Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехника и комплексная автоматизация»

КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

НА ТЕМУ:

«Разработка чат-бота для банковских систем»

Студент РК6-81Б _____ **Гунько Н.М.**
(Группа) (Подпись, дата) (Фамилия И.О.)

Руководитель ВКР _____ **Витюков Ф.А.**
(Подпись, дата) (Фамилия И.О.)

Нормоконтролёр _____ **Грошев С.В.**
(Подпись, дата) (Фамилия И.О.)

2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой РК6
(Индекс)
_____ Карпенко А.П.
(Фамилия И.О.)
«15» февраля 2023 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы бакалавра

Студент группы РК6-81Б

Гунько Никита Макарович
(фамилия, имя, отчество)

Тема квалификационной работы: «Разработка чат-бота для банковских систем»

Источник тематики (НИР кафедры, заказ организаций и т. п.): ООО “ЮБС”.

Тема квалификационной работы утверждена распоряжением по факультету РК №___ от «__» _____ 2021 г.

Техническое задание

Часть 1. *Аналитическая часть*

Исследовать архитектуру и алгоритм работы чат-ботов. Рассмотреть основные этапы разработки чат-ботов. Определить роль машинного обучения в разработке виртуальных ассистентов. Рассмотреть две архитектуры нейронных сетей: LSTM и Transformer. Выявить возможные варианты использования чат-ботов и выполнить сравнение существующих решений в банковской сфере. Определить роль языковых моделей в разработке чат-ботов и выполнить обзор наиболее развитой для генерации текста модели GPT.

Часть 2. *Практическая часть 1. Разработка чат-бота*

Разработать прототип чата с чат-ботом с использованием предобученной языковой модели GPT3.5-turbo через API OpenAI для генерации естественного языка и Embedding модели для встраивания дополнительного контекста.

Часть 3. *Практическая часть 2. Разработка чат-бота, работающего без использования сети Интернет*

Улучшить решение из первой практической части, перейдя от использования OpenAI API к локальному решению, не требующему подключения к сети Интернет.

Оформление выпускной квалификационной работы

Расчетно-пояснительная записка на 64 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

Работа содержит 5 графических листов формата А1.

Дата выдачи задания: «10» февраля 2023 г.

В соответствии с учебным планом выпускную квалификационную работу выполнить в полном объеме в срок до «21» июня 2023 г.

Руководитель квалификационной работы _____ **Витюков Ф.А.**
(Подпись, дата) (Фамилия И.О.)

Студент _____ **Гунько Н.М.**
(Подпись, дата) (Фамилия И.О.)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ РК

КАФЕДРА РК6

ГРУППА РК6-81Б

УТВЕРЖДАЮ

Заведующий кафедрой РК6
(Индекс)

_____ Карпенко А.П.
(Фамилия И.О.)

«15» февраля 2023 г.

КАЛЕНДАРНЫЙ ПЛАН
выполнения выпускной квалификационной работы

Студент группы РК6-81Б

Гунько Никита Макарович
(фамилия, имя, отчество)

Тема квалификационной работы: «Разработка чат-бота для банковских систем»

№ п/п	Наименование этапов выпускной квалификационной работы	Сроки выполнения этапов		Отметка о выполнении	
		план	факт	Должность	ФИО, подпись
1.	Задание на выполнение работы. Формулирование проблемы, цели и задач работы	10.02.2023	_____	Руководитель ВКР	Витюков Ф.А.
2.	1 часть. Аналитическая часть	18.02.2023	_____	Руководитель ВКР	Витюков Ф.А.
3.	Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач	28.02.2023	_____	Заведующий кафедрой	Карпенко А.П.
4.	2 часть. Практическая часть 1	21.04.2023	_____	Руководитель ВКР	Витюков Ф.А.
5.	3 часть. Практическая часть 2	23.05.2023	_____	Руководитель ВКР	Витюков Ф.А.
6.	1-я редакция работы	28.05.2023	_____	Руководитель ВКР	Витюков Ф.А.
7.	Подготовка доклада и презентации	04.06.2023	_____	Студент	Гунько Н.М.
8.	Заключение руководителя	10.06.2023	_____	Руководитель ВКР	Витюков Ф.А.
9.	Допуск работы к защите на ГЭК (нормоконтроль)	16.06.2023	_____	Нормоконтролер	Грошев С.В.
10.	Внешняя рецензия	19.06.2022	_____		
11.	Защита работы на ГЭК	21.06.2022	_____		

Студент _____ Гунько Н.М.
(подпись, дата) (Фамилия И.О.)

Руководитель ВКР _____ Витюков Ф.А.
(подпись, дата) (Фамилия И.О.)

АННОТАЦИЯ

Работа посвящена разработке чат-бота для корпоративных банковских систем. В работе приведена классификация виртуальных помощников и рассмотрены их архитектуры. Показана роль алгоритмов машинного обучения при разработке чат-ботов. Выделены и описаны две основные архитектуры нейронных сетей для решения задачи генерации человеческой речи – LSTM и Transformer. Рассмотрено понятие языковой модели и описана ее роль в разработке чат-ботов. Выполнено сравнение двух последних версий наиболее развитой в наше время языковой модели для генерации речи GPT. Выполнена разработка виртуального ассистента с использованием инструментов, предлагаемых компанией OpenAI и далее проделана модернизация этого решения для работы чат-бота без использования сети Интернет средствами языковых моделей в открытом доступе.

Тип работы: выпускная квалификационная работа.

Тема работы: «Разработка чат-бота для банковских систем».

Объекты исследований: процесс создания чат-ботов, внедрение готовых языковых моделей, запуск крупных языковых моделей на домашнем пк с небольшими ресурсами, разработка виртуального ассистента с использованием языковой модели.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Bot (Бот) – автономная программа, которая может взаимодействовать с компьютерными системами, программами или пользователями. В большинстве случаев находится под прямым или косвенным управлением человека.

Chat-Bot (Чат-бот) – это автономная программа, которая имитирует реальный разговор с пользователем. Чат-боты позволяют общаться с помощью текстовых или аудио сообщений на сайтах, в мессенджерах, мобильных приложениях или по телефону.

Natural Language Processing (NLP) – это направление в машинном обучении, посвященное распознаванию, генерации и обработке устной и письменной человеческой речи.

Естественный язык – это хранящаяся в сознании человека сложная система правил, в соответствии с которыми происходит речевая деятельность, т.е. порождение и понимание текстов.

ML (англ. Machine Learning – Машинное обучение) – это подраздел ИИ о разработке алгоритмов и моделей, способных решать задачи через обобщение множества схожих примеров. В рамках машинного обучения система собирает информацию, учится на ней, а затем использует то, чему научилась, для принятия решений.

Backpropagation (Метод обратного распространения ошибки) – метод обучения нейронных сетей, относится к методам обучения с учителем. Цель метода проста – отрегулировать веса пропорционально тому, насколько он способствует общей ошибке. Является одним из наиболее известных алгоритмов машинного обучения. На каждой итерации происходит два прохода сети – прямой и обратный. На прямом методе входной вектор распространяется от входов сети к ее выходам и формирует некоторый выходной вектор, соответствующий текущему (фактическому) состоянию весов. Затем вычисляется ошибка нейронной сети как разность между фактическим и целевым значениями. На обратном проходе эта ошибка распространяется от

выхода сети к ее входам, и производится коррекция весов нейронов в соответствии с правилом.

Dataset (Набор данных) – набор структурированных данных, предназначенных для обучения моделей нейронных сетей.

API (англ. Application Programming Interface – программный интерфейс приложения) – это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.

OpenAI – это исследовательская организация, которая специализируется на искусственном интеллекте и машинном обучении. Она была основана в 2015 году группой выдающихся исследователей и разработчиков, в том числе Илоном Маском, Сэмом Альтманом и Джоном Скальза. OpenAI стремится повысить понимание искусственного интеллекта и применять его для решения сложных проблем в мире.

Hugging Face – это платформа с коллекцией готовых современных предварительно обученных Deep Learning моделей. А библиотека Transformers предоставляет инструменты и интерфейсы для их простой загрузки и использования. Это позволяет вам экономить время и ресурсы, необходимые для обучения моделей с нуля.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В наше время, эпоху всеобщего удаленного доступа, большинство людей при использовании банковских услуг сталкиваются с рядом проблем, в частности:

- Опасность заражения каким-либо вирусом в периоды повышения уровня заболеваемости;
- Удаленность банковского отделения от места жительства в некоторых регионах;
- Длительное время ожидания;
- Несоответствие: разные агенты по обслуживанию клиентов дают разные ответы.

Многие из приведенных проблем может решить дистанционное обслуживание клиентов. Банковский виртуальный собеседник или другими словами чат-бот – одно из таких решений, которое автоматизирует процесс взаимодействия с клиентами.

Одной из проблем большинства существующих чат-ботов является то, что общение с ними очень похоже на общение с роботами, которые запрограммированы использовать ограниченный набор слов и инструкций. Люди предпочитают общаться с другими людьми, а не с роботами. Именно поэтому, на текущем этапе развития чат-ботов, они предпочитают обратиться в отделение банка или позвонить оператору. Взаимодействие с чат-ботами зачастую неотличимо от работы с пользовательским интерфейсом банковского приложения, где приходится переключаться между различными окнами с помощью кнопок.

Хотя можно было бы сказать, что все функции легко доступны в банковском приложении, на самом деле это не совсем так. В идеальном мире разработки все функции действительно были бы четко представлены в пользовательском интерфейсе с понятными объяснениями, и банковские продукты были бы настолько простыми, что их описание помещалось бы на

полстраницы. Однако, учитывая объем разработки и количество подразделений, работающих над продуктом, это обычно не так. Документация по банковским продуктам содержит сотни страниц, полными деталей.

Именно поэтому крупные компании, такие как Сбер и Тинькофф, внедряют чат-ботов в свою работу, чтобы снизить нагрузку на операторов и помочь пользователям разобраться в интерфейсе. Чат-боты также способны помочь пользователям справиться с сложными продуктами, отвечая на вопросы так, будто ими занимается живой человек.

Цель данной выпускной квалификационной работы заключается в разработке чат-бота, который будет соответствовать описанным потребностям и предоставлять поддержку пользователям.

В современной банковской сфере, где конкуренция растет, создание конкурентного преимущества является приоритетом для банковских организаций. Разработка банковского чат-бота, способного вести естественный диалог с клиентами, становится ключевым инновационным шагом для достижения этой цели. Банковский чат-бот помогает повысить операционную эффективность, улучшить качество обслуживания и удовлетворить растущие потребности клиентов в мгновенном доступе к информации и поддержке. Также, актуальность темы обусловлена потребностью предприятия ООО «ЮБС», которое является поставщиком банковского программного обеспечения.

В последнее время внедрение языковых моделей, основанных на искусственном интеллекте, стало популярным подходом для улучшения качества речи чат-ботов. Языковая модель, обученная на больших объемах текстовых данных, способна генерировать более естественные и связные ответы, что делает взаимодействие с чат-ботом более приятным и продуктивным для пользователей.

Цель работы: разработка чат-бота для банковских систем.

1 Архитектура и алгоритм работы чат-ботов

1.1 Обзор концепций компьютерного бота

Бот – это виртуальный робот или искусственный интеллект, работающий по набору алгоритмов, который описан в виде компьютерной программы. Он автоматически выполняет определенные задачи, заложенные в него разработчиком.

Существует большое количество разновидностей ботов, которые отличаются наборами выполняемых задач: чат-боты, которые имитируют разговор с человеком, боты для совершения покупок, которые осуществляют отслеживание цен и выполняют поиск лучшей цены на продукты, интересные пользователю, боты-поисковики, боты-загрузчики и т.д.

Можно выделить следующие плюсы использования компьютерных и интернет-ботов:

- Быстрее людей выполняют повторяющиеся задачи;
- Доступны круглосуточно (24/7);
- Приложения для обмена сообщениями позволяют компаниям общаться с большим количеством людей;
- Есть и ряд минусов использования компьютерных и интернет-ботов:
- Ботов нельзя настроить для выполнения определенных задач, в которых есть риск неправильно понять пользователей и вызвать у них разочарование в процессе;
- Для управления ботами по-прежнему требуются люди. Также участие человека необходимо в случае возникновения непонимания;
- Боты могут быть запрограммированы на совершение вредоносных действий;
- Ботов можно использовать для рассылки спама.

1.2 Основные определения и классификация

Чат-бот – разновидность ботов, программа, которая имитирует реальный разговор с пользователем. Они позволяют общаться с помощью текстовых или аудио сообщений на сайтах, в мессенджерах, мобильных приложениях или по телефону.

Чат-боты – это специальные аккаунты, за которыми не закреплен какой-либо человек, а сообщения, отправленные с них или на них, обрабатываются внешней системой. Кроме того, для пользователя общение с ботом выглядит как обычная переписка с реальным человеком [9].

Чат-боты помогают автоматизировать некоторые задачи, работая по заданному алгоритму. Первые программы, имитирующие общение людей, появились в 1966 году. Виртуальный собеседник Elisa достаточно убедительно пародировал диалог с психотерапевтом. С ростом популярности мессенджеров в 2010-х годах чат-боты обрели новую жизнь. Большинство работает на платформах популярных мессенджеров: Facebook, Telegram, WhatsApp, “ВКонтакте” и другие [1].

Можно выделить два общих типа классификации чат-ботов: бизнес-классификация и классификация чат-бот приложений по техническому типу. Диаграмма бизнес-классификации чат-ботов приведена на рисунке 1.

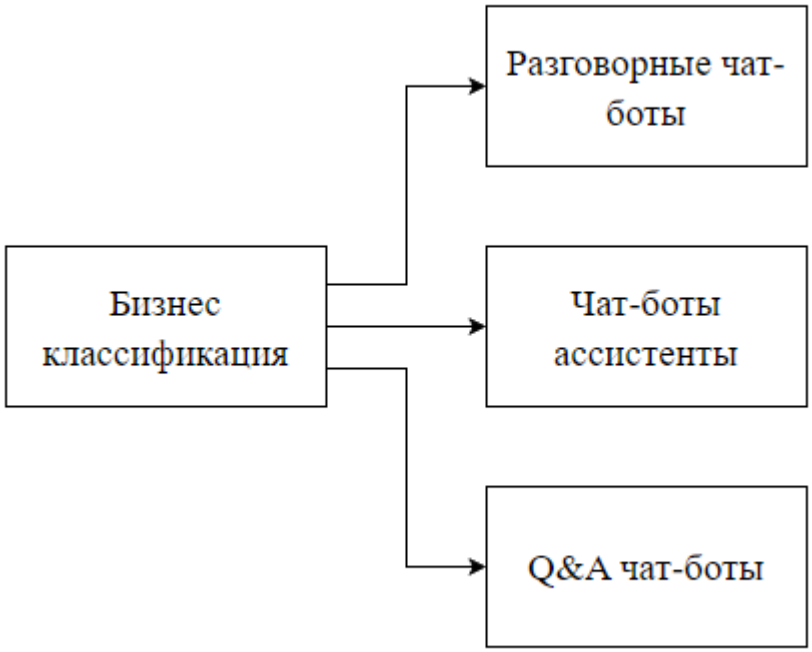


Рисунок 1 – Бизнес классификация чат-ботов

Рассмотрим каждый из типов более подробно:

1. **Разговорные чат-боты** созданы для общения подобно разговору с человеком. Не имеют конкретной цели;

2. **Чат-боты ассистенты** имеют конкретную, заранее определенную цель. Из пользовательских сообщений выделяются данные, которые используются для достижения определенных целей. Могут служить заменой или помощниками (ассистентами) в получении банковской выписки или подбора выгодного кредита on-line;

3. **Q&A** (question and answer) чат-боты, созданные для ответа на вопросы по принципу “1 вопрос – 1 ответ”. Могут служить заменой FAQ раздела различных сайтов.

Рисунок 2 – Техническая классификация чат-ботов

Чат-ботов разделяют по алгоритму работы на **ограниченных**, **саморазвивающихся** и **гибридных**. Ограниченные (простые) чат-боты взаимодействуют с пользователем на основе запрограммированных сценариев с множественным выбором. Они имеют ограниченные возможности и обычно называются ботами, основанными на правилах. Например, опция А ведет к опции В и так далее. Таких чат-ботов легче создавать, потому что зачастую они используют простой алгоритм true-false для понимания запросов пользователей и предоставления соответствующих ответов. Недостатком таких чат-ботов является то, что они не могут отвечать ни на какие вопросы, выходящие за рамки установленных правил. Также они не обучаются посредством взаимодействия с пользователями [2].

В основе саморазвивающихся чат-ботов лежит искусственный интеллект, который “понимает” контекст и цель вопроса, прежде чем формулировать ответ. Такие компьютерные ассистенты используют машинное обучение для выявления моделей общения. Благодаря постоянному взаимодействию с людьми они учатся подражать реальным разговорам и реагируют на устные или

письменные запросы, помогая найти ответы. Поскольку чат-боты используют искусственный интеллект, то понимают язык, а не просто команды. Таким образом, после каждого диалога они становятся умнее и лучше взаимодействуют с пользователями [3].

Третья группа виртуальных помощников – гибридные. Они представляют из себя комбинацию простых и умных чат-ботов. И простые, и умные чат-боты являются крайностями в спектре чат-ботов. Постоянно будет потребность в том, чтобы простые чат-боты были умнее, а умные чат-боты – проще. Гибридные чат-боты соответствуют этой золотой середине. У гибридных чат-ботов есть некоторые задачи, основанные на правилах, и они могут понимать намерения и контекст. Это делает их сбалансированным инструментом взаимодействия бизнеса с клиентами [4].

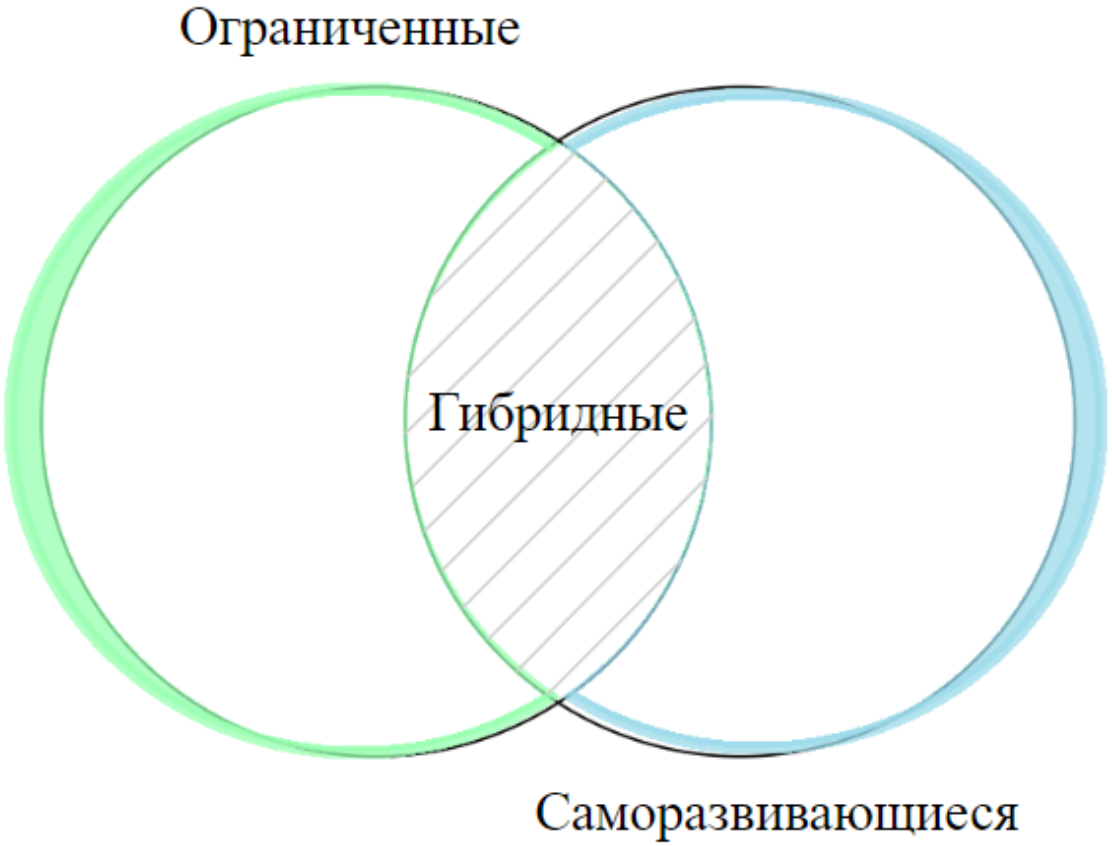


Рисунок 3 – Гибридные чат-боты

Умные, а также гибридные чат-боты используют в своей логике алгоритмы обработки и понимания естественного языка, а также алгоритмы генерации ответа, каждый из которых использует методы машинного обучения.

2 Обработка естественного языка

2.1 Основные понятия

Обработка естественного языка (Natural Language Processing) – возникла из компьютерной лингвистики, использует методы из различных дисциплин, таких как информатика, искусственный интеллект, лингвистика и наука о данных, чтобы позволить компьютерам понимать человеческий язык как в письменной, так и в устной форме. В то время как компьютерная лингвистика больше сосредоточена на аспектах языка, обработка естественного языка делает упор на использование машинного обучения и методов глубокого обучения для выполнения таких задач, как языковой перевод или ответы на вопросы. Обработка естественного языка работает, беря неструктурированные данные и преобразовывая их в формат структурированных данных. Это достигается за счет идентификации именованных сущностей (процесс, называемый распознаванием именованных сущностей) и выявления шаблонов слов с использованием таких методов, как токенизация, выделение корней и лемматизация, которые исследуют корневые формы слов [6].

2.2 Понимание естественного языка

Понимание естественного языка (Natural Language Understanding) – это часть обработки естественного языка, которая использует синтаксический и семантический анализ текста и речи для определения значения предложения. Синтаксис относится к грамматической структуре предложения, а семантика

указывает на его предполагаемое значение. NLU также устанавливает соответствующую онтологию: структуру данных, которая определяет отношения между словами и фразами. В то время как люди, естественно, делают это во время разговора, комбинация этих анализов требуется для того, чтобы машина понимала предполагаемое значение различных текстов. Наша способность различать омонимы и омофоны хорошо иллюстрирует нюансы языка.

Например, возьмем следующие два предложения:

1. Алиса плывет против течения;
2. Текущая версия отчета находится в папке.

В первом предложении слово течение является существительным. Глагол, который предшествует ему, плавать, предоставляет читателю дополнительный контекст, позволяя нам сделать вывод о том, что мы имеем в виду течение воды в водоеме. Во втором предложении слово текущая используется, но как прилагательное. Описываемое им существительное, версия, обозначает несколько итераций отчета, что позволяет нам определить, что мы имеем в виду наиболее актуальный статус файла.

Эти подходы также широко используются в интеллектуальном анализе данных, чтобы понять отношение потребителей. В частности, анализ настроений позволяет брендам более внимательно отслеживать отзывы своих клиентов, позволяя им группировать положительные и отрицательные комментарии в социальных сетях и отслеживать чистые оценки промоутеров. Просматривая негативные комментарии, компании могут быстрее выявлять и устранять потенциальные проблемные области в своих продуктах или услугах.

2.3 Генерация естественного языка

Генерация естественного языка (Natural Language Generation) – еще одно подмножество обработки естественного языка. В то время как понимание

естественного языка сосредоточено на понимании компьютерного чтения, генерация естественного языка позволяет компьютерам писать. NLG — это процесс создания текстового ответа на человеческом языке на основе некоторых входных данных. Этот текст также можно преобразовать в речевой формат с помощью служб преобразования текста в речь. NLG также включает в себя возможности суммирования текста, которые генерируют сводки из входящих документов, сохраняя при этом целостность информации.

Как и в случае с NLU, приложения NLG должны учитывать языковые правила, основанные на морфологии, лексике, синтаксисе и семантике, чтобы сделать выбор в отношении того, как правильно формулировать ответы. Они решают эту задачу в три этапа:

- Планирование текста: на этом этапе формулируется и логически упорядочивается общее содержание;
- Планирование предложений: на этом этапе учитываются пунктуация и поток текста, разбивка содержания на абзацы и предложения и включение местоимений или союзов, где это уместно;
- Реализация: этот этап учитывает грамматическую точность, гарантируя соблюдение правил пунктуации и спряжения.

3 Машинное обучение

3.1 Основные принципы

Машинное обучение (Machine Learning – ML) – это использование математических моделей данных, которые помогают компьютеру обучаться без непосредственных инструкций. Оно считается одной из форм искусственного интеллекта. При машинном обучении с помощью алгоритмов выявляются закономерности в данных. На основе этих закономерностей создается модель данных для прогнозирования новых, не встречавшихся ранее случаев, исход которых неизвестен. Чем больше данных обрабатывает такая модель и чем

дольше она используется, тем точнее становятся результаты. Это очень похоже на то, как человек оттачивает навыки на практике [7].

3.2 Взаимосвязь в рамках технологий ИИ

Как мы уже выяснили, машинное обучение считается одной из форм искусственного интеллекта. В дискуссиях об искусственном интеллекте вообще и о машинном обучении в частности обычно смешиваются нейронные сети, машинное и глубокое обучение.

Рисунок 4 – Иерархия терминов из области искусственного интеллекта

- Нейронные сети – один из видов машинного обучения.
- Глубокое обучение – это один из видов архитектуры нейронных сетей.

Рисунок 5 – Иерархия машинного обучения в рамках технологий искусственного интеллекта

3.3 Основные виды

3.3.1 Классическое обучение

Это простейшие алгоритмы, которые являются прямыми наследниками вычислительных машин 1950-х годов. Они изначально решали формальные задачи – такие, как поиск закономерностей в расчетах и вычисление траектории объектов. Сегодня алгоритмы на базе классического обучения – самые распространенные. Именно они формируют блок рекомендаций на многих платформах.

Классическое обучение подразделяется на следующие типы:

Обучение с учителем – когда у машины есть некий учитель, который знает, какой ответ правильный. Это значит, что исходные данные уже размечены (отсортированы) нужным образом, и машине остается лишь определить объект с нужным признаком или вычислить результат.

Такие модели используют в спам-фильтрах, распознавании языков и рукописного текста, выявлении мошеннических операций, расчете финансовых показателей, скоринге при выдаче кредита. В медицинской диагностике классификация помогает выявлять аномалии – то есть возможные признаки заболеваний на снимках пациентов.

Обучение без учителя – когда машина сама должна найти среди хаотичных данных верное решение и отсортировать объекты по неизвестным признакам. Например, определить, где на фото собака.

Эта модель возникла в 1990-х годах и на практике используется гораздо реже. Ее применяют для данных, которые просто невозможно разметить из-за их колоссального объема. Такие алгоритмы применяют для риск-менеджмента, сжатия изображений, объединения близких точек на карте, сегментации рынка, прогноза акций и распродаж в ретейле, мерчендайзинга. По такому принципу работает алгоритм iPhoto, который находит на фотографиях лица (не зная, чьи они) и объединяет их в альбомы.

3.3.2 Обучение с подкреплением

Это более сложный вид обучения, где ИИ нужно не просто анализировать данные, а действовать самостоятельно в реальной среде – будь то улица, дом или видеоигра. Задача робота – свести ошибки к минимуму, за что он получает возможность продолжать работу без препятствий и сбоев. Обучение с подкреплением инженеры используют для беспилотников, роботов-пылесосов, торговли на фондовом рынке, управления ресурсами компании. Именно так алгоритму AlphaGo удалось обыграть чемпиона по игре Го: просчитать все возможные комбинации, как в шахматах, здесь было невозможно.

3.3.3 Ансамбли

Это группы алгоритмов, которые используют сразу несколько методов машинного обучения и исправляют ошибки друг друга. Их получают тремя способами:

- Стекинг – когда разные алгоритмы обучают по отдельности, а потом передают их результаты на вход последнему, который и принимает решение;
- Беггинг – когда один алгоритм многократно обучают на случайных выборках, а потом усредняют ответы;
- Бустинг – когда алгоритмы обучают последовательно, при этом каждый обращает особое внимание на ошибки предыдущего.

Ансамбли работают в поисковых системах, компьютерном зрении, распознавании лиц и других объектов.

3.3.4 Нейронные сети и глубокое обучение

Самый сложный уровень обучения ИИ. Нейросети моделируют работу человеческого мозга, который состоит из нейронов, постоянно формирующих

между собой новые связи. Очень условно можно определить их как сеть со множеством входов и одним выходом.

Нейроны образуют слои, через которые последовательно проходит сигнал. Все это соединено нейронными связями – каналами, по которым передаются данные. У каждого канала свой «вес» – параметр, который влияет на данные, которые он передает.

ИИ собирает данные со всех входов, оценивая их вес по заданным параметрами, затем выполняет нужное действие и выдает результат. Сначала он получается случайным, но затем через множество циклов становится все более точным.

Хорошо обученная нейросеть работает, как обычный алгоритм или точнее. Настоящим прорывом в этой области стало *глубокое обучение*, которое обучает нейросети на нескольких уровнях абстракций.

Здесь используют две главных архитектуры:

- **Сверточные нейросети** первыми научились распознавать неразмеченные

изображения – самые сложные объекты для ИИ. Для этого они разбивают их на блоки, определяют в каждом доминирующие линии и сравнивают с другими изображениями нужного объекта;

- **Рекуррентные нейросети** отвечают за распознавание текста и речи. Они

выявляют в них последовательности и связывают каждую единицу – букву или звук – с остальными.

Нейросети с глубоким обучением требуют огромных массивов данных и технических ресурсов. Именно они лежат в основе машинного перевода, чат-ботов и голосовых помощников, создают музыку и дипфейки, обрабатывают фото и видео [10].

3.4 Классы задач

Регрессия – это прогнозирование числового значения на основе выборки объектов с различными признаками. Например, оценка платёжеспособности заёмщика, ожидаемого дохода компании или цены квартиры на рынке недвижимости.

Классификация – отнесение объектов на основе имеющихся параметров к одному из предопределённых классов. В рамках работы “Центра изучения и сетевого мониторинга молодёжи” именно качественная классификация помогает выявить деструктивный контент среди текстовых или визуальных объектов. Ежедневно благодаря машинному обучению анализируется более миллиона изображений и текстов.

Кластеризация – объединение похожих данных в группы (кластеры). Например, поиск сообществ, похожих по контенту, или объединение схожих по смыслу постов в социальной сети.

Прогнозирование временного ряда – работа с данными, полученными в определённый период времени, и предсказание на их основе значений в задаваемый исследуемый период. Решение этой задачи позволяет спрогнозировать сейсмическую активность или изменение стоимости ценных бумаг.

Также существуют вспомогательные задачи, которые можно решить с помощью машинного обучения – распознавание текста на изображениях, детекция символов, идентификация речи и так далее [8].

3.5 Применение в разработке чат-ботов

Проанализировав основные направления методов машинного обучения, можно сделать вывод, что они все могут быть использованы при разработке чат-ботов.

Обработка естественного языка (NLP) используется для того, чтобы чат-боты могли понимать, интерпретировать и генерировать человекоподобную речь. Методы NLP включают в себя моделирование языка, анализ настроений, распознавание сущностей и тегирование частей речи.

Алгоритмы контролируемого обучения используются для обучения чат-ботов понимать и реагировать на пользовательский ввод. При контролируемом обучении виртуальный ассистент тренируется на наборе данных, содержащем маркированные примеры пользовательских вводов и соответствующих ответов. Чат-бот учится определять закономерности в данных и обобщать их, чтобы делать прогнозы по новым входным данным.

Алгоритмы обучения без учителя могут использоваться для группировки похожих пользовательских данных и объединения их в категории. Это может помочь чат-ботам определить общие темы и вопросы, которые интересуют пользователей.

Алгоритмы обучения с подкреплением могут использоваться для обучения чат-ботов на основе обратной связи. При обучении с подкреплением помощник получает вознаграждение за действия, которые приводят к желаемому результату, и наказание за действия, которые приводят к нежелательному результату. Это позволяет чат-боту учиться методом проб и ошибок и со временем улучшать свою работу.

В целом, чат-боты опираются на сочетание методов машинного обучения, чтобы понимать и реагировать на входные данные пользователя подобно человеку.

3.6 Языковое моделирование

Языковое моделирование (Language Modeling) – это использование различных статистических и вероятностных методов для определения вероятности появления определенной последовательности слов в предложении.

Языковая модель – это тип модели машинного обучения, обученной проводить распределение вероятностей по словам. Проще говоря, модель пытается предсказать следующее наиболее подходящее слово для заполнения пробела в предложении или фразе, исходя из контекста данного текста. Языковые модели анализируют массивы текстовых данных, чтобы обеспечить основу для своих предсказаний слов. Они используются в приложениях обработки естественного языка (NLP), особенно в тех, которые генерируют текст в качестве вывода. Некоторые из этих приложений включают в себя, машинный перевод и ответы на вопросы [14], [15].

3.7 Нейронные сети LSTM и Transformer

Нейронные сети LSTM и Transformer – это два из самых популярных типов нейронных сетей, которые используются для обработки последовательных данных. LSTM была разработана для решения проблемы затухания и взрыва градиента в рекуррентных нейронных сетях, а Transformer – для обработки последовательностей с использованием механизма внимания.

LSTM (Long Short-Term Memory) – это архитектура рекуррентной нейронной сети, которая позволяет сохранять долгосрочные зависимости в данных. Она была разработана в 1997 году Хохрайтером и Шмидхубером.

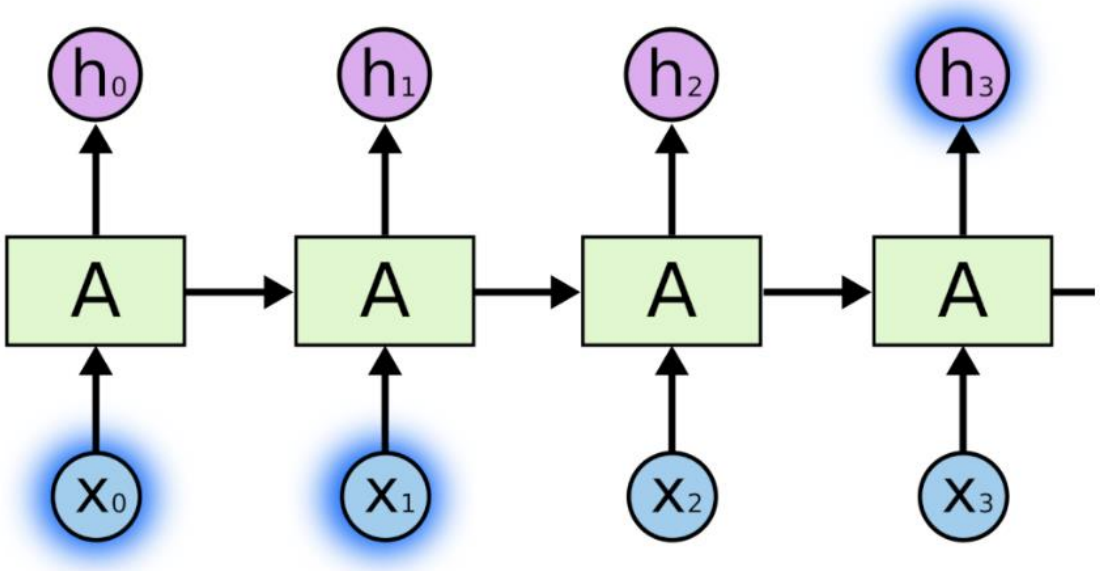


Рисунок 6 – Развертка цикла в рекуррентных нейронных сетях

Все рекуррентные нейронные сети имеют форму цепочки повторяющихся модулей нейронной сети. В стандартных РНС этот повторяющийся модуль имеет простую структуру, например, один слой **tanh**.

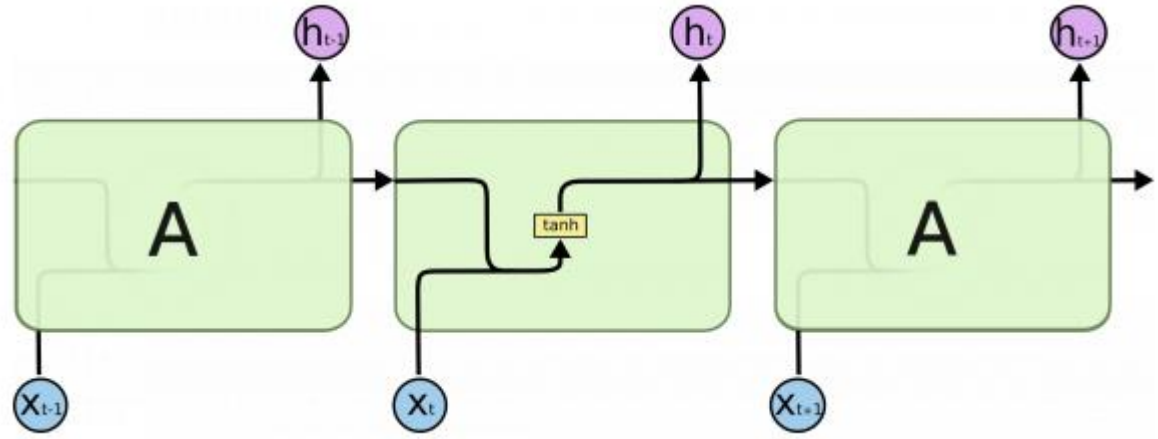


Рисунок 7 – Повторяющийся модуль стандартной РНС, состоящий из одного слоя

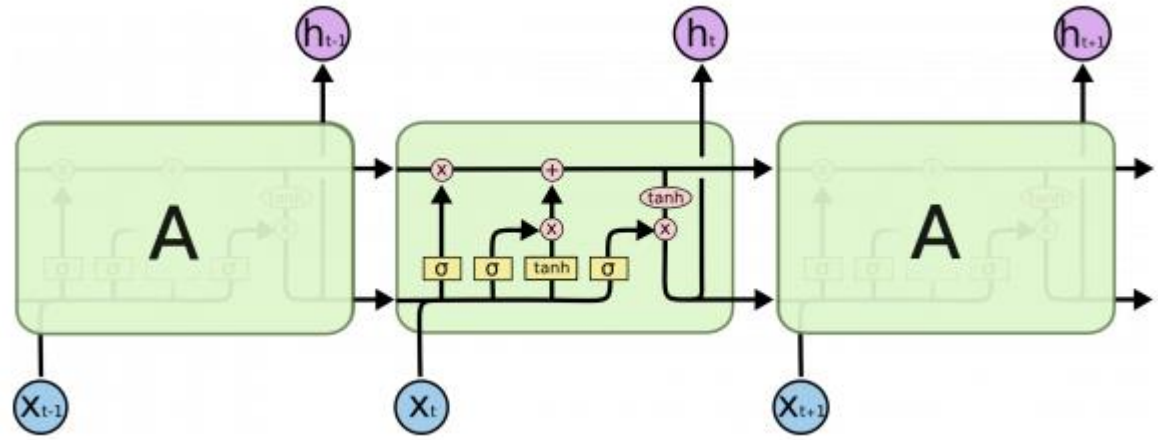


Рисунок 8 – Повторяющийся модуль LSTM, состоящий из четырех слоев

Основная идея LSTM заключается в использовании специальных блоков памяти, которые могут добавлять или удалять информацию в зависимости от ситуации. Каждый блок памяти состоит из трех компонентов: забывания, входа и выхода.

Алгоритм работы LSTM начинается с ввода входных данных в сеть. Далее данные поступают на входной уровень, где они проходят через ряд слоев, в каждом из которых происходит обработка. В случае LSTM, наиболее важным

является блок памяти, который определяет, какую информацию следует сохранить и какую следует забыть.

Входной слой сети принимает информацию от предыдущего временного шага и текущего входа. Затем эта информация проходит через четыре уровня, каждый из которых выполняет определенные функции: забывание, добавление новой информации, обновление состояния памяти и вывод.

Забывание осуществляется за счет использования сигмоидальной функции, которая решает, какую информацию нужно забыть. Затем выполняется процесс добавления новой информации, который определяет, какую информацию нужно сохранить. Для этого используется гиперболический тангенс, который возвращает новую информацию, которую необходимо добавить в память.

Обновление состояния памяти осуществляется путем использования ранее полученных результатов и новых данных, которые были приняты на входном уровне. Наконец, вывод позволяет выбрать, какую информацию нужно передать на следующий временной шаг.

Нейросеть Transformer – это одна из наиболее популярных архитектур глубокого обучения, используемых для обработки последовательностей данных, таких как тексты или звуковые сигналы. Общий алгоритм работы Transformer состоит из двух частей: энкодера и декодера.

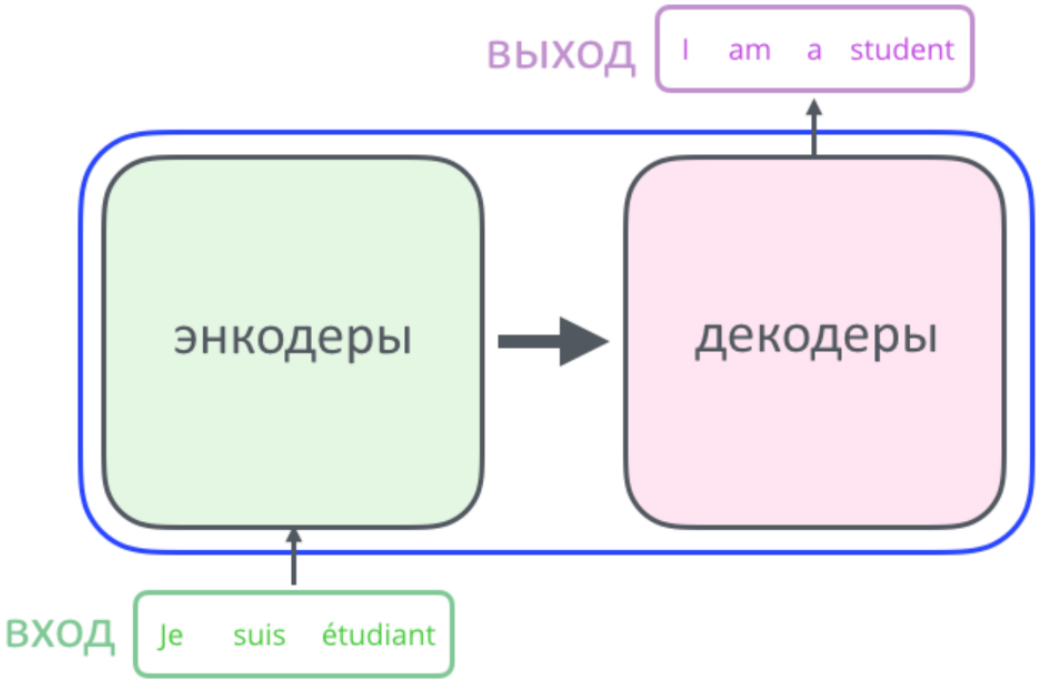


Рисунок 9 – Общий алгоритм работы трансформера

Энкодер в нейросети Transformer работает следующим образом. На вход энкодеру подается последовательность данных, которую необходимо обработать. Сначала каждый элемент последовательности преобразуется в вектор фиксированной размерности, называемый эмбеддингом. Затем эти эмбеддинги проходят через Positional Encoding – метод, используемый для добавления информации о позиции каждого элемента входной последовательности. Далее данные проходят несколько слоев нейросети, которые последовательно вычисляют некоторые преобразования. На каждом слое используется механизм внимания, который позволяет энкодеру фокусироваться на наиболее важных элементах последовательности. На выходе энкодера получается набор векторов, которые содержат информацию о каждом элементе последовательности.

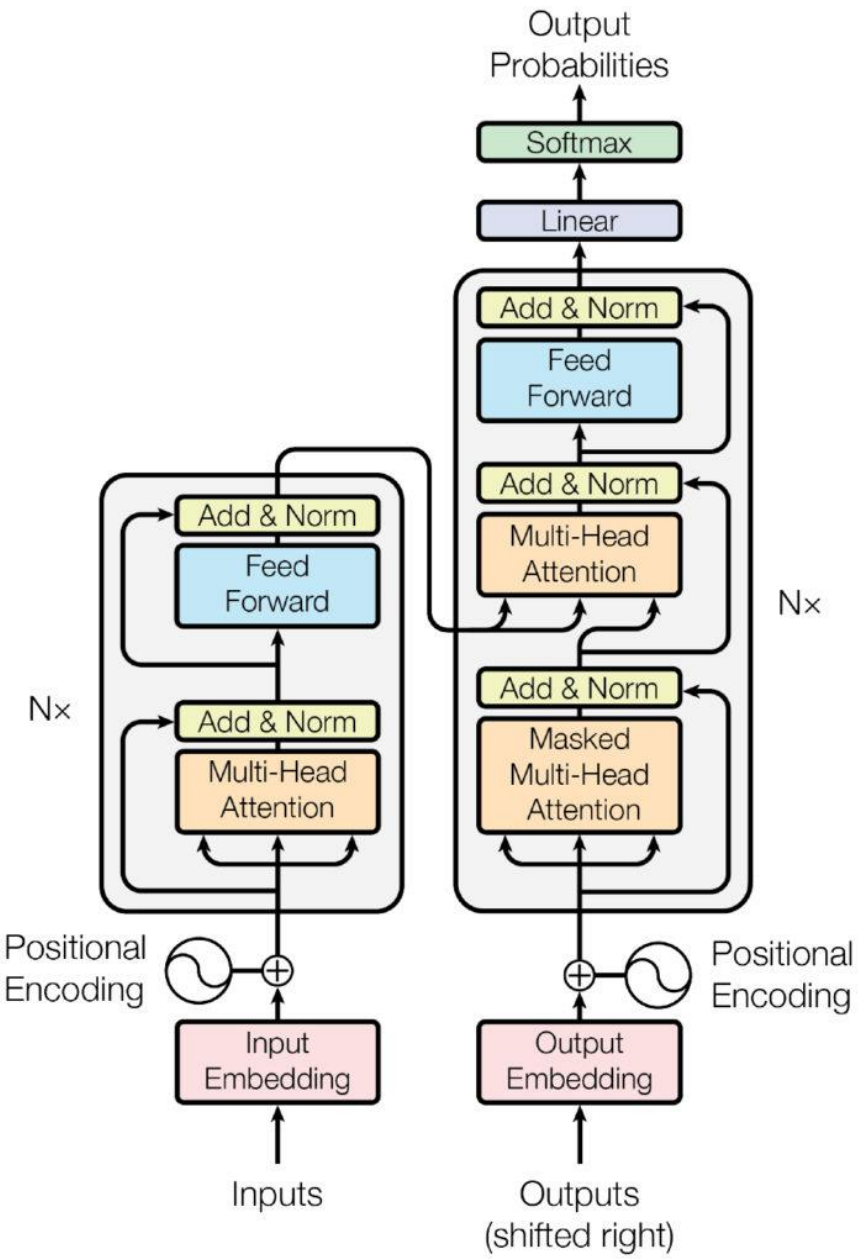


Рисунок 10 – Архитектура трансформера

Внимание в нейросети Transformer представляет собой механизм, который позволяет энкодеру или декодеру обращаться к определенным элементам последовательности, наиболее важным для решения задачи. В частности, внимание позволяет нейросети определять важность каждого элемента последовательности и вычислять взвешенные суммы этих элементов. Механизм внимания включает в себя несколько слоев, в каждом из которых вычисляются веса, отображающие важность каждого элемента последовательности.

Декодер в нейросети Transformer работает следующим образом. На вход декодеру также подается последовательность данных, но в отличие от энкодера,

декодер получает еще и выход энкодера, который содержит информацию о важности каждого элемента входной последовательности. Декодер постепенно генерирует выходную последовательность, элемент за элементом. На каждом шаге декодер использует механизм внимания, чтобы определить, на какие элементы входной последовательности следует сосредоточиться для генерации очередного элемента выходной последовательности. По мере генерации каждого элемента, декодер получает все больше информации об исходной последовательности и использует ее для генерации последующих элементов.

Основное преимущество Transformer заключается в его способности использовать параллельную обработку для ускорения вычислений, что делает его особенно полезным для больших наборов данных и высокопроизводительных приложений.

В целом, как LSTM, так и Transformer являются мощными инструментами для обработки последовательных данных. Они используются для решения широкого спектра задач в различных областях, включая естественный язык, компьютерное зрение и анализ данных.

3.8 GPT

GPT расшифровывается как генеративный предварительно-обученный трансформер (GPT) – это тип модели машинного обучения, используемой для задач обработки естественного языка. Эти модели предварительно обучаются на больших объемах данных, таких как книги и веб-страницы, для создания контекстуально релевантного и семантически связного языка.

В основе языковой модели GPT лежит несколько блоков нейронной сети архитектуры Transformer.

Начнем с того, что генеративные модели получают в качестве входных данных какое-то количество токенов, и создают один токен в качестве выходных данных (рис. 11).

Рисунок 11 – Входные и выходные данные генеративной модели

Это кажется довольно простой концепцией, но, чтобы понять ее, нам нужно знать, что такое токен. Токен – это фрагмент текста. В контексте моделей GPT компании OpenAI общие и короткие слова обычно соответствуют одному токену, например, слово “Мы” на рисунке 12. Длинные и редко используемые слова обычно разбиваются на несколько токенов. Например, слово “антропоморфизация” на рисунке 2 разбито на три токена. Аббревиатуры, такие как “ChatGPT”, могут быть представлены одним токеном или разбиты на несколько, в зависимости от того, насколько часто буквы появляются вместе.

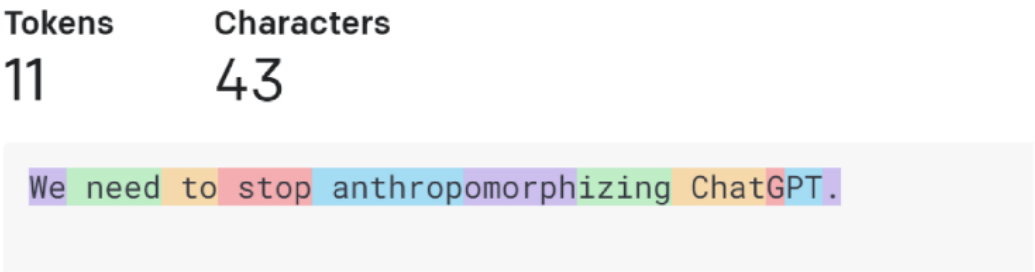


Рисунок 12 – Пример разделения на токены

3.9 Сравнение GPT-3 и GPT-4

Для сравнения моделей GPT-3 и GPT-4 опишем характеристики каждой. Модель GPT-3 была выпущена компанией OpenAI в 2020 году. Имея 175 миллиардов параметров, GPT-3 более чем в 100 раз больше, чем GPT-1, и более чем в десять раз больше, чем GPT-2.

GPT-3 обучается на различных источниках данных, включая BookCorpus, Common Crawl и Wikipedia. Наборы данных содержат почти триллион слов, что позволяет GPT-3 генерировать сложные ответы на широкий спектр задач NLP, даже без предоставления каких-либо предварительных данных.

Одним из основных улучшений GPT-3 по сравнению с предыдущими моделями является его способность генерировать связный текст, писать компьютерный код и даже создавать произведения искусства. В отличие от предыдущих моделей, GPT-3 понимает контекст данного текста и может генерировать соответствующие ответы. Возможность создавать естественно звучащий текст имеет огромное значение для таких приложений, как чат-боты, создание контента и языковой перевод. Одним из таких примеров является ChatGPT, диалоговый бот с искусственным интеллектом, который почти за одну ночь превратился из безвестности в известность.

Хотя GPT-3 может делать невероятные вещи, у него все же есть недостатки. Например, модель может возвращать предвзятые, неточные или неуместные ответы. Эта проблема возникает из-за того, что GPT-3 обучается на большом количестве текста, который может содержать предвзятую и неточную информацию. Также бывают случаи, когда модель генерирует совершенно нерелевантный текст для подсказки, что указывает на то, что модель все еще испытывает трудности с пониманием контекста и фоновых знаний.

Возможности GPT-3 также вызвали озабоченность по поводу этических последствий и потенциального неправильного использования таких мощных языковых моделей. Эксперты обеспокоены возможностью использования модели в злонамеренных целях, таких как создание поддельных новостей, фишинговых писем и вредоносного ПО. Действительно, мы уже видели, как преступники используют ChatGPT для создания вредоносных программ.

OpenAI также выпустила улучшенную версию GPT-3, GPT-3.5, до официального запуска GPT-4.

GPT-4 – последняя модель в серии GPT, выпущенная 14 марта 2023 года. Это значительный шаг вперед по сравнению с предыдущей моделью GPT-3,

которая уже производила впечатление. Хотя особенности обучающих данных и архитектуры модели официально не объявлены, она, безусловно, опирается на сильные стороны GPT-3 и преодолевает некоторые из ее ограничений.

Выдающейся особенностью GPT-4 являются его мультимодальные возможности. Это означает, что модель теперь может принимать изображение в качестве входных данных и понимать его как текстовую подсказку. Например, во время прямой трансляции запуска GPT-4 инженер OpenAI передал модели изображение нарисованного от руки макета веб-сайта, и модель неожиданно предоставила рабочий код для веб-сайта.

Модель также лучше понимает сложные подсказки и демонстрирует производительность на уровне человека в нескольких профессиональных и традиционных тестах. Кроме того, у него больше окно контекста и размер контекста, который относится к данным, которые модель может сохранить в своей памяти во время сеанса чата.

GPT-4 раздвигает границы того, что в настоящее время возможно с помощью инструментов ИИ, и, вероятно, найдет применение в самых разных отраслях. Однако, как и в случае с любой мощной технологией, существуют опасения по поводу потенциального неправильного использования и этических последствий такого мощного инструмента [16].

На основе этих данных можно сделать сравнение моделей:

1) Параметры:

- GPT-3: 175 миллиардов параметров.
- GPT-4: Почти триллион параметров.

Модель GPT-4 имеет значительно больше параметров, что делает его более продвинутым и потенциально более точным, и быстрым по сравнению с GPT-3.

2) Размер набора данных:

- GPT-3: 17 гигабайт обучающих данных.
- GPT-4: 45 гигабайт обучающих данных.

Модель GPT-4 имеет больше данных для обучения, что может привести к более точным результатам по сравнению с GPT-3.

3) Характеристики/производительность:

- GPT-3: Может выполнять задачи обработки естественного языка, создавать тексты и имеет некоторые ограничения в интерпретации идиоматических выражений и сарказма.
- GPT-4: Ожидается, что сможет выполнять более сложные задачи, такие как написание эссе и статей, создание музыки и произведений искусства. Ожидается, что он устранит недостатки предыдущих моделей, такие как понимание сарказма и идиоматических выражений.

Модель GPT-4 предположительно будет обладать лучшей производительностью и расширенными возможностями по сравнению с GPT-3.

В результате можно сделать общий вывод о том, что GPT-4, с его большим количеством параметров, большим набором данных и ожидаемыми улучшениями в производительности и функциональности, может быть значительным шагом вперед по сравнению с GPT-3. Он предоставит более точные результаты и будет иметь больше потенциальных применений в различных областях.

4 Чат-боты в банковской сфере

4.1 Возможные варианты использования

В сегодняшней тенденции автоматизации банковский мир постепенно ориентируется на самообслуживание, чтобы удовлетворить потребности и требования клиентов, разбирающихся в цифровых технологиях. Таким образом, включение чат-ботов в финансовую отрасль является замечательным явлением, которое в значительной степени снижает общую банковскую задачу. С помощью чат-бота клиенты банка могут без особых хлопот совершать любые финансовые операции с помощью текстового или голосового сообщения.

Какое отношение банк имеет к чат-ботам? Ответ довольно прост: для автоматизации сервисов. Как видите, сервисы в наши дни работают довольно медленно и иногда даже неприятно, поскольку люди относительно более склонны к непониманию и ошибкам, чем компьютерные программы.

Таким образом, диалоговый чат-бот может помочь вам обеспечить исключительное обслуживание клиентов, поскольку он доступен 24/7, никогда ничего не забывает, никогда не болеет и никогда не становится непродуктивным. Виртуальный помощник для банков может быть установлен для выполнения повседневных операций и повышения качества обслуживания клиентов в секторе цифрового банкинга [5].

На рисунке 13 приведены некоторые примеры использования чат-ботов в банковской сфере:

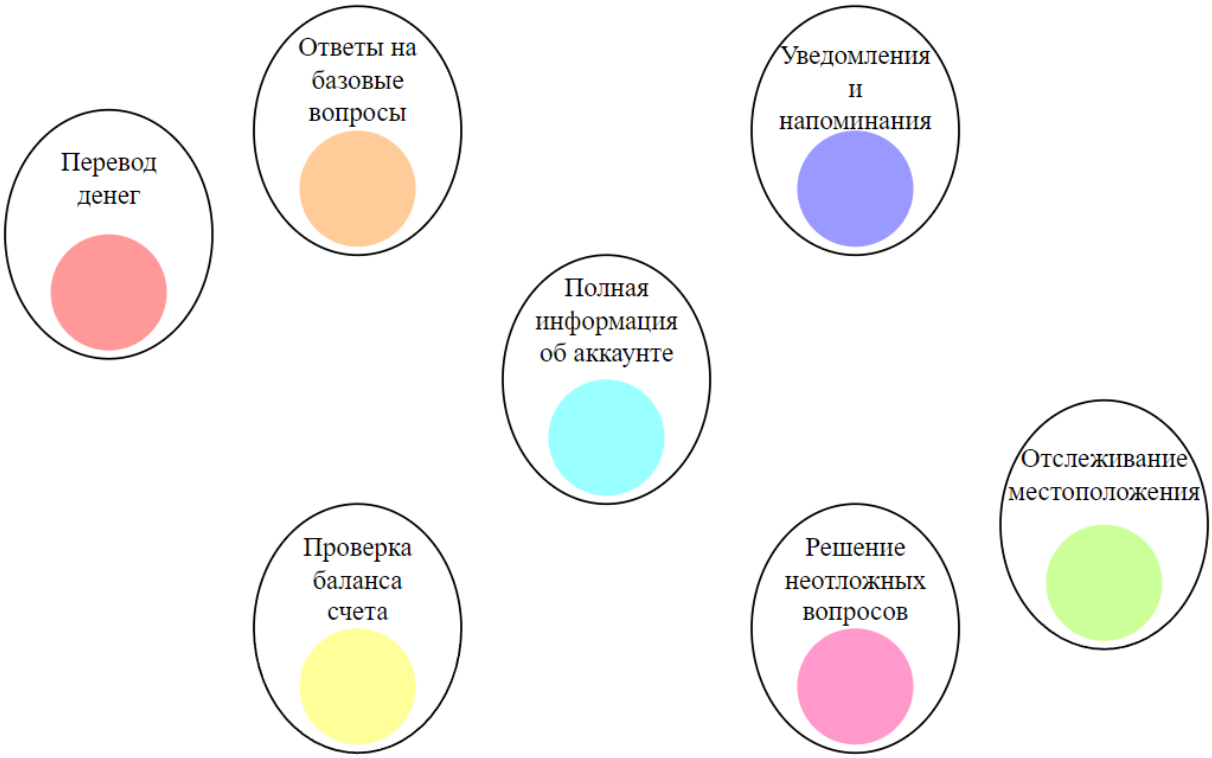


Рисунок 13 – Направления использования чат-бота в банкинге

Более подробно про каждое из направлений использования:

- Перевод денег. Пользователи могут использовать чат-ботов для быстрого перевода денег, написав ему всего одну фразу, к примеру “переведи

2000 Петру Иванову”. Также можно просить компьютерного ассистента отменить какую-либо транзакцию и т.д.;

- Ответы на базовые вопросы. Чат-боты могут отвечать на разные фундаментальные вопросы, касающихся счетов клиентов или банковских продуктов. Например, они могут отвечать на такие вопросы, как “Как я могу подать заявку на получение кредитной карты?”;
- Уведомления и напоминания. Большинство банков используют чат-ботов, чтобы отправлять своим клиентам своевременные напоминания и регулярные уведомления об их банковских счетах. Некоторые из частых напоминаний, которые часто получают клиенты, касаются сроков оплаты счетов, предложения кредита в последний день и так далее. Все эти напоминания предназначены для того, чтобы информировать клиентов обо всех действиях, которые могут принести им пользу, и оставаться с ними;
- Проверка баланса счета. Пользователи могут попросить чат-ботов предоставить им информацию о балансе счета под своим именем;
- Предоставить полную информацию. Помимо остатка на счете, пользователи также могут запрашивать другие детали счетов, такие как регулярные платежи и расходы, бонусные баллы по карте и лимиты денежных переводов. Можно также восстановить данные своей учетной записи и внести изменения, такие как обновление текущего адреса или номера телефона;
- Отслеживание местоположения в режиме реального времени. В зависимости от местоположения ответы на вопросы пользователей могут различаться. Например, если пользователь спросит: “Где ближайшее отделение банка?” В этом случае чат-бот будет отвечать в зависимости от местоположения пользователя. Кроме того, чат-боты могут отслеживать местоположение с помощью мобильного GPS, тем самым каждый раз давая правильные ответы;
- Решать неотложные вопросы в приоритете. Чат-боты в банковской сфере могут помочь клиентам с проблемами, которые могут быть несложными, но срочными. Эти проблемы включают разблокировку или блокировку карт, сброс, проверку банковских выписок и выполнение денежных переводов. Чат-

бот с искусственным интеллектом позволяет клиентам завершить весь процесс, не дожидаясь ответа по телефону.

4.2 Будущее чат-ботов в банковской сфере

Доля банков, использующих ИИ-решения и, в частности, чат-ботов, постоянно растет. В качестве еще одного фактора, использование смартфонов и других интеллектуальных устройств также является быстро растущей тенденцией. Эти две движущие силы определяют ближайшее будущее помощников искусственного интеллекта в банковской сфере.

Качество чат-ботов определенно улучшится в ближайшие несколько лет. Они станут более «человечными» и научатся гораздо лучше интерпретировать просьбы. В качестве дальнейшего развития чат-боты будут более точно предсказывать поведение человека и использовать эту информацию для самообучения.

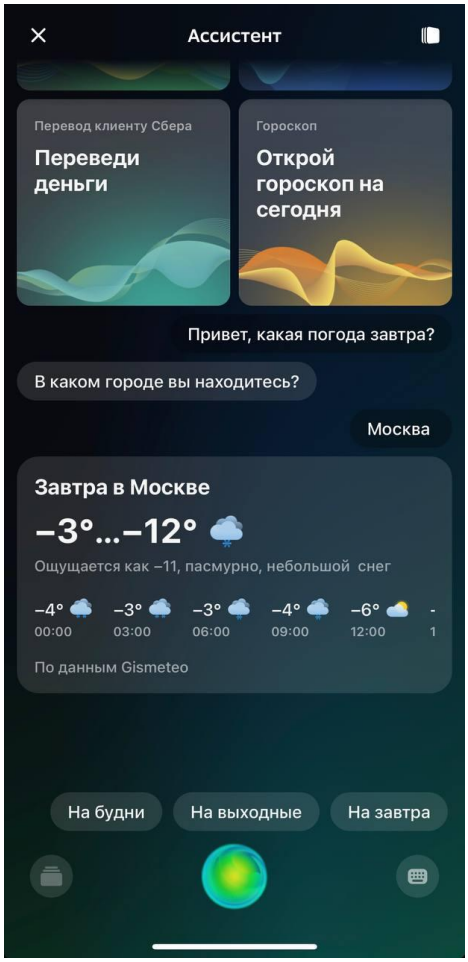
5 Обзор существующих банковских решений

В банковской индустрии уже есть чат-бот решения. Мы рассмотрим виртуальных ассистентов, которые существуют в мобильных приложениях ведущих банков России. Для оценки чат-ботов будет руководствоваться следующими критериями:

- возможность общаться в свободной форме;
- умение переключаться между тематиками без потери контекста;
- возможность перейти на оператора по запросу или после ошибки чат-бота;
- наличие кнопок-подсказок в чате, ускоряющих консультацию.

5.1 Сбер Банк

Сбер Банк – крупнейший универсальный банк России и Восточной Европы. У него есть мобильное приложение, в котором можно найти умного чат-бота. Проанализировав компьютерного помощника, было определено, что это не узкоспециализированный банковский ассистент, который сможет помочь только с банковскими вопросами, а полноценный помощник, способный подобрать билеты в кино, поставить будильник на утро, подобрать выгодный вклад и посоветовать хороший кредит.



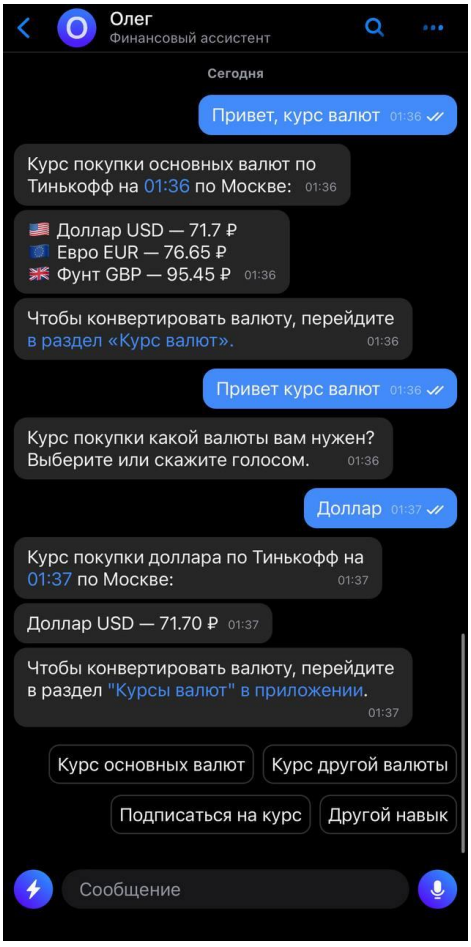
Рисунки 14-15 – Взаимодействие с виртуальным ассистентом Сбер Банка

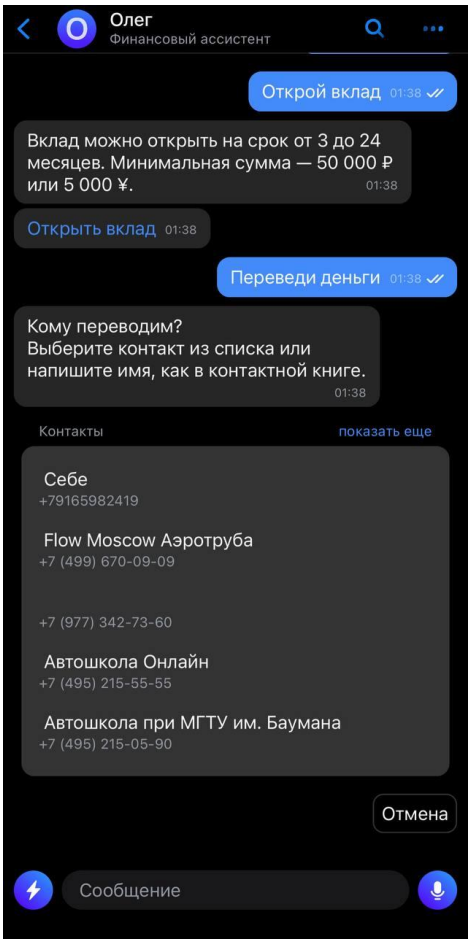
Ему можно отправить как текстовое сообщение, так и произнести команду голосом. Написав несколько команд, можно сделать вывод, что он очень хорошо определяет контекст запроса и предлагает нужные решения. Если попытаться его запутать или написать с ошибкой – он все равно поймет, что вы имели ввиду.

При попытке, к примеру, оформить вклад по рекомендации чат-бота, он открывает действие открытия вклада, как если бы вы нашли кнопку для этого действия в приложении без помощи ассистента. Это сделано для того, чтобы человек осознанно и самостоятельно производил операцию.

5.2 Тинькофф Банк

Тинькофф Банк – российский коммерческий банк, сфокусированный полностью на дистанционном обслуживании, не имеющий розничных отделений. Он считается крупнейшим в мире онлайн-банком по количеству клиентов. В его банковском приложении во вкладке “Чат” можно найти виртуального помощника Олега. Если обратиться к нему, вы получите ответы только на вопросы, связанные с банковской тематикой.



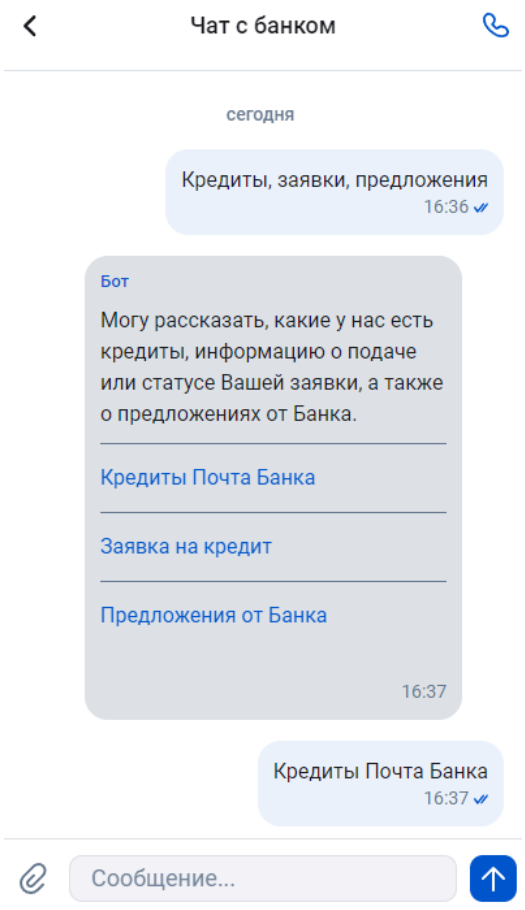


Рисунки 16-17 – Общение с чат-ботом Тинькофф Банка

Этот ассистент сможет рассказать вам о кредитах, вкладах или сообщит курс доллара к рублю. Если сравнивать его возможности с предыдущим конкурентом – ботом в Сбер Банке, то они не такие широкие. По оценочным критериям, приведенным выше, можно сделать вывод, что чат-бот Олег не очень способен общаться в свободной форме, однако имеет различные кнопки-подсказки, которые ускоряют консультацию.

5.3 Почта Банк

Почта Банк – универсальный розничный банк, созданный в 2016 году группой ВТБ и Почтой России. Ключевая цель Почта Банка — повышение доступности финансовых услуг для жителей России. Чат-бота этого банка можно найти в мобильном приложении или на сайте. Во время общения с виртуальным помощником Дмитрием, пользователь должен выбирать фразы из перечня предложенных ботом (рис. 9-10).



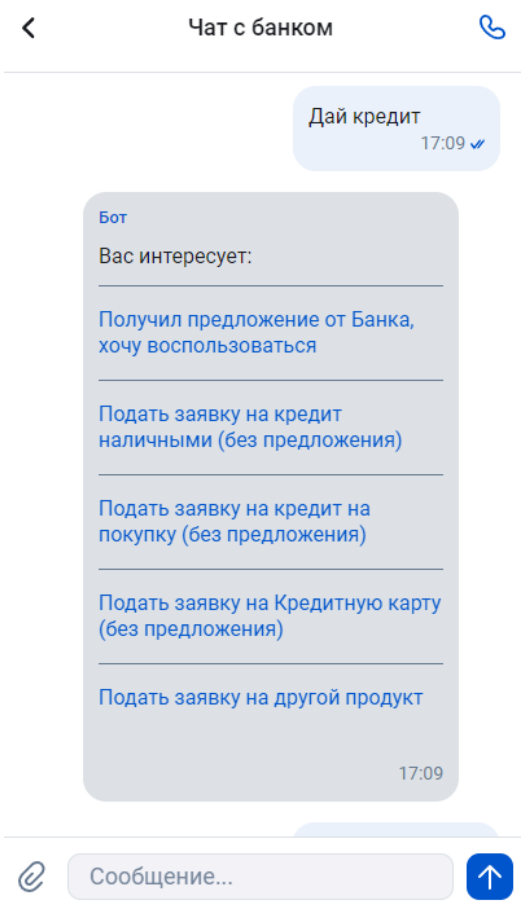


Рисунок 18-19 – Общение с виртуальным ассистентом Почта Банка

Чат-бот этого банка хорошо распознает намерение пользователя и предлагает наиболее полезные возможности. Он понимает свободную форму общения, не только при помощи кнопок. Помощник Дмитрий переключает вас к оператору, если он не понял сообщения от пользователя или клиента не устроило ничего из предложенных вариантов.

6 Выбор подхода к разработке

Первый вопрос, которым мы зададимся, заключается в том, следует ли нам создавать “с нуля” и предварительно обучать LLM самостоятельно или использовать существующий. Существует три основных подхода:

Вариант 1: Использовать API коммерческого LLM, например GPT-3 от компании OpenAI.

Вариант 2: Использовать существующую LLM с открытым исходным кодом, например GPT-J.

Вариант 3: Предварительное обучение LLM самостоятельно.

В таблице 1 приведены плюсы и минусы каждого из вариантов.

Таблица 1 – Плюсы и минусы подходов к внедрению языковой модели в алгоритм чат-бота

Вариант 1	
Плюсы	Минусы
• Требует минимум технических навыков, необходимых для получения степени магистра права; • Минимальное предварительное обучение/стоимость; • Наименее требовательный к данным вариант;	• Коммерческие услуги LLM могут стать дорогими из-за большого объема задач тонкой настройки или логических выводов; • Многие отрасли/варианты использования запрещают использование коммерческих служб

• Всего несколько примеров (или вообще никаких примеров) необходимы моделям для выполнения логического вывода; • Может использовать наиболее эффективные LLM на рынке и показывать превосходный результат.	LLM в качестве конфиденциальных данных; • Ограниченные возможности настройки модели (тонкая настройка становится дорогостоящей), ограниченные возможности для постоянного улучшения модели.
Вариант 2	
Плюсы	Минусы
• Хороший способ использовать то, что LLM узнал из огромного количества интернет-данных, и строить на их основе, не платя за использование;	• Не так требовательно, как создание собственного, но все же требует много навыков эксперта в предметной области для обучения, тонкой настройки и размещения LLM

Продолжение таблицы 1

Вариант 2	
Плюсы	Минусы
• По сравнению с первым вариантом мы в меньшей степени зависим от будущего направления деятельности поставщиков LLM и, таким образом, имеем больше контроля; По сравнению с третьим вариантом у нас гораздо более быстрое время окупаемости, учитывая, что мы не создаем LLM с нуля, что также приводит к сокращению объема	с открытым исходным кодом. Воспроизводимость LLM по-прежнему является серьезной проблемой, поэтому нельзя недооценивать количество необходимого времени и работы; • Модели с открытым исходным кодом обычно отстают по производительности от коммерческих моделей на месяцы/годы.

данных, времени обучения и необходимого бюджета на обучение.	
Вариант 3	
Плюсы	Минусы
• По сравнению с вариантами один и два, у вас есть максимальный контроль над производительностью вашего LLM и будущим направлением, что дает вам большую гибкость для инноваций в методах и / или адаптации к вашим последующим задачам; • Получите полный контроль над обучающими наборами данных, используемыми для	• Очень дорогое решение с высокими рисками. Нужны междисциплинарные знания, охватывающие NLP/ML, предметные знания, опыт в программном и аппаратном обеспечении. Ошибки, особенно на поздних стадиях обучения, трудно исправить; • Мы начинаем с нуля и нам нужно много высококачественных/разнообразных

Продолжение таблицы 1

Вариант 3	
Плюсы	Минусы
предварительного обучения, что напрямую влияет на качество модели. Для сравнения, эти проблемы менее контролируемы в первом или втором варианте; • Обучение вашего собственного LLM также дает вам глубокий ров: превосходная производительность LLM как в горизонтальных сценариях использования, так и с учетом вашей	наборов данных для модели, чтобы получить хорошие возможности.

вертикали, что позволяет вам создать устойчивое преимущество.	

Основываясь на преимуществах и недостатках описанных подходов, в этой работе решено сделать следующее: сначала построим прототип чата с чат-ботом с использованием готовой языковой модели по API, далее найдем языковую модель в открытом доступе и внедрим ее в наш прототип банковского чата, вместо API.

7 Прототип чата с чат-ботом

7.1 Выбор инструментов для разработки

В реализации прототипа чата для банковского приложения будем использовать готовые модели, которые предоставляет интерфейс OpenAI API. Используя этот инструмент, мы будем иметь доступ к некоторым из самых мощных языковых моделей, которые когда-либо создавались.

Однако, эти модели не могут генерировать предложения на банковскую тематику конкретного банка, так как не обучены на таких узконаправленных данных. В наши задачи входит каким-то образом научить этому используемую готовую модель.

В настоящее время существует два основных способа расширить базу знаний языковой модели:

- Тонкая настройка (fine-tuning) – это практика модификации существующей предварительно обученной языковой модели путём её обучения (под наблюдением) конкретной задаче;
- Быстрое проектирование и встраивание – дополнение входящего в языковую модель запроса информацией из собранной базы знаний. Таким образом, мы не изменяем модель, а предоставляем ей данные, на основе которых она генерирует ответ на естественном языке.

В нашей реализации мы будем использовать второй способ, так как он проще и дешевле, чем первый. Однако мы сталкиваемся с проблемой: на вход готовой языковой модели мы не можем подать всю базу знаний, которую мы имеем, так как она может быть очень большой. Решить эту задачу можно при помощи поиска и встраиванием в запрос наиболее релевантной информации. Такой механизм можно реализовать с использованием инструмента OpenAI Embeddings.

OpenAI Embeddings – это набор предобученных векторных представлений слов, которые были получены с использованием глубоких нейронных сетей на огромных корпусах текстов. Эти векторные представления слов позволяют выразить семантические отношения между словами или предложениями в числовой форме. Таким образом, преобразуя входной запрос в векторные данные, мы будем находить семантически-близкие данные из нашей базы знаний отправлять их на вход языковой модели.

Для разработки пользовательского интерфейса прототипа чата была выбрана платформа Windows Forms. Это библиотека классов .NET, которая предназначена для создания приложений с графическим интерфейсом (GUI) на языке C# или других языках программирования.

Windows Forms предоставляет разработчикам готовые элементы управления, такие как кнопки, текстовые поля, списки, таблицы и др., которые могут быть использованы для создания пользовательского интерфейса. Кроме того, Windows Forms позволяет создавать свои собственные элементы управления и кастомизировать уже существующие.

Плюсы использования Windows Forms:

- Простота создания приложений с графическим интерфейсом. Windows Forms позволяет быстро и легко создавать приложения с графическим интерфейсом без необходимости изучения сложных технологий.
- Быстрое развертывание. Приложения, созданные с помощью Windows Forms, могут быть быстро развернуты на многих платформах, таких как Windows, Linux и macOS.

- Богатый функционал. Windows Forms предоставляет широкий набор готовых элементов управления и возможностей кастомизации.

7.2 Этапы разработки

7.2.1 Подготовка данных

База знаний формировалась на основе информации на сайте Сбер Банка. Брались все текстовые данные со всех вкладок, показанных на рисунке X: Вклады, Кредиты, Кредитные карты, Дебетовые карты, Переводы и т. д. Текст сохранялся в файл построчно.

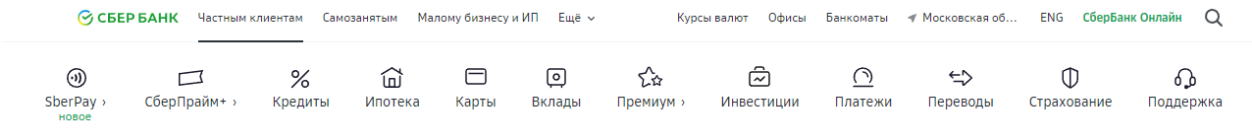


Рисунок 20 – Вкладки на сайте Сбер Банка

Информация сохранялась в файл формата .csv без разрыва по смыслу по следующему правилу: в каждой строке – абзац с сайта Сбер Банка про какую-либо услугу или ответ на какой-либо вопрос. В результате мы имеем файл с банковскими данными Сбер Банка, длиной в 349 237 символов и 471 строку. Часть этого файла представлена на рисунке 21.

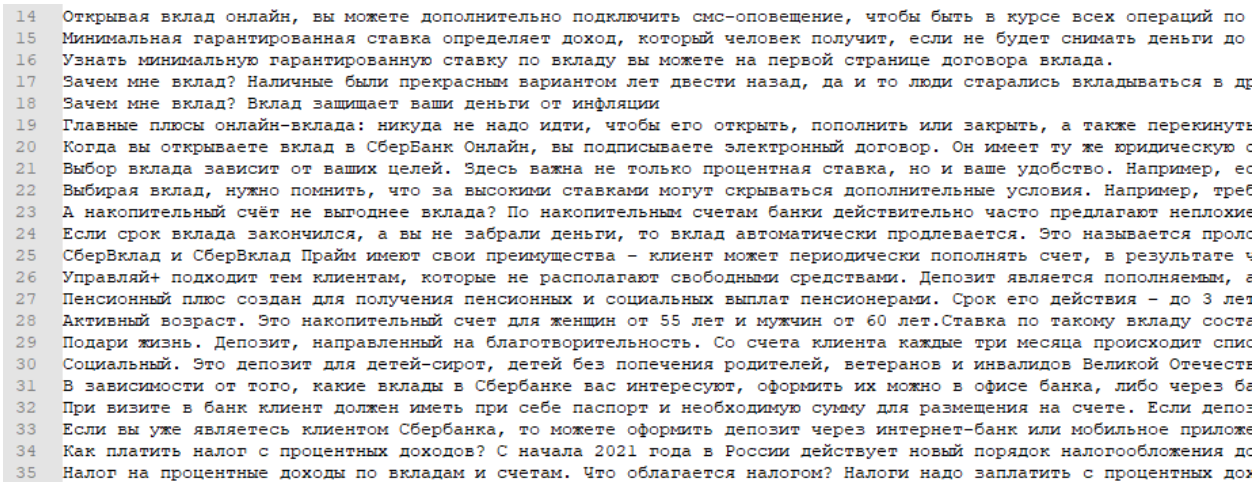


Рисунок 21 – Отрывок из файла с базой знаний

На следующем шаге мы пропускаем эти данные через модель OpenAI Embeddings для преобразования каждого предложения в числовой вектор. Для

этого я воспользовался платформой Google Collab и языком программирования python.

Импортируем необходимые библиотеки, указываем личный OpenAI Token, и определяем название модели, которую мы будем использовать [11].

```
[ ]  import openai
     from openai.embeddings_utils import get_embedding
     import pandas as pd

     openai.api_key = "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"
     # models
     EMBEDDING_MODEL = "text-embedding-ada-002"
```

Рисунок 22 – Инициализация начальных данных

Далее загружаем нашу базу знаний на платформу и передаем ее на вход модели, которая преобразует данные в соответствующие векторные представления и сохраняет все в отдельный файл, как показано на рисунке 20.

```
[ ]  sber_deposits = pd.read_csv('sber.csv', sep='|')
     sber_deposits
```

```
[ ]  sber_deposits['embedding'] = sber_deposits['text'].apply(lambda x: get_embedding(x, engine=EMBEDDING_MODEL))
     sber_deposits.to_csv('sber_data_embeddings.csv', sep='|')
     sber_deposits
```

	text	embedding
0	Где можно открыть вклад или счёт? Вклады и сче...	[-0.007171741221100092, -0.008707172237336636,...
1	Где выгоднее открывать вклады — в офисе или он...	[-0.008761981502175331, 0.0006015222170390189,...
2	Как подобрать вклад? Чтобы правильно выбрать в...	[-0.009093747474253178, -0.009491111151874065,...
3	Можно ли открыть на своё имя сразу несколько в...	[-0.009806495159864426, -0.024453753605484962,...
4	Можно ли открыть вклады 'Домклик' или 'Хорошее...	[-0.014559488743543625, -0.016061553731560707,...
...	...	...
465	Автопереводы. Как я узнаю, что мой автоперевод...	[-0.03163580223917961, -0.016078708693385124, ...
466	Автопереводы. Что произойдёт, если в день испо...	[-0.02542446367442608, -0.008891872130334377, ...
467	Автопереводы. Почему не исполняется автоперево...	[-0.021153846755623817, -0.014690170995891094,...
468	Автопереводы. Можно ли отключить автоперевод? ...	[-0.02793269231915474, -0.015137887559831142, ...
469	Автопереводы. Можно ли отменить исполненный ав...	[-0.030668094754219055, -0.029448792338371277,...

470 rows × 2 columns

Рисунок 23 – Загрузка данных на платформу, формирование векторного представления и сохранение результата в новый файл

В результате мы имеем следующий файл в формате .csv, готовый к использованию. Часть этого файла представлена на рисунке 24.

```
|text|embedding
0|Где можно открыть вклад или счёт? Вклады и счета можно открыть в личном кабинете и мобильном приложении
1|Где выгоднее открывать вклады — в офисе или онлайн? В Сбербанк Онлайн ставки по вкладам выше, чем при от
2|Как подобрать вклад? Чтобы правильно выбрать вклад, определитесь с целью, которой вы хотите достичь, и д
3|Можно ли открыть на своё имя сразу несколько вкладов или счетов? Да, вы можете открыть любое количество
4|Можно ли открыть вклады 'Домклик' или 'Хорошее начало' на имя другого человека? Да, но данные вклады дол
5|Можно ли открыть вклад сразу на несколько человек, например, семейный? Вклад открывается только на одног
6|В какой валюте можно открыть вклад или счёт? Зависит от вклада или счёта. Вклады СберВклад, СберВклад Пр
7|Что такое капитализация процентов? При капитализации проценты, начисленные за прошедший период, добавляк
8|Что такое неснижаемый остаток по вкладам? Неснижаемый остаток по вкладу – это минимальная сумма, которая
9|Какие льготы есть для пенсионеров при открытии вклада? Женщины от 55 лет и мужчины от 60 лет могут откры
10|Я гражданин другого государства, но временно проживаю в России и хочу оформить вклад. Какие документы я
11|Безопасно ли хранить деньги на вкладе, открытом онлайн? Мы заботимся о безопасности ваших денег: круглс
12|Открывая вклад онлайн, вы можете дополнительно подключить смс-оповещение, чтобы быть в курсе всех опера
13|Минимальная гарантированная ставка определяет доход, который человек получит, если не будет снимать ден
14|Узнать минимальную гарантированную ставку по вкладу вы можете на первой странице договора вклада.|[-0.0
15|Зачем мне вклад? Наличные были прекрасным вариантом лет двести назад, да и то люди старались вкладывать
16|Зачем мне вклад? Вклад защищает ваши деньги от инфляции|[-0.01802617684006691, -0.00935601256787777, -0
17|Главные плюсы онлайн-вклада: никуда не надо идти, чтобы его открыть, пополнить или закрыть, а также пер
18|Когда вы открываете вклад в СберБанк Онлайн, вы подписываете электронный договор. Он имеет ту же юридич
19|Выбор вклада зависит от ваших целей. Здесь важна не только процентная ставка, но и ваше удобство. Напри
20|Выбирая вклад, нужно помнить, что за высокими ставками могут скрываться дополнительные условия. Наприме
21|А накопительный счёт не выгоднее вклада? По накопительным счетам банки действительно часто предлагают н
22|Если срок вклада закончился, а вы не забрали деньги, то вклад автоматически продлевается. Это называетс
```

Рисунок 24 – Отрывок из предобработанного файла

Можно заметить, структура файла состоит из множества строк и трех колонок: Индекс, Текстовые данные и Векторное представление.

7.2.2 Реализация пользовательского интерфейса

С использованием среды разработки Visual Studio 2022 был создан проект на основе платформы .NET 6.0 Windows Forms.

В редакторе была сделана форма прототипа чата для банковского приложения. Для этого на главное окно пользовательского интерфейса были установлены следующие элементы управления: два текстовых поля и две кнопки (рисунок 25).

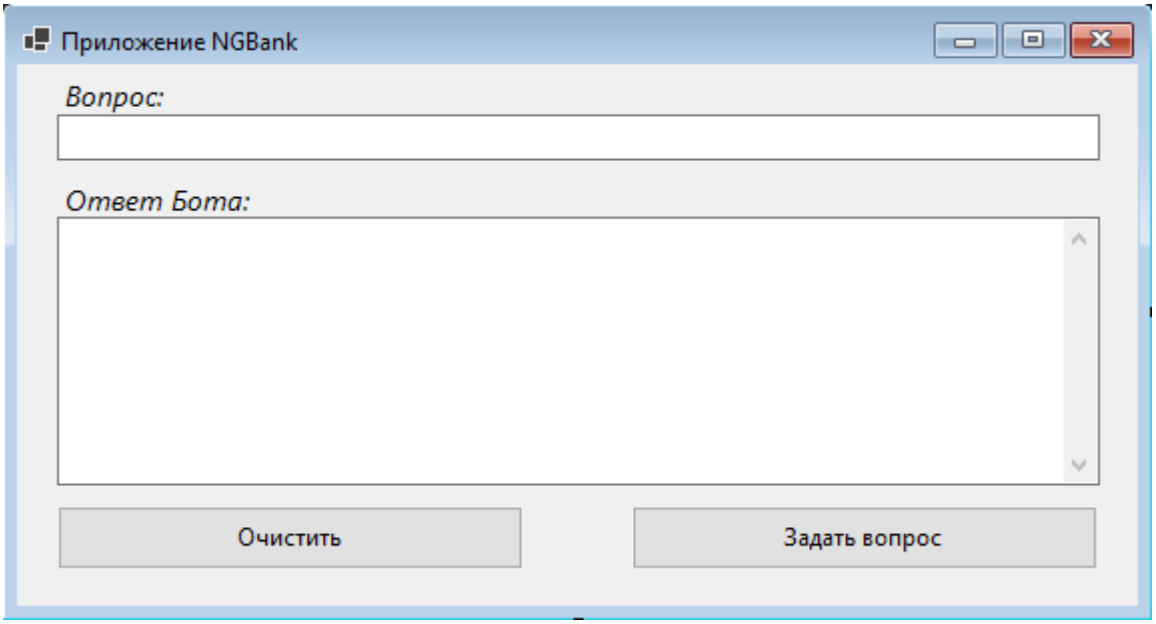


Рисунок 25 – Форма главного окна прототипа

В классе созданной формы описаны соответствующие события, которые возникают при взаимодействии с формой. А именно события нажатия на кнопки. При нажатии на кнопку “Очистить”, стираются данные из всех текстовых полей на форме. При нажатии на кнопку “Задать вопрос”, поле “Вопрос” очищается, выполняется вызов метода объекта чата, который формирует ответ и выводит его в текстовое поле “Ответ бота”.

7.2.3 Логика поиска в базе знаний и встраивание

Для реализации поиска наиболее релевантной информации в базе знаний и генерации ответа на входящий запрос был реализован класс ChatModel, который имеет конструктор без параметров и несколько методов.

В конструкторе выполняется начальная инициализация объекта OpenAIService, который используется для взаимодействия с готовыми моделями OpenAI.

Класс ChatModel имеет несколько методов:

- метод для загрузки предобработанного файла с данными (после п.п. 4.2.1) в объект для работы с файлами DataFrame;
- метод для формирования вспомогательного контекста, который будет отправляться языковой модели вместе с входящим запросом пользователя;
- метод для расчета расстояния между векторами, который будет вычислять расстояния между векторами данных в базе знаний и вектором входящего запроса для определения в дальнейшем семантически наиболее близкой информации. Другими словами, будет выполнять поиск наиболее релевантных данных. Сходство между данными мы будем определять при помощи косинуса угла между соответствующими векторами по формуле:

$$\mathrm{S}(\mathrm{x},\mathrm{y}) = \frac{x \cdot y}{\|x\|\|y\|} = \frac{\sum_{i=0}^{n-1} x_i y_i}{\sqrt{\sum_{i=0}^{n-1}(x_i)^2} \times \sqrt{\sum_{i=0}^{n-1}(y_i)^2}} \quad (1)$$

7.3 Тестирование и анализ решения

Для тестирования разработанного прототипа, чат-боту было задано несколько вопросов на банковскую тематику, и на любую другую тему. На все вопросы ассистент ответил правильно и справился со своей задачей. Результаты работы программы представлены в приложении А.

В результате назовем положительные и отрицательные стороны полученного решения и сделаем вывод.

Плюсы разработки:

1) Использование обученной языковой модели GPT-3.5-turbo от OpenAI позволяет получить ответы, которые могут быть более качественными и естественными в сравнении с более простыми алгоритмами чат-ботов. Модель обучена на огромном объеме текстовых данных и способна генерировать релевантные и информативные ответы на широкий спектр вопросов.

2) Встраивание дополнительного контекста из базы знаний при помощи модели Embedding в запросы к языковой модели GPT-3.5-turbo позволяет более точно формулировать вопросы и получать ответы, которые связаны с заданной темой. Это может улучшить релевантность ответов и помочь избежать несвязанных или неправильных ответов.

3) Благодаря мощности серверов компании OpenAI, мы получаем быстрые ответы на вопросы пользователей. Это особенно важно в интерактивных ситуациях, где пользователь ожидает мгновенных ответов.

Минусы разработки:

1) Один из главных минусов использования общедоступного OpenAI API заключается в том, что встроенные вспомогательные данные, а также запросы пользователей, могут быть переданы на сервера компании OpenAI. Это может вызвать опасения в отношении конфиденциальности и безопасности данных, особенно в случае, если эти данные содержат чувствительную банковскую информацию.

Разработанный прототип чата с чат-ботом на банковскую тематику, использующий модель GPT-3.5-turbo для генерации ответов и модель Embedding для поиска релевантной информации из базы знаний, имеет несколько преимуществ, таких как качественная генерация ответов и скорость генерации. Однако важно учитывать, что передача данных на сторонние серверы может быть проблемой в отношении конфиденциальности и безопасности данных. Перед дальнейшим развитием проекта рекомендуется внимательно рассмотреть вопросы безопасности и поискать способы минимизации передачи чувствительных данных на сторонние серверы.

8 Полностью локальный прототип

Многие клиенты все еще опасаются использовать онлайн-сервисы из-за возможности утечки их личных данных или несанкционированного доступа к ним.

Для избежание этого, мы избавимся от взаимодействия с OpenAI API и разработаем полностью локальную версию банковского чат-бота, который позволит клиентам задавать вопросы чат-боту, а самому виртуальному ассистенту работать без необходимости подключения к интернету. В решении будут использованы возможности LLM (Large Language Model) для создания интеллектуального ассистента, работающего исключительно внутри клиентской среды выполнения. Это означает, что все данные остаются строго конфиденциальными и никогда не покидают устройство пользователя.

8.1 Выбор инструментов для разработки

Для решения поставленного задания была использована платформа Hugging Face с коллекцией готовых моделей машинного обучения под разные задачи [12]. На этой площадке можно подобрать и использовать в своих целях модели разного размера и качества, требующие различных ресурсов. Таким

образом, используя Hugging Face, можно подобрать модель, которая будет удовлетворять почти любым требованиям. В нашем случае необходимо найти две модели:

1) Языковая модель, способная генерировать осмысленный текст на основе дополнительного контекста;

2) Embedding модель для поиска наиболее близкой по смыслу информации в базе знаний. Нужна для того, чтобы формировать дополнительный контекст на основе наиболее релевантной информации.

Использовать выбранные модели с платформы Hugging Face можно благодаря библиотеке Transformers, которой мы и будем пользоваться. способная генерировать текст. Она есть только в реализации на языке программирования Python. Он и будет выбран для написания кода программы.

Также необходимо как-то хранить нашу базу знаний, описанную в пункте 4.2.1, которая останется без изменений. Для взаимодействия с данными были выбраны следующие инструменты:

- LangChain – инструменты для анализа документа со знаниями и локального создания вложений с помощью Hugging Face Embeddings (SentenceTransformers). Затем он сохраняет результат в локальной базе данных векторов.
- Chroma – это база данных, которая используется для хранения и поиска эмбеддингов документов. Эмбеддинги представляют собой числовое представление документов, которое делает их “понятными” для модели машинного обучения. В контексте Chroma, эмбеддинги позволяют документам и запросам с одинаковым содержанием быть “близкими” друг к другу, что упрощает их поиск. Chroma поддерживает различные методы создания эмбеддингов. По умолчанию Chroma использует Sentence Transformers для создания эмбеддингов, но также можно использовать эмбеддинги от OpenAI, Cohere или создать свои собственные эмбеддинги.

8.2 Модернизация данных

По сравнению с предыдущим решением, собранные данные переведены на английский язык по причине того, что русский язык в переводе на токены языковых моделей занимает больше места. Это происходит по следующим причинам:

1) Русский язык использует кириллическую азбуку, которая имеет больше символов по сравнению с английским латинским алфавитом. Включение дополнительных символов кириллицы влечет за собой увеличение числа уникальных токенов, что увеличивает размер словаря и требует больше места для хранения;

2) Русский язык обладает более богатым набором грамматических форм и склонений, чем английский. Каждая грамматическая форма может требовать свой собственный токен, чтобы модель могла правильно учитывать грамматические особенности и синтаксис русского языка. Это также приводит к увеличению числа токенов и размера словаря для русского языка;

3) В среднем, русские слова имеют больше символов, чем английские слова. Это связано с грамматическими особенностями русского языка, такими как склонения и суффиксы. Длинные слова требуют больше символов для представления в токенах, что увеличивает размер текстовых последовательностей и, следовательно, потребление места.

В результате, использование русского языка в токенах языковых моделей может требовать больше места и ресурсов по сравнению с английским языком, из-за более широкого алфавита, богатого грамматического набора и длинных слов [13].

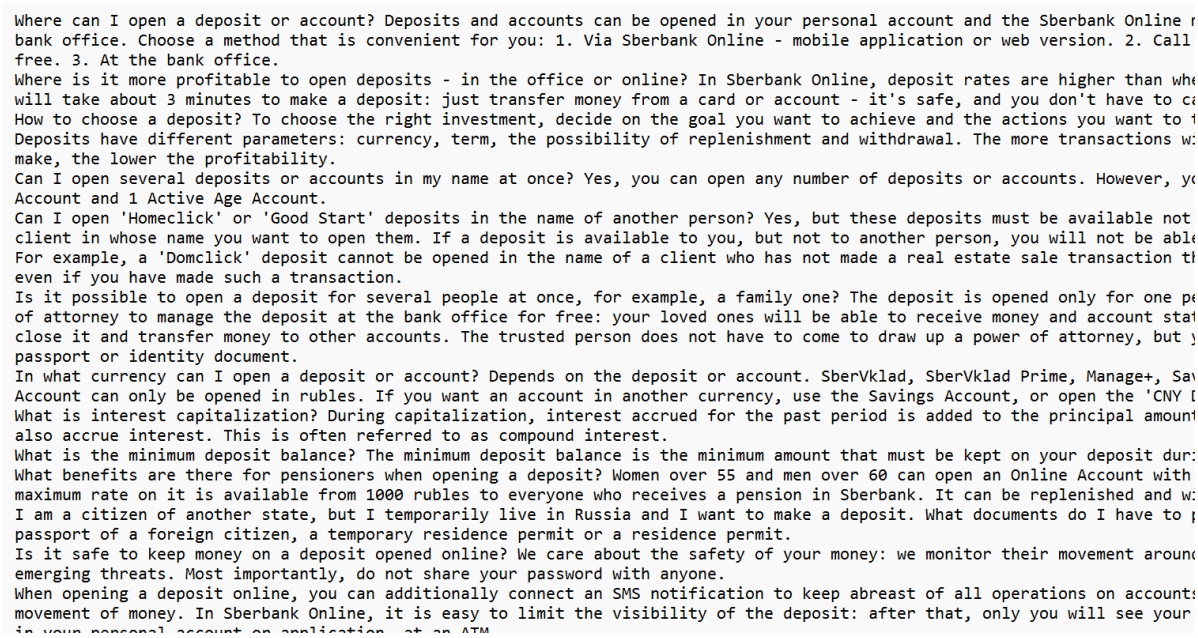


Рисунок 26 – Часть подготовленных данных на латинском языке

8.3 Алгоритм работы

Алгоритм работы этой программы почти не отличается от прошлой версии прототипа с чат-ботом. Опишем его:

Алгоритм работы следующий:

1) Программа начинает свою работу с импорта необходимых библиотек, таких как LangChain, Transformers и Chroma. Эти библиотеки обеспечивают функциональность анализа документов, создания вложений и работу с готовыми языковыми моделями;

2) Первый запуск программы начинается с анализа прикрепленных документов. Программа использует библиотеку LangChain для анализа входящих документов. LangChain обрабатывает текстовые данные и разделяет их на токены, выполняет поиск шаблонов и извлекает ключевые слова. Результаты анализа сохраняются в памяти для дальнейшей обработки;

3) Далее программа использует модель эмбеддингов для создания локальных вложений на основе обработанных текстовых данных;

4) С использованием библиотеки Chroma программа создает локальное хранилище векторов, где сохраняются полученные вложения. Хранилище векторов служит в качестве базы данных, в которой векторные представления связаны с соответствующими контекстами из документов;

5) Программа загружает локально предварительно обученную языковую модель при помощи библиотеки Transformer. Эта модель будет использоваться для понимания вопросов пользователей и генерации ответов;

6) Программа использует алгоритм для нахождения наиболее релевантного к запросу пользователя контекста из документов. Это позволяет языковой модели генерировать правдивый ответ на естественном языке;

7) На основе найденного контекста языковая модель генерирует ответ на заданный вопрос.

8.4 Программная реализация

Проект реализован в виде трех файлов и директории с документами, в которых хранятся данные. Первые два файла – это программы на языке программирования Python. В первом файле ingest.py реализована логика для подготовки векторного пространства для дальнейшего поиска в нем наиболее релевантной информации. Второй файл privateGPT.py отвечает за основую часть работы программы, а именно взаимодействие с загруженной языковой моделью. Третий файл называется .env. В нем собраны основные глобальные переменные, которые используются в описанных программах.

Во время первого запуска файла ingest.py необходимо подключение к сети Интернет, так как выполняется загрузка указанной эмбеддинговой модели. При дальнейших активациях этого файла не требуется подключение к сети Интернет. После загрузки модели или проверки ее наличия, создается директория database, куда помещается обработанный документ из директории source_documents в векторном виде. Реализация не имеет ограничений по количеству документов.

Все загруженные знания будут собраны в локальной базе данных вложений. Во время загрузки никакие данные не покидают локальную среду (рис. 27).

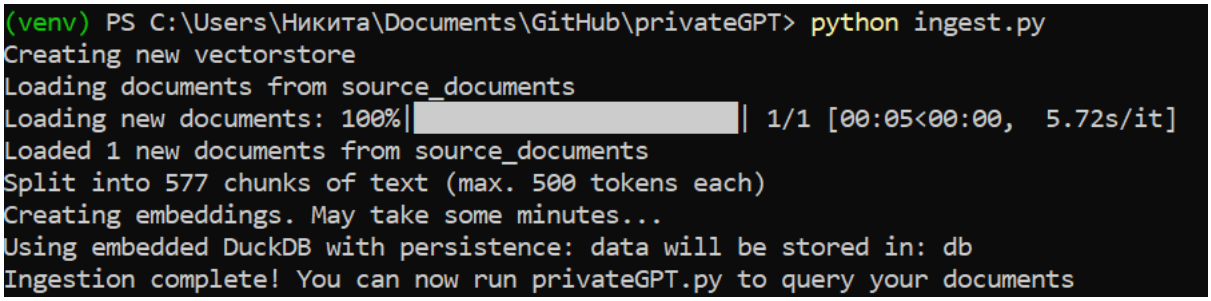


Рисунок 27 – Загрузка нового документа в векторную базу знаний

При запуске файла privateGPT.py будет показан этап активации языковой модели (рис. 28) и выведена строка для ввода запросов.

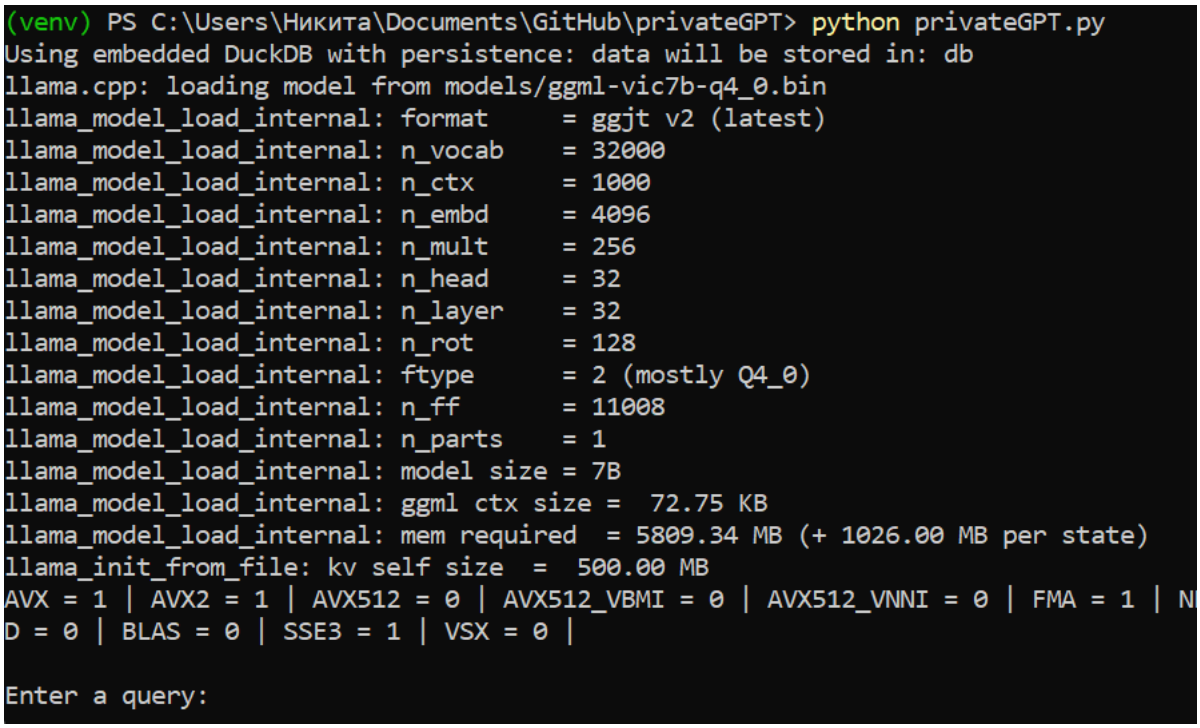


Рисунок 28 – Инициализация языковой модели при запуске программы

8.5 Тестирование

Для тестирования разработанного прототипа, чат-боту было задано несколько вопросов на банковскую тематику, и на любую другую тему. Был активирован режим показа дополнительного контекста, на который опирается языковая модель при генерации ответа на вопросы. На все вопросы ассистент

ответил правильно и справился со своей задачей. Результаты работы программы представлены в приложении А.

8.6 Анализ решения

В результате была разработана полностью локальная реализация банковского чат-бота без необходимости обращения в интернет. Используя мощность и возможности современных библиотек и инструментов, таких как LangChain, Transformers, Chroma и загруженных моделей, которые находятся в открытом доступе, был разработан проект чата с ботом, который позволяет клиентам задавать вопросы и получать на них правильные ответы, обеспечивая полную конфиденциальность данных.

В зависимости от системных характеристик рабочей станции можно использовать различные модели языкового моделирования, от размера которых будет зависеть качество формулирования естественной речи.

ЗАКЛЮЧЕНИЕ

В заключение можно отметить, что одной из основных проблем существующих чат-ботов в банковской сфере является их недостаточная естественность и ограниченность в коммуникации, что приводит к предпочтению людьми общения с операторами.

В рамках данной работы была поставлена целью разработка банковского чат-бота, способного предоставлять естественную поддержку пользователям.

Во время проведения исследования, был сделан вывод о том, что внедрение в алгоритм работы чат-бота языковой модели, основанной на искусственном интеллекте, позволяет значительно улучшить качество речи виртуального ассистента. Языковая модель, обученная на больших объемах текстовых данных, способна генерировать более естественные и связные ответы, что делает взаимодействие с чат-ботом более приятным и продуктивным для пользователей.

В процессе достижения поставленной цели был разработан прототип чата с чат-ботом со встроенной языковой моделью GPT-3.5-turbo компании OpenAI по API. Были выявлены положительные стороны и недостатки полученного решения. Главным минусом считается утечка данных на серверах компании, предоставляющей готовую модель.

Ключевым шагом для улучшения полученного решения стало внедрение языковой модели в открытом доступе с платформы Hugging Face.

В результате проделанной работы был разработан банковский чат-бот, способный обслуживать клиентов и предоставлять им поддержку в их потребностях без использования сети Интернет, то есть без утечки корпоративных данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Антонов С. Что такое чат-боты и зачем они нужны? // Inform Бюро [Электронный ресурс]. Режим доступа: https://informburo.kz/cards/chto-takoe-chat-boty-i-zachem-oni-nuzhny.html (дата обращения 15.11.2022);

2. Mirant Hingrajia. How do Chatbots work? A Guide to Chatbot Architecture // maruti techlabs [Электронный ресурс]. Режим доступа: https://marutitech.com/chatbots-work-guide-chatbot-architecture/ (дата обращения 20.11.2022);

3. Jenna Alburger. Rule-Based Chatbots vs. AI Chatbots: Key Differences // hubtype [Электронный ресурс]. Режим доступа: https://www.hubtype.com/blog/rule-based-chatbots-vs-ai-chatbots (дата обращения 20.11.2022);

4. Types of chatbots // freshworks [Электронный ресурс]. Режим доступа: https://www.freshworks.com/live-chat-software/chatbots/three-types-of-chatbots/ (дата обращения 20.11.2022);

5. Shambhavi Sinha. Chatbot for Banking: Everything you Need to Know // AMEYO [Электронный ресурс]. Режим доступа: https://www.ameyo.com/blog/chatbot-for-banking-everything-you-need-to-know/#:~:text=Chatbots%20in%20banking%20industries%20can,without%20waiting%20on%20the%20phone (дата обращения 22.11.2022);

6. Анна Юченко. How do chatbots work? Often with a little help from AI // TechArt [Электронный ресурс]. Режим доступа: https://www.itechart.com/blog/how-do-chatbots-really-work/ (дата обращения 22.11.2022);

7. Что такое машинное обучение? // Azure [Электронный ресурс]. Режим доступа: https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-machine-learning-platform/#:~:text=%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0

%B8%D0%B5%20(ML)%20%E2%80%94%20%D1%8D%D1%82%D0%BE,%D0%B8%D0%B7%20%D1%84%D0%BE%D1%80%D0%BC%20%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B8%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82%D0%B0%20(%D0%98%D0%98) (дата обращения 05.12.2022);

8. Какие задачи позволяет решать машинное обучение? // ЦИСМ [Электронный ресурс]. Режим доступа: https://www.cism-ms.ru/poleznye-materialy/kakie-zadachi-pozvolyaet-reshat-mashinnoe-obuchenie-/ (дата обращения 05.12.2022);

9. Чат-боты – кто они и что умеют? // EFSOL [Электронный ресурс]. Режим доступа: https://efsol.ru/articles/messendzhery-i-chat-boty-dlya-biznesadostavki.html (Дата обращения: 06.12.2022);

10. Ася Зуйкова. Что такое машинное обучение и как оно работает // РБК Тренды [Электронный ресурс]. Режим доступа: https://trends.rbc.ru/trends/industry/60c85c599a7947f5776ad409 (дата обращения 24.12.2022).

11. Руководство по языку программирования Python [Электронный ресурс] // URL: https://metanit.com/python/tutorial/ (дата обращения: 25.05.2023);

12. Hugging Face. The AI community [Электронный ресурс] // URL: https://huggingface.co/ (дата обращения: 25.05.2023);

13. How to create a private ChatGPT with your own data? [Электронный ресурс] // URL: https://medium.com/@imicknl/how-to-create-a-private-chatgpt-with-your-own-data-15754e6378a1 (дата обращения: 25.05.2023);

14. Ben Lutkevich. Language modeling // TechTarget [Электронный ресурс]. Режим доступа: https://www.techtarget.com/searchenterpriseai/definition/language-modeling (дата обращения 20.05.2023);

15. Language Models, Explained: How GPT and Other Models Work // altexsoft [Электронный ресурс]. Режим доступа: https://www.altexsoft.com/blog/language-models-gpt/ (дата обращения 20.05.2023);

16. Fawad Ali. GPT-1 to GPT-4: Each of OpenAI's GPT Models Explained and Compared // MakeUseOf [Электронный ресурс]. Режим доступа: https://www.makeuseof.com/gpt-models-explained-and-compared/ (дата обращения 21.05.2023).

ПРИЛОЖЕНИЕ А