одно другому не противоречит»: что происходит с Telegram в России. [Электронный ресурс]. URL: https://www.gazeta.ru/tech/2019/10/24/12774110/everything_is_permitted.shtml (дата обращения: 06.12.2019).
8. Джаваид О., Тырина Е. Новое поколение чат-ботов: умное, функциональное, интегрированное. — 2018. [Электронный ресурс]. URL: <https://clck.ru/GU5MM> (дата обращения 07.06.2019).
9. Зачем бизнесу нужны чат-боты и как они работают. [Электронный ресурс]. URL: <https://clck.ru/NbSud> (дата обращения 16.05.2020).
10. Искусственный интеллект в банках: 2019. [Электронный ресурс]. URL: <https://clck.ru/GiR4e> (дата обращения 13.06.2019).

11. Исследование топ-50 чат-бот платформ и виртуальных ассистентов 2019 года // Блог компании МФТИ. [Электронный ресурс]. URL: <https://clck.ru/LckAP> (дата обращения 15.12.2019).

12. Кондратьев А. Бизнес ждет чат-ботов с человеческим лицом. [Электронный ресурс]. URL: <https://clck.ru/GU4za> (дата обращения 07.06.2019).

13. Кривошапко Ю. Бот в помощь // Российская газета – Федеральный выпуск № 202(7960). [Электронный ресурс]. URL: <https://clck.ru/J2zri> (дата обращения 12.12.2019).

14. Мамчиц Р. Чат-бот успешно отвечает на 83% поступающих вопросов. [Электронный ресурс]. URL: <https://clck.ru/NbRTe> (дата обращения 16.05.2020).

15. Матвеева Н.Ю., Золотарюк А.В. Технологии создания и применения чат-ботов // Научные записки молодых исследователей. — М.: ФГБУ «Финансовый университет при Правительстве РФ». — 2018. — № 1. — С. 28-30.

16. Нашествие ботов: как банки используют мессенджеры. [Электронный ресурс]. URL: <https://clck.ru/NbTAb> (дата обращения 17.05.2020).

17. Пилька М. Чат-боты в 2019 году. [Электронный ресурс]. URL: <https://clck.ru/LcgFk> (дата обращения 13.12.2019).

18. Пономарев А. Кейс: мы внедрили чат-бота — теперь он экономит нам около 250 рабочих часов в месяц. [Электронный ресурс]. URL: <https://clck.ru/LbDqz> (дата обращения: 06.12.2019).

19. Самойлова Ю. Виртуальные помощники // Журнал FoodService. — 2018. — № 11. [Электронный ресурс]. URL: <https://clck.ru/GU5fy> (дата обращения 12.06.2019).

20. Самообучающийся чат-бот Промсвязьбанка поможет МСБ с открытием счетов и оформлением документов. [Электронный ресурс]. URL: <https://clck.ru/GU53V> (дата обращения 07.06.2019).

21. Скобелев В. Без лишних разговоров. Чат-боты набирают популярность среди ретейлеров и отельеров. — 2019. [Электронный ресурс]. URL: <https://clck.ru/GU55c> (дата обращения 07.06.2019).

22. Старкова Т. Разновидности чат-ботов для бизнеса. [Электронный ресурс]. URL: <https://clck.ru/NbSfy> (дата обращения 17.05.2020).

23. Степанова Н. Преимущества (плюсы чат-ботов). — 2018. [Электронный ресурс]. URL: <https://clck.ru/GUvVe> (дата обращения 08.06.2019).

24. Фомичев Г. Банковские чат-боты: российский опыт и мировые тенденции. [Электронный ресурс]. URL: <https://clck.ru/LbCF5> (дата обращения: 05.12.2019).

25. Чат-боты: 2017. [Электронный ресурс]. URL: <https://clck.ru/GiRXN> (дата обращения 18.06.2019).

26. Чат-боты в банкинге: ТОП-7 примеров успешного использования: 2018. [Электронный ресурс]. URL: <https://clck.ru/GiQxD> (дата обращения 19.06.2019).

27. Чат-боты для бизнеса // Бизнес & информационные технологии. — 2017. — Вып. № 04 (67). [Электронный ресурс]. URL: <https://clck.ru/GU5yS> (дата обращения 07.06.2019).

28. Чат-боты и искусственный интеллект в обучении: конец живого общения // Журнал EduTech. — 2018. — № 6 (18). [Электронный ресурс]. URL: <https://clck.ru/GU5BA> (дата обращения 07.06.2019).

29. Шабаетова А. Чат-боты: что бизнес думает о виртуальных помощниках. [Электронный ресурс]. URL: <https://clck.ru/GU4t2> (дата обращения 07.06.2019).

30. Шошин П. Зачем банкам боты. [Электронный ресурс]. URL: <https://clck.ru/GiRce> (дата обращения 22.06.2019).

Приложение 1. Сценарии диалогов для прототипа чат-бота

1. Assistant-trn.json

```
[
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Адрес главного офиса банка"
    },
    {
      "speaker": 2,
      "text": "ул. А. Невского 14",
      "act": "address_msg"
    },
    {
      "speaker": 1,
      "text": "Благодарю"
    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет, бот"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",

```

```

    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Время работы банка"
  },
  {
    "speaker": 2,
    "text": "Банк работает до 19:00 в буднии дни",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Хорошо. Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Доброе утро"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Банк работает в воскресенье?"
  },
  {
    "speaker": 2,
    "text": "Банк работает до 19:00 в буднии дни",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Жаль"
  },

```

```

{
  "speaker": 2,
  "text": "Какая жалость! В следующий раз, может быть. Хорошего дня!",
  "act": "bad_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Приветик"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Где главный офис?"
  },
  {
    "speaker": 2,
    "text": "ул. А. Невского 14",
    "act": "address_msg"
  },
  {
    "speaker": 1,
    "text": "Это находится далеко от меня"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! В следующий раз, может быть. Хорошего дня!",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Добрый вечер"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  }
]

```

```

},
{
  "speaker": 1,
  "text": "Где главный офис банка?"
},
{
  "speaker": 2,
  "text": "ул. А. Невского 14",
  "act": "address_msg"
},
{
  "speaker": 1,
  "text": "Отлично"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Банк работает до 19:00 в буднии дни",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Сегодня не успею"
  },
  {

```



```

    "speaker": 2,
    "text": "Какая жалость! В следующий раз, может быть. Хорошего дня!",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Как дела, бот?"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "До сколько работает банк?"
  },
  {
    "speaker": 2,
    "text": "Банк работает до 19:00 в буднии дни",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Отлично, спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Бот, привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },

```

```

{
  "speaker": 1,
  "text": "Адрес"
},
{
  "speaker": 2,
  "text": "ул. А. Невского 14",
  "act": "address_msg"
},
{
  "speaker": 1,
  "text": "Далеко"
},
{
  "speaker": 2,
  "text": "Какая жалость! В следующий раз, может быть. Хорошего дня!",
  "act": "bad_bye"
}
]
]

```

2. Assistant-val.json

```

[
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Адрес главного офиса банка"
    },
    {
      "speaker": 2,
      "text": "ул. А. Невского 14",
      "act": "address_msg"
    },
    {

```

```

    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
]
]

```

3. Assistant-tst.json

```

[
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Время работы банка"
    },
    {
      "speaker": 2,
      "text": "Банк работает до 19:00 в буднии дни",
      "act": "time_bank"
    },
    {
      "speaker": 1,
      "text": "Я не успею"
    },
    {
      "speaker": 2,
      "text": "Какая жалость! В следующий раз, может быть. Хорошего дня!",
      "act": "bad_bye"
    }
  ]
]

```

]

Приложение 2. Расширенная база сценариев диалогов

1. Assistant-trn.json

```
[
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Адрес главного офиса банка"
    },
    {
      "speaker": 2,
      "text": "Банк располагается на ул. А. Невского 14.",
      "act": "addr_bank"
    },
    {
      "speaker": 1,
      "text": "Благодарю"
    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет, бот"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    }
  ]
]
```

```

    },
    {
      "speaker": 1,
      "text": "Время работы банка"
    },
    {
      "speaker": 2,
      "text": "Вас интересует полное расписание или конкретный день?",
      "act": "time_bank"
    },
    {
      "speaker": 1,
      "text": "Полное"
    },
    {
      "speaker": 2,
      "text": "ПН: с 09:00 до 18:00 ВТ: с 09:00 до 18:00 СР: с 09:00 до 18:00 ЧТ: с 09:00 до 18:00 ПТ: с 09:00 до 18:00 СБ: с 10:00 до 17:00 ВС: выходной день",
      "act": "all_time_bank"
    },
    {
      {
        "speaker": 1,
        "text": "Спасибо за помощь"
      },
      {
        "speaker": 2,
        "text": "Рад был помочь! До свидания!",
        "act": "good_bye"
      }
    },
  ],
  [
    {
      "speaker": 1,
      "text": "Здравствуйте"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Банк работает в воскресенье?"
    },
  ],

```

```

{
  "speaker": 2,
  "text": "ВС: выходной день",
  "act": "sunday_time"
},
{
  "speaker": 1,
  "text": "Жаль"
},
{
  "speaker": 2,
  "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
  "act": "bad_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Где главный офис банка?"
  },
  {
    "speaker": 2,
    "text": "Банк располагается на ул. А. Невского 14.",
    "act": "addr_bank"
  },
  {
    "speaker": 1,
    "text": "Это находится далеко от меня"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
]

```

```
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Где главный офис банка?"
  },
  {
    "speaker": 2,
    "text": "Банк располагается на ул. А. Невского 14.",
    "act": "addr_bank"
  },
  {
    "speaker": 1,
    "text": "Отлично"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
```



```

    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Вторник"
  },
  {
    "speaker": 2,
    "text": "ВТ: с 09:00 до 18:00",
    "act": "tuesday_time"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет, бот"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "До сколько работает банк?"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,

```

```

    "text": "Суббота"
  },
  {
    "speaker": 2,
    "text": "СБ: с 10:00 до 17:00",
    "act": "saturday_time"
  },
  {
    "speaker": 1,
    "text": "Не успею"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банка"
  },
  {
    "speaker": 2,
    "text": "Банк располагается на ул. А. Невского 14.",
    "act": "addr_bank"
  },
  {
    "speaker": 1,
    "text": "Далеко"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
]

```

```

    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Адрес банкомата"
    },
    {
      "speaker": 2,
      "text": "Банкомат находится на ул. 1812 года.",
      "act": "addr_bankomat"
    },
    {
      "speaker": 1,
      "text": "Далеко"
    },
    {
      "speaker": 2,
      "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
      "act": "bad_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    }
  ]

```

```

    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
      "act": "ex_rates"
    },
    {
      "speaker": 1,
      "text": "Злоты"
    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 15,86. ПРОДАЖА: 16,80",
      "act": "pln_rate"
    },
    {
      "speaker": 1,
      "text": "Спасибо"
    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",

```

```

    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Евро"
  },
  {
    "speaker": 2,
    "text": "ПОКУПКА: 67,61. ПРОДАЖА: 71,10",
    "act": "euro_rate"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро, злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Доллар"
  }
]

```

```

    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 60,53. ПРОДАЖА: 63,64",
      "act": "dollar_rate"
    },
    {
      "speaker": 1,
      "text": "Спасибо"
    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
      "act": "ex_rates"
    },
    {
      "speaker": 1,
      "text": "Злоты"
    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 15,86. ПРОДАЖА: 16,80",
      "act": "pln_rate"
    }
  ]

```

```

    },
    {
      "speaker": 1,
      "text": "Не выгодно"
    },
    {
      "speaker": 2,
      "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
      "act": "bad_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
      "act": "ex_rates"
    },
    {
      "speaker": 1,
      "text": "Евро"
    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 67,61. ПРОДАЖА: 71,10",
      "act": "euro_rate"
    },
    {
      "speaker": 1,
      "text": "Не выгодно"
    },
  ],

```

```

{
  "speaker": 2,
  "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
  "act": "bad_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро, злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Доллар"
  },
  {
    "speaker": 2,
    "text": "ПОКУПКА: 60,53. ПРОДАЖА: 63,64",
    "act": "dollar_rate"
  },
  {
    "speaker": 1,
    "text": "Не выгодно"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
]

```



```

],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },

```

```

{
  "speaker": 2,
  "text": "Здравствуйте, чем я могу вам помочь?",
  "act": "welcome_msg"
},
{
  "speaker": 1,
  "text": "Адрес банкомата"
},
{
  "speaker": 2,
  "text": "Банкомат находится на ул. 1812 года.",
  "act": "addr_bankomat"
},
{
  "speaker": 1,
  "text": "Время работы банкомата"
},
{
  "speaker": 2,
  "text": "Банкомат работает круглосуточно.",
  "act": "time_bankomat"
},
{
  "speaker": 1,
  "text": "Отлично"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
{
  "speaker": 1,
  "text": "Привет"
},
{
  "speaker": 2,
  "text": "Здравствуйте, чем я могу вам помочь?",
  "act": "welcome_msg"
},
{

```

```

    "speaker": 1,
    "text": "Где находится банкомат?"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",

```

```

    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Понедельник"
  },
  {

```

```

{
  "speaker": 2,
  "text": "ПН: с 09:00 до 18:00",
  "act": "monday_time"
},
{
  "speaker": 1,
  "text": "Спасибо"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
{
  "speaker": 1,
  "text": "Здравствуйте"
},
{
  "speaker": 2,
  "text": "Здравствуйте, чем я могу вам помочь?",
  "act": "welcome_msg"
},
{
  "speaker": 1,
  "text": "Режим работы банка"
},
{
  "speaker": 2,
  "text": "Вас интересует полное расписание или конкретный день?",
  "act": "time_bank"
},
{
  "speaker": 1,
  "text": "Среда"
},
{
  "speaker": 2,
  "text": "СР: с 09:00 до 18:00",
  "act": "wednesday_time"
},
{

```

```

    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Четверг"
  },
  {
    "speaker": 2,
    "text": "ЧТ: с 09:00 до 18:00",
    "act": "thursday_time"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",

```

```

    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Пятница"
  },
  {
    "speaker": 2,
    "text": "ПТ: с 09:00 до 18:00",
    "act": "friday_time"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,

```

```

    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Суббота"
  },
  {
    "speaker": 2,
    "text": "СБ: с 10:00 до 17:00",
    "act": "saturday_time"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
]
]

```

2. Assistant-val.json

```

[
  [
    {
      "speaker": 1,

```



```

    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес главного офиса банка"
  },
  {
    "speaker": 2,
    "text": "Банк располагается на ул. А. Невского 14.",
    "act": "addr_bank"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет, бот"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Время работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  }
]

```

```

    },
    {
      "speaker": 1,
      "text": "Полное"
    },
    {
      "speaker": 2,
      "text": "ПН: с 09:00 до 18:00 ВТ: с 09:00 до 18:00 СР: с 09:00 до 18:00 ЧТ: с 09:00 до 18:00 ПТ: с 09:00 до 18:00 СБ: с 10:00 до 17:00 ВС: выходной день",
      "act": "all_time_bank"
    },
    {
      "speaker": 1,
      "text": "Спасибо за помощь"
    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Здравствуйте"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Банк работает в воскресенье?"
    },
    {
      "speaker": 2,
      "text": "ВС: выходной день",
      "act": "sunday_time"
    },
    {
      "speaker": 1,
      "text": "Жаль"
    },
  ],

```

```

{
  "speaker": 2,
  "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
  "act": "bad_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Где главный офис банка?"
  },
  {
    "speaker": 2,
    "text": "Банк располагается на ул. А. Невского 14.",
    "act": "addr_bank"
  },
  {
    "speaker": 1,
    "text": "Это находится далеко от меня"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  }
]

```

```

},
{
  "speaker": 1,
  "text": "Где главный офис банка?"
},
{
  "speaker": 2,
  "text": "Банк располагается на ул. А. Невского 14.",
  "act": "addr_bank"
},
{
  "speaker": 1,
  "text": "Отлично"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Вторник"
  },
  {

```

```

    "speaker": 2,
    "text": "ВТ: с 09:00 до 18:00",
    "act": "tuesday_time"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет, бот"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "До сколько работает банк?"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Суббота"
  },
  {
    "speaker": 2,
    "text": "СБ: с 10:00 до 17:00",
    "act": "saturday_time"
  },
  {
    "speaker": 1,

```

```

    "text": "Не успею"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банка"
  },
  {
    "speaker": 2,
    "text": "Банк располагается на ул. А. Невского 14.",
    "act": "addr_bank"
  },
  {
    "speaker": 1,
    "text": "Далеко"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,

```

```

    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Далеко"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,

```

```

    "text": "Злоты"
  },
  {
    "speaker": 2,
    "text": "ПОКУПКА: 15,86. ПРОДАЖА: 16,80",
    "act": "pln_rate"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро, злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Евро"
  },
  {
    "speaker": 2,
    "text": "ПОКУПКА: 67,61. ПРОДАЖА: 71,10",

```



```

    "act": "euro_rate"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Доллар"
  },
  {
    "speaker": 2,
    "text": "ПОКУПКА: 60,53. ПРОДАЖА: 63,64",
    "act": "dollar_rate"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  }
]

```

```

    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
      "act": "ex_rates"
    },
    {
      "speaker": 1,
      "text": "Злоты"
    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 15,86. ПРОДАЖА: 16,80",
      "act": "pln_rate"
    },
    {
      "speaker": 1,
      "text": "Не выгодно"
    },
    {
      "speaker": 2,
      "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
      "act": "bad_bye"
    }
  ]

```

```

    }
  ]
]

```

3. Assistant-tst.json

```

[
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Время работы банка"
    },
    {
      "speaker": 2,
      "text": "Вас интересует полное расписание или конкретный день?",
      "act": "time_bank"
    },
    {
      "speaker": 1,
      "text": "Я не успею"
    },
    {
      "speaker": 2,
      "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
      "act": "bad_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",

```

```

    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Злоты"
  },
  {
    "speaker": 2,
    "text": "ПОКУПКА: 15,86. ПРОДАЖА: 16,80",
    "act": "pln_rate"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  }
]

```

```

    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
      "act": "ex_rates"
    },
    {
      "speaker": 1,
      "text": "Евро"
    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 67,61. ПРОДАЖА: 71,10",
      "act": "euro_rate"
    },
    {
      "speaker": 1,
      "text": "Спасибо"
    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",

```

```

    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Доллар"
  },
  {
    "speaker": 2,
    "text": "ПОКУПКА: 60,53. ПРОДАЖА: 63,64",
    "act": "dollar_rate"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро, злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Злоты"
  }
]

```

```

    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 15,86. ПРОДАЖА: 16,80",
      "act": "pln_rate"
    },
    {
      "speaker": 1,
      "text": "Не выгодно"
    },
    {
      "speaker": 2,
      "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
      "act": "bad_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
      "act": "ex_rates"
    },
    {
      "speaker": 1,
      "text": "Евро"
    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 67,61. ПРОДАЖА: 71,10",
      "act": "euro_rate"
    }
  ]

```

```

    },
    {
      "speaker": 1,
      "text": "Не выгодно"
    },
    {
      "speaker": 2,
      "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
      "act": "bad_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
      "act": "ex_rates"
    },
    {
      "speaker": 1,
      "text": "Доллар"
    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 60,53. ПРОДАЖА: 63,64",
      "act": "dollar_rate"
    },
    {
      "speaker": 1,
      "text": "Не выгодно"
    },
  ],

```



```

{
  "speaker": 2,
  "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
  "act": "bad_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],

```

```
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Отлично"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
```

```

    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Где находится банкомат?"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,

```

```

    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  }
]

```

```
},
{
  "speaker": 1,
  "text": "Понедельник"
},
{
  "speaker": 2,
  "text": "ПН: с 09:00 до 18:00",
  "act": "monday_time"
},
{
  "speaker": 1,
  "text": "Спасибо"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
]
]
```

Приложение 3. Обучение прототипа чат-бота

```

from deeppavlov.dataset_readers.dstc2_reader import SimpleDSTC2DatasetReader

class AssistantDatasetReader(SimpleDSTC2DatasetReader):

    url = "http://files.deeppavlov.ai/datasets/tutor_assistant_data.tar.gz"

    @staticmethod
    def _data_fname(datatype):
        assert datatype in ('val', 'trn', 'tst'), "wrong datatype name"
        return f"assistant-{datatype}.json"

data = AssistantDatasetReader().read('assistant_data')

!dir assistant_data

file_js = open("assistant_data\\assistant-trn.json", "r")

from deeppavlov.dataset_iterators.dialog_iterator import DialogDatasetIterator
iterator = DialogDatasetIterator(data)

from pprint import pprint
for dialog in iterator.gen_batches(batch_size=1, data_type='train'):
    turns_x, turns_y = dialog

    print("User utterances:\n-----\n")
    pprint(turns_x[0], indent=4)
    print("\nSystem responses:\n-----\n")
    pprint(turns_y[0], indent=4)

    break

print("\n-----")
print(f"{len(iterator.get_instances('train')[0])} dialog(s) in train.")
print(f"{len(iterator.get_instances('valid')[0])} dialog(s) in valid.")
print(f"{len(iterator.get_instances('test')[0])} dialog(s) in test.")

file2_js = open("assistant_data\\assistant-templates.txt", "r")

file_js = open("assistant_data\\assistant-trn.json", "r")

```

```

from deeppavlov import configs
from deeppavlov.core.common.file import read_json
gobot_config = read_json(configs.go_bot.gobot_dstc2_minimal)

from deeppavlov.download import download_resource
download_resource(url="http://files.deeppavlov.ai/embeddings/glove.6B.100d.txt",
                  dest_paths=['assistant_bot/'])

gobot_config['chainer']['pipe'][-1]['embedder'] = {
    "class_name": "glove",
    "load_path": "assistant_bot/glove.6B.100d.txt"
}

gobot_config['chainer']['pipe'][-1]['template_path'] = 'assistant_data\\assistant-templates.txt'

gobot_config['dataset_reader']['class_name'] = '__main__:AssistantDatasetReader'
gobot_config['metadata']['variables']['DATA_PATH'] = 'assistant_data'
gobot_config['metadata']['variables']['MODEL_PATH'] = 'assistant_bot'

from deeppavlov import train_model

gobot_config['train']['batch_size'] = 4 # set batch size
gobot_config['train']['max_batches'] = 30 # maximum number of training batches
gobot_config['train']['val_every_n_batches'] = 30 # evaluate on full 'valid' split every 30 epochs
gobot_config['train']['log_every_n_batches'] = 5 # evaluate on full 'train' split every 5 batches
train_model(gobot_config);

from deeppavlov import build_model
bot = build_model(gobot_config)

```

Приложение 4. Сценарии диалогов для чат-бота

```
[
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Адрес главного офиса банка"
    },
    {
      "speaker": 2,
      "text": "Банк располагается на ул. А. Невского 14.",
      "act": "addr_bank"
    },
    {
      "speaker": 1,
      "text": "Благодарю"
    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет, бот"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
  ],
]
```



```

{
  "speaker": 1,
  "text": "Время работы банка"
},
{
  "speaker": 2,
  "text": "Вас интересует полное расписание или конкретный день?",
  "act": "time_bank"
},
{
  "speaker": 1,
  "text": "Полное"
},
{
  "speaker": 2,
  "text": "ПН: с 09:00 до 18:00 ВТ: с 09:00 до 18:00 СР: с 09:00 до 18:00 ЧТ: с
09:00 до 18:00 ПТ: с 09:00 до 18:00 СБ: с 10:00 до 17:00 ВС: выходной день",
  "act": "all_time_bank"
},
{
  "speaker": 1,
  "text": "Спасибо за помощь"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
{
  "speaker": 1,
  "text": "Здравствуйте"
},
{
  "speaker": 2,
  "text": "Здравствуйте, чем я могу вам помочь?",
  "act": "welcome_msg"
},
{
  "speaker": 1,
  "text": "Банк работает в воскресенье?"
},
{

```

```

    "speaker": 2,
    "text": "ВС: выходной день",
    "act": "sunday_time"
  },
  {
    "speaker": 1,
    "text": "Жаль"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Где главный офис банка?"
  },
  {
    "speaker": 2,
    "text": "Банк располагается на ул. А. Невского 14.",
    "act": "addr_bank"
  },
  {
    "speaker": 1,
    "text": "Это находится далеко от меня"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[

```

```

{
  "speaker": 1,
  "text": "Привет"
},
{
  "speaker": 2,
  "text": "Здравствуйте, чем я могу вам помочь?",
  "act": "welcome_msg"
},
{
  "speaker": 1,
  "text": "Где главный офис банка?"
},
{
  "speaker": 2,
  "text": "Банк располагается на ул. А. Невского 14.",
  "act": "addr_bank"
},
{
  "speaker": 1,
  "text": "Отлично"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,

```

```

    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Вторник"
  },
  {
    "speaker": 2,
    "text": "ВТ: с 09:00 до 18:00",
    "act": "tuesday_time"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет, бот"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "До сколько работает банк?"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Суббота"
  }
]

```

```

},
{
  "speaker": 2,
  "text": "СБ: с 10:00 до 17:00",
  "act": "saturday_time"
},
{
  "speaker": 1,
  "text": "Не успею"
},
{
  "speaker": 2,
  "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
  "act": "bad_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банка"
  },
  {
    "speaker": 2,
    "text": "Банк располагается на ул. А. Невского 14.",
    "act": "addr_bank"
  },
  {
    "speaker": 1,
    "text": "Далеко"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
]

```

```

],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Далеко"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },

```

```

{
  "speaker": 2,
  "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
  "act": "ex_rates"
},
{
  "speaker": 1,
  "text": "Злоты"
},
{
  "speaker": 2,
  "text": "ПОКУПКА: 15,86. ПРОДАЖА: 16,80",
  "act": "pln_rate"
},
{
  "speaker": 1,
  "text": "Спасибо"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро,
злоты.",
    "act": "ex_rates"
  }
]

```

```

},
{
  "speaker": 1,
  "text": "Евро"
},
{
  "speaker": 2,
  "text": "ПОКУПКА: 67,61. ПРОДАЖА: 71,10",
  "act": "euro_rate"
},
{
  "speaker": 1,
  "text": "Спасибо"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро, злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Доллар"
  },

```



```

{
  "speaker": 2,
  "text": "ПОКУПКА: 60,53. ПРОДАЖА: 63,64",
  "act": "dollar_rate"
},
{
  "speaker": 1,
  "text": "Спасибо"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
{
  "speaker": 1,
  "text": "Привет"
},
{
  "speaker": 2,
  "text": "Здравствуйте, чем я могу вам помочь?",
  "act": "welcome_msg"
},
{
  "speaker": 1,
  "text": "Курс валют"
},
{
  "speaker": 2,
  "text": "Какая валюта Вас интересует? Могу предложить доллар, евро, злоты.",
  "act": "ex_rates"
},
{
  "speaker": 1,
  "text": "Злоты"
},
{
  "speaker": 2,
  "text": "ПОКУПКА: 15,86. ПРОДАЖА: 16,80",
  "act": "pln_rate"
},

```

```

    {
      "speaker": 1,
      "text": "Не выгодно"
    },
    {
      "speaker": 2,
      "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
      "act": "bad_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Привет"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Курс валют"
    },
    {
      "speaker": 2,
      "text": "Какая валюта Вас интересует? Могу предложить доллар, евро, злоты.",
      "act": "ex_rates"
    },
    {
      "speaker": 1,
      "text": "Евро"
    },
    {
      "speaker": 2,
      "text": "ПОКУПКА: 67,61. ПРОДАЖА: 71,10",
      "act": "euro_rate"
    },
    {
      "speaker": 1,
      "text": "Не выгодно"
    },
    {

```

```

    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Курс валют"
  },
  {
    "speaker": 2,
    "text": "Какая валюта Вас интересует? Могу предложить доллар, евро, злоты.",
    "act": "ex_rates"
  },
  {
    "speaker": 1,
    "text": "Доллар"
  },
  {
    "speaker": 2,
    "text": "ПОКУПКА: 60,53. ПРОДАЖА: 63,64",
    "act": "dollar_rate"
  },
  {
    "speaker": 1,
    "text": "Не выгодно"
  },
  {
    "speaker": 2,
    "text": "Какая жалость! Я могу Вам еще чем-то помочь?",
    "act": "bad_bye"
  }
],

```

```
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
```

```

    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Отлично"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Привет"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,

```

```

    "text": "Где находится банкомат?"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  },
  {
    "speaker": 1,
    "text": "Время работы банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат работает круглосуточно.",
    "act": "time_bankomat"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Адрес банкомата"
  },
  {
    "speaker": 2,
    "text": "Банкомат находится на ул. 1812 года.",
    "act": "addr_bankomat"
  }
]

```

```

},
{
  "speaker": 1,
  "text": "Время работы банкомата"
},
{
  "speaker": 2,
  "text": "Банкомат работает круглосуточно.",
  "act": "time_bankomat"
},
{
  "speaker": 1,
  "text": "Спасибо"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Понедельник"
  },
  {

```

```

    "speaker": 2,
    "text": "ПН: с 09:00 до 18:00",
    "act": "monday_time"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Среда"
  },
  {
    "speaker": 2,
    "text": "СР: с 09:00 до 18:00",
    "act": "wednesday_time"
  },
  {
    "speaker": 1,

```



```

    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
],
[
  {
    "speaker": 1,
    "text": "Здравствуйте"
  },
  {
    "speaker": 2,
    "text": "Здравствуйте, чем я могу вам помочь?",
    "act": "welcome_msg"
  },
  {
    "speaker": 1,
    "text": "Режим работы банка"
  },
  {
    "speaker": 2,
    "text": "Вас интересует полное расписание или конкретный день?",
    "act": "time_bank"
  },
  {
    "speaker": 1,
    "text": "Четверг"
  },
  {
    "speaker": 2,
    "text": "ЧТ: с 09:00 до 18:00",
    "act": "thursday_time"
  },
  {
    "speaker": 1,
    "text": "Спасибо"
  },
  {
    "speaker": 2,
    "text": "Рад был помочь! До свидания!",
    "act": "good_bye"
  }
]

```

```

    }
  ],
  [
    {
      "speaker": 1,
      "text": "Здравствуйте"
    },
    {
      "speaker": 2,
      "text": "Здравствуйте, чем я могу вам помочь?",
      "act": "welcome_msg"
    },
    {
      "speaker": 1,
      "text": "Режим работы банка"
    },
    {
      "speaker": 2,
      "text": "Вас интересует полное расписание или конкретный день?",
      "act": "time_bank"
    },
    {
      "speaker": 1,
      "text": "Пятница"
    },
    {
      "speaker": 2,
      "text": "ПТ: с 09:00 до 18:00",
      "act": "friday_time"
    },
    {
      "speaker": 1,
      "text": "Спасибо"
    },
    {
      "speaker": 2,
      "text": "Рад был помочь! До свидания!",
      "act": "good_bye"
    }
  ],
  [
    {
      "speaker": 1,
      "text": "Здравствуйте"
    }
  ]

```

```

},
{
  "speaker": 2,
  "text": "Здравствуйте, чем я могу вам помочь?",
  "act": "welcome_msg"
},
{
  "speaker": 1,
  "text": "Режим работы банка"
},
{
  "speaker": 2,
  "text": "Вас интересует полное расписание или конкретный день?",
  "act": "time_bank"
},
{
  "speaker": 1,
  "text": "Суббота"
},
{
  "speaker": 2,
  "text": "СБ: с 10:00 до 17:00",
  "act": "saturday_time"
},
{
  "speaker": 1,
  "text": "Спасибо"
},
{
  "speaker": 2,
  "text": "Рад был помочь! До свидания!",
  "act": "good_bye"
}
]
]

```

Приложение 5. Код чат-бота

```

{
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 1,
      "metadata": {},
      "outputs": [],
      "source": [
        "from deeppavlov.dataset_readers.dstc2_reader import SimpleDSTC2Da-
datasetReader\n",
        "\n",
        "class AssistantDatasetReader(SimpleDSTC2DatasetReader):\n",
        "    \n",
        "    url = \"http://files.deeppavlov.ai/datasets/tutor_assistant_data.tar.gz\"\n",
        "    \n",
        "    @staticmethod\n",
        "    def _data_fname(datatype):\n",
        "        assert datatype in ('val', 'trn', 'tst'), \"wrong datatype name\"\n",
        "        return f\"assistant-{datatype}.json\"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {},
      "outputs": [
        {
          "name": "stderr",
          "output_type": "stream",
          "text": [
            "2020-03-13 14:17:31.169 INFO in 'deeppavlov.dataset_readers.dstc2_read-
er'['dstc2_reader'] at line 290: [loading dialogs from assistant_data\\assistant-
trn.json]\n",
            "2020-03-13 14:17:31.213 INFO in 'deeppavlov.dataset_readers.dstc2_read-
er'['dstc2_reader'] at line 290: [loading dialogs from assistant_data\\assistant-
val.json]\n",
            "2020-03-13 14:17:31.229 INFO in 'deeppavlov.dataset_readers.dstc2_read-
er'['dstc2_reader'] at line 290: [loading dialogs from assistant_data\\assistant-
tst.json]\n",

```

```

"2020-03-13 14:17:31.291 INFO in 'deeppavlov.dataset_readers.dstc2_reader['dstc2_reader'] at line 282: There are 24 samples in train split.\n",
"2020-03-13 14:17:31.295 INFO in 'deeppavlov.dataset_readers.dstc2_reader['dstc2_reader'] at line 283: There are 3 samples in valid split.\n",
"2020-03-13 14:17:31.298 INFO in 'deeppavlov.dataset_readers.dstc2_reader['dstc2_reader'] at line 284: There are 3 samples in test split.\n"
]
}
],
"source": [
"data = AssistantDatasetReader().read('assistant_data')"
],
{
"cell_type": "code",
"execution_count": 3,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Том в устройстве С не имеет метки.\n",
"Серийный номер тома: 4EDB-D969\n",
"\n",
"Содержимое папки C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_data\n",
"\n",
"07.03.2020 13:17 <DIR> .\n",
"07.03.2020 13:17 <DIR> ..\n",
"28.12.2019 14:06 0 .done\n",
"28.12.2019 17:02 361 assistant-templates.txt\n",
"28.12.2019 17:05 5я233 assistant-trn.json\n",
"28.12.2019 16:32 679 assistant-tst.json\n",
"28.12.2019 16:32 611 assistant-val.json\n",
" 5 файлов по 6884 байт \n",
" 2 папки 144 727 556 096 байт свободно\n"
]
}
],
"source": [
"!dir assistant_data"
],
},

```

```

{
  "cell_type": "code",
  "execution_count": 4,
  "metadata": {},
  "outputs": [],
  "source": [
    "file_js = open(\"assistant_data\\\\\\\\assistant-trn.json\", \"r\")"
  ]
},
{
  "cell_type": "code",
  "execution_count": 5,
  "metadata": {},
  "outputs": [],
  "source": [
    "from deeppavlov.dataset_iterators.dialog_iterator import DialogDatasetIterator\n",
    "\n",
    "iterator = DialogDatasetIterator(data)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 6,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "User utterances:\n",
        "-----\n",
        "\n",
        "[ {'prev_resp_act': None, 'text': 'Здравствуйте!'},\n",
        "  {'prev_resp_act': 'welcome_msg',\n",
        "    'text': 'Режим работы банка'},\n",
        "  {'prev_resp_act': 'time_bank', 'text': 'Сегодня не успею'}]\n",
        "\n",
        "System responses:\n",
        "-----\n",
        "\n",
        "[ {'act': 'welcome_msg',\n",
        "   'text': 'Здравствуйте!'\n",
        "   'text': 'Чем я могу Вам помочь?'},\n",
        "  {'act': 'time_bank',

```

```

"      'text': 'Время работы банка'}],\n",
"    { 'act': 'bad_bye',\n",
"      'text': 'Какая жалость! Я могу Вам еще чем-то помочь?'\n",
"        'Какая жалость, в следующий раз, может быть. Хорошего дня!'}],\n",
"\n",
"-----\n",
"49 dialog(s) in train.\n",
"10 dialog(s) in valid.\n",
"10 dialog(s) in test.\n"
]
}
],
"source": [
"from pprint import pprint\n",
"\n",
"for dialog in iterator.gen_batches(batch_size=1, data_type='train'):\n",
"    turns_x, turns_y = dialog\n",
"    \n",
"    print("\nUser utterances:\n-----\n")\n",
"    pprint(turns_x[0], indent=4)\n",
"    print("\nSystem responses:\n-----\n")\n",
"    pprint(turns_y[0], indent=4)\n",
"    \n",
"    break\n",
"\n",
"print("\n-----\n") \n",
"print(f"{len(iterator.get_instances('train')[0])} dialog(s) in train.}")\n",
"print(f"{len(iterator.get_instances('valid')[0])} dialog(s) in valid.}")\n",
"print(f"{len(iterator.get_instances('test')[0])} dialog(s) in test.}")
]
},
{
"cell_type": "code",
"execution_count": 7,
"metadata": {},
"outputs": [],
"source": [
"file2_js = open(\"assistant_data\\\\assistant-templates.txt\", \"r\")"
]
},
{
"cell_type": "code",
"execution_count": 8,
"metadata": {},

```

```

"outputs": [],
"source": [
  "file_js = open(\"assistant_data\\\\assistant-trn.json\", \"r\")"
],
{
  "cell_type": "code",
  "execution_count": 9,
  "metadata": {},
  "outputs": [],
  "source": [
    "from deeppavlov import configs\n",
    "from deeppavlov.core.common.file import read_json\n",
    "\n",
    "gobot_config = read_json(configs.go_bot.gobot_dstc2_minimal)"
  ],
},
{
  "cell_type": "code",
  "execution_count": 10,
  "metadata": {},
  "outputs": [
    {
      "name": "stderr",
      "output_type": "stream",
      "text": [
        "2020-03-13 14:18:28.876 INFO in 'deeppavlov.download'['download'] at line
117: Skipped http://files.deeppavlov.ai/embeddings/glove.6B.100d.txt download be-
cause of matching hashes\n"
      ]
    }
  ],
  "source": [
    "from deeppavlov.download import download_resource\n",
    "\n",
    "download_resource(url=\"http://files.deeppavlov.ai/embed-
dings/glove.6B.100d.txt\", \n",
    "                  dest_paths=['assistant_bot/'])"
  ],
},
{
  "cell_type": "code",
  "execution_count": 11,
  "metadata": {},

```



```

"outputs": [],
"source": [
  "gobot_config['chainer']['pipe'][-1]['embedder'] = {\n",
  "  \"class_name\": \"glove\",\n",
  "  \"load_path\": \"assistant_bot/glove.6B.100d.txt\"\n",
  "}"
],
},
{
  "cell_type": "code",
  "execution_count": 12,
  "metadata": {},
  "outputs": [],
  "source": [
    "gobot_config['chainer']['pipe'][-1]['template_path'] = 'assistant_data\\\\\\\\assistant-templates.txt'"
  ],
},
{
  "cell_type": "code",
  "execution_count": 13,
  "metadata": {},
  "outputs": [],
  "source": [
    "gobot_config['dataset_reader']['class_name'] = '__main__:AssistantDatasetReader'\n",
    "gobot_config['metadata']['variables']['DATA_PATH'] = 'assistant_data'\n",
    "\n",
    "gobot_config['metadata']['variables']['MODEL_PATH'] = 'assistant_bot'"
  ],
},
{
  "cell_type": "code",
  "execution_count": 14,
  "metadata": {},
  "outputs": [
    {
      "name": "stderr",
      "output_type": "stream",
      "text": [
        "2020-03-13 14:21:08.239 INFO in 'deeppavlov.dataset_readers.dstc2_reader['dstc2_reader'] at line 290: [loading dialogs from C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_data\\assistant-trn.json]\n",

```

```

"2020-03-13 14:21:08.255 INFO in 'deeppavlov.dataset_readers.dstc2_reader'['dstc2_reader'] at line 290: [loading dialogs from C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_data\\assistant-val.json]\\n",
"2020-03-13 14:21:08.268 INFO in 'deeppavlov.dataset_readers.dstc2_reader'['dstc2_reader'] at line 290: [loading dialogs from C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_data\\assistant-tst.json]\\n",
"2020-03-13 14:21:08.281 INFO in 'deeppavlov.dataset_readers.dstc2_reader'['dstc2_reader'] at line 282: There are 24 samples in train split.\\n",
"2020-03-13 14:21:08.291 INFO in 'deeppavlov.dataset_readers.dstc2_reader'['dstc2_reader'] at line 283: There are 3 samples in valid split.\\n",
"2020-03-13 14:21:08.307 INFO in 'deeppavlov.dataset_readers.dstc2_reader'['dstc2_reader'] at line 284: There are 3 samples in test split.\\n",
"[nltk_data] Downloading package punkt to\\n",
"[nltk_data] C:\\Users\\Лиза\\AppData\\Roaming\\nltk_data...\\n",
"[nltk_data] Package punkt is already up-to-date!\\n",
"[nltk_data] Downloading package stopwords to\\n",
"[nltk_data] C:\\Users\\Лиза\\AppData\\Roaming\\nltk_data...\\n",
"[nltk_data] Package stopwords is already up-to-date!\\n",
"[nltk_data] Downloading package perluniprops to\\n",
"[nltk_data] C:\\Users\\Лиза\\AppData\\Roaming\\nltk_data...\\n",
"[nltk_data] Package perluniprops is already up-to-date!\\n",
"[nltk_data] Downloading package nonbreaking_prefixes to\\n",
"[nltk_data] C:\\Users\\Лиза\\AppData\\Roaming\\nltk_data...\\n",
"[nltk_data] Package nonbreaking_prefixes is already up-to-date!\\n",
"2020-03-13 14:23:13.991 INFO in 'deeppavlov.core.data.simple_vocab'['simple_vocab'] at line 115: [loading vocabulary from C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_bot\\word.dict]\\n",
"2020-03-13 14:23:14.38 INFO in 'deeppavlov.core.data.simple_vocab'['simple_vocab'] at line 101: [saving vocabulary to C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_bot\\word.dict]\\n",
"2020-03-13 14:26:58.309 INFO in 'deeppavlov.models.embedders.glove_embedder'['glove_embedder'] at line 52: [loading GloVe embeddings from `C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_bot\\glove.6B.100d.txt`]\\n",
"Using TensorFlow backend.\\n"
]
},
{
"name": "stdout",
"output_type": "stream",
"text": [
"WARNING:tensorflow:From C:\\Anaconda\\lib\\site-packages\\tensorflow\\python\\framework\\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.\\n",
"Instructions for updating:\\n",

```

```

    "Colocations handled automatically by placer.\n"
]
},
{
    "name": "stderr",
    "output_type": "stream",
    "text": [
        "2020-03-13 14:30:00.239 INFO in 'deeppavlov.models.go_bot.network'['net-
work'] at line 165: [loading templates from C:\\Users\\Лиза\\Desk-
top\\bot_test\\test_ru\\assistant_data\\assistant-templates.txt]\n",
        "2020-03-13 14:30:00.244 INFO in 'deeppavlov.models.go_bot.network'['net-
work'] at line 168: 5 templates loaded.\n",
        "2020-03-13 14:30:00.254 INFO in 'deeppavlov.models.go_bot.network'['net-
work'] at line 214: Calculated input size for `GoalOrientedBotNetwork` is 129\n"
    ]
},
{
    "name": "stdout",
    "output_type": "stream",
    "text": [
        "WARNING:tensorflow:From C:\\Anaconda\\lib\\site-packages\\deeppav-
lov\\models\\go_bot\\network.py:687: dense (from tensorflow.python.layers.core) is
deprecated and will be removed in a future version.\n",
        "Instructions for updating:\n",
        "Use keras.layers.dense instead.\n",
        "WARNING:tensorflow:From C:\\Anaconda\\lib\\site-packages\\deeppav-
lov\\models\\go_bot\\network.py:737: LSTMCell.__init__ (from tensorflow.py-
thon.ops.rnn_cell_impl) is deprecated and will be removed in a future version.\n",
        "Instructions for updating:\n",
        "This class is equivalent as tf.keras.layers.LSTMCell, and will be replaced by that
in Tensorflow 2.0.\n",
        "WARNING:tensorflow:From C:\\Anaconda\\lib\\site-packages\\deeppav-
lov\\models\\go_bot\\network.py:746: dynamic_rnn (from tensorflow.python.ops.rnn)
is deprecated and will be removed in a future version.\n",
        "Instructions for updating:\n",
        "Please use `keras.layers.RNN(cell)`, which is equivalent to this API\n",
        "WARNING:tensorflow:From C:\\Anaconda\\lib\\site-packages\\tensorflow\\py-
thon\\ops\\rnn.py:626: to_int32 (from tensorflow.python.ops.math_ops) is deprecated
and will be removed in a future version.\n",
        "Instructions for updating:\n",
        "Use tf.cast instead.\n",

```

```

"WARNING:tensorflow:From C:\\Anaconda\\lib\\site-packages\\deeppav-
lov\\models\\go_bot\\network.py:251: checkpoint_exists (from tensorflow.py-
thon.training.checkpoint_management) is deprecated and will be removed in a future
version.\\n",
  "Instructions for updating:\\n",
  "Use standard file APIs to check for files with this prefix.\\n"
]
},
{
  "name": "stderr",
  "output_type": "stream",
  "text": [
    "2020-03-13 14:30:10.308 INFO in 'deeppavlov.models.go_bot.network'['net-
work'] at line 255: [initializing `GoalOrientedBot` from scratch]\\n",
    "2020-03-13 14:30:15.154 INFO in 'deeppavlov.core.trainers.nn_train-
er'['nn_trainer'] at line 198: Initial best per_item_dialog_accuracy of 0.0\\n"
  ]
},
{
  "name": "stdout",
  "output_type": "stream",
  "text": [
    "{\\\"valid\\\": {\\\"eval_examples_count\\\": 1, \\\"metrics\\\": {\\\"per_item_dialog_accu-
racy\\\": 0.0}, \\\"time_spent\\\": \\\"0:00:05\\\", \\\"epochs_done\\\": 0, \\\"batches_seen\\\": 0,
\\\"train_examples_seen\\\": 0, \\\"impatience\\\": 0, \\\"patience_limit\\\": 10}}\\n",
    "{\\\"train\\\": {\\\"eval_examples_count\\\": 8, \\\"metrics\\\": {\\\"per_item_dialog_accu-
racy\\\": 0.0}, \\\"time_spent\\\": \\\"0:00:09\\\", \\\"epochs_done\\\": 2, \\\"batches_seen\\\": 5,
\\\"train_examples_seen\\\": 20, \\\"learning_rate\\\": 0.003, \\\"momentum\\\": 0.95, \\\"loss\\\":
1.5553321361541748}}\\n",
    "{\\\"train\\\": {\\\"eval_examples_count\\\": 8, \\\"metrics\\\": {\\\"per_item_dialog_accu-
racy\\\": 0.0}, \\\"time_spent\\\": \\\"0:00:10\\\", \\\"epochs_done\\\": 4, \\\"batches_seen\\\": 10,
\\\"train_examples_seen\\\": 40, \\\"learning_rate\\\": 0.003, \\\"momentum\\\": 0.95, \\\"loss\\\":
1.3782984018325806}}\\n",
    "{\\\"train\\\": {\\\"eval_examples_count\\\": 8, \\\"metrics\\\": {\\\"per_item_dialog_accu-
racy\\\": 0.0}, \\\"time_spent\\\": \\\"0:00:11\\\", \\\"epochs_done\\\": 7, \\\"batches_seen\\\": 15,
\\\"train_examples_seen\\\": 60, \\\"learning_rate\\\": 0.003, \\\"momentum\\\": 0.95, \\\"loss\\\":
1.0438369750976562}}\\n",
    "{\\\"train\\\": {\\\"eval_examples_count\\\": 8, \\\"metrics\\\": {\\\"per_item_dialog_accu-
racy\\\": 0.0}, \\\"time_spent\\\": \\\"0:00:11\\\", \\\"epochs_done\\\": 9, \\\"batches_seen\\\": 20,
\\\"train_examples_seen\\\": 80, \\\"learning_rate\\\": 0.003, \\\"momentum\\\": 0.95, \\\"loss\\\":
0.6769822657108306}}\\n",
    "{\\\"train\\\": {\\\"eval_examples_count\\\": 8, \\\"metrics\\\": {\\\"per_item_dialog_accu-
racy\\\": 0.0}, \\\"time_spent\\\": \\\"0:00:12\\\", \\\"epochs_done\\\": 12, \\\"batches_seen\\\": 25,

```

```

{"train_examples_seen": 100, "learning_rate": 0.003, "momentum": 0.95, "loss": 0.49409288763999937}}\n",
  {"train": {"eval_examples_count": 8, "metrics": {"per_item_dialog_accuracy": 0.0}, "time_spent": "\"0:00:12\"", "epochs_done": 14, "batches_seen": 30, "train_examples_seen": 120, "learning_rate": 0.003, "momentum": 0.95, "loss": 0.45719830989837645}}\n"
]
},
{
  "name": "stderr",
  "output_type": "stream",
  "text": [
    "2020-03-13 14:30:22.204 INFO in 'deeppavlov.core.trainers.nn_trainer['nn_trainer'] at line 211: Did not improve on the per_item_dialog_accuracy of 0.0\n"
  ]
},
{
  "name": "stdout",
  "output_type": "stream",
  "text": [
    "{\"valid\": {\"eval_examples_count\": 1, \"metrics\": {\"per_item_dialog_accuracy\": 0.0}, \"time_spent\": \"0:00:12\", \"epochs_done\": 14, \"batches_seen\": 30, \"train_examples_seen\": 120, \"impatience\": 1, \"patience_limit\": 10}}\n"
  ]
},
{
  "name": "stderr",
  "output_type": "stream",
  "text": [
    "2020-03-13 14:30:22.597 INFO in 'deeppavlov.core.data.simple_vocab['simple_vocab'] at line 115: [loading vocabulary from C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_bot\\word.dict]\n",
    "2020-03-13 14:30:22.607 INFO in 'deeppavlov.models.embedders.glove_embedder['glove_embedder'] at line 52: [loading GloVe embeddings from `C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_bot\\glove.6B.100d.txt`]\n",
    "2020-03-13 14:32:28.519 INFO in 'deeppavlov.models.go_bot.network['network'] at line 165: [loading templates from C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_data\\assistant-templates.txt]\n",
    "2020-03-13 14:32:28.530 INFO in 'deeppavlov.models.go_bot.network['network'] at line 168: 5 templates loaded.\n",
    "2020-03-13 14:32:28.535 INFO in 'deeppavlov.models.go_bot.network['network'] at line 214: Calculated input size for `GoalOrientedBotNetwork` is 129\n",
    "2020-03-13 14:32:30.908 INFO in 'deeppavlov.models.go_bot.network['network'] at line 255: [initializing `GoalOrientedBot` from scratch]\n"
  ]
}

```

```

]
},
{
  "name": "stdout",
  "output_type": "stream",
  "text": [
    "{\"valid\": {\"eval_examples_count\": 1, \"metrics\": {\"per_item_dialog_accu-  
racy\": 0.0}, \"time_spent\": \"0:00:01\"}}\n",
    "{\"test\": {\"eval_examples_count\": 1, \"metrics\": {\"per_item_dialog_accu-  
racy\": 0.0}, \"time_spent\": \"0:00:01\"}}\n"
  ]
},
{
  "name": "stderr",
  "output_type": "stream",
  "text": [
    "2020-03-13 14:32:31.459 INFO in 'deeppavlov.core.data.simple_vocab'['sim-  
ple_vocab'] at line 115: [loading vocabulary from C:\\Users\\Лиза\\Desk-  
top\\bot_test\\test_ru\\assistant_bot\\word.dict]\n",
    "2020-03-13 14:32:31.531 INFO in 'deeppavlov.models.embedders.glove_embed-  
der'['glove_embedder'] at line 52: [loading GloVe embeddings from `C:\\Us-  
ers\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_bot\\glove.6B.100d.txt`]\n",
    "2020-03-13 14:34:42.102 INFO in 'deeppavlov.models.go_bot.network'['net-  
work'] at line 165: [loading templates from C:\\Users\\Лиза\\Desk-  
top\\bot_test\\test_ru\\assistant_data\\assistant-templates.txt]\n",
    "2020-03-13 14:34:42.106 INFO in 'deeppavlov.models.go_bot.network'['net-  
work'] at line 168: 5 templates loaded.\n",
    "2020-03-13 14:34:42.112 INFO in 'deeppavlov.models.go_bot.network'['net-  
work'] at line 214: Calculated input size for `GoalOrientedBotNetwork` is 129\n",
    "2020-03-13 14:34:44.954 INFO in 'deeppavlov.models.go_bot.network'['net-  
work'] at line 255: [initializing `GoalOrientedBot` from scratch]\n"
  ]
}
],
"source": [
  "from deeppavlov import train_model\n",
  "\n",
  "gobot_config['train']['batch_size'] = 4 # set batch size\n",
  "gobot_config['train']['max_batches'] = 30 # maximum number of training  
batches\n",
  "gobot_config['train']['val_every_n_batches'] = 30 # evaluate on full 'valid' split  
every 30 epochs\n",
  "gobot_config['train']['log_every_n_batches'] = 5 # evaluate on full 'train' split every  
5 batches\n",

```

```

"\n",
"train_model(gobot_config);"
]
},
{
"cell_type": "code",
"execution_count": 15,
"metadata": {},
"outputs": [
{
"name": "stderr",
"output_type": "stream",
"text": [
"2020-03-13 14:36:14.882 INFO in 'deeppavlov.core.data.simple_vocab'['simple_vocab'] at line 115: [loading vocabulary from C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_bot\\word.dict]\n",
"2020-03-13 14:36:14.893 INFO in 'deeppavlov.models.embedders.glove_embedder'['glove_embedder'] at line 52: [loading GloVe embeddings from `C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_bot\\glove.6B.100d.txt`]\n",
"2020-03-13 14:38:33.16 INFO in 'deeppavlov.models.go_bot.network'['network'] at line 165: [loading templates from C:\\Users\\Лиза\\Desktop\\bot_test\\test_ru\\assistant_data\\assistant-templates.txt]\n",
"2020-03-13 14:38:33.22 INFO in 'deeppavlov.models.go_bot.network'['network'] at line 168: 5 templates loaded.\n",
"2020-03-13 14:38:33.25 INFO in 'deeppavlov.models.go_bot.network'['network'] at line 214: Calculated input size for `GoalOrientedBotNetwork` is 129\n",
"2020-03-13 14:38:35.722 INFO in 'deeppavlov.models.go_bot.network'['network'] at line 255: [initializing `GoalOrientedBot` from scratch]\n"
]
}
],
"source": [
"from deeppavlov import build_model\n",
"\n",
"bot = build_model(gobot_config)"
]
},
{
"cell_type": "code",
"execution_count": 17,
"metadata": {},
"outputs": [
{
"data": {

```

```

    "text/plain": [
      "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
    ]
  },
  "execution_count": 17,
  "metadata": {},
  "output_type": "execute_result"
},
{
  "source": [
    "bot(['Привет'])"
  ],
  "cell_type": "code",
  "execution_count": 18,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
        ]
      },
      "execution_count": 18,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Привет, бот'])"
  ],
  "cell_type": "code",
  "execution_count": 19,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
        ]
      },

```



```

    "execution_count": 19,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Когда работает банк?'])"
]
},
{
  "cell_type": "code",
  "execution_count": 20,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
        ]
      },
      "execution_count": 20,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Привет'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 19,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Банк работает до 19:00 в буднии дни]"
        ]
      },
      "execution_count": 19,
      "metadata": {},
      "output_type": "execute_result"
    }
  ]
}

```

```

],
"source": [
  "bot(['Банк работает в воскресенье?'])"
],
{
  "cell_type": "code",
  "execution_count": 20,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Банк работает до 19:00 в буднии дни]"
        ]
      },
      "execution_count": 20,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Отлично, спасибо'])"
  ],
  {
    "cell_type": "code",
    "execution_count": 32,
    "metadata": {},
    "outputs": [
      {
        "data": {
          "text/plain": [
            "[Здравствуйте, чем я могу вам помочь?]"
          ]
        },
        "execution_count": 32,
        "metadata": {},
        "output_type": "execute_result"
      }
    ],
    "source": [
      "bot(['Банк работает в воскресенье?'])"
    ]
  }

```

```

},
{
  "cell_type": "code",
  "execution_count": 18,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Здравствуйте, чем я могу вам помочь?\"]"
        ]
      },
      "execution_count": 18,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Здравствуйте'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 20,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Здравствуйте, чем я могу вам помочь?\"]"
        ]
      },
      "execution_count": 20,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Привет'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 21,

```

```

"metadata": {},
"outputs": [
  {
    "data": {
      "text/plain": [
        "[Здравствуйте, чем я могу вам помочь?]"
      ]
    },
    "execution_count": 21,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Где находится главный офис?'])"
],
},
{
  "cell_type": "code",
  "execution_count": 22,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Здравствуйте, чем я могу вам помочь?]"
        ]
      },
      "execution_count": 22,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Адрес главного офиса банка'])"
  ],
},
{
  "cell_type": "code",
  "execution_count": 26,
  "metadata": {},
  "outputs": [
    {
      "data": {

```

```

    "text/plain": [
      "[Здравствуйте, чем я могу вам помочь?]"
    ]
  },
  "execution_count": 26,
  "metadata": {},
  "output_type": "execute_result"
},
{
  "source": [
    "bot(['Привет'])"
  ],
  "cell_type": "code",
  "execution_count": 72,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Рад был помочь! До свидания!]"
        ]
      },
      "execution_count": 72,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Спасибо'])"
  ],
  "cell_type": "code",
  "execution_count": 73,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Банк работает до 19:00 в буднии дни]"
        ]
      },

```

```

    "execution_count": 73,
    "metadata": {},
    "output_type": "execute_result"
  }
],
  "source": [
    "bot(['Время работы банка'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 23,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "['Здравствуйте, чем я могу вам помочь?']"
        ]
      },
      "execution_count": 23,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Спасибо'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 75,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "['Рад был помочь! До свидания!']"
        ]
      },
      "execution_count": 75,
      "metadata": {},
      "output_type": "execute_result"
    }
  ]
}

```

```

],
"source": [
  "bot(['Добрый день'])"
],
{
  "cell_type": "code",
  "execution_count": 76,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Банк работает до 19:00 в буднии дни]"
        ]
      },
      "execution_count": 76,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Привет'])"
  ],
  {
    "cell_type": "code",
    "execution_count": 77,
    "metadata": {},
    "outputs": [
      {
        "data": {
          "text/plain": [
            "[Здравствуйте, чем я могу вам помочь?]"
          ]
        },
        "execution_count": 77,
        "metadata": {},
        "output_type": "execute_result"
      }
    ],
    "source": [
      "bot(['Банк работает в воскресенье'])"
    ]
  }

```

```

},
{
  "cell_type": "code",
  "execution_count": 24,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Здравствуйте, чем я могу вам помочь?\"]"
        ]
      },
      "execution_count": 24,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Банк работает в воскресенье?'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 25,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Здравствуйте, чем я могу вам помочь?\"]"
        ]
      },
      "execution_count": 25,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Банк работает в воскресенье?'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 80,

```



```

"metadata": {},
"outputs": [
  {
    "data": {
      "text/plain": [
        "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
      ]
    },
    "execution_count": 80,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Спасибо'])"
],
},
{
  "cell_type": "code",
  "execution_count": 81,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
        ]
      },
      "execution_count": 81,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['приветик'])"
  ],
},
{
  "cell_type": "code",
  "execution_count": 27,
  "metadata": {},
  "outputs": [
    {
      "data": {

```

```

    "text/plain": [
      "[Здравствуйте, чем я могу вам помочь?]"
    ]
  },
  "execution_count": 27,
  "metadata": {},
  "output_type": "execute_result"
},
{
  "source": [
    "bot(['Где главный офис?'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 28,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Здравствуйте, чем я могу вам помочь?]"
        ]
      },
      "execution_count": 28,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Время работы банка'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 85,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Рад был помочь! До свидания!]"
        ]
      },

```

```

    "execution_count": 85,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Время работы банка'])"
],
},
{
  "cell_type": "code",
  "execution_count": 86,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Рад был помочь! До свидания!]"
        ]
      },
      "execution_count": 86,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Время банка'])"
  ],
},
{
  "cell_type": "code",
  "execution_count": 87,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Рад был помочь! До свидания!]"
        ]
      },
      "execution_count": 87,
      "metadata": {},
      "output_type": "execute_result"
    }
  ]
}

```

```

],
"source": [
  "bot(['Адрес'])"
],
},
{
  "cell_type": "code",
  "execution_count": 21,
  "metadata": {},
  "outputs": [],
  "source": [
    "bot.reset()"
  ],
},
{
  "cell_type": "code",
  "execution_count": 89,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Банк работает до 19:00 в буднии дни\"]"
        ]
      },
      "execution_count": 89,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Привет'])"
  ],
},
{
  "cell_type": "code",
  "execution_count": 90,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Рад был помочь! До свидания!\"]"
        ]
      }
    }
  ],
  "source": [
    "bot(['Привет'])"
  ],
},

```

```

    },
    "execution_count": 90,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Адрес'])"
],
},
{
  "cell_type": "code",
  "execution_count": 91,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Рад был помочь! До свидания!\"]"
        ]
      },
    },
    "execution_count": 91,
    "metadata": {},
    "output_type": "execute_result"
  ]
},
"source": [
  "bot(['Адрес'])"
],
},
{
  "cell_type": "code",
  "execution_count": 92,
  "metadata": {},
  "outputs": [],
  "source": [
    "bot.reset()"
  ]
},
{
  "cell_type": "code",
  "execution_count": 93,
  "metadata": {},
  "outputs": [

```

```
{
  "data": {
    "text/plain": [
      "['Банк работает до 19:00 в буднии дни']"
    ]
  },
  "execution_count": 93,
  "metadata": {},
  "output_type": "execute_result"
},
{
  "source": [
    "bot(['Адрес'])"
  ],
  "cell_type": "code",
  "execution_count": 29,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "['Здравствуйте, чем я могу вам помочь?']"
        ]
      },
      "execution_count": 29,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Привет'])"
  ],
  "cell_type": "code",
  "execution_count": 30,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "['Здравствуйте, чем я могу вам помочь?']"
        ]
      }
    }
  ]
}
```

```

    ]
  },
  "execution_count": 30,
  "metadata": {},
  "output_type": "execute_result"
},
{
  "source": [
    "bot(['Адрес банка'])"
  ],
  "cell_type": "code",
  "execution_count": 33,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Здравствуйте, чем я могу вам помочь?\"]"
        ]
      },
      "execution_count": 33,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Адрес банка?'])"
  ],
  "cell_type": "code",
  "execution_count": 35,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Здравствуйте, чем я могу вам помочь?\"]"
        ]
      },
      "execution_count": 35,
      "metadata": {}
    }
  ]
}

```

```

    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Привет'])"
]
},
{
  "cell_type": "code",
  "execution_count": 36,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Здравствуйте, чем я могу вам помочь?\"]"
        ]
      },
      "execution_count": 36,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Адрес банка'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 21,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Банк работает до 19:00 в буднии дни\"]"
        ]
      },
      "execution_count": 21,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [

```



```

    "bot(['Адрес банка'])"
  ],
  {
    "cell_type": "code",
    "execution_count": 22,
    "metadata": {},
    "outputs": [
      {
        "data": {
          "text/plain": [
            "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
          ]
        },
        "execution_count": 22,
        "metadata": {},
        "output_type": "execute_result"
      }
    ],
    "source": [
      "bot(['Спасибо'])"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": 23,
    "metadata": {},
    "outputs": [
      {
        "data": {
          "text/plain": [
            "[Банк работает до 19:00 в буднии дни]"
          ]
        },
        "execution_count": 23,
        "metadata": {},
        "output_type": "execute_result"
      }
    ],
    "source": [
      "bot(['Я не успею'])"
    ]
  },
  {

```

```

"cell_type": "code",
"execution_count": 24,
"metadata": {},
"outputs": [
  {
    "data": {
      "text/plain": [
        "[\"Банк работает до 19:00 в буднии дни\"]"
      ]
    },
    "execution_count": 24,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Я не смогу сегодня'])"
]
},
{
  "cell_type": "code",
  "execution_count": 25,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Банк работает до 19:00 в буднии дни\"]"
        ]
      },
      "execution_count": 25,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Привет'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 26,
  "metadata": {},
  "outputs": [

```

```

{
  "data": {
    "text/plain": [
      "[\"Банк работает до 19:00 в буднии дни\"]"
    ]
  },
  "execution_count": 26,
  "metadata": {},
  "output_type": "execute_result"
},
{
  "source": [
    "bot(['Спасибо за помощь'])"
  ],
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
},
{
  "cell_type": "code",
  "execution_count": 27,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Банк работает до 19:00 в буднии дни\"]"
        ]
      },
      "execution_count": 27,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Когда работает банк?'])"
  ],
  "cell_type": "code",

```

```

"execution_count": 28,
"metadata": {},
"outputs": [
  {
    "data": {
      "text/plain": [
        "[\"Банк работает до 19:00 в буднии дни\"]"
      ]
    },
    "execution_count": 28,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Жаль'])"
],
},
{
  "cell_type": "code",
  "execution_count": 30,
  "metadata": {},
  "outputs": [],
  "source": [
    "bot.reset()"
  ]
},
{
  "cell_type": "code",
  "execution_count": 31,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Ул. А. Невского 14\"]"
        ]
      },
      "execution_count": 31,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [

```

```

    "bot(['Привет'])"
  ],
  {
    "cell_type": "code",
    "execution_count": 32,
    "metadata": {},
    "outputs": [
      {
        "data": {
          "text/plain": [
            "['Ул. А. Невского 14']"
          ]
        },
        "execution_count": 32,
        "metadata": {},
        "output_type": "execute_result"
      }
    ],
    "source": [
      "bot(['Привет'])"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": 33,
    "metadata": {},
    "outputs": [
      {
        "data": {
          "text/plain": [
            "['Банк работает до 19:00 в буднии дни']"
          ]
        },
        "execution_count": 33,
        "metadata": {},
        "output_type": "execute_result"
      }
    ],
    "source": [
      "bot(['Банк работает в воскресенье?'])"
    ]
  },
  {

```

```

"cell_type": "code",
"execution_count": 34,
"metadata": {},
"outputs": [
  {
    "data": {
      "text/plain": [
        "[\"Банк работает до 19:00 в буднии дни\"]"
      ]
    },
    "execution_count": 34,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "bot(['Отлично, спасибо'])"
]
},
{
  "cell_type": "code",
  "execution_count": 35,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[\"Банк работает до 19:00 в буднии дни\"]"
        ]
      },
      "execution_count": 35,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Когда банк закрывается?'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": 22,
  "metadata": {},
  "outputs": [

```

```
{
  "data": {
    "text/plain": [
      "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
    ]
  },
  "execution_count": 22,
  "metadata": {},
  "output_type": "execute_result"
},
{
  "source": [
    "bot(['Здравствуйте'])"
  ],
  "cell_type": "code",
  "execution_count": 23,
  "metadata": {},
  "outputs": [],
  "source": [
    "bot.reset()"
  ],
  "cell_type": "code",
  "execution_count": 24,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
        ]
      },
      "execution_count": 24,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Когда работает банк'])"
  ],
}
```

```

{
  "cell_type": "code",
  "execution_count": 25,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "[Какая жалость! В следующий раз, может быть. Хорошего дня!]"
        ]
      },
      "execution_count": 25,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "bot(['Спасибо'])"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
}
],
"metadata": {
  "kernel_spec": {
    "display_name": "Python 3",
    "language": "python",
    "name": "python3"
  },
  "language_info": {
    "codemirror_mode": {
      "name": "ipython",
      "version": 3
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",
    "nbconvert_exporter": "python",
    "pygments_lexer": "ipython3",

```



```

    "version": "3.7.2"
  }
},
"nbformat": 4,
"nbformat_minor": 2
}

```

main.py

```

import random
import json
import vk_api
from vk_api.longpoll import VkLongPoll, VkEventType
from openpyxl import load_workbook
from commander.commander import Commander
from vk_bot import VkBot

def write_msg(user_id, message):
    vk.method('messages.send', {'user_id': user_id, 'message': message, 'random_id':
    random.randint(0, 2048)})

# API-ключ
token =
"40bd2757f0d8ffdce488c78b0d5f33e5e2af9eeb090724bf7933a7e0f6e2cf48c9c0b170
26115d464e0c1"

# Авторизуемся как сообщество
vk = vk_api.VkApi(token=token)

# Работа с сообщениями
longpoll = VkLongPoll(vk)

# Подключение базы клиентов из Excel
wb_val = load_workbook(filename = 'Банк.xlsx', data_only = True)
sheet_val = wb_val['Клиенты']
#D2_val = sheet_val['D2'].value
for i in range(2, 8):
    print(i, sheet_val.cell(row=i, column=4).value)

commander = Commander()
print("Server started")
for event in longpoll.listen():

    if event.type == VkEventType.MESSAGE_NEW:

```

```

if event.to_me:

    print(f'New message from {event.user_id}', end=")

    bot = VkBot(event.user_id)

    if event.text[0] == "/":
        write_msg(event.user_id, commander.do(event.text[1::]))
    else:
        write_msg(event.user_id, bot.new_message(event.text))

    print('Text message: ', event.text)
    print("-----")

```

vk_bot.py

```

import bs4 as bs4
import requests
from bs4 import BeautifulSoup

URL_currency = 'https://kaliningrad.bankiros.ru/bank/sberbank/currency'
HEADERS = {'user-agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:76.0) Gecko/20100101 Firefox/76.0',
            'accept': '*//*'}
HOST = 'https://kaliningrad.bankiros.ru'

class VkBot:

    def __init__(self, user_id):
        print("\nСоздан объект бота!")
        self._USER_ID = user_id
        self._USERNAME = self._get_user_name_from_vk_id(user_id)
        self._COMMANDS = ["ПРИВЕТ", "ВРЕМЯ"]

    # Получение имени пользователя
    def _get_user_name_from_vk_id(self, user_id):
        request = requests.get("https://vk.com/id"+str(user_id))
        bs = bs4.BeautifulSoup(request.text, "html.parser")
        user_name = self._clean_all_tag_from_str(bs.findAll("title")[0])
        return user_name.split()[0]

    # Сообщение
    def new_message(self, message):
        if message.upper() == self._COMMANDS[0]:

```

```

        return f"Здравствуйте, {self._USERNAME}!"
    elif message.upper() == self._COMMANDS[1]:
        return self._get_time()
    else:
        return "Не понимаю о чем вы..."
# Банк (ширина, долгота)
def bank_location(message):
    lat_b = message.location.latitude
    lon_b = message.location.longitude
    print(format(lat_b, lon_b))
# Банкомат (ширина, долгота)
def bankomat_location(self, message):
    lat_bt = message.location.latitude
    lon_bt = message.location.longitude
    print(format(lat_bt, lon_bt))

URL_currency = 'https://kaliningrad.bankiros.ru/bank/sberbank/currency'
HEADERS = {'user-agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:76.0)
Gecko/20100101 Firefox/76.0',
            'accept': '*//*'}
HOST = 'https://kaliningrad.bankiros.ru'

# Парсинг сайта
def get_html(url, params = None):
    r = requests.get(url, headers = HEADERS, params = params)
    return r
# Содержимое сайта
def get_content(html):
    soup = BeautifulSoup(html, 'html.parser')
    items = soup.find_all('a', class_='actual-currency-item')
    currency = []
    for item in items:
        sale = item.find('span', class_='size15')
        if sale:
            sale = sale.get_text()
        else:
            sale = 'Course to clarify'
        currency.append({
            'title': item.find('div', class_='actual-currency-name').get_text(strip = True),
            'link': HOST + item.find('span', class_='link').get('href'),
            'pur1': item.find('strong', class_='dollar').get_text(),
            'sale1': item.find('strong', class_='dollar').get_text(),
            'pur2': item.find('strong', class_='euro').get_text(),
            'sale2': item.find('strong', class_='euro').get_text(),

```

```

        'pur3': item.find('strong', class_='zl').get_text(),
        'sale3': item.find('strong', class_='zl').get_text(),
        'gold': item.find('strong', class_='metal').get_text(),
        'silver': item.find('strong', class_='metal').get_text(),
        'platinum': item.find('strong', class_='metal').get_text(),
        'palladium': item.find('strong', class_='metal').get_text()
    })
    print(currency)

def parse():
    html = get_html(URL_currency)
    if html.status_code == 200:
        get_content(html.text)
    else:
        print('Unable to access the site')
parse()

@staticmethod
def _clean_all_tag_from_str(string_line):
    """
    Очистка строки stringLine от тэгов и их содержимых
    :param string_line: Очищаемая строка
    :return: очищенная строка
    """

    result = ""
    not_skip = True
    for i in list(string_line):
        if not_skip:
            if i == "<":
                not_skip = False
            else:
                result += i
        else:
            if i == ">":
                not_skip = True
    return result

```

commander.py

```

import os

class Commander:

```

```

def __init__(self):
    self.command_list = []
    self.create_command_list()
    filename_trn = "assistant_data/assistant-trn.json"
    filename_tst = "assistant_data/assistant-tst.json"
    filename_val = "assistant_data/assistant-val.json"
    file_trn = open(filename_trn, "r", encoding="UTF-8")
    file_tst = open(filename_tst, "r", encoding="UTF-8")
    file_val = open(filename_val, "r", encoding="UTF-8")
    data_trn = file_trn.read().split("\n")
    data_tst = file_tst.read().split("\n")
    data_val = file_val.read().split("\n")

    @staticmethod
    def command(command):
        os.system(command)

    # Обращение к файлу с командами
    def create_command_list(self):
        filename = "commander/assistant-templates.txt"
        file = open(filename, "r", encoding="UTF-8")
        res = file.read().split("\n")
        for data in res:
            data = data.split("@")
            self.command_list.append([data[1], data[0].split(",")])

    def get_command(self, inp):
        for data in self.command_list:
            if inp in data[1]:
                return data[0]

        return None

    def do(self, user_input):
        command = self.get_command(user_input)

        if command is None:
            return "Error command"
        else:
            os.system(command)
            return "Done!"

```

```

@staticmethod
def compare(name: str, array: list, upper: bool = True) -> bool:
    """
    Сравнивает значение переданного слова со значениями массива. Так же
    учитываются возможные опечатки,
    но только позиционно. То есть каждая позиция проверяется с соответствующей.

    :param name: проверяемое слово
    :param array: массив, где хранятся возможные значения слова
    :param upper: если истина, то не обращает внимания на регистр, иначе различает
    :return: если хотя бы одно значение с массива совпадает со словом, возвращает True, иначе False
    """
    if upper:
        name = name.upper()
        for i in range(len(array)):
            array[i] = array[i].upper()

    for i in array:
        k = 0 # считывание разницы в символах (посимвольно, позиционно)
        if len(i) > len(name):
            for j in range(len(name)):
                if name[j] == i[j]:
                    pass
                else:
                    k = k + 1
            else:
                for j in range(len(i)):
                    if i[j] == name[j]:
                        pass
                    else:
                        k = k + 1

        k = k + abs(len(i) - len(name)) # добавление разницы в недостающих символах

    # Обработка возможной опечатки
    if 7 > len(name) > 4 and k < 3:
        return True
    elif 7 <= len(name) < 12 and k < 5:
        return True
    elif len(name) > 11 and k < 7:

```

```

    return True
elif len(name) <= 4 and k < 1:
    return True

return False

```

assistant-templates.txt

```

welcome_msg    Здравствуйте, {self._USERNAME}! Чем я могу вам помочь?
time_bank      Вас интересует полное расписание или конкретный день недели?
addr_bank      Банк располагается на ул. А. Невского 14. {lat_b, lon_b}
good_bye       Рад был помочь! До свидания!
bad_bye        Какая жалость! Я могу Вам еще чем-то помочь?
addr_bankomat  Банкомат находится на ул. Театральная 35, этаж 1 {lat_bt, lon_bt}
ex_rates       Какая валюта Вас интересует? Могу предложить доллар, евро, злотый.
Вы также можете узнать курс других валют на сайте https://kalinin-grad.bankiros.ru/bank/sberbank/currency
dollar_rate    ПОКУПКА: {pur1} ПРОДАЖА: {sale1}
euro_rate      ПОКУПКА: {pur2} ПРОДАЖА: {sale2}
pln_rate       ПОКУПКА: {pur3} ПРОДАЖА: {sale3}
gold {gold}
silver {silver}
platinum {platinum}
palladium {palladium}
all_time_bank  ПН: с 09:00 до 18:00 ВТ: с 09:00 до 18:00 СР: с 09:00 до 18:00 ЧТ:
с 09:00 до 18:00 ПТ: с 09:00 до 18:00 СБ: с 10:00 до 17:00 ВС: выходной день
monday_time    с 09:00 до 18:00
tuesday_time   с 09:00 до 18:00
wednesday_time с 09:00 до 18:00
thursday_time  с 09:00 до 18:00
friday_time    с 09:00 до 18:00
saturday_time  с 10:00 до 17:00
sunday_time    выходной день
mtl_rates      Какой драгоценный металл Вас интересует (золото, серебро, платина,
палладий)? Вы также можете узнать курс драгметаллов на сайте ЦБ
http://mfd.ru/centrobank/preciousmetals/

```