Оглавление

Введение

В настоящее время обработка и анализа естественного языка является важным инструментом для обработки и понимания большого количества неструктурированных текстовых данных. В современном мире обработка естественного языка (Natural Language Processing, NLP) представляет собой отдельное направление искусственного интеллекта и математической лингвистики.

Применение различных методов в решении задач автоматической обработки естественного языка в последние десятилетия активно развивается благодаря симбиозу технологий машинного обучения, нейронных сетей и технического прорыва в области вычислительной техники. В связи этим тенденций по исследованию и применению различных методов для построения эффективных систем по обработке и анализа естественного языка является актуальной задачей [1].

Учитывая современную потребность в решении задач обработки и анализу естественного языка со стороны компаний, в частности ПАО АКБ «Приморье», находящийся в мире быстро меняющегося рынка, необходимость в исследовании и разработки систем по автоматической обработке естественного языка, является одним из направления по минимизации расходов и увеличения скорости принятия решений по актуальным вопросам в режиме реального времени.

Целью данной выпускной квалификационной работы является исследование процессов обработки естественного языка и применение систем автоматической обработки естественного языка на реестре обращений клиентов ПАО АКБ «Приморье».

В соответствии с поставленной целью исследования были сформулированы следующие задачи:

– исследовать теоретические аспекты и задачи обработки естественного языка;

- проанализировать существующее системы автоматической обработки естественного языка;
- описать общую характеристику компании ПАО АКБ «Приморье»;
- реализовать этапы обработки и анализа естественного языка;
- применить систему искусственного интеллекта для решения задач обработки естественного языка;
- в заключении сделать вывод по проведенной работе.

Объектом исследования является реестр обращений клиентов предоставленный компанией ПАК АКБ «Приморье», который представляет текстовые данные о вопросах/проблемах или иные сообщениях клиентов банку.

Предметом исследования является область искусственного интеллект в задачах обработки и анализа естественного языка.

Работа состоит из введения, трех глав, заключения, списка литературы и одного приложения.

1 Теоретические аспекты обработки естественного языка

1.1 Задачи обработки естественного языка

Обработка естественного языка (Natural language processing, NLP) – симбиоз искусственного интеллекта и математической лингвистики направленная на создание систем синтеза и анализа естественного языка.

Основные понятия: Естественный язык – язык, на котором говорят и пишут текст люди. Корпус – подобранная и обработанная по определённым правилам совокупность текстов, используемых в качестве базы для исследования языка. Токен – текстовая единица, например слово, словосочетание, предложение.

Задачи обработки естественного языка (Natural Language Processing, NLP) имеют множество прикладных применений. Хорошая система NLP – это система, которая решает комплекс задач. Основными направлениями обработки естественного языка, можно считать:

- Распознавание речи (Speech Recognition);
- Понимание естественного языка (Natural Language Understanding);
- Генерация естественного языка (Natural Language Generation).

Из этих направлений вытекают классические задачи обработки естественного языка:

Машинный перевод (Machine Translation, MT). Задача преобразования предложения/фразы из исходного языка (например, немецкого) в целевой язык (например, английский). Это очень сложная задача, поскольку разные языки имеют очень разные морфологические структуры, следовательно, это не взаимно однозначное преобразование;

Классификация теста (Text Classification, TC). Задача имеет множество вариантов использования, таких как обнаружение спама, классификация новостных статей (например, политические, технологические и спортивные) и распознавание отзывов о продукте (например, положительные или

отрицательные). Это достигается обучением модели классификации на помеченных данных (то есть на обзорах, аннотированных людьми).

Извлечение именованных сущностей (Named Entity Recognition, NER). Задача пытается извлечь сущности (например, человека, местоположение и организацию) из заданного текста или текстового корпуса. Например, предложение «Джон дал Мэри два яблока в школе в понедельник» будет преобразовано в [Джон]имя дал [Мэри]имя [два]число яблока в [школе]организация в [понедельник]время. Без NER невозможно обойтись в таких областях, как поиск информации и представление знаний.

Генерация естественного языка (Natural Language Generation, NLG). Компьютерная модель, например нейронная сеть, с помощью текстового корпуса обучается генерации новых текстов. Например, можно сгенерировать совершенно новый научно-фантастический рассказ, используя для обучения модели существующие рассказы.

Вопросно-ответные системы (Question Answering, QA). Технологии вопросно-ответных систем имеют высокую коммерческую ценность и лежат в основе чат-ботов и виртуальных помощников (например, Google Assistant и Apple Siri). Чат-боты широко используются для ответов на вопросы и решения простых проблем клиентов (например, изменения тарифного плана мобильной связи), которые могут быть выполнены без вмешательства человека. Реализация QA-систем охватывает обширные аспекты NLP, такие как поиск информации и представление знаний [2, с. 22].

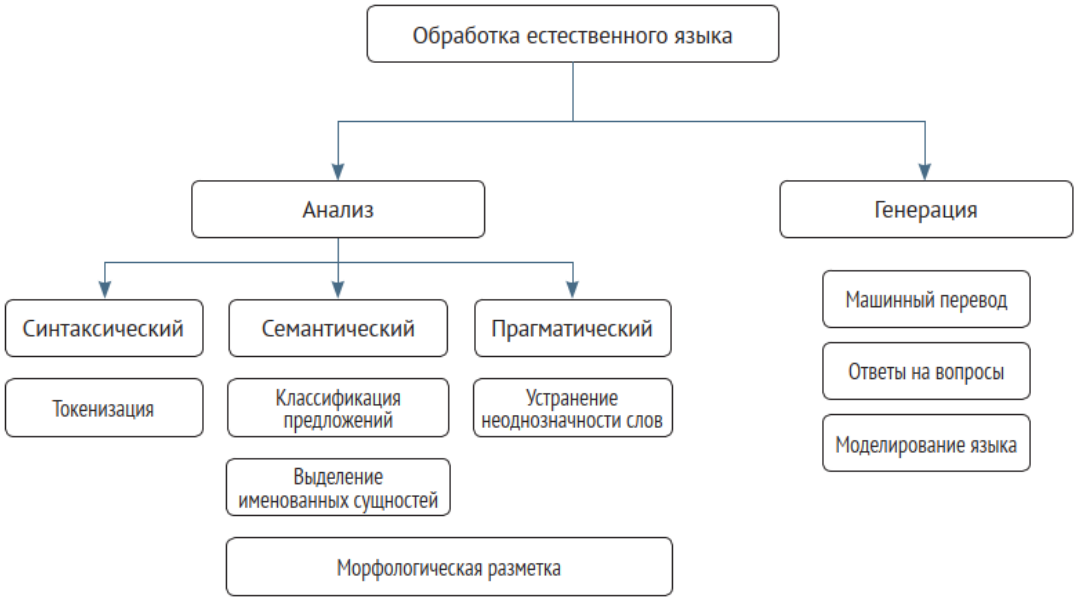


Рисунок 1 – Таксономия классических задач обработки естественного языка

Исходя из таксономии (рисунок 1), задачи обработки естественного языка можно разделить на две широкие категории: анализ существующего текста и генерация нового текста. Анализ в свою очередь, подразделяется на три подкатегории: синтаксический (задачи, основанные на структуре языка), семантический (задачи, основанные значении языка) и прагматический (открытые проблемы, которые трудно решить).

Таким образом это безусловно не весь список задач обработки естественного языка. Их десятки. По большому счету, все, что можно делать с текстом на естественном языке, можно отнести к задачам обработки естественного языка.

1.2 Этапы обработки и анализа естественного языка

При решении задач обработки естественного языка запускается цепочка задач состоящее из методов обработки и анализа естественного языка. Методы обработки естественного языка отвечают за изменение и преобразование текстового корпуса, в то время как методы анализа естественного языка позволяют извлечь информацию из исследуемого объекта корпуса, такого как слова, предложения или документы.

Процессы обработки естественного языка включают:

Предобработку: процесс изменения и удаления в текстовом корпусе, туда входит изменение регистра слов, удаление стоп-слов (слово, которое не несет особого значения), цифр, пунктуации и пробелов в корпусе;

Токенизацию: процесс разделения текстового корпуса на неделимые единицы, например слова;

Векторизацию: преобразования текстового корпуса в векторные числовые представления, например в виде математической матрицы TF-IDF;

Стемминг: грубый эвристический процесс, который отрезает «лишнее» от корня слов, часто это приводит к потере словообразовательных суффиксов, например «обнаружена» в «обнаруж»;

Лемматизация*:* процесс преобразования слово, когда оно возвращается к базовой форме в контексте морфологического анализа, что в итоге приводит слово к его канонической форме – лемме, например «обнаружена» в «обнаружить»;

Процессы анализа естественного языка включают:

Лексический анализ: процесс разбора слова с точки зрения его значения, происхождения, употребления, наличия у него синонимов, антонимов, омонимов и паронимов.

Морфологический анализ: процесс определение морфологических признаков слова, включая его форму, часть речи, падеж, род, число и т.д. В это процесс входят лемматизация и стемминг. Они заключаются в обработке слово с точки зрения грамматической формы, т.к. в тесте могут быть разные формы одного и того же слова, или быть однокоренные слова.

Синтаксический анализ: процесс сопоставления линейной последовательности предложения или токенов естественного языка. Результатом может являться дерево (синтактическое дерево) где каждый узел представляет собой слово или группу слов, а дуги между узлами показывают синтаксические отношения между ними. Или может в виде графа (синтактический граф) зависимостей, где каждый узел представляет собой

слово, а дуги между узлами показывают синтаксические отношения между ними [3, с. 207].

Например: разберем следующее корпуса – [«Илья Сегалович является клиентом банка и имеет положительную кредитную историю.»]. Синтактическое дерево корпуса изображено на рисунке 2

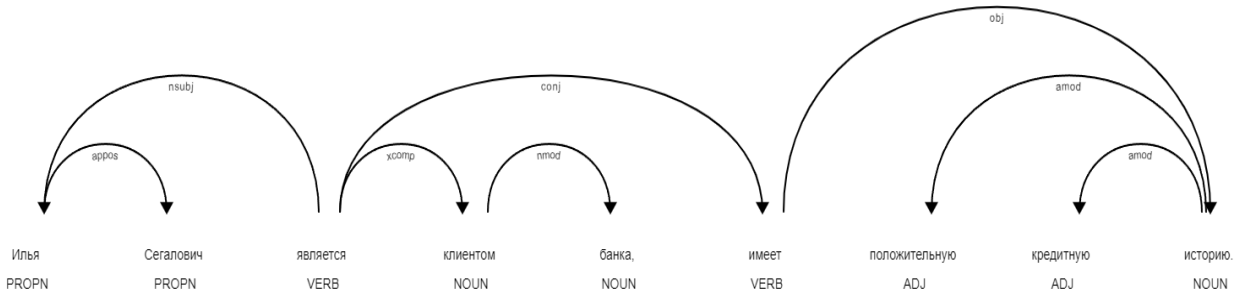


Рисунок 2 – Пример синтактического дерева

Семантический анализ: процесс в последовательности действий алгоритма автоматического понимания текстов, заключающийся в выделении семантических отношений, формировании семантического представления текстов. Один из возможных вариантов в себя включает идентификацию значимых слов, фраз и предложений, а также определение связей между ними. Семантический анализ в рамках одного предложения называется локальным семантическим анализом. Он позволяет определить семантическую близость между словами в контексте предложения [4, с. 104].

Например, проведем семантический анализ двух схожих предложений: [Илья Сегалович является клиентом банка и имеет положительную кредитную историю.] и [Сегал Ильявич считается клиентом компании и обладает хорошую историю кредитных выплат.] Близость данных предложений по смыслу составляет 0.815.

Таким образом мы перечислить основные этапы обработки и анализа естественного языка, которые протекают при решении задач, описанных ранее в подразделе 1.1.

1.3 Модели искусственного интеллекта для обработки естественного языка

Одним из направлений искусственного интеллекта является обработка естественного языка. В этой области основной задачей является понимание нашего мира. За последние десятилетия благодаря исследованиям и симбиозу технологий, таких как машинного обучения, нейронных сетей и технический прорыв, удалось найти эффективное решение в виде больших языковых моделей.

Большие языковые модели (Large Language Model, LLM**)** – это математическая модель нейронных сетей, обученная методом глубокого обучения на большом объёме данных. Они состоят из свыше миллиарда параметров и показывают высокую точность при решении задач обработки естественного языка.

Большие языковые модели появились примерно в 2018 году и показали себя эффективными в решении с широкого спектра задач обработки естественного языка. Это привело смешению фокус с парадигмы обучения специализированных контролируемых моделей для конкретных задач, на изучение универсальной модели, что в результате привело к новым открытиям и улучшениям в обработке естественного языка [4].

В таблице 1 представлен список современных больших языковых моделей.

Языковые модели, обученные на больших текстовых данных, позволяет им изучать сложные зависимости между словами. Это дает им способность решать широкий круг задач, включая обработку и генерацию текста, изображений и звуков на основе входных данных.

Таблица 1 – Список современных больших языковых моделей

Имя	Дата выхода	Создатель	Кол-во параметров	Кол-во токенов	Примечание
BERT	2018	Google	340 млн.	3.3 млрд.	Ранняя и влиятельная языковая модель, но предназначенная только для кодировщика и, следовательно, не предназначенная для подсказок или генерации.
GPT-2	2019	OpenAI	1.5 млрд	10 млрд.	Универсальная модель на базе трансформаторной архитектуры
GPT-3	2020	OpenAI	175 млрд	300 млрд.	Доработанный вариант GPT-3, получивший название GPT-3.5, стал общедоступным через веб-интерфейс под названием ChatGPT в 2022 году.
GPT-J	06.2021	EleutherAI	6 млрд.	825 гиб	Языковая модель в стиле GPT-3
Megatron-Turing NLG	10.2021	Microsoft, Nvidia	580 млрд.	339 мдрд.	Стандартная архитектура, но обучение на суперкомпьютерном кластере.
GLaM	12.2021	Google	1.2 трл.	1.6 трл.	Модель с разреженной смесью экспертов, что делает ее более дорогой для обучения, но более дешевой для проведения логического вывода по сравнению с GPT-3.
OPT	05.2022	Meta	175 млрд.	180 млрд.	Архитектура GPT-3 с некоторыми адаптациями от Megatron.
YaLM	06.2022	Yandex	100 млрд	1.7 трб	Англо-русская модель на базе Megatron-LM от Microsoft.
AlexaTM	11.2022	Amazon	20 млрд	1.3 трл.	Двунаправленная архитектура «последовательность к последовательности»
LLaMA	02.2023	Meta	65 млрд.	1.4 трл.	Прошел обучение на большом корпусе из 20 языков, чтобы добиться лучшей производительности при меньшем количестве параметров. Исследователи из Стэнфордского университета обучили отлаженную модель, основанную на весах LLaMA, под названием Alpaca.
GPT-4	03.2023	OpenAI	~ 1 трл.	-	Доступно для пользователей ChatGPT Plus и используется в нескольких продуктах.

Основные технологии, которые внесли вклад в создание эффективных больших языковых моделей, включают в себя:

Технология машинного обучения (Machine Learning, ML) – это наука о разработке алгоритмов и статистических моделей, которую вычислительная система используют для выполнения задач без явных инструкций, полагаясь вместо этого на шаблоны и логические выводы. Вычислительная система используют алгоритмы машинного обучения для обработки больших объемов статистических данных и выявления шаблонов данных.

Глубокое обучение (Deep Learning, DL) – это подраздел машинного обучения, который использует нейронные сети, моделированные по образцу человеческого мозга, для обработки данных и распознавания сложных закономерностей в них. Глубокое обучение характеризуется как класс алгоритмов машинного обучения, который использует многослойную систему нелинейных фильтров для извлечения признаков с преобразованиями. Каждый последующий слой получает на входе выходные данные предыдущего слоя. Процесс глубокого обучения состоит из двух основных этапов: обучения и формирования выводов [5].

Обучение происходит в несколько этапов:

- ввод данных: данные подаются на входную часть нейронной сети;
- прямое распространение: данные проходят через каждый слой нейронной сети, где они обрабатываются узлами (искусственными нейронами);
- расчет ошибки: система сравнивает результаты выходного слоя нейронной сети с правильными ответами и вычисляет ошибку;
- обновление весов: система корректирует веса каждого узла в соответствии с вычисленной ошибкой, используя алгоритмы градиентного спуска;
- обучение повторяется: процесс обучения повторяется несколько раз, пока результаты не станут достаточно точными.

Фазу обучения следует рассматривать как метод маркировки больших объемов данных и определение их соответствующих характеристик. Система

сравнивает эти характеристики и запоминает их, чтобы сделать правильные выводы, когда она столкнется с подобными данными в следующий раз.

Нейронные сети (Neural Network, NN) – это программные алгоритмы, которые повторяют работу биологических нейронных сетей в мозгу человека, они работают на принципах математических вычислений и представляют из себя искусственные нейронные сети, благодаря этому они способны обрабатывать, анализировать и запоминать информацию.

Глубокое обучение (DL) и нейронные сети (NN) имеют очень тесную связь. Нейронная сеть является математической моделью, которая может быть обучена с помощью глубокого обучения. Глубокое обучение, в свою очередь, является подкатегорией машинного обучения, в которой используются многослойные нелинейные алгоритмы для извлечения признаков с преобразованиями. Каждый последующий слой получает на входе выходные данные предыдущего слоя.

Таким образом при рассмотрении направления искусственного интеллекта в обработке естественного языка, мы обнаружил что языковые модели эффективны для решения нашей задачи исследования. Далее рассмотрим одну их языковых моделей.

1.4 Языковая модель BERT для обработки естественного языка

В последние годы трансферное обучение стало популярным подходом в области обработки естественного языка. Одним из наиболее успешных методов трансферного обучения является языковая модель BERT.

BERT (Bidirectional Encoder Representations from Transformers) – это языковая модель, основанная на архитектуре трансформера. Она представляет собой нейронную сеть, которая обучается на больших объемах текста и имеет способность понимать контекстный смысл слов и связи между ними.

Базовая модель семейства BERT достаточно долго обучается на огромных корпусах текстов, пропуская через себя миллионы документов и

постепенно осваивая язык, грамматику и сущность слов. В дальнейшем модель можно дообучить на пользовательских наборах данных для выполнения конкретной прикладной задач обработки естественного языка [6, с. 142].

Подобные сети основаны на архитектуре «трансформер», которая используется для моделирования задач понимания языка, полагаясь на механизмы внимания (attention) для построения глобальных зависимостей между входами и выходами.

Рассмотрим подробнее принципы работы механизма внимания.

Пусть на вход сети передано предложение, состоящие из слов $x_1, x_2, \dots, x_n$, при это уже в векторном представлении. Предположим, что необходимо установить зависимость x_4 от всех остальных слов. Обозначим зависимость как y_4, которая вычисляет по выражению 1.

$$y_3 = \sum_{i=1}^{n} w_{4i} * x_i \quad (1)$$

где w_{4i} – веса семантической близости слова, получены как скалярное произведение слова x_4 со словом x_i – векторное представление некоторого слова, входящего в состав предложения.

При этом можно установить, каким образом слово x_i оказывает «внимание» на слово x_4. Операция внимания входит в блок трансформера, и с каждым блоком происходит переход на абстракцию более высокого уровня относительно исходных слов, что позволяет лучше устанавливать связь слов друг с другом.

В чистом виде блок трансформера состоит из двух компонентов — кодировщика и декодера. Первый компонент считывает входные данные, а второй выполняет задачу предсказания (рисунок 3).

Внутренняя архитектура блока состоит из следующих элементов:

– с начала текст разбивается на слова, а потом слова сопоставляются с их векторными представлениями;
– позиционные кодировщики вводят информацию о позиции входного слова;

– уровень внимания кодирует информацию о входной последовательности с учетом контекста;
– слой прямого распространения сигнала, который работает как статическая память, одна из его выходных последовательностей является константой.
– перекрестное внимание декодирует выходную последовательность различных входов и модальностей.

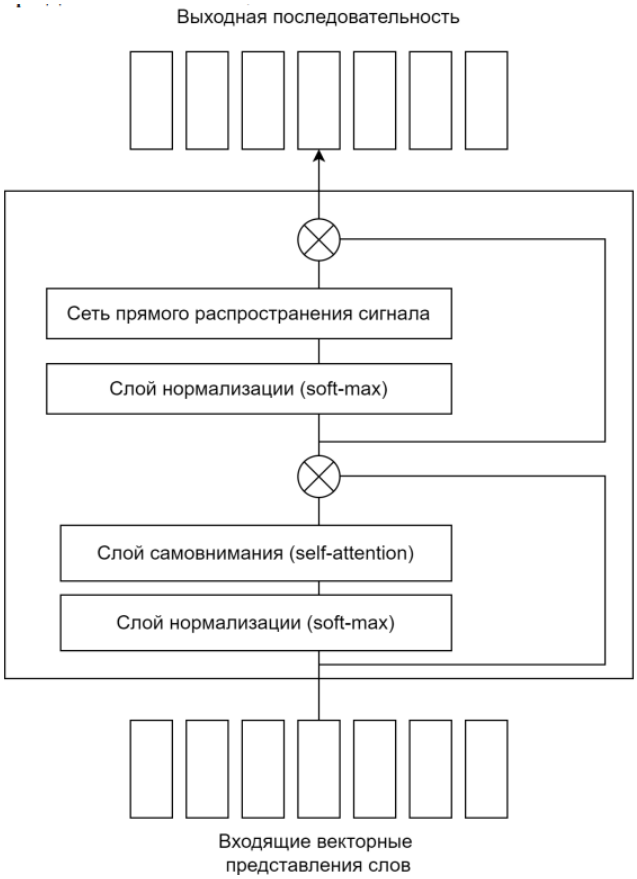


Рисунок 3 – Внутреннее устройство блока трансформера для кодера

Слой внимания принимает n входов и возвращает n выходов. Слой внимания создает три вектора для каждого входящего числового представления слова: вектор запроса, вектор ключа и вектор значения. Эти векторы создаются с помощью перемножения входящего вектора на три матрицы, которые были получены при обучении. В итоге после перемножения мы получаем проекции, W^Q, W^K, W^K для каждого слова в составе входящего предложения. Далее рассчитываются коэффициенты внимания для каждого слова, входящего в предложения, путем скалярного произведения вектора запроса на вектор ключа этого слова. Полученные

скалярные величины делятся на квадратный корень размерности вектора ключа – $\sqrt{64}$, а затем полученный результат пропускается через функцию нормализации. После применения функции скалярные величины представляются вещественным числом в интервале [0,1] и их сумма равна 1. Каждый вектор значения умножается на коэффициент нормализации. Далее взвешенные векторы значения складываются, образуя выход слоя внимания для этого слова. После всех вычислений мы получаем вектор, который можно дальше передать в сеть прямого распространения сигнала. На практике для ускорения вычислений используются матрицы вместо векторов.

Таким образом появляется возможность учитывать при обучении позицию слова в контексте предложения. Для этого необходимо добавить в начала векторного представления слова вектор-позицию такой же размерности (рисунок 4).

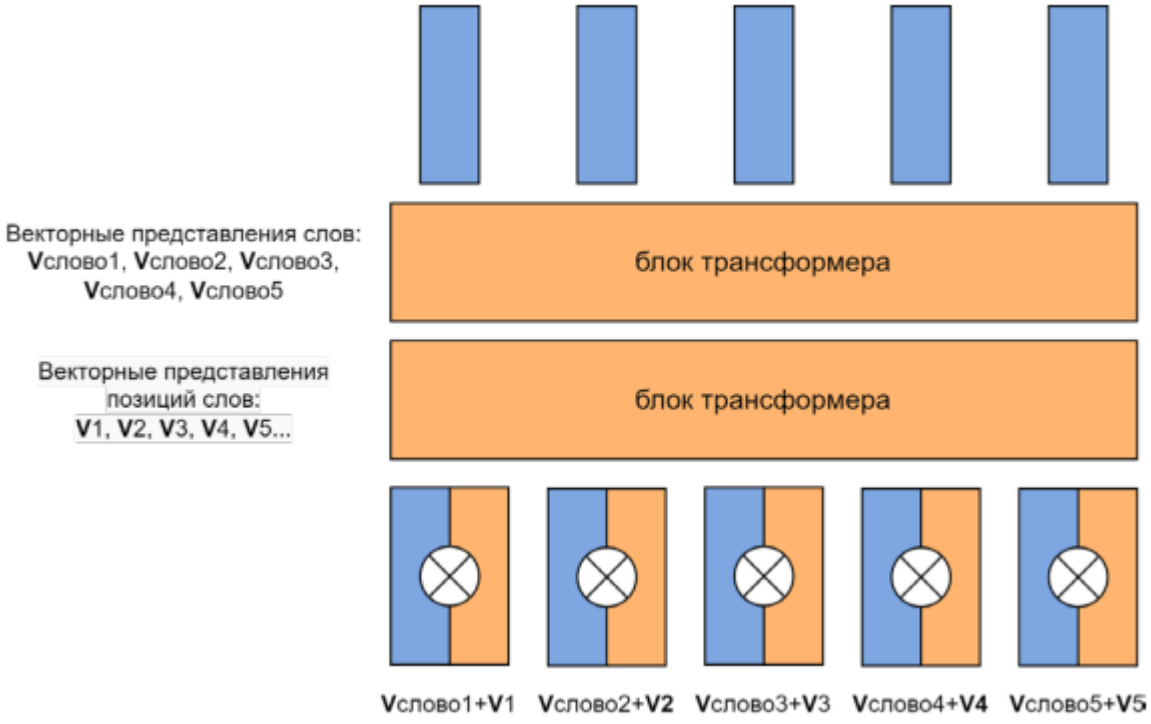


Рисунок 4 – Входные векторы предложений, учитывающие порядок слов

В результате для каждого слова генерируется вектор, заключающий в себе значения слова и номер позиции в предложении. Похожие по смыслу слова имеют близкие числовые значения внутри векторов.

Этап предварительной подготовки данных для BERT

Для каждого кодировщика BERT существует своя модель предварительной обработки данных – набор операторов для приведения исходного текста в числовые представления, ожидаемые кодировщиком на входе. Каждая из моделей поставляется со своими заранее сформированным словарем и связанной с ним логикой нормализации текст, на практике не требует точной настройки параметров обучения.

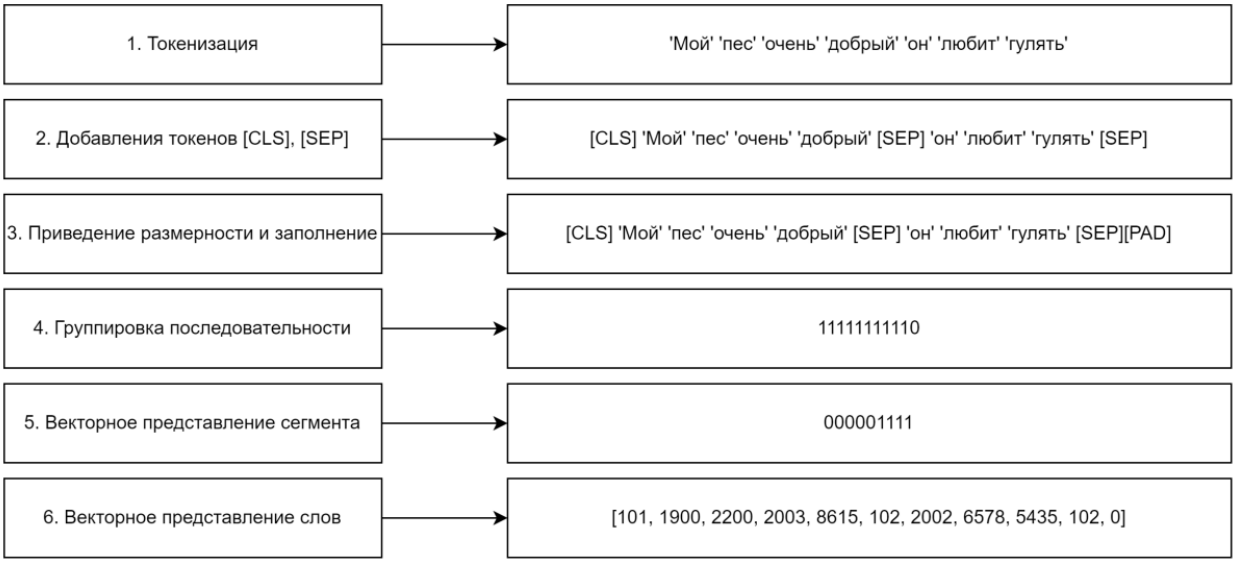


Рисунок 5 – Блок операции предварительной подготовки естественного языка

Обработка текста условно разделена на шесть этапов [7], как показано на рисунке 5. Первый этап — разбиение слов на токены. В базовой модели BERT использует словарь в размере 30552 слов. Процесс разбиения на слова токены включает в себя разбиение входного потока текстовых данных на список слов [8], доступных в словаре. Отсутствующие слова постепенно разбиваются на морфемы, а затем представляются группой морфем. Поскольку морфемы являются частью словаря, можно получить векторное представление этих морфем, а контекст слова – это просто комбинации этих морфем. Далее все предложения усекаются до единой длины – это еще одно из важных условий для успешного обучения.

На выходе блока предварительной подготовки данных остается числовое представление вектора с указанием индекса слов в словаре с учетом их семантической близости.

Инкапсулированные базовые операции этапа предварительной подготовки текста внутри моделей BERT позволяют на выходе получить более чистый и лаконичный программный код при реализации этапов обучения и прогнозирования пользовательских значений в реализуемых задачах.

Обучение моделей BERT

Основная задача состоит в том, чтобы протестировать наиболее производительные модели из семейства BERT для выполнения задачи классификации данных в рамках реестра обращений клиентов банка [6].

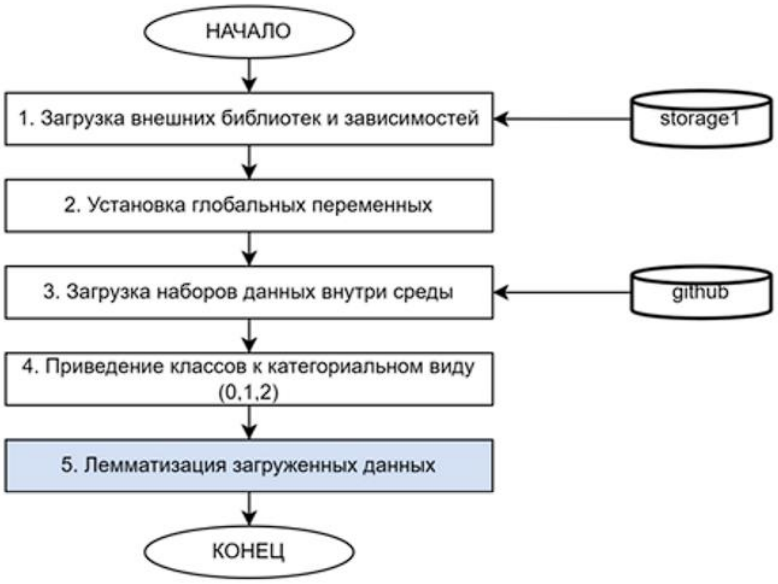


Рисунок 6 – Схема этапов предварительной подготовки данных

Техническая часть исследования будет разделена на два этапа. Первый этап относится к выбору подходящего набора данных для выполнения поставленной задачи и его первичная обработка (рисунок 6). Этап включает в себя загрузку пользовательских данных и библиотек для работы с ними, предварительная подготовка данных для обучения модели, включая дополнительные этапы нормализации (лемматизация данных).

Следующий этап включает в себя работу с предобученными моделями на базе механизма трансформеров. Алгоритм работы с моделями включает себя следующие под этапы: выбор необходимой модели с сайта-репозитория, выгрузка данных в рабочую среду, векторное представление входных последовательностей (токенизация), обучение выбранной модели, тестирование моделей.

После тестирования модели необходимо убедиться, удовлетворяет ли модель необходимым критериям качества для решения задачи классификации данных, если нет, то есть несколько вариантов дальнейшего развития событий — или возвращаемся на этап выбора моделей, для выбора модели, обученной на другом корпусе текстов, или переходим к этапу точной настройки параметров обучения модели.

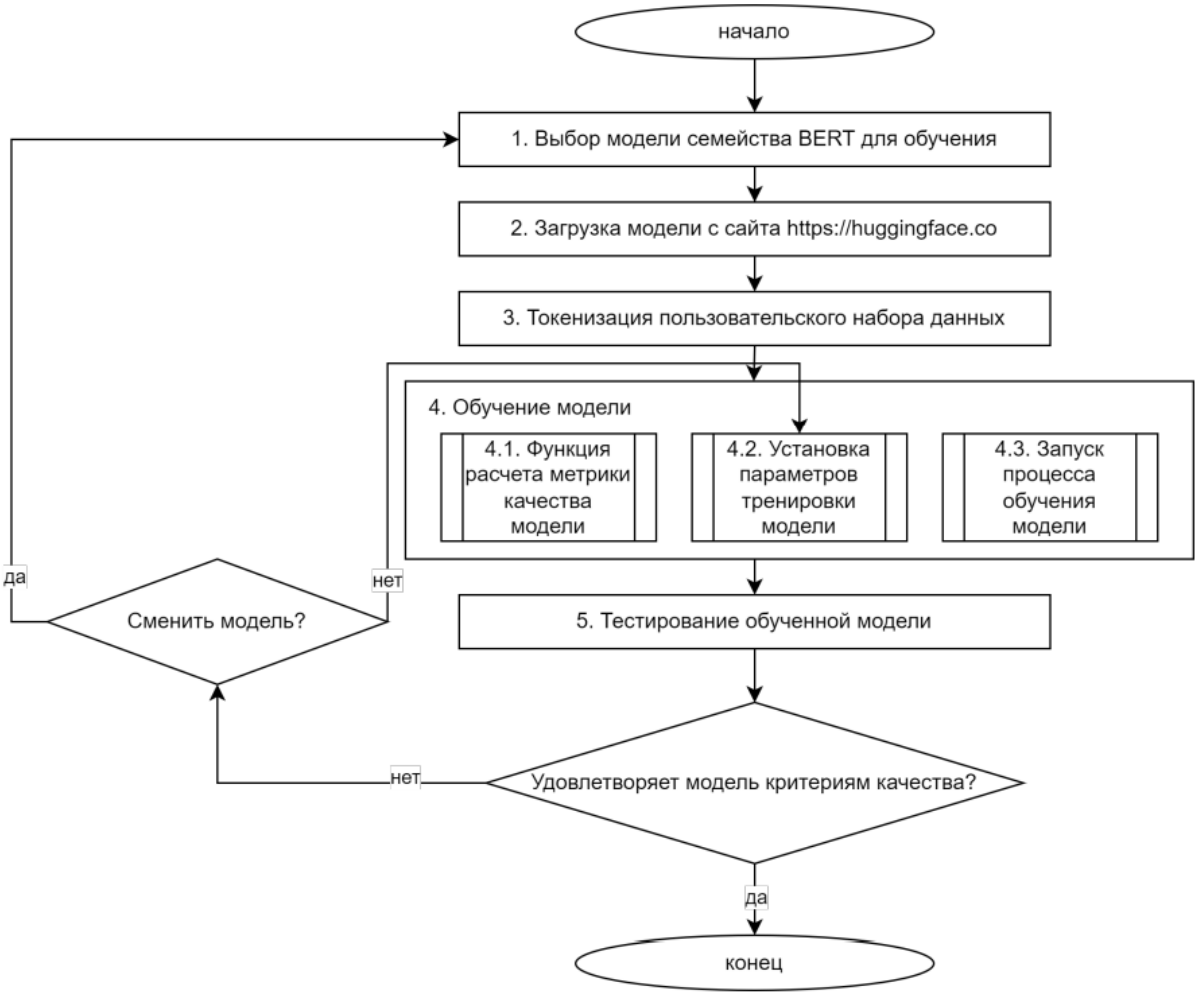


Рисунок 7 – Алгоритм обучения модели BERT

Как показывает практика, то многообразие настроек обучения (рисунок 7, блок 4.2–4.3) сводится к выбору количества эпох обучения сети, размера обучающей выборки и выбор метода оптимизации нейронных сетей.

Преимущества модели BERT включают повышение точности в различных задачах обработки естественного языка, таких как оценка общего понимания языка и набор данных ответов на вопросы. BERT также позволяет учитывать связи между предложениями и выполнять задачи классификации двух предложений.

Таким образом, модель BERT представляет собой эффективный подход к обработке естественного языка, который использует трансферное обучение для достижения передовых результатов в различных задачах. BERT позволяет учитывать контекст и взаимосвязи между словами, а также выполнять задачи классификации и предсказания следующего предложения.

2 Обработка и анализ реестра обращений клиентов ПАО АКБ «Приморье»

2.1 Общая характеристика ПАО АКБ «Приморье»

Основанный в 1994 году, Банк «Приморье» все эти годы развивался вместе с дальневосточным бизнесом, выступая партнером и поддерживая начинания многих региональных компаний и предпринимателей. В настоящее время ПАО АКБ «Приморье» является одним из крупнейших банков в России, оперирующих на региональном уровне. Банк имеет высокую степень устойчивости и компетентности и является надежным финансовым партнером для деловых кругов, органов государственной власти и населения различных регионов страны, включая Приморский и Хабаровский края, Магаданскую, Сахалинскую и Иркутскую области, а также Москву и Санкт-Петербург [9].

В приоритете — развитие продуктовой линейки для частных и корпоративных клиентов, а также расширение географии присутствия. В конце 2018 года Банк «Приморье» открыл второй офис в Хабаровске и первый в Южно-Сахалинске, в начале 2019 года — представительство в Магадане, летом 2019 — офис в Иркутске. Первый офис в Москве Банк «Приморье» открыл в июле 2020 года, а год спустя, в сентябре 2021 — в Санкт-Петербурге, в 2022 году в Новосибирске. Банк поддерживает своих клиентов в их развитии, освоении новых рынков: клиенты Банка расширяют свою деятельность, выходят в новые регионы Дальнего Востока, Сибири и другие федеральные округа, а Банк, в свою очередь, готов обеспечивать им комфортное и выгодное банковское обслуживание там, где удобно.

Сегодня большая часть клиентов банка представляет предприятия из различных отраслей экономики Дальнего Востока и Восточной Сибири. Банк известен своим высоким качеством и технологичностью услуг. Он предлагает коммерческие банковские продукты для корпоративных и частных клиентов, что является приоритетным направлением его бизнеса.

В банке существует несколько управлений, в обязанности сотрудников которых входит полное обслуживание клиентов по всем видам оказываемых банком услуг. Сотрудники банка разделены на три управления: по работе с физическими лицами, обслуживанию юридических лиц и специальное подразделение для VIP-клиентов. Банк также имеет разделение на фронт-офис, отвечающий за работу с клиентами и проведение операций, и бэк-офис, который занимается оформлением документации и другими задачами, связанными с обслуживанием работы фронт-офиса.

Также стоит отметить, что организационная структура банка представлена в приложении А, которое, содержит более подробную информацию о структуре и функциях отдельных подразделений банка.

2.2 Обзор реестра обращений клиентов банка

Рассмотрим реестр обращений клиентов банка ПАО АКБ «Приморье», данные, предоставленные компанией, приставляют собой excel-файл, в котором отражены обращения клиентов с 2019 г. по 2023 г.

Набор данных содержит 29 тыс. записей. В таблице 1 представлены атрибуты.

Таблица 2 – Атрибуты реестра обращений

Название атрибута	**Пояснение**
Id	идентификационный номер обращения
Date	дата обращения
About	текст обращения
Subject	предмет обращения (*25 уникальных*)
Answer	ответ на обращение
Depart	департаменты к зоне ответственности, к которой относится обращение (*286/89 уникальных*)
Method	метод обращения (*15 уникальных)*

Для наглядного реестра обращений клиентов банка построим соответствующие графики с использование библиотеки *matplotlib*:

1. График частоты общений клиентов (рисунок 8).
2. Столбчатые/круговые диаграммы по «method», «subject» и «depart» обращений клиентов банка (рисунки 9,10 и 11).

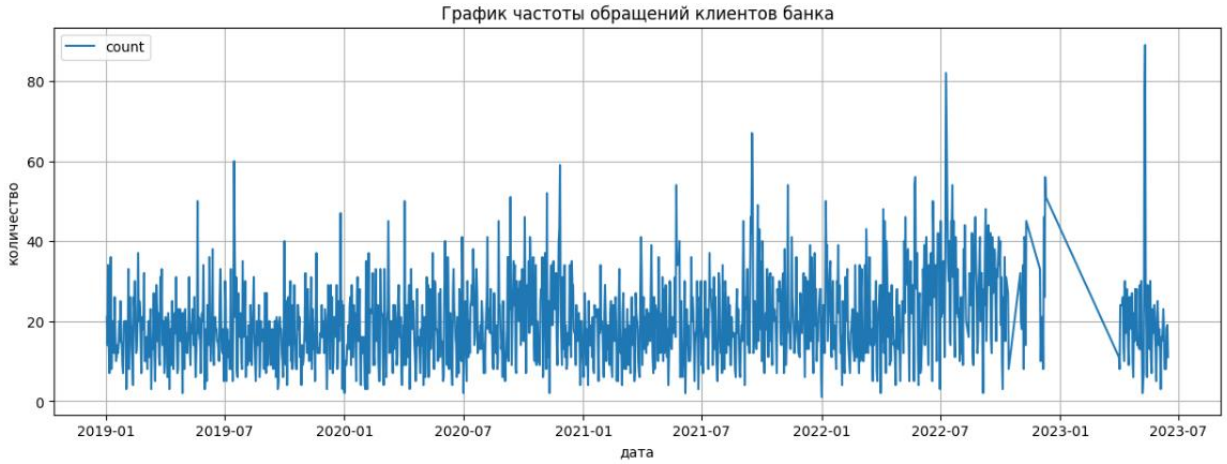
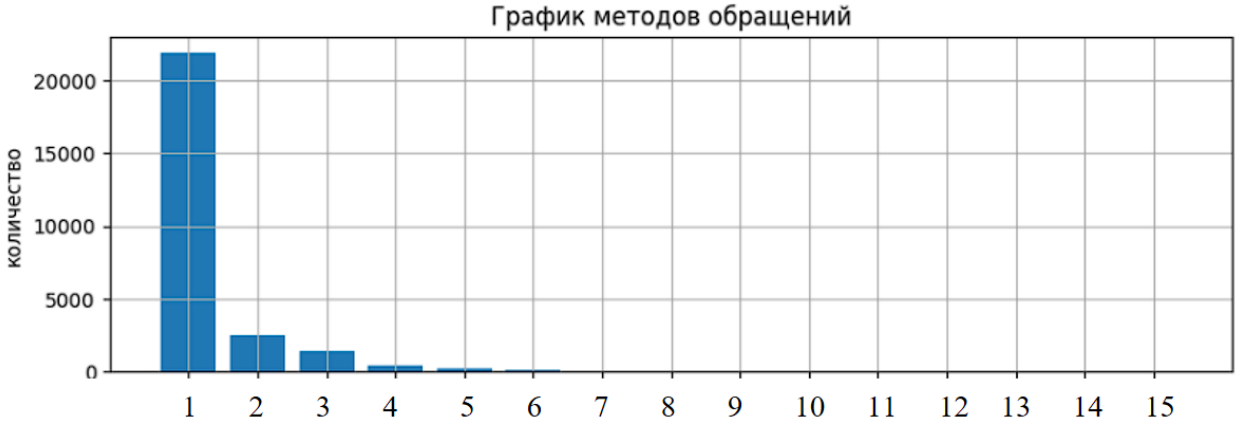


Рисунок 8 – График частоты обращений клиентов банка

На рисунке 8 видно, что график равномерно распределен, и в среднем в день поступает около 18 обращении. В конце 2022 года в начале 2023 есть разрыв в отсутствии данных, это вязано с тем, что компания разрабатывала соответствующее ПО единого реестр обращений клиентов и в это промежуток времени она внедряла новую систему.



1. звонок (21873);
2. обращение, оставленное на сайте банка (2461);
3. заявление (1358);
4. обращение, направленное на общий адрес банка mail@primbank.ru (360);
5. анкета (220);
6. письмо (154);
7. email (40);
8. письмо от сот-ка банка по клиенту (20);
9. Play market/appstore/appgallery (15);
10. цб (11);
11. обращение, направленное на общий адрес кц contact@primbank.ru (10);
12. заявление, пришедшее на почтовый адрес банка (9);
13. ibank (6);
14. прокуратура (1); роспотребнадзор (1).

Рисунок 9 – График методов обращений клиентов банка

Из рисунка 9 видно, что основной метод обращений клиентов – звонки, их более 80%, а на другие приходится около 20%.

1.карта (9531);
2.п/к (4264);
3.atm (4102);
4.ипт/бпт/птбск (3064);
5.смс (1606);
6.мобильное приложение (1457);
7.pos (995);
8.переводы (723);
9.интернет банк (693);
10.сервис (647);
11.кредитование фл (391);
12.рко фл (324);
13.касса (274);
14.вклады (242);
15.прочее (193);
16.рко юл (132);
17.сайт (99);
18.офис (86);
19.кредитование юл (54);
20.о банке (52);
21.электронная очередь (38);
22.ibank2 (25);
23.динамический pin (10);
24.запрос (10);
25.страхование (1);

Рисунок 10 – График предметов обращений клиентов

Из рисунка 10, видно предмет обращений не сбалансирован – в каждой категории представлено разное число обращений с сильным разбросом по их количеству.

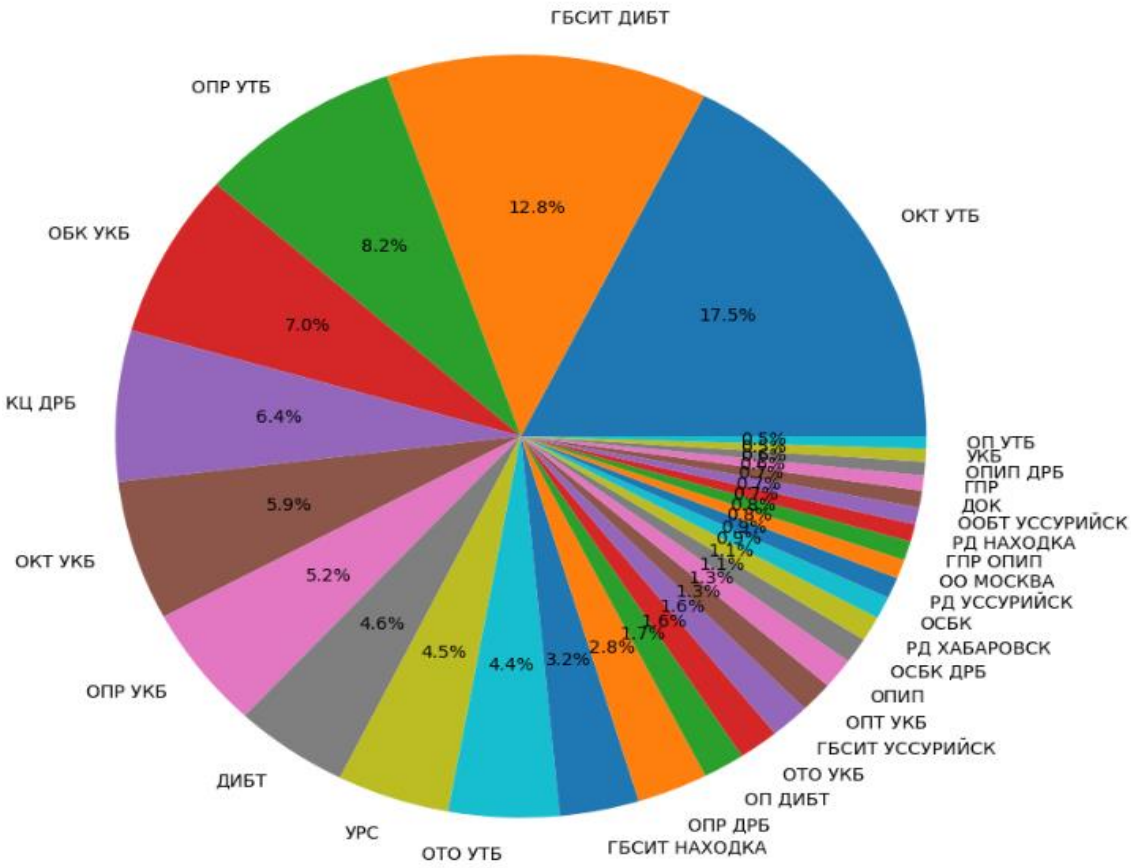


Рисунок 11 – Круговая диаграмма департаментов обращений

На рисунке 11 изображена круговая диаграмма только тех департаментов, у которых количество обращений превышает 100. Здесь так же обращения не сбалансированы – в каждой категории представлено разное число обращений с сильным разбросом.

Очевидно, что при распознавании обращений, относящихся к категориям с малым количеством данных, будут возникать трудности.

2.3 Обработка и анализ реестра обращений клиентов

В данном подразделе проведены этапы обработки и анализа на реестре обращений клиентов, теоретическою часть которой мы описали в разделе 1.2. Для обработки и анализа реестра обращений рассмотрим «About» т. е. текст обращений клиентов.

Опишем процесс предобработка текста обращений клиентов. Для перевода текста в нижний регистр и удаления стоп-слов, цифр, пунктуации и пробелов в корпусе, напишем скрипт на языке python (рисунок 12).

```
#библиотеки
import re
from nltk.corpus import stopwords
import nltk
nltk.download('stopwords')
```

```
%%time
# 1. нижний регистр
# 2. удаление из текста символов из заранее заданного набора
# 3. удаление стоп слов (stopwords)
df_1['re_ABOUT'] = 0
df_1['stop_ABOUT'] = 0

for i in range(len(df_1)):
  df_1.loc[:, ('re_ABOUT')][i] = re.sub(r'[.,"\'-?:!;N%•–«»]',"",
                                        str(df_1['ABOUT'][i])).lower()
  df_1.loc[:, ('stop_ABOUT')][i] = " ".join([word for word in str(df_1['re_ABOUT'][i]).split()
                                             if word.lower() not in set(stopwords.words("russian"))])
```

Рисунок 12 – Программный скрипт для предобработки на языке python

Опишем процесс лемматизации текст обращений клиентов. Для этого процесса необходимо загрузить русский словарь с базовых форм слов, с помощью команды spacy.load('ru_core_news_sm'). Программный скрипт представлен на рисунке 13.

```
#библиотеки
!python -m spacy download ru_core_news_sm
import spacy
import pymorphy2
nlp = spacy.load('ru_core_news_sm')
```

```
%%time
# 4. Лемматизация (pymorphy2)
df_1['lemma_ABOUT'] = 0

for i in range(len(df_1)):
    df_1.loc[:, ('lemma_ABOUT')][i] = " ".join([token.lemma_ for token in nlp(df_1['stop_ABOUT'][i])])
```

Рисунок 13 – Программный скрипт для лемматизации на языке python

Опишем процесс векторизации текста обращений клиентов. Для начала проведем маркировку предложений в корпусе и создадим словарь, содержащий слова с соответствующим ими частоты в корпусе.

Напишем функцию print_bow(), которая создает словарь под названием bow{}. Далее мы повторяем каждое предложение в корпусе. Предложение маркируется словами. Далее мы повторяем каждое слово в предложении. Если слово не существует в словаре bow{}, мы добавим слово в качестве ключа и установим значение слова как 1. В противном случае, если слово уже существует в словаре, мы просто увеличим количество ключей на 1. Программный скрипт представлен на рисунке 14.

```
def print_bow(sentence: str) -> None:
    bow = {}
    for sentence in sentence:
        tokens = nltk.word_tokenize(sentence)
        for token in tokens:
            if token not in bow.keys():
                bow[token] = 1
            else:
                bow[token] += 1
    return bow
```

```
bow = print_bow(t)
print(f"Мешок слов:\n{bow}")
print(f'Обнаружено {len(bow)} уникальных токенов.')
```

Рисунок 14 – Программный скрипт по созданию словаря обращений клиентов на языке python

Для наглядности словаря слов обращений клиентов банка построим график облака ключевых слов.

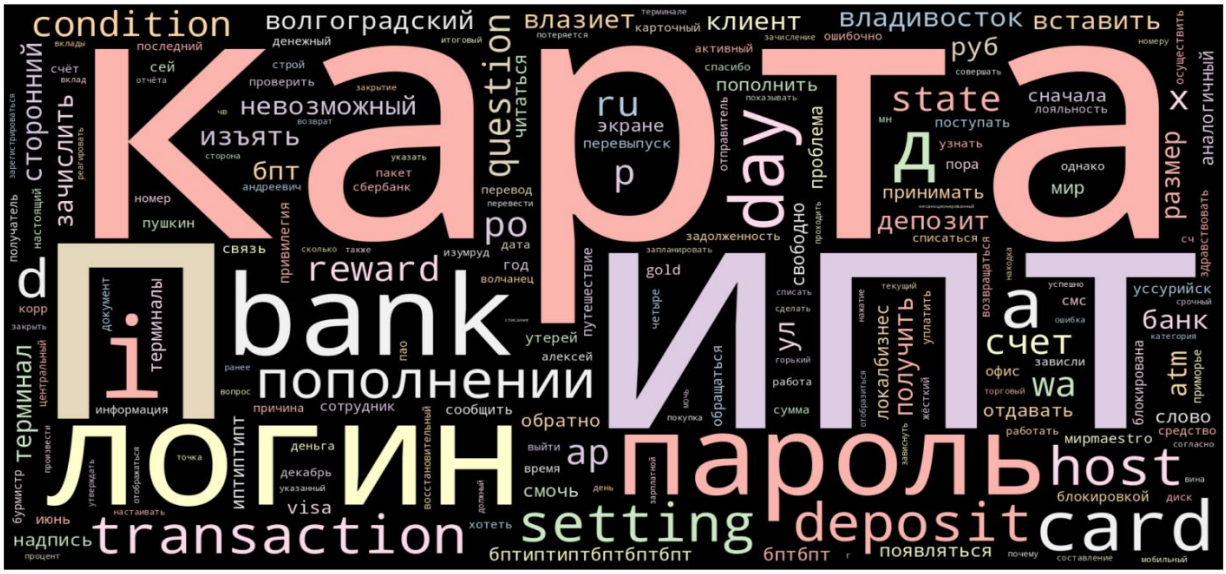


Рисунок 15 – Облако ключевых слов

Далее реализуем метод TF-IDF. Что будет представлять матрицу с оценками важности слова в корпусе. Первый шаг – это поиск значения IDF для наиболее часто вручаемых слов в корпусе (используем 50 наиболее часто встречающихся слов). Вторым шагом является создание словаря TF для каждого слова. В словаре TF ключом будут наиболее часто встречающиеся слова. Третий шаг, когда уже есть значения IDF всех слов, а также значения TF каждого слова в предложениях. Заключается в простом умножении значения IDF на значения TF [10, с. 298].

В итоге получаем 50x17395 – мерную матрицу A, где строки соответствуют векторам TF-IDF:

$$A = \begin{bmatrix} 0. & 0.3 & 0. & \dots & 0. & 0. & 0. \\ 0. & 0.18 & 0.35 & \dots & 0. & 0. & 0. \\ 0. & 0.11 & 0. & \dots & 0. & 0. & 0. \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0.1 & 0. & 0. & \dots & 0. & 0. & 0. \\ 0. & 0.09 & 0. & \dots & 0. & 0. & 0. \\ 0. & 0. & 0. & \dots & 0. & 0. & 0. \end{bmatrix}$$

Таким образом модель TF-IDF была реализована в Python, она является одной из наиболее используемых моделей преобразование теста в цифры. Далее эту модель можно использовать с применением алгоритмов машинного обучения и методов глубокого обучения, для решения задач классификации обращений клиентов.

3 Применение языковой модели BERT для классификации реестра обращений

При решении задачи классификации реестра обращений клиентов банка, был выбран атрибут предмет обращений «SUBJECT» для ее классификации, на основе текста обращений «ABOUT».

Для сравнительного тестирования были отобраны несколько моделей, обученных на данных разного размера. Приведём их основные характеристики.

1. Bert-base-multilingual-cased. Полноценная нейронная сеть BERT, обученная на большом корпусе текстов из Википедии и поддерживающая 104 языка, включая русский. Модель можно 3 использовать как для извлечения признаков, так и для решения каких-либо задач. Также эта модель чувствительна к регистру слов [11].

2. DeepPavlov/rubert-base-cased-sentence. Модель, основанная на BERT и обученная специально для русского языка. Может взаимодействовать с предложениями и чувствительна к регистру [12].

3. Sberbank-ai/sbert_large_nlu_ru. Реализация BERT, обученная на расширенном наборе данных, предназначена для получения векторных представлений предложений. Модель не чувствительна к регистру [13].

Вся техническая реализация осуществлялась на языке программирования Python при использовании дистрибутива Anaconda с помощью которой создавалась среда разработки и тестирования. В рамках этой среды были установлены различные библиотеки, специализированные в области машинного обучения, обработки естественного языка и визуализации результатов исследования.

3.1 Предварительная подготовка реестра обращений

Реализуем предварительный этап подготовки реестра обращений для модели BERT описанный в подразделе 1.4.

1. Этап токенизации. Для этого воспользуемся уже предобученным токенизатором, идущим вместе с моделью семейства BERT.

2. Этап добавления токенов [CLS], [SEP]. Каждый текст обращения мы приведём к виду, который требуется Берту, затем разобъём выборку на обучающую и тестовую, для последующей проверки качества.

Специальные токены Берта:

[CLS] - начало последовательности | [SEP] - разделение двух предложений

Мы будем окружать наши обращения этими токенами. Далее загрузим наш токенизатор. Он идёт вместе с предобученной моделью. Загружать будем с помощью универсального метода Pytorch torch.hub.load.

После токенизации, нам нужно выбрать размер входной последовательности, подаваемой нейросети. Он должен быть фиксированным, но наши обращения имеют различную длину. Поэтому построим гистограмму распределение количества токенов в обращении и попытаемся определить оптимальный размер (рисунок 16).

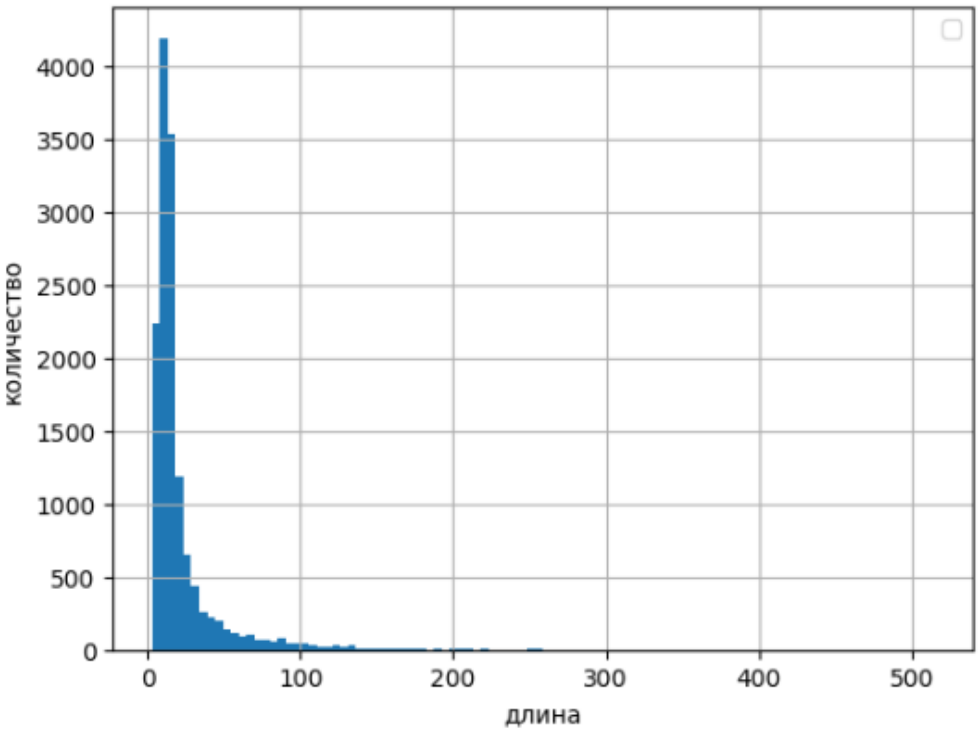


Рисунок 16 – Распределении количества обращений на длину обращений

В нашей задаче используются длинные последовательности, поэтому мы не можем брать максимальную длину, это может быть критично - мы должны учитывать, хватит ли машине памяти для обработки последовательностей большого размера.

Выбрав размер (100), мы приведём все обращения к единому виду: последовательности большого размера будут обрезаны, последовательности малого размера дополнятся паддингами – специальными нулевыми токенами.

В конечном этапе получим векторное представление слов, т. е. матрицу размерностью (длина)х(количество) обращений.

3.2 Обучение модели BERT

Далее загружаем веса предобученных моделей и запускаем процесс дообучения на наших данных. Так как в результате мы хотим получить не языковую модель, а мультиклассовый классификатор, укажем это при настройках. В библиотеке Transformers уже имплементированы классы для различных задач. Нам понадобится AutoModelForSequenceClassification.

Использовав этот класс, мы возьмём предобученный BERT, добавив ему на выход один полносвязный слой, который и будет решать нашу задачу классификации. По умолчанию обёртка AutoModelForSequenceClassification (или BertForSequenceClassification) использует бинарную классификацию. Нам же нужна мультиклассовая, поэтому укажем это в конфигурации.

Обучение модели происходит за 4 эпохи: так как набор данных небольшой, при большем количестве эпох модель может переобучиться, а при меньшем можно не увидеть динамики обучения. После каждой эпохи модель сохраняется, чтобы была возможность проанализировать результаты, полученные на разных эпохах.

На рисунке 17 изображен процесс обучения моделей, как видно модели обучились достаточно быстро т. к. последние итерации идут лишь флуктуации без улучшений качества модели.

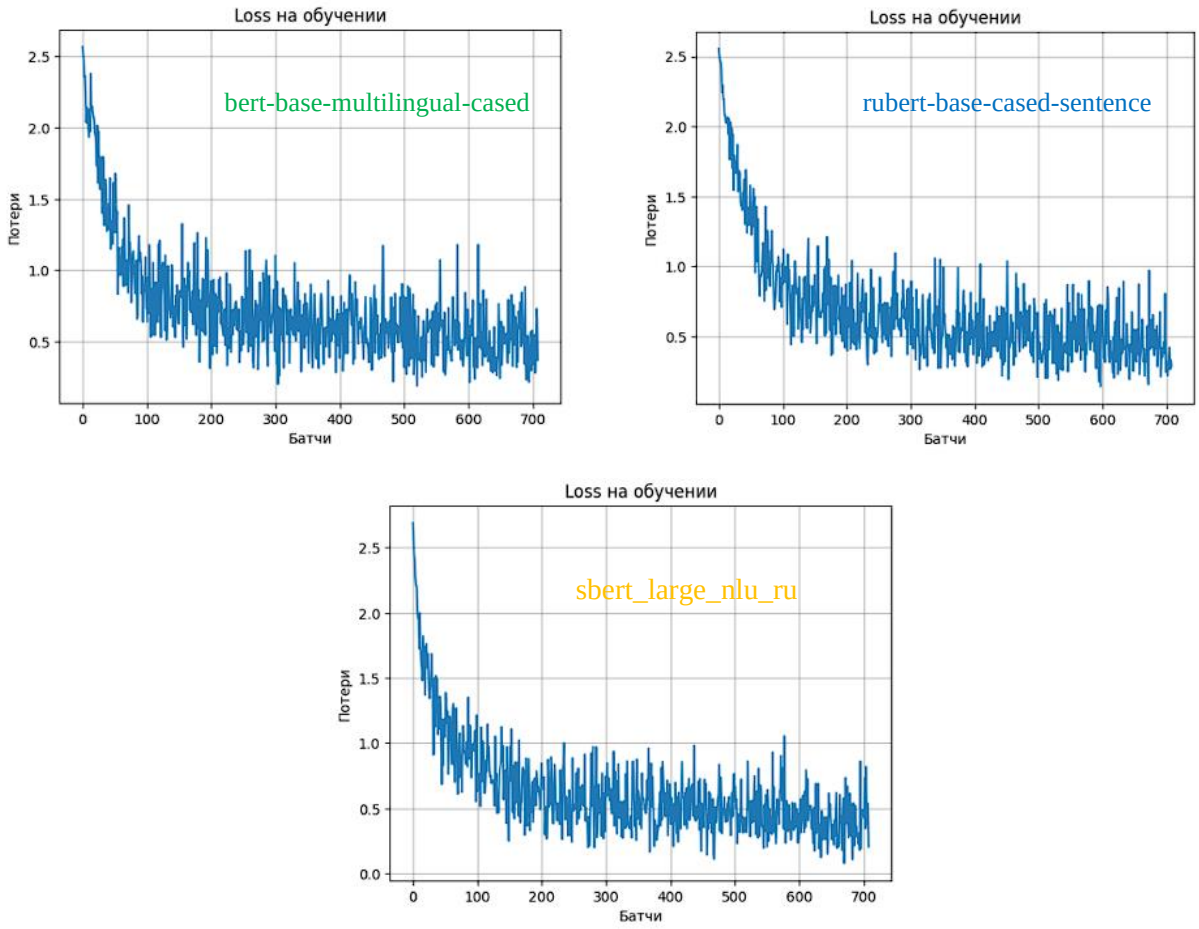


Рисунок 17 – Графики функции потерь во время обучении моделей BERT

3.3 Оценка качества модели BERT

Для оценки качества модели, в силу несбалансированности исходных данных, использование доли правильных результатов (accuracy) в качестве метрики эффективности модели может дать необъективный результат, поэтому использовались оценка качества классификации каждого класса из валидационной выборки данных по следующим метрикам:

F_1-score хороша тем, что близка к 1, когда точность и полнота близки к единице, и близка к 0, если один из аргументов близок к нулю, которая вычисляет по выражению 2.

Precision (точность, 3) – это доля меток, действительно принадлежащих этому классу, ко всем меткам, которые модель посчитала принадлежащими к данному классу.

Recall (полнота, 4) – это доля объектов, определенных как принадлежащие к данному классу, ко всем объектам этого класса из выборки.

$$F_1 = 2\,\frac{\text{Точность} * \text{Полнота}}{\text{Точноть} + \text{Полнота}} \tag{2}$$

$$\text{Точноть} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Полнота} = \frac{TP}{TP + FN} \tag{4}$$

В таблице 2 представлен окончательный отчет о качестве классификации моделей разработанной системой. Где 1 – модель Bert-base-multilingual-cased, 2 – модель rubert-base-cased-sentence и 3 – модель sbert_large_nlu_ru. Колонка «Support» отражает количество элементов данного класса в выборке.

Модель верно определяет предмет обращения в 86% случаев.

Таблица 2 – Отчет качества классификации модели на тестовой выборке

Название класса	**Precision**			**Recall**			**F1-score**			**Support**
	1	2	3	1	2	3	1	2	3	
карта	0.98	0.98	0.96	0.91	0.91	0.92	0.95	0.95	0.94	858
п/к	0.66	0.67	0.67	0.74	0.81	0.77	0.70	0.74	0.72	384
atm	0.99	0.98	1.00	0.99	0.98	0.99	0.99	0.98	0.99	369
ипт/бтп/птбск	0.99	0.99	0.98	1.00	0.99	1.00	0.99	0.99	0.99	276
смс	0.78	0.78	0.88	0.88	0.91	0.90	0.83	0.84	0.89	144
мобильное приложение	0.70	0.70	0.70	0.73	0.85	0.86	0.71	0.77	0.77	131
pos	0.63	0.69	0.81	0.66	0.63	0.63	0.64	0.66	0.71	90
переводы	0.49	0.74	0.71	0.49	0.60	0.49	0.49	0.66	0.58	65
интернет банк	0.61	0.76	0.77	0.74	0.63	0.65	0.67	0.69	0.70	62
сервис	0.60	0.66	0.55	0.78	0.60	0.79	0.68	0.63	0.65	58
кредитование фл	0.84	0.89	0.72	0.46	0.69	0.60	0.59	0.77	0.66	35
касса	0.00	0.64	0.50	0.00	0.28	0.04	0.00	0.39	0.07	25
вклады	0.90	0.88	0.81	0.41	0.32	0.59	0.56	0.47	0.68	22
Общий взвешенный	**0.85**	**0.87**	**0.87**	**0.85**	**0.87**	**0.87**	**0.85**	**0.87**	**0.86**	2519

Как видно, лучше всего с задачей справилась реализация BERT, предобученная специально для русского языка. Благодаря этому набору модели было проще выделить закономерности, характерные для нашей конкретной задачи.

Заметим, что дисбаланс классов сильно повлиял на качество моделей. Для получения более точных предсказаний необходимо обогащать обучающий набор данных новыми наблюдениями.

В ходе проведения экспериментов также были предприняты попытки исключить из набора данных те классы, которые представлены малым количеством записей из-за чего модель их плохо предсказывала. Это помогло улучшить общие показателей качества модели. Представленная выше стратегия является компромиссом между точностью предсказаний и учетом количества классов в модели.

Заключение

Настоящая выпускная квалификационная работа посвящена исследованию обработки естественного языка и применение систем автоматической обработки естественного языка в рассматриваемых задачах. После анализа различных методов, было принято решение использовать языковые модели в качестве основы разработки моделей для решения задачи классификации.

В результате была разработана система на языке Python, которая автоматизирует процесс классификации предмета обращения на основе текста обращения клиента банка, на представленных данных компанией ПАО АКБ «Приморье», обученные модели показали хорошую точность на сбалансированных классах.

Практически полученные результаты удовлетворяют цели работы, модели справляются с поставленной задачей и их можно использовать на практике. Исследование показало, что языковые модели семейства BERT не сложны в реализации и обладают высокой эффективностью при решении задач обработки естественного языка.

Для успешного выполнения роботы были выполнены все задачи, которые были поставлены в начале:

- исследованы теоретические аспекты и задачи обработки естественного языка;
- проанализированы существующие системы автоматический обработки естественного языка;
- описана общая характеристика компании ПАО АКБ «Приморье»;
- реализованы этапы обработки и анализа естественного языка;
- применена система искусственного интеллекта для решения задач обработки естественного языка;
- сделаны вывод по проведенной работе.

Таким образом, все поставленные цели и задачи настоящей работы выполнены.

Список литературы

1. Рассел, М. Data Mining. Извлечение информации из Facebook, Twitter, LinkedIn, Instagram, GitHub: практическое руководство / М. Рассел, М. Классен. - Санкт-Петербург: Питер, 2020. - 464 с. - (Серия «IT для бизнеса»). - ISBN 978-5-4461-1246-3. - Текст: электронный. - URL: https://znanium.com/catalog/product/1856794 (дата обращения: 01.06.2023).
2. Ганегедара, Т. Обработка естественного языка с TensorFlow: монография / Т. Ганегедара; пер. с анг. В. С. Яценкова. - Москва: ДМК Пресс, 2020. - 382 с. - ISBN 978-5-97060-756-5. - Текст: электронный. - URL: https://znanium.com/catalog/product/1094940 (дата обращения: 01.06.2023).
3. Васильев, Ю. Обработка естественного языка. Python и spaCy на практике. — СПб.: Пи тер, 2021. — 256 с.: ил. — (Серия «Библиотека программиста»). - ISBN 978-5-4461-1506-8. (дата обращения: 01.06.2023).
4. Бахтизин А. Р. , Брагин А. В. , Макаров В. Л. Большие языковые модели четвёртого поколения как новый инструмент в научной работе // Искусственные общества. – 2023. – Т. 18. – Выпуск 1. URL: https://artsoc.jes.su/s207751800025046-9-1/. DOI: 10.18254/S207751800025046-9 (дата обращения: 01.06.2023).
5. Гольдберг Й. Нейросетевые методы в обработке естественного языка /пер. с анг. А. А. Слинкина. – М.: ДМК Пресс, 2019. – 282 с.: ил. - ISBN 978-5-97060-754-1. (дата обращения: 01.06.2023).
6. Косых Н.Е. Применение модели дистилляций знаний BERT для анализа настроений текста // Вестник воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2022. – С. 139-151. – ISSN: 1995-5499. URL: https://elibrary.ru/item.asp?id=49612811 (дата обращения: 01.06.2023).
7. Ярушкина, Н. Г. Применение языковых моделей word2vec и BERT в задаче сентимент-анализа текстовых сообщений социальных сетей / Н. Г. Ярушкина, В. С. Мошкин, А. А. Константинов // Автоматизация

процессов управления. – 2020. – № 3(61). – С. 60-69. – DOI 10.35752/1991-2927-2020-3-61-60-69. (дата обращения: 01.06.2023).

8. Бессмертный, И. А. Методы квантового формализма в информационном поиске и обработке текстов на естественных языках / И. А. Бессмертный, А. В. Васильев, Ю. А. Королева [и др.] // Известия высших учебных заведений. Приборостроение. – 2019. – Т. 62. – № 8. – С. 702–709. – DOI 10.17586/0021-3454-2019-62-8-702-709. (дата обращения: 01.06.2023).

9. О банке [Электронный ресурс] // – URL: https://www.primbank.ru/o-banke (дата обращения – 01.06.2023)

10. Паттерсон Дж., Гибсон А. Глубокое обучение с точки зрения практика / пер. с анг. А. А. Слинкина. – М.: ДМК Пресс, 2018. – 418 с.: ил. - ISBN 978-5-97060-481-6. (дата обращения: 01.06.2023).

11. Документация модели bert-base-multilingual-cased [Электронный ресурс]: документация. – Режим доступа URL: https://huggingface.co/bertbase-multilingual-cased (дата обращения – 01.06.2023)

12. Документация модели DeepPavlov/rubert-base-cased-sentence [Электронный ресурс]: документация. – Режим доступа URL: https://huggingface.co/DeepPavlov/rubert-base-cased-sentence (дата обращения – 01.06.2023)

13. Документация модели sberbank-ai/sbert_large_nlu_ru [Электронный ресурс]: документация. – Режим доступа URL: https://huggingface.co/sberbank-ai/sbert_large_nlu_ru (дата обращения – 01.06.2023)

Приложение А

Организационная структура ПАО АКБ «Приморье»