**СОДЕРЖАНИЕ**

[СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ 9](#_Toc168324869)

[ВВЕДЕНИЕ 10](#_Toc168324870)

[1 Анализ предметной области 12](#_Toc168324871)

[1.1 Рассмотрение задачи анализа тональности текста 12](#_Toc168324872)

[1.1.1 Методы анализа тональности текста 12](#_Toc168324873)

[1.1.1 Проблемы анализа тональности текста 13](#_Toc168324874)

[1.2 Обзор методов анализа тональности текста 14](#_Toc168324875)

[1.2.1 Методы, основанные на словарях и правилах 14](#_Toc168324876)

[1.2.2 Методы, основанные на машинном обучении без учителя 15](#_Toc168324877)

[1.2.3 Методы, основанные на машинном обучении с учителем 15](#_Toc168324878)

[1.2.4 Гибридный метод 17](#_Toc168324879)

[1.3 Сравнение методов анализа тональности текста 17](#_Toc168324880)

[1.3.1 Логистическая регрессия, LSTM, CNN, Transformers 18](#_Toc168324881)

[1.3.2 Bag of Words 21](#_Toc168324882)

[1.3.3 Методы, основанные на словарях и правилах 22](#_Toc168324883)

[1.3.4 Итоговое сравнение методов 23](#_Toc168324884)

[1.4 Обзор моделей на основе архитектуры Transformers 23](#_Toc168324885)

[1.5 Формирование палитры эмоций 25](#_Toc168324886)

[2 Проектирование 29](#_Toc168324887)

[2.1 Диаграмма компонентов UML клиентской части 31](#_Toc168324888)

[2.2 Диаграмма компонентов UML серверной части 31](#_Toc168324889)

[2.3 Общая диаграмма компонентов UML 32](#_Toc168324890)

[2.4 Рассмотрение приложения в нотации IDEF0 33](#_Toc168324891)

[2.5 Результаты проектирования системы 35](#_Toc168324892)

[3 Формирование датасета 36](#_Toc168324893)

[3.1 Датасеты с расширенной маркировкой 37](#_Toc168324894)

[3.2 Датасеты с тернарной маркировкой 39](#_Toc168324895)

[3.3 Датасеты с бинарной маркировкой 40](#_Toc168324896)

[3.4 Результат рассмотрения датасетов 42](#_Toc168324897)

[3.5 Формирование датасета, принципы обработки данных 42](#_Toc168324898)

[4 Реализация 44](#_Toc168324899)

[4.1 Обучение моделей 44](#_Toc168324900)

[4.1.1 Токенизация 45](#_Toc168324901)

[4.1.2 Валидация 46](#_Toc168324902)

[4.1.3 Рассмотрение гиперпараметров 47](#_Toc168324903)

[4.1.4 Петля обучения 47](#_Toc168324904)

[4.2 Сравнение выбранных моделей 48](#_Toc168324905)

[4.3 Реализация бота 48](#_Toc168324906)

[ЗАКЛЮЧЕНИЕ 51](#_Toc168324907)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 52](#_Toc168324908)

[Приложение А 54](#_Toc168324909)

# СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

SVM – метод опорных векторов

AUC – площадь под кривой ошибок

CNN – сверточные нейронные сети

LSTM – длительная кратковременная память

GRU – управляемые рекуррентные нейроны

ЕЯ – естественный язык

BoW – Bag of words

# ВВЕДЕНИЕ

Анализ и обработка текстовой информации является достаточно актуальным направлением в сфере информационных технологий. В наше время оно активно развивается, проводится множество исследований, разработок и поисков прикладного применения технологий данной среды. Одним из основных разделов данного направления является анализ тональности текста. Он активно применяется при сборе мнений пользователей о каком-либо продукте, либо же бренде, деятельности других людей и т. п. Анализ тональности текста может помочь более четко понять эмоциональную окраску, настроение человека, общий контекст.

На данный момент одним из самым популярных способов общения являются мессенджеры, люди частично переходят от формата живого общения, звонков и писем к взаимодействию в интернет-пространстве. Всемирная Сеть даёт людями возможность общаться из различных точек мира, при таком формате взаимодействия мы не видим и не слышим собеседника, из-за чего отсутствует возможность оценить настроение человека привычными методами – посмотреть на язык тела, услышать эмоциональную окраску голоса. При общении посредством текста человек легко может не заметить, что его слова оскорбили собеседника или вызвали негативные эмоции. Именно в этом может помочь анализ тональности переписок, данная технология позволит лучше оценивать настроение собеседника и контролировать собственные высказывания.

Технологии анализа тональности текста способствуют повышению качества коммуникации, в значительной степени в ситуациях, когда понимание эмоций и намерений собеседника имеет ключевую роль. В качестве примера, в сфере обслуживания клиентов, посредством сбора обратной связи и анализа эмоциональной окраски ответов можно понять удовлетворенность клиентов, а следовательно, и повлиять на нее. В сфере виртуального общения, помимо личных диалогов, анализ тональности текста может помочь на различных форумах или в крупных беседах для выявления комментариев, нарушающих правила сообщества более быстрым и эффективным методом.

Для корректной работы моделей машинного обучения требуется подобрать подходящий датасет, а также входные параметры. В случае, верно, подобранной комбинации результат окажется достаточно точным, чтобы его можно было применять в реальном общении. Помимо определения используемой технологии, датасетов и входных параметров также необходимо выбрать платформу, которая будет выступать в качестве интерфейса взаимодействия между системой и пользователем. В качестве такового будем использоваться мессенджер VK, в связи с тем, что он является одним из наиболее популярных в нашей стране.

Целью данной работы является проектирование и последующая разработка чат-бота для VK, который будет анализировать эмоциональный окрас текста в полученном текстовом сообщении.

Для достижения данной цели необходимо:

1. провести анализ существующих методов для определения эмоциональной окраски текста;
2. сравнить результаты эффективности различных методов;
3. подобрать или сформировать наиболее подходящие для данной задачи датасеты;
4. спроектировать архитектуру и функциональность приложения;
5. провести обучение модели на сформированном датасете;
6. реализовать спроектированную систему;
7. провести тестирование и оценку обученной модели.

# 1 Анализ предметной области

Данный раздел будет включать в себя рассмотрение предметной области анализа эмоциональной окраски текста. В частности, будут исследованы различные методы анализа тональности, проведено их сравнение на основе выбранных критериев.

## Рассмотрение задачи анализа тональности текста

Анализ тональности текста – частная задача компьютерной лингвистики, целью которой является определение субъективных мнений и эмоций человека. Её можно разбить на две основные группы: общий анализ и таргетированный анализ [1]. Общий анализ подразумевает под собой определение тональности текста целиком или его фрагментов. Таргетированный анализ направлен на определение отношения субъекта к конкретной цели. Таким образом, общий анализ необходим в случае, когда нам интересен эмоциональный окрас текста пользователя, для того чтобы лучше понять его настроение и состояние на момент написания текста. Таргетированный анализ необходим для выявления личного отношение человека к деталям, например в отзыве о посещении автосервиса следует обратить внимание на отношение пользователя к обслуживанию, скорости и качестве работы.

* + 1. Методы анализа тональности текста

Основные подходы в анализе тональности [2]:

1. основанный на словарях, основной целью является выявление тональных единиц в тексте и их подсчет;
2. основанный на правилах, анализ происходит исходя из набора правил, составленного экспертом-лингвистом;
3. основанный на машинном обучении, включает в себя три основные задачи: обучение с учителем, обучение без учителя. В анализе тональности наиболее распространен первый тип задачи;
4. гибридный, может включать в себя объединение вышеизложенных методов.
   * 1. Проблемы анализа тональности текста

Сложность анализа тональности разнится от выбора текста, например книги, документы, рецензии оценить проще чем короткие сообщения в переписке между людьми. Это можно объяснить наличием неформальных слов, сокращений, длиной сообщений при общении между людьми.

Можно выделить основные восемь проблем при анализе эмоциональности текста [3]:

1. неявные чувства и сарказм, некоторые предложения могут не нести в себе явной эмоциональной окраски, но могут содержать в себе неявное настроение;
2. мировоззрение, некоторые предложения могут быть эмоционально интерпретированы исходя из личного отношения субъекта к тому или иному явлению при отсутствии дополнительных маркеров;
3. контекст, некоторые предложения могут содержать слова, имеющие определенную эмоциональную окраску, но их нельзя использовать в анализе из-за контекста;
4. национальные особенности, некоторые слова могут нести различную эмоциональную окраску в зависимости от национальной принадлежности человека их написавшего;
5. ирреалис, слова, относящиеся к косвенному наклонению, являются противопоставлением достоверной ситуации, в связи с чем их эмоциональный окрас может определяться полярно;
6. субъективность, различие между фактом и личным мнением субъекта;
7. идентификация субъекта, предложение может содержать несколько субъектов для которых эмоциональная окраска одного слова будет разной;
8. работа с отрицанием, существуют слова, которые при употреблении перед другими могут изменить эмоциональную окраску, однако их использование не всегда приводит к вышеизложенному результату, иногда они могут не изменить полярность.

## Обзор методов анализа тональности текста

В данном разделе будут более детально рассмотрены методы анализа тональности текста, описанные в пункте 1.1.1, указаны их плюсы и минусы.

* + 1. Методы, основанные на словарях и правилах

В основе данных методов лежат три этапа [4]:

1. создание словаря оценочной лексики;
2. определить параметры словаря оценочной лексики;
3. классифицировать тексты по тональности на основе созданного словаря.

При создании словаря оценочной лексики используются корпуса текста, которые вручную обрабатываются экспертом. После обработки для всех слов, в соответствии с их классом тональности, вычисляются веса по методу RF [5]

, (1)

где – вес i-го термина для класса тональности C, a – количество документов, содержащих i-й термин и относящихся к классу C, b – количество документов, содержащих термин и не относящихся к классу C, – коэффициент, равный отношению количества текстов класса C, в которых встречается i-й термин, к количеству текстов противоположного класса, в которых этот термин также присутствует.

После того, как все слова получают свой вес, происходит их упорядочивание по убыванию. Из полученного листа выбираются слова, имеющие наиболее яркую эмоциональную окраску, после чего подбираются однокоренные к ним. Как только, все слова однокоренные слова добавлены, начинается последний этап – добавление слов, усиливающих или ослабляющих эмоциональную окраску, для них также необходимо установить веса.

Далее на основе словаря дается оценка корпусам текста, чтобы получить итоговую оценку текста достаточно сложить все весовые значения корпусов.

Основным плюсом данных методов является достаточно высокая точность при большом наборе словарей. Также можно отметить, что он является достаточно простым в использовании.

Основными недостатками являются: высокая трудоемкость при составлении словарей, недостаточно точное взаимодействие с контекстом, привязка к языку анализируемого текста.

* + 1. Методы, основанные на машинном обучении без учителя

Данные методы основаны на том, что общую тональность текста определяют слова, которые наиболее часто встречаются в нем. Алгоритмы, обучающиеся на тестовой выборке не знают присвоенных кластерам тональностей и ищут слова, которые наиболее часто встречаются в конкретном тексте, при этом встречающиеся лишь в ограниченном количестве текстов выборки. Таким образом данным словам присваивается наибольший вес, после чего определяется их тональность, на основе которой делается вывод о тональности всего текста.

Основным достоинством данных методов является простота подготовки датасета.

Основным недостатком является сравнительно низкая точность.

* + 1. Методы, основанные на машинном обучении с учителем

Методы, основанные на машинном обучении с учителем в отличии предыдущих, требуют обучающую выборку. Она представляет собой некие размеченные данные. Модель предварительно обучается на этой выборке, а затем способна решать задачи, работая с не тестовыми данными. Существует несколько классических моделей, относящихся к данному типу:

1. SVM;
2. логистическая регрессия;
3. случайный лес;
4. наивный байесовский классификатор;
5. метод максимума энтропии;
6. деревья решений.

Исследования в задаче определения тональности текста использовали вышеперечисленные модели, в качестве входных данных были использованы отзывы о товарах с интернет-магазинов, в качестве метрики оценивания – точность(accuracy) [6]. Результаты исследования отображены в таблице 1.

Таблица 1 – Результаты исследования

|  |  |
| --- | --- |
| Модель | Accuracy, % |
| Метод максимума энтропии | 72.60 |
| Случайный лес | 88.39 |
| Наивный байесовский классификатор | 75.50 |
| SVM | 91.15 |

Второе исследование, где в качестве входных данных были корпуса англоязычных текстов, а метрика оценивания – AUC показало результаты, отображенные в таблице 2.

Таблица 2 – Результаты исследования

|  |  |  |
| --- | --- | --- |
|  | AUC | |
| Модель | Обучающая выборка | Тестовая выборка |
| Логистическая регрессия | 0.93445 | 0.93445 |
| Деревья решений | 0.68204 | 0.65000 |
| Случайный лес | 0.90799 | 0.84000 |
| SVM | 0.89416 | 0.86167 |

На основе данных исследований можно сказать, что методы машинного обучения с учителем обладают достаточно высокой точностью.

Основными плюсами данных методов являются: высокая точность, возможность дообучения, возможность изменения тестовых выборок под конкретные задачи.

Основным недостатком является необходимость размечать данные для обучения.

* + 1. Гибридный метод

Гибридный метод объединяет в себе методы машинного обучения и анализа тональности текста. После формирования словаря и взвешивания происходит обучение классификатора.

Данный метод является крайне трудоемким, поскольку объединяет в себе две самые время затратные области из вышеописанных подходов при этом его точность находится на уровне ранее описанных.

## Сравнение методов анализа тональности текста

В предыдущем пункте были рассмотрены основные методы анализа тональности текста, коротко упомянуты их достоинства и недостатки. В общих чертах, методы, основанные на словарях и методы, основанные на машинном обучении с учителем, обладают достаточно высокой точностью, однако имеют общий недостаток – трудоёмкость при подготовлении данных. Кроме этого, методы, основанные на словарях, также являются узконаправленными. Методы, основанные на машинном обучении без учителя напротив, не требует особой подготовки данных, однако выдают не столь высокую точность. Гибридный метод является достаточно трудоемким, при этом не обладая точностью, значительно превышающей остальные. Для выбора наиболее эффективного метода необходимо установить критерии оценки.

Критериями будут выбраны: точность, сложность подготовки данных, универсальность, необходимость корпуса для обучения, трудоемкость, возможность эффективной работы при небольшом количестве входных данных.

Среди данных критериев ключевыми будут: точность, универсальность, возможность эффективной работы при небольшом количестве входных данных.

В рамках данной работы рассматриваются различные архитектуры, используемые для решения задач анализа тональности текста. Среди описанных ранее основных групп выделим некоторые архитектуры и методы. К методам, основанным на машинном обучении с учителем, будут отнесены: Логистическая регресссия, LSTM, CNN, Transformers. К методам, основанным на машинном обучении без учителя, будет отнесен классический представитель – Bag of Words. Поскольку словарные методы схожи друг с другом конкретная модель не будет выбрана. В процессе рассмотрения выбранных архитектур и методов также будут выделены их преимущества и недостатки на основе некоторых из вышеописанных критериев.

* + 1. Логистическая регрессия, LSTM, CNN, Transformers

Логистическая регрессия – это метод построения линейного классификатора, который широко применяется для вероятности принадлежности объектов к различным классам. В рамках данного метода для снижения эффекта переобучаемости, зачастую используется регуляризация. Метод регуляризации основан на рассмотрении параметров модели в качестве случайных векторов, с заданной плотностью распределения. Подобный подход дает более высокую стабильность результатов, а также позволяет уменьшить влияние параметров регуляризации на результат. После обучения модели происходит оценка, как правило, с использованием метрики AUC, поскольку для данного типа моделей данная метрика является типичной и эффективной.

LSTM – это разновидность рекуррентной нейронной сети, отличительной особенностью которой является возможность учитывать долгосрочные зависимости между элементами данных [8]. Данная архитектура нейронной сети эффективно справляется с проблемой затухания градиента при обучении на долгих последовательностях данных. Возможность решения вышеназванной проблемы кроется в особенности архитектуры LSTM. В частности, данный вид сетей имеет механизмы запоминания и забывания информации, которые состоят из четырех основных элементов: cell state – хранит в себе долгосрочную информацию, практически исключая ее изменения, forget state – решает, какую информацию необходимо забыть, input gate – решает, какую информацию необходимо добавить, output gate – решает.

Данная архитектура является широко распространённой в сфере обработки ЕЯ, она показывает достаточно высокую точность, однако сильно зависит от подбора параметров.

CNN – архитектура, которая требует малое количество параметров, что сокращает время на её обучение, при этом выдаёт достаточно высокую точность на некоторых выборках. Принцип её работы, заключается в присвоении корпусам тональности и разложении их на вектора признаков, которые исследует модель.

Данная архитектура популярна в решении задач анализа тональности, однако несмотря на удобства при работе с подбором параметров, существует необходимость предварительной подготовки текстовых корпусов, в частности составление их векторных представлений.

Transformers – архитектура, которой представлена на рисунке 1, состоит из кодировщика и декодировщик. Кодировщик состоит из стека, включающего в себя шесть одинаковых слоёв, каждый из которых имеет два подслоя. Первый подслой представляет механизм внимания, а второй – позиционную полностью связанную сеть, обеспечивающую прямую связь. Кодировщик получает на вход текст на исходном языке, после направляя его в декодировщик. Декодировщик также состоит из стека шести слоёв, однако помимо двух подслоев он обладает третьим, который выполняет функцию механизма внимания на выходе из стека. За счет наличия прямой связи между слоями и механизма понимания возможно обеспечить эффективный учет контекста для каждого отдельного слова, при этом гарантируя, что i-й слой, может «смотреть» только на предшествующие ему [10].

Модели, основывающиеся на архитектуре Transformers предварительно обучены под конкретные задачи, что упрощает работу, как с подбором параметров, так и с корпусами текста. При этом они обладают высокой точностью. Для дальнейшего использования в рамках конкретных задач NLP происходит fune-tuning моделей с использованием специально составленных датасетов.

Сравнивая представленные выше архитектуры, можно выделить ряд ключевых моментов:

1. логистическая регрессия достаточно проста в использовании, а также ее можно легко интерпретировать, основным же минусом является ограничения при работе с нелинейными зависимостями;
2. LSTM обладает высокой точностью, а также избегает проблему затухания градиента, однако требует значительное время для подбора параметров, в связи с чем данная архитектура значительно хуже подойдет для решения задач в условиях ограниченных временных ресурсов;
3. CNN не требует большого количества времени на подбор параметров, а также достаточно быстро обучается, как следствие они хорошо применимы в задачах, где скорость обучения крайне важна, однако для её эффективного использования необходимы большие трудозатраты при подготовке данных;
4. Transformers обладает крайне высокой точностью и возможностью учитывать контекст, относительной простотой в подготовке данных, однако требует значительные вычислительные ресурсы, что может ограничивать их использование в условиях ограниченных технических возможностей.

Обобщая вышеизложенное, каждая из рассмотренных архитектур обладает рядом преимуществ и недостатков. В зависимости от рассматриваемых задач в области анализа тональности текста каждая из модели, основанные на каждой из архитектур могут показывать высокую эффективность.

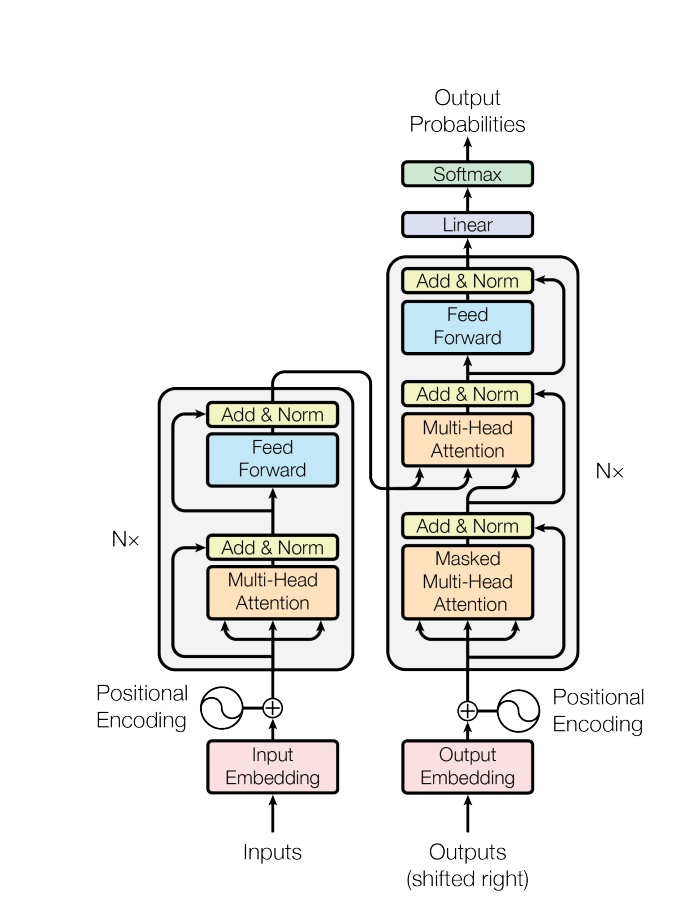


Рисунок 1 – Архитектура Transformers

* + 1. Bag of Words

Метод Bag of Words представляет собой подход, при котором происходит подсчет частоты появления каждого слова в тексте. Основной идеей данного метода является присвоение наибольшего веса тем словам, которые встречаются в тексте наибольшее количество раз. На основе проставленных таким образом весов определяется тональность текста. Таким образом данный метод не учитывает значительный ряд факторов, однако несмотря на это Bag of Words также применим в ряде задач анализа тональности текста. В сравнении с более сложными методами, которые были описаны ранее, он показывает меньшую точность, однако на определённых текстовых данных может быть равен, при этом его использование проще.

Причина простоты использования моделей на основе данного метода заключается в отсутствии необходимости значительных трудозатрат на подготовку текстовых корпусов. Этот факт значительно упрощает применение моделей, спроектированных на основе Bag of Words. Важно отметить, что данный метод крайне ограничен конкретными задачами, поскольку не способен учитывать контекстуальные зависимости между словами, а следовательно плохо применим к сложным задачам анализа тональности текста.

Обобщая вышеизложенное, основным преимуществом метода Bag of Words является его простота и низкая трудоемкость, однако в сравнении с методами из прошлой главы он крайне ограничен в применимости, а также показывает меньшую точность.

* + 1. Методы, основанные на словарях и правилах

Методы, основанные на словарях и правилах способны показывать высокую точность, сопоставимую и порой превосходящую методы машинного обучения с учителем, однако для их успешного использования требуется подготовка большого количества специализированных словарей. Этот процесс является крайне трудоемким и занимает достаточно большое количество времени, более того данные методы сильно зависимы как от конкретного языка, так и от тематики текста. Данный факт в значительной степени ограничивает их универсальность и применимость для различных контекстов. Однако данные методы крайне просты и легко интерпретируемы, также результаты их оценки как правило достаточно легко предсказать, что обеспечивает их надежность. Также методы, основанные на словарях и правилах не требуют больших наборов данных для обучения.

Обобщая вышеизложенное, словарные методы крайне надежны и предсказуемы, однако являются наименее универсальными из всех методов, о которых ранее говорилось. Для их эффективного использования важно предварительно оценивать применимость для каждой конкретной задачи.

* + 1. Итоговое сравнение методов

Итоговое сравнение вышеописанных методов представлено в таблице 3.

Таблица 3 – Сравнение методов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Метод | Точность | Сложность подготовки даннных | Универсальность | Необходимость корпуса для обучения | Трудоемкость | Возм. Эфф.  Раб. |
| Лог. Рег. | + | + | +- | + | +- | - |
| LSTM | + | - | + | + | + | - |
| CNN | + | + | + | + | + | - |
| Transf. | + | +- | + | + | +- | + |
| BoW | +- | +- | - | + | - | - |
| Слов.  Прав. | +- | + | - | - | + | - |

В результате проведения сравнительного анализа представленных архитектур можно утверждать, что наиболее эффективной является архитектура Transformers. Она обладает крайне высокой точностью, благодаря возможности учитывать контекст и сложные зависимости между словами, является крайне универсальной и применимой для широкого спектра задач анализа тональности текста благодаря возможности учитывать контекст, а также показывает достаточно высокую эффективность при работе с небольшим объемом данных для обучения благодаря механизмам внимания.

Основным минусом архитектуры Transformers является необходимость использовать значительное количество вычислительных ресурсов.

В трех критериях, которые были выделены ключевыми, архитектура Transformers обладает значительными преимуществами перед другими рассмотренными. При этом основной ее недостаток достаточно легко компенсировать за счет современных ресурсов, предоставляющих значительные вычислительные мощности.

## Обзор моделей на основе архитектуры Transformers

Архитектура Transformers все чаще применяется в решении задач анализа тональности текста. Модели, основанные на ней, показывают крайне высокую точность и производительность в различных задачах классификации текста. В частности, они крайне эффективны для решения задач, связанных с анализом тональности текста. В данном пункте будут рассмотрены следующие модели: rubert-tiny, rubert-base-cased-sentence, bert-base-multilingual, rubert-base и rubert-base-cased-conversational.

Целью данного обзора является определение наиболее эффективной модели для решения поставленной задачи. Для ее достижения будут описаны особенности выбранных моделей, на основе которых будет сформировано предложения о наиболее подходящей под задачу.

RuBert-tiny является версией модели Bert-base-multilingual-cased для анализа текстов на русском и английском языках. Данная модель была обучена на датасетах, содержащих пары коротких сообщений. В отличии от base моделей, RuBert-tiny включает в себя 3 слоя и работает приблизительно в 10 раз быстрее чем модели базового размера, однако ее точность ниже. Данная модель хорошо подойдет для решения простых NLP задач, а также в ситуациях, когда скорость и экономия вычислительных ресурсов важнее точности. Для использования данной модели в рамках прикладных задач необходимо провести обучение на маркированном датасете.

RuBert-base является классической версией моделей RuBert. Двумя наиболее популярными моделями являются RuBert-base от DeepPavlov и RuBert-base от ai-forever. Обе эти модели обучены на данных из Википедии, новостей, книг и некоторых других электронных ресурсах. Принципиальное отличие данных моделей заключается в вычислительных мощностях, использованных при обучении, модель от DeepPavlov обучена при маленьком параметре batch\_size, а также ограниченном количестве GPU, модель от ai-forever напротив. При этом обе модели показывают хорошие результате в задачах анализа тональности.

Bert-base-multilingual является классической моделью BERT, созданной Google. Для обучения использовались текстовые корпуса из Википедии. Данные для обучения не были промаркированы. Данная модель поддерживают работу с 102 языками. Поскольку она является мультиязычной, вероятно точность ее определения для прикладной задачи конкретного языка будет ниже, чем у модели, обученной специально под этот язык.

От DeepPavlov также есть модели RuBert-base-sentence и RuBert-base-conversational, обученные на основе его версии RuBert-base, однако корпуса текстов на которых они были обучены, совпадают с данными на которых обучена модель RuBert-base от ai-forever, поэтому остальные модели от DeepPavlov не будут рассматриваться в данной работе.

В результате обзора можно заключить, что для решения поставленной задачи большего всего подходит модель RuBert-base от ai-forever, эту же модель от DeepPavlov также можно использовать, однако, она вероятно покажет результат несколько хуже. Тем не менее далее на этапе реализации модель от DeepPavlov также будет рассмотрена. Модель RuBert-tiny можно использовать в случае ограниченности вычислительных ресурсов, поскольку Google предоставляет бесплатные вычислительные мощности данная модель не будет рассмотрена, а модель Bert-base-multilingual больше подходят для задач, в которых фигурируют несколько языков, однако также будет рассмотрена на этапе реализации. Суммируя – били выбраны три модели для дальнейшего рассмотрения, а именно Ru-bert-base от ai-forever и DeppPavlov, а также Bert-base-multilingual. Итоговый выбор модели на основе метрик будет происходить на этапе реализации.

## Формирование палитры эмоций

Задача выделения ключевых человеческих эмоций актуальна уже несколько десятилетий[11]. В работах многих ученых представляются различные классификации базовой эмоциональной палитры, а также причины возникновения эмоций. Одной из классических работ является «Descartes' Error: Emotion, Reason, and the Human Brain», автором которой является Антонио Дамасио. В данной книге рассматриваются причины возникновения эмоций с точки зрения физиологии, в качестве базовых эмоций выделены те, которые вызывают реакцию организма при их проявлении, например учащенное сердцебиение и резкое прерывистое дыхание, когда человек испытывает страх. Помимо страха Дамасио также выделяет радость, грусть и злость. Каждая из этих эмоций является крайне сильной и находит отражение на человеке, как в моральном плане, так и в реакции его тела.

Рассмотрим описанные выше эмоции с точки зрения задачи анализа тональности текста. Поскольку данные эмоции является крайне сильными, мы зачастую испытываем именно их, в различной степени.

Первоначально рассмотрим страх – данную эмоцию в тексте можно вполне однозначно классифицировать по наличию ряда ключевых слов: жутко, стремно, боюсь, пугать, страшно, кошмар, а также различные их формы и знаменательные части речи.

Злость можно отличить по следующим ключевым словам: ненавижу, сердиться, ярость, злиться, жестокий, сердитый, различные ругательства, а также различные их формы и знаменательные части речи.

Радость можно отличить по следующим ключевым словам: веселье, восторг, счастье, довольный, радостный, приятно, нравится, а также различные их формы и знаменательные части речи.

Грусть можно отличить по следующим ключевым словам: грустно, печально, мрачность, расстроенный, не в духе, а также различные их формы и знаменательные части речи.

Таким образом эмоции, названные в данной работе базовыми, не только оказывают сильный моральный и физический эффект на человека, а также достаточно четко отличимы и легко передаваемы в текстовой форме общения. Исходя из вышеизложенного данная базовая палитра эмоций отлично подходит для задачи анализа тональности текста.

Многие авторы, изучающие тему классификации эмоций, также выделяют вышеописанные, как основные, но помимо них включают и ряд других, в зависимости от метода исследования, например Пол Экман выделяет также отвращение и удивление. Достаточно популярной является классификация Роберта Платчика, в которой помимо описанной выше базовой палитры эмоции также выделяется другая вариация базовых эмоций, их различные степени и смежные между ними. Крайне интересной является классификация Лизы Фельдман Баррет, она значительно отличается тем, что оперирует не с категорией эмоций, а с чувствами, выделяя четыре основных: чувство безопасности, чувство угрозы, чувство уверенности, чувство неуверенности.

Рассмотрим данные вариации расширения базовой палитры эмоций с точки зрения задачи анализа тональности и вероятности определения данных эмоций в тексте.

Говоря о классификации Лизы Фельдман Баррет, чувства достаточно сложно классифицировать в тексте, поскольку они являются личными ощущениями каждого человека. Помимо этого, ощущение безопасности или же угрозы крайне сложно определить в текстовом формате. Чувства уверенности и неуверенности можно понять скорее по вербальным и невербальным признакам, их также достаточно тяжело определить в тексте.

Одной из ярких эмоций, которую выделяют в своих Классификациях Платчик и Экман, является отвращение, она достаточно легко определима по невербальным признакам, также её можно достаточно просто классифицировать в тексте по наличию следующих ключевых слов: ненавижу, гадость, мерзость, не нравиться, отвратительный, отталкивающий, фу, а также различные их формы и знаменательные части речи.

Помимо эмоций, которые оказывают сильное воздействие на человека, есть и менее заметные, однако крайне подходящие для рассмотрения в рамках задачи анализа тональности текста. Таковой является заинтересованность, данную эмоцию сложнее отличить по невербальным признакам, однако значительно проще по вербальным и тексту. Словами маркерами для её определения являются: интерес, любопытство, увлечение, вовлеченность, нравиться, а также различные их формы и знаменательные части речи.

В качестве характеристик для отбора эмоций, попавших в итоговую палитру, были выбраны: количество слов маркеров, а также частота их совпадений между различными эмоциями, представленными в таблице 4.

Таблица 4 – Выделение эмоций

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Эмоция | Страх | Злость | Радость | Грусть | Отвращение | Заинтересованность | Удивление | Возбуждение |
| Слова маркеры | Жутко,  стремно, боюсь, пугать, страшно, кошмар, тревога, паника, | ненавижу, сердиться, ярость, злиться, жестокий, различные ругательства, возмущение | веселье, восторг, счастье, довольный, радостный, приятно, нравится, | грустно, печально, мрачность, расстроенный, не в духе, жаль, тяжело, волнение | ненавижу, гадость, мерзость, не нравится, отвратительный, отталкивающий, фу, противный | интерес, любопытство, увлечение, вовлеченность, вау, удивительно | удивительно, неожиданно, внезапно, вау | волнение, экстаз, восторг, нетерпение, |
| Количество | 8 | 7 | 7 | 8 | 8 | 6 | 4 | 4 |
| Количество совпадений | 0 | 1 | 2 | 1 | 1 | 2 | 2 | 2 |

Обширное количество слов маркеров позволяет более однозначно определить принадлежность текста к конкретной эмоции. Совпадение слов в свою очередь вводит неоднозначность в определении, поэтому слова, у которых меньше всего слов маркеров и при этом больше повторений не попали в итоговую палитру.

В текстовом формате общения нередко встречаются содержащие исключительно фактическую информацию. Таким образом не все сообщения можно классифицировать с точки зрения наличия в них какой-либо эмоции. Для решения данной проблемы, в расширенную палитру эмоций добавлен «нулевой маркер», который будет говорить о нейтральности текста. Исходя из вышеизложенного в палитру эмоций добавляется нейтральность.

В результате данной работы была составлена следующая палитра эмоций: страх, злость, радость, грусть, отвращение, заинтересованность, нейтральность, которая позволяет классифицировать ряд основных, при этом различных человеческих эмоций.

# Проектирование

В данном разделе будут детально рассмотрены все этапы проектирования системы, начиная от общей архитектуры и заканчивая взаимодействием с пользователем и внутренними процессами. Описанием функциональных требований к приложению, принципов работы Longpoll VK API.

В данной системе выделяются две основные части: серверная, отвечающая за внутренние процессы, и клиентская, представляющая интерфейс взаимодействия с пользователем. Архитектура системы будет представлена в виде диаграмм UML, поскольку они представляют собой визуализацию структуры системы и взаимосвязей между ее компонентами, достаточно легко понять и проанализировать общую архитектуру. В работе будут представлены 3 диаграммы UML: диаграмма компонентов клиентской части, диаграмма компонентов серверной части и общая диаграмма компонентов. Помимо представления архитектуры будут описаны процессы взаимодействия пользователя с системой и внутренние процессы. Для этого будет использована нотация IDEF0, которая позволяет детально описать функции системы и их взаимосвязи. В частности, будут рассмотрены процессы взаимодействия пользователя с системой, состоящие из инициализации взаимодействия и обработки сообщений пользователя, а также внутренние системные процессы, состоящие из процесса анализа текста и процесса предсказания и отправки результатов.

Функциональные требования к разрабатываемой системе следующие:

1. система должна поддерживать инициализацию бота и подключение к серверам VK при запуске локальной машины;
2. наличие возможности аутентификации и авторизации на сервере VK через Longpoll API с помощью ключа доступа;
3. система должна быть способна получать сообщения от пользователей через VK;
4. система должна уметь различать типы сообщений, текстовые, фото, голосовые;
5. для текстовых сообщений система должна уметь анализировать их содержимое и определять эмоциональную окраску;
6. система должна отправлять пользователю ответ, содержащий эмоцию сообщения, которое он отправил;
7. система должна отправлять пользователю уведомление об ошибке, в виде текстового сообщения, если тип сообщения пользователя являлся не текстовым;
8. система должна стабильно поддерживать связь с сервером VK в процессе своей работы;
9. поскольку система использует Longpoll API она должна минимизировать количество запросов к серверу VK;
10. в процессе работы система должна вести базу данных с информацией о пользователях, в частности их идентификатор;
11. система должна эффективно обрабатывать сообщения в реальном времени, без значительных задержек;
12. система должна предоставлять удобный и понятный интерфейс.

В качестве интерфейса взаимодействия будет выступать VK бот, основанный на Longpoll API. Метод Longpoll позволяет взаимодействовать с сервером VK посредством отслеживания и обработки различных событий: добавление нового сообщения, установка/замена флагов сообщения, проверка статуса пользователя и некоторые другие, без необходимости постоянно опрашивать сервер.

Основной принцип работы Longpoll API выглядит следующим образом: отправление запроса на сервер VK, получение ответа о произошедшем событии, либо пустой ответ в обратном случае, повторное отправление запроса. При этом ответ ожидается в течение конкретно заданного времени.

Благодаря подобной структуре можно быть постоянно подключенным к серверу, получать обновления в реальном времени, при этом запросы к серверу минимизируются, а следовательно, происходит экономия ресурсов.

## Диаграмма компонентов UML клиентской части

Архитектура клиентской части состоит из нескольких ключевых элементов. Во-первых, бот, который выступает в качестве интерфейса для взаимодействия с пользователем. Во-вторых, метод отправления сообщений, который обеспечивает передачу информации пользователю. В-третьих, база данных с информацией о пользователях. В-четвертых, общая конфигурация, которая объединяет все компоненты и настраивает их взаимодействие. Основной принцип работы данной части системы заключается в следующем – бот получает сообщение от пользователя, а также его VK id, после чего отправляет эти данные на серверную часть для последующей обработки, а также отвечает пользователю результатом, который приходит с серверной части. Схема архитектуры клиентской части представлена на рисунке 2.

Изображение выглядит как диаграмма, линия, План, Технический чертеж

Автоматически созданное описание

Рисунок 2 – UML диаграмма клиентской части

## Диаграмма компонентов UML серверной части

Архитектура серверной части состоит из нескольких ключевых элементов. Во-первых, анализатор текста, который является ключевым компонентом и служит для передачи поступающих в данных в компоненты, которые занимаются их обработкой. Следующим элементом является токенизатор, который преобразует полученные данные для дальнейшей обработки. Также ключевым слоем является классификатор, который выдает предсказания на основе полученных от токенизатора данных, а также полученной от модели развесовки данных. Модель обеспечивает корректную интерпретацию и анализ. Схема архитектуры серверной части представлена на рисунке 3.

Изображение выглядит как диаграмма, линия, снимок экрана, График

Автоматически созданное описание

Рисунок 3 – UML диаграмма серверной части

## Общая диаграмма компонентов UML

Объединяя две эти части получаем общую архитектуру системы. Общий принцип работы всей системы выглядит следующим образом:

1. Бот получает сообщение от пользователя, которое попадает в компонент Text Analyzer и записывает в файл Database идентификатор пользователя. Соединение с VK происходит за счет ключа, хранящегося в файле Config
2. Text Analyzer проверяет тип данных сообщения и отправляет его в компонент Tokenize.
3. Tokenize производит токенизацию текста, разделяя его на элементы, для дальнейшей обработки.
4. Компонент Classificator проводит развесовку слов на основе компонента Model, что позволяет классифицировать текст.
5. Компонент Text Analyzer передает боту предсказанную эмоцию, предсказанную после анализа текста.
6. Компонент Bot отправляет эмоцию клиенту с помощью компонента Sender

Общая архитектура представлена на рисунке 4.

Изображение выглядит как диаграмма, линия, зарисовка, План

Автоматически созданное описание

Рисунок 4 – Общая UML диаграмма компонентов

## Рассмотрение приложения в нотации IDEF0

Для более детального рассмотрения основных процессов взаимодействия пользователя с системой, а также внутренних процессов используется нотация IDEF0. Схема выглядит следующим образом: пользователь отправляет сообщение, после чего происходит его обработка, затем формируется ответ, который содержит в себе эмоцию, и отправляется обратно пользователю. Общая схема взаимодействия представлена на рисунке 5.

Изображение выглядит как текст, линия, снимок экрана, диаграмма

Автоматически созданное описание

Рисунок 5 – Общая схема IDEF0

После того, как пользователь открыл чат с ботом, ему предлагается нажать на кнопку «Начать». Система проверяет, есть ли пользователь в базе данных, в случае отсутствия записей о пользователе, отправляется приветственное сообщение, записи о пользователе есть в базе данных, система ждет сообщения от пользователя для дальнейшего взаимодействия.

Схема начала работы с пользователем представлена на рисунке 6.

Изображение выглядит как текст, диаграмма, линия, чек

Автоматически созданное описание

Рисунок 6 – Получение сообщения от пользователя

После получения сообщения от пользователя система определяет тип данных, которые в нем содержатся. Если полученные данные являются текстовыми, сообщение перенаправляется в анализатор, если тип данных иной, в частности: голосовое сообщение, фотография, видео, пользователю отсылается сообщение с ошибкой и просьбой ввести корректные данные. Схема реакции системы на сообщения от пользователя представлена на рисунке 7.

Изображение выглядит как текст, диаграмма, линия, чек

Автоматически созданное описание

Рисунок 7 – Обработка сообщения пользователя

После получения и обработки сообщения оно направляется в анализатор, где осуществляется токенизация текста. В процессе токенизации текст разбивается на отдельные элементы – токены, которые передаются в классификатор. Классификатор использую внутренние алгоритмы и модель производит предсказание эмоциональной окраски текста. На основе этого предсказания формируется ответное сообщение, которое отправляется пользователю. Схема работы анализатора представлена на рисунке 8.

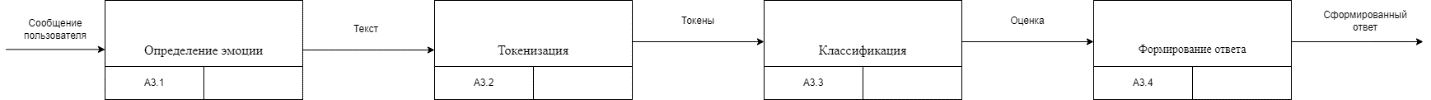


Рисунок 8 – Работа анализатора

## Результаты проектирования системы

В ходе данной работы была спроектирована и разработана система, состоящая из двух основных частей: серверной и клиентской. Данная система обеспечивает эффективное взаимодействие пользователя с ботом VK посредством Longpoll API. Описание архитектуры системы было представлено в виде UML диаграмм, а процессы взаимодействия пользователя и внутренние системные процессы описаны с помощью нотации IDEF0.

Архитектура клиентской части, включающая бот-интерфейс, метод отправки сообщений, базу данных и конфигурационный файл, обеспечивает получение сообщений от пользователей и взаимодействие с серверной частью для обработки данных. Серверная часть, состоящая из анализатора текста, токенизатора, классификатора и модели, отвечает за обработку поступающих данных и предсказание эмоций на основе введённых пользователем текстов.

Использование Longpoll API позволяет системе поддерживать постоянное подключение к серверу VK, минимизируя количество запросов и обеспечивая экономию ресурсов. Это позволяет системе оперативно реагировать на действия пользователей и обеспечивать обмен сообщениями в реальном времени.

Объединяя все элементы системы, мы получаем целостную архитектуру, способную эффективно обрабатывать текстовые сообщения, предсказывать эмоции и взаимодействовать с пользователем через удобный интерфейс бота VK.

# Формирование датасета

Для обучения модели помимо подбора гиперпараметров также необходимо подобрать или сформировать датасет, в котором данные будут качественно промаркированы, а также будут соответствовать поставленной задаче. От качества маркировки в значительной степени зависит итоговый результат точности модели на реальных задачах.

Этот раздел будет включать в себя рассмотрение нескольких датасетов из открытой библиотеки HuggingFace, которые будут содержать в себе различные маркировки и данные представленные на различных языках. Для каждого датасета будет подробное рассмотрение его применимости для решения задачи определения эмоций пользователя на основе сообщений. В случае, если рассмотренные датасеты по какой-либо причине не будут подходить для решения поставленной задачи в полной мере, будет сформирован датасет путем объединения корпусов данных из различных готовых наборов с последующей ручной маркировкой. Рассматриваемые датасеты представлены в таблице 5.

Таблица 5 – Рассматриваемые датасеты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название датасета | Описание | Вид маркировки | Количество данных | Источник |
| 1 | 2 | 3 | 4 | 5 |
| Imdb | Набор данных рецензий на фильм. Все данные на английском языке. | Бинарная | 25000 | HuggingFace |
| Auditor\_sentiment | Набор данных из англоязычных финансовых новостей. | Тернарная | 3880 | HuggingFace |
| Healthcare\_facilities\_reviews | Набор данных содержит отзывы пользователей о медицинских учреждениях. | Бинарная | 70600 | HuggingFace |

Продолжение таблицы 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| cedr | Набор данных из различных социальных источников | Расширенный | 9410 | HuggingFace |
| Ru\_go\_emotions | Перевод датасета Go\_emotions | Расширенный | 211000 | HuggingFace |
| Russian\_twitter\_sentiment | Набор данных, содержащий сообщения из Твиттера | Бинарная | 274890 | Kaggle |
| Twitter Sentiment Analysis | Набор данных, содержащий мнение пользователей об играх | Тернарная | 69491 | Kaggle |
| Sentiment Analysis in Russian | Новости на русском языке | Тернарная | 8000 | Kaggle |

Все представленные датасеты можно разделить на 3 категории по виду маркировки: бинарная, тернарная, расширенная. Рассматриваемая задача подразумевает под собой использование расширенной маркировки, однако датасетов с подобной маркировкой в открытом доступе достаточно мало. Достаточно важным критерием является язык, на котором составлен набор данных, поскольку полноценный перевод отнимает достаточно много времени, более того, в результате могут возникнуть несоответствия с маркировкой, что приведет к еще большим временным затратам.

## Датасеты с расширенной маркировкой

Датасет Ru\_go\_emotions является самым большим среди всех корпусов текстов, переведенных на русский язык и находящихся в открытом доступе, в сфере анализа тональности эмоций. Данный датасет является переводом датасета от Google – GoEmotions, который содержит в себе сообщения с форума Reddit. В Ru\_go\_emotions промаркированы 27 эмоций: восхищение, веселье, злость, раздражение, одобрение, забота, непонимание, любопытство, желание, разочарование, неодобрение, отвращение, смущение, возбуждение, страх, признательность, горе, радость, любовь, нервозность, оптимизм, гордость, осознание, облегчение, раскаяние, грусть, удивление, нейтральность. Данная палитра эмоций содержит в себе некоторые элементы, которые достаточно тяжело отличить друг от друга в текстовом формате, в частности: неодобрение и разочарование, разочарование и грусть, забота и любовь, восхищение и радость и т. д. Однако большая часть данных из этого набора крайне эффективно подходит для решения задачи определения эмоций пользователя, поскольку сами данные представляют собой короткие сообщения от пользователей форума/социальной сети, текста из данного корпуса затрагивают обширное количество тем, в связи с чем формат данных в этом наборе соответствует требованиям для обучения выбранной модели. Одной из ключевых проблем данного датасета являются частые повторы сообщений, в связи с чем требуется тщательная обработка данных, помимо этого разметка датасета не соответствует методу разметки, используемой сети, следовательно маркировку необходимо проводить вручную.

Датасет Cedr состоит из сообщений с различных социальных ресурсов, данные в нем промаркированы по 5 эмоциям: радость, грусть, удивление, страх и злость. Каждый элемент в данной палитре эмоций крайне легко отличим от другого, что является несомненным плюсом этого текстового корпуса. Ещё одним достоинством является проведенная для всех сообщений лемматизация, представление в отдельном столбце набора данных. Достаточно часто в процессе общения в интернете возникают сообщения, которые не несут никакой эмоциональной окраски, в них могут отсутствовать слова маркеры или же явный контекст, указывающий на принадлежность текста к той или иной эмоции. В таком случае необходимо нулевое состояние – нейтральность, которая в данном датасете обозначается отсутсвием данных. Формат разметки не подходит для выбранной модели, следовательно маркировку придется проводить вручную.

Датасеты относящиеся к расширенной маркировке наиболее близко подходят для поставленной задачи, несмотря на то что они требуют тщательной обработки и ручной маркировки, сами данные, которые в них содержаться являются достаточно удобными, поскольку уже содержат в себе ряд эмоций, рассматриваемых в данной работе.

## Датасеты с тернарной маркировкой

Датасеты, которые содержат в себе маркировку из трех эмоций: позитивная, нейтральная, негативная, относятся к тернарным, содержащим три класса тональности. Как правило для определения тональности текста достаточно данного количества классов, в связи с чем подобные датасеты встречаются значительно чаще, чем наборы данных содержавшие маркировки с большим количеством эмоций. В рамках поставленной задачи данные датасеты также могу быть полезны, нередко текстовые данные, которые содержаться в них могут быть расширены до большего количества маркировок.

Датасет Auditor\_sentiment – небольшой набор данных, состоящий из англоязычных финансовых новостей, промаркированных с использованием тернарной классификации. Текстовые блоки в этом датасете не являются большими, что удовлетворяют условию поставленной задачи, однако их содержание является совершенно не применимым для анализа эмоций сообщений в социальных сетях. Маркировки проставлены в соответствии с настроениями рынка, также в данных полностью отсутствуют слова маркеры, на основе которых можно было бы выделить эмоции.

Датасет Twitter Sentiment Analysis состоит из англоязычных сообщений с социальной сети Twitter. Основная часть данных представляет собой короткие тексты, большая часть из которых имеет явную эмоциональную окраску. Маркировку данного текстового корпуса возможно расширить по меньшей мере до 5 эмоций: радость, грусть, нейтральность, злость, веселье. Все этим эмоции, за исключением веселья, подходят для решения поставленной в работе задачи. Основной проблемой данного датасета является крайне большое количество практически полностью идентичных сообщений, как следствие требуется очень тщательная обработка, которая может занять достаточно много времени.

Датасет Sentiment Analysis in Russian состоит из русскоязычных новостей, большая часть предложений которого являются очень длинными. Несмотря на наличие большого количества слов, содержащих яркую эмоциональную окраску, данные из данного корпуса текста тяжело применимыми к поставленной задаче. Набор данных из Sentiment Analysis in Russian возможно использовать в рамках задачи определения эмоций сообщений при тщательной предварительной обработке, а именно разбиении текстовых корпусов на более маленькие, с сохранением основной эмоциональной нагрузки. Однако даже при таком методе взаимодействия достаточно большое количество данных тяжело интерпретировать вне рамок тернарной классификации.

Подводя итоги рассмотрения датасетов с тернарной маркировкой, в первую очередь они могут выступать источником дополнительных данных в случае возникновения трудностей с обработкой текстов датасетов, размеченных несколькими эмоциями. Однако важно учитывать, есть ли возможность расширения эмоционального спектра. Также практически во всех, рассмотренных наборах данных достаточно много нейтральных эмоций, при использовании датасетов из данного пункта для формирования итогового, будет необходимо значительно сократить количество маркеров нейтральности.

## Датасеты с бинарной маркировкой

Бинарная маркировка содержит в себе две основные эмоции: негативная и позитивная, на основе которой происходит оценка принадлежности текста к одному из этих классов. С одной стороны бинарная классификация уступает тернарной поскольку достаточно часто встречаются сообщения, не имеющие явной эмоциональной окраски, поэтому при решении сложных задач анализа тональности данное разбиение является малоэффективным, с другой стороны уменьшение количества классов при анализе увеличивает его точно для имеющихся классов. Существует достаточно много датасетов с бинарной маркировкой, в данном пункте будут рассмотрены три. Как и в случае с датасетами, данные которых размечены тернарной маркировкой, наборы данных с бинарной могут быть адаптируемы для расширения эмоционального спектра.

Датасет Imdb включает в себя англоязычные отзывы о фильмах с сайта Imdb. Данные в данном наборе являются достаточно специфичными, каждый текст представляет собой длинное и связное повествование, в рамках которого, оценивается фильм. Слова маркеры, а также контекстно эмоционально окрашенные фрагменты могут значительно меняться несколько раз в рамках одного текста. Это приводит к трудностям при попытке адаптировать данные из данного набора для поставленной в работе задачи.

Датасет healthcare\_facilities\_reviews представляет собой отзывы о посещении клиентами различных медицинских учреждений, данный набор данных, как и предыдущий является достаточно специфичным, однако может быть применен в рамках поставленной задачи. Поскольку тексты в данном наборе являются не очень длинными, а также содержат в себе явные слова маркеры, некоторые из этих данных могут быть использованы для расширения итогового датасета.

Датасет Russian\_twitter\_sentiment содержит в себе сообщения с социальной сети Twitter. Несмотря на бинарную маркировку, данный датасет может быть легко адаптирован под задачу анализа эмоций сообщений людей. Текста в этом наборе данных представляют собой короткие, ярко эмоционально окрашенные сообщения пользователей социальной сети, при рассмотрении ряда из них легко заметить, что в большинстве содержаться различные эмоции, в частности: радость, грусть, нейтральность, страх, злость, заинтересованность и некоторые другие. Для этого датасета можно провести расширение эмоционального спектра и успешно интегрировать его часть в итоговый. Более того, Russian\_twitter\_sentiment включает в себя данные, изначально собранные на русском языке, что значительно упрощает работу с ними. Таким образом, часть данного датасета будет рассмотрена для использования при формировании текстового корпуса для решения поставленной в работе задачи.

Подводя итоги рассмотрения датасетов с бинарной маркировкой, первый рассмотренной набор данных не может быть адаптирован под задачу данной работы, поскольку текстовые данные в нем не соответствуют поставленной задаче, второй, хоть и содержит часть, которую можно использовать все ещё затруднителен в обработке, поскольку потребуется значительное время для отделения данных, которые модно использовать, третий текстовый корпус можно использовать, поскольку его маркировку достаточно легко расширить до необходимой, а также формат данных отлично подходит к поставленной задаче.

## Результат рассмотрения датасетов

Среди рассмотренных наборов данных наиболее удобными в использовании являются Ru\_go\_emotions и Russian\_twitter\_sentiment. Первый текстовый корпус уже содержит в себе маркировку значительного количества эмоций, формат данных полностью удовлетворяет задаче, а за счет готовой разметки поиск необходимых сообщений значительно упрощается, второй корпус содержит в себе комментарии, изначально написанные на русском языке, что упрощает работу. При формировании итогового набора данных будут использоваться эти два датасета.

## Формирование датасета, принципы обработки данных

Рассмотрим сложности обработки выбранных датасетов. Ru\_go\_emotions содержит в себе значительное количество повторений, а также является переводом англоязычного датасета в связи с чем возникают следующие проблемы: наличие непереведенных текстов, наличие текстов, которые невозможно классифицировать ввиду наличия слишком большого количества слов маркеров или различных эмоциональных контекстов, наличие сообщений состоящих исключительно из символов, наличие сообщений которые невозможно интерпретировать.

Примеры проблем в текстах:

1. «HEY! NO OC IN r/ComedyCemetery» – не переведен;
2. «Мне нравиться, так грустно, ахаха, идиот» - невозможно классифицировать;
3. «#################» - состоит исключительно из символов;
4. «Стул,занание,дом,бред,чел» - невозможно интерпретировать.

Сообщения, которые не переведены можно также разделить на две категории: текст полностью на английском языке, текст частично переведен. В качестве решения проблемы выбран перевод данных сообщений, если это возможно. В ситуации, когда текст не может быть классифицирован или интерпретирован, сообщение удаляется, после чего его место заполняется данными из другого датасета.

Ru\_twitter\_sentiment содержит в себе тексты, которые достаточно легко интерпретировать, однако он промаркирован с использованием бинарной классификации, следовательно единственная проблема связанная с его использованием – время застраченное на ручную разметку.

При формировании итогового датасета за основу будут взяты данные из Ru\_go\_emotions, в случае, когда требуется удаление данных, пустые поля будут заполняться текстами из Ru\_twitter\_sentiment. В качестве палитры эмоций будут использоваться: нейтральность, радость, грусть, страх, злость, заинтересованность, отвращение, промаркированные от 0 до 6 в соответствии с представленным порядком. Для каждого текстового поля в новом датасете будет присвоено значение от 0 до 6.

В результате обработки и коррекции данных был получен датасет, содержащий приблизительно 14000 значений с новой маркировкой. Сформированный набор данных будет использоваться для дальнейшего обучения выбранной модели.

# Реализация

В данном разделе будет описана техническая реализация приложения, в частности: обучение и сравнение моделей Ru-bert-base от ai-forever и DeepPavlov, а также Bert-multilingual, на основе результатов которого будет выбрана итоговая модель для дальнейшей интеграции в приложение. Сравнение моделей будет произведено на основе метрики accuracy. Описание токенизации. Также будет описание fine-tuning классов BERT. Описаны технические детали взаимодействия с Longpoll VK API, написание бота, настройка группы VK, функциональная часть бота.

## Обучение моделей

Рассматриваемые модели относятся к модели BERT и являются ее различными модификациями. Данная модель крайне эффективна для решения задач NLP, в рамках которой область применимости BERT не ограничена. Для конкретных решений необходимо провести fine-tuning модели, основной частью которого является подбор качественных данных для обучения модели. По своему принципу работы BERT создает векторные представления слов.

В связи с тем, что модели основанные на архитектуре Transformers, в том числе и BERT требуют значительных вычислительных ресурсов процесс дообучение будет происходит с использованием сервиса от Google – Collab, предоставляющий возможность использования технических средств с высокой производительностью, в частности бесплатно предоставляется графический процессор Tesla T4, который можно подключить с использованием библиотеки PyTorch и метода device.

После получения доступа к вычислительным ресурсам необходимо загрузить библиотеку datasets, которая позволяет загрузить датасет в Collab. Для взаимодействия с набором данных используется библиотека pandas. Для подачи данных в модель необходимо провести токенизацию.

* + 1. Токенизация

Для корректного проведения токенизации необходимо использовать родной токенайзер модели, в противном случае могут возникать ошибки при индексации. Пример токенизации представлен на рисунке 9.

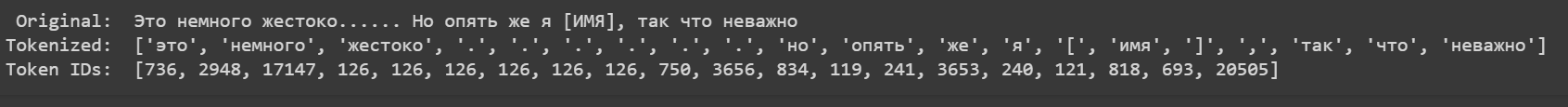


Рисунок 9 – Пример токенизации

Принцип токенизации заключается в следующем:

1. Добавление специальных токенов CLS и SEP, в начало и конец каждого предложения соответственно
2. Приведение каждого предложения к единой длине путем добавления или усечения
3. Отделение padding токенов от реальных, с помощью «attention mask»

Для приведения предложений к одной длине используются Pad-токены, по своей сути это просто нулевой токен, которые заполняет пустой пространство до установленного параметром max length. Для того, чтобы модель не рассматривала пустые части блоков существует механизм attention mask, который делает так, что сеть не рассматривала Pad-токены. В библиотеке AutoTokenizer есть метод tokenizer.encode\_plus, который позволяет в удобном виде проводить токенизацию. Для данного метода необходимо указать следующие параметры:

1. add\_special\_tokens – имеет два значения True или False, отвечает за проставление CLS и SEP токенов;
2. max\_length – указывает длину к которой будут приводиться предложения;
3. pad\_to\_max\_length - имеет два значения True или False, отвечает за добавление Pad-токенов;
4. return\_attention\_mask - имеет два значения True или False, отвечает за вывод attention masks;
5. return\_tensors – указывает какой вид тензоров используется;
6. truncation - имеет два значения True или False, отвечает за урезание предложений.

В результате токенизации получается два списка: первый содержит в себе список тензора с идентификаторами, второй attention mask для этих идентификаторов.

* + 1. Валидация

Для подбора гиперпараметров, выявления переобучения модели необходим метод, позволяющий произвести оценку качества работы модели. Для этого необходима валидация данных. Для разбиения будет использоваться классический метод валидации – hold out. С помощью Dataloader создаем два набора данных train\_dataloader и validation\_dataloader, с помошью метода RandomSampler добиваемся случайности распределения данных. Итоговое соотношения 9 к 1. Также для проведения валидации необходимо установить параметр batch size, который отвечает за количество данных, обрабатываемых за одну итерацию.

Для использования валидации в процессе обучения модель переводится в режим eval, что обеспечивает отсутствие зависимости от случайных факторов, влияющих на обучение, также используется менеджер torch.no\_grad, отключающий вычисление градиентов с целью экономии ресурсов. После инициализируются переменные total\_eval\_accuracy, total\_eval\_loss. nb\_eval\_steps, которые за накопление точности, потерь и количества обработанных батчей соответственно. Процесс валидации представлен в листинге 1.

Листинг 1 – Процесс валидации

for batch in validation\_dataloader:

b\_input\_ids = batch[0].to(device)

b\_input\_mask = batch[1].to(device)

b\_labels = batch[2].to(device)

with torch.no\_grad():

res = model(b\_input\_ids, token\_type\_ids=None, attention\_mask=b\_input\_mask, labels=b\_labels)

loss = res['loss']

logits = res['logits']

total\_eval\_loss += loss.item()

logits = logits.detach().cpu().numpy()

label\_ids = b\_labels.to('cpu').numpy()

total\_eval\_accuracy += flat\_accuracy(logits, label\_ids)

validation\_time = format\_time(time.time() - t0)

В результате рассчитываются значения средней точности и потерь на валидационных данных avg\_val\_accuracy = total\_eval\_accuracy / len(validation\_dataloader), avg\_val\_loss = total\_eval\_loss / len(validation\_dataloader).

* + 1. Рассмотрение гиперпараметров

Эффективность обучения модели напрямую зависит от подбора гиперпараметров. Для моделей BERT используется optimizer AdamW, для которого устанавливаются изначальные параметры learning rate и epsilon. В процессе обучения данные значения изменяются за счет scheduler. Помимо learning rate есть ещё два гиперпараметра: batch size и количество эпох. Первый влияет на количество итераций в рамках одной эпохи, а как следствие на количество перезаписей весов, вторая регулирует общее количество итераций.

* + 1. Петля обучения

Петля обучения состоит из двух основных элементов: обучение и оценка. Поскольку оценка модели с помощью валидации была описана ранее, в данном пункте будет представлено описание части обучения. Для обучения модели используется следующая последовательность действий:

1. распаковать обучающие данные и лейблы;
2. загрузить данные на GPU;
3. обнулить градиенты с предыдущего шага;
4. передать данные неиронной сети и пропросить их вперед;
5. рассчитать градиенты по всем параметрам;
6. обновить параметры с помощью оптимайзера;
7. рассчитать все статистики.

## Сравнение выбранных моделей

Для выбора итоговой модели необходимо провести сравнительный анализ моделей, выбранных ранее, требуется провести обучение каждой из них на сформированном датасете, подобрать наиболее эффективные гиперпараметры для каждой из моделей. Далее провести оценку модели на тестовом датасете и определить метрику accuracy. Результаты сравнительного анализа представлены в таблице 6.

Таблица 6 – Результаты сравнения

|  |  |  |  |
| --- | --- | --- | --- |
|  | RuBert-base by ai-forever | RuBert-base by DeepPavlov | Bert-base-multilingual |
| Batch size | 16 | 8 | 16 |
| Количество эпох | 3 | 2 | 2 |
| Learning rate | 2e-5 | 2e-5 | 2e-5 |
| Accuracy | 0.75 | 0.71 | 0.69 |

После полного сравнения, для использования в итоговом приложении будет использоваться модель RuBert-base от ai-forever, помимо того что она показала наиболее высокую точность, в процессе обучения данная модель также была наиболее стабильной среди всех.

## Реализация бота

Для реализации бота используется Longpoll VK API. Основной функцией бота является взаимодействие с сообщениями пользователей, для этого необходимо корректно настроить группу VK. Настройка приведена на рисунках 10,11.

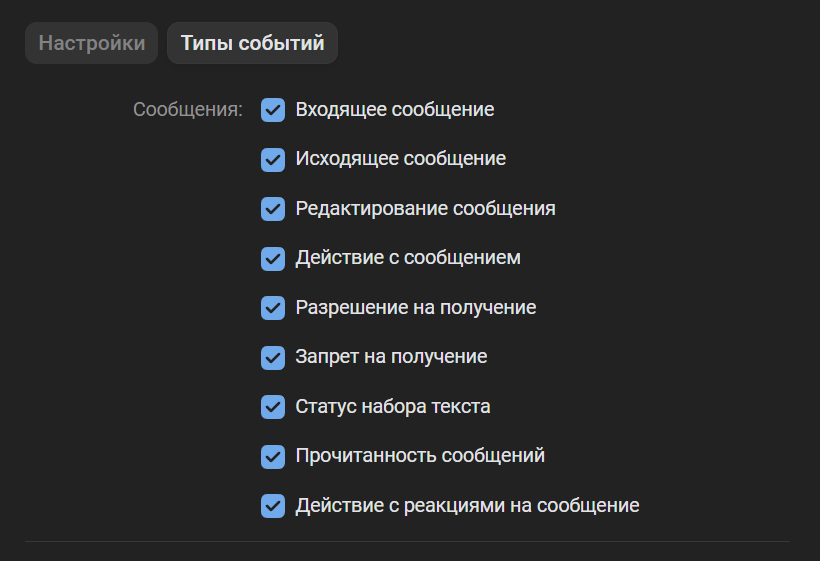


Рисунок 10 – Настройка типов событий

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

Рисунок 11 – Настройки для бота

Для соединения бота с серверами ВКонтакте используется токен, который храниться в конфигурации бота. Для отслеживания ивентов используется функция longpoll.listen, после чего происходит проверка на тип ивента. Бот должен реагировать только на текстовые сообщения, также предлагается, что бот может быть добавлен в беседу, где не все сообщения будут адресованы ему, для корректного реагирования бота происходят проверки event.to\_me и event.text. Для того чтобы бот просматривал новые сообщения происходит проверка с помощью VkEventType с использование параметра MESSAGE\_NEW. Если новое сообщение не проходит проверку event.to\_me то бот не будет на него реагировать, если же сообщение не проходит проверку event.text, то бот отправит ответ о том, что пользователь отправил не текст и попросит присылать сообщения только в текстовом формате. Если сообщение проходит обе проверки, то бот передает текст в модель, где происходит анализ, после чего с использованием функции sender, основанная на методе vk.message.send, бот отправляет результат пользователю.

Для использования модели в боте происходит загрузка ее конфигурации в отдельный файл. Инициализируются: модель с весами, токенизатор и метод BertForSequenceClassification. После того, как бот получает сообщение от пользователя активируется функция tokenize, которая проводит токенизацию полученного сообщения, затем активируется функция definition\_emote, которая определяет принадлежность эмоции к одному из классов.

Для хранения данных о пользователях используется база данных sqlite3. Она включает в себя таблицу user\_status, которая содержит два столбца: user\_id, greeted. База данных нужна для отправки приветственного сообщения пользователям, которые ранее не использовали данного бота. Как только пользователь отправляет сообщение происходит проверка наличия ID в базе данных, если он находится, то система ждет следующего сообщения, если ID отсутствует в базе данных, то активируется функция hello, которая добавляет ID в базу и отправляет приветственный текст.

# ЗАКЛЮЧЕНИЕ

В результате выполненной работы были получены следующие результаты на основе поставленных задач:

1. произведен анализ предметной области, в рамках которого рассмотрены следующие методы анализа предметной области: методы машинного обучения с учителем, а именно LSTM, CNN, Transformers, Логистическая регрессия, метод машинного обучения без учителя – BoW, методы основанные на словарях и правилах, в результате сравнения для дальнейшего рассмотрения была выбрана архитектура Transformers;
2. сформирован собственный датасет, промаркированный от 0 до 6 в соответствии с эмоциями: нейтральность, радость, грусть, страх, злость, заинтересованность, отвращение. Общее количество данных примерно равняется 14000, основой датасета является Ru\_go\_emotions;
3. спроектирована система, основными компонентами которой являются клиентская и серверная части, сформулированы функциональные требования, архитектура система представлена в виде диаграмм компонентов UML, процессы взаимодействия пользователя с системой, а также внутренние процессы описаны с помощью нотации IDEF0;
4. дообучена модель RuBert от ai-forever и подобраны гиперпараметры: batch size = 16, lr = 1e-5;
5. реализован бот для социальной сети ВКонтакте;
6. проведено тестирование и оценка модели, с использованием тестового датасета, метрика точности превышает 0.7;
7. готовое приложение опубликовано в открытом репозитории.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Н. В. Лукашевич, Автоматический анализ тональности текстов: проблемы и методы // Интеллектуальные системы. Теория и приложения, 2022, том 26, выпуск 1, 50–61 URL: (дата обращения: 18.02.2024)
2. Семина Т.А. Анализ тональности текста: современные подходы и существующие проблемы // Социальные и гуманитарные науки. Отечественная и зарубежная литература. Сер. 6, Языкознание: Реферативный журнал. 2020. №4. URL: (дата обращения: 18.02.2024).
3. Ермоленко Т. В., Кравченко О. А. ПРОБЛЕМЫ АНАЛИЗА ТОНАЛЬНОСТИ ТЕКСТА В СИСТЕМЕ АНАЛИЗА МНЕНИЙ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ ГЛУБОКОГО ОБУЧЕНИЯ //Донецкие чтения 2022: образование, наука, инновации, культура и вызовы современности. – 2022. – С. 244-246. URL: https://elibrary.ru/download/elibrary\_49968311\_65490243.pdf (дата обращения: 18.02.2024)
4. Плетнёва М. В. Подход автоматического анализа тональности текстов, основанный на словаре эмоциональной лексики //ОБЩЕСТВО. НАУКА. ИННОВАЦИИ (НПК-2017). – 2017. – С. 1779-1786. URL: https://elibrary.ru/download/elibrary\_30535620\_31472451.pdf (дата обращения: 18.02.2024)
5. Lan M., Tan C. L., Su J., Lu Y. Supervised and Traditional Term Weighting Methods for Automatic Text Categorization // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, Vol. 31, no. 4, pp. 721–735.
6. Самигулин Тимур Русланович, Джурабаев Анвар Эркин Угли АНАЛИЗ ТОНАЛЬНОСТИ ТЕКСТА МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ // Научный результат. Информационные технологии. 2021. №1. URL: https://cyberleninka.ru/article/n/analiz-tonalnosti-teksta-metodami-mashinnogo-obucheniya (дата обращения: 19.02.2024).
7. Ермаков П. Д., Федянин Р. В. Исследование методов машинного обучения в задаче автоматического определения тональности текстов на естественном языке // Новые информационные технологии в автоматизированных системах. 2015. №18. URL: https://cyberleninka.ru/article/n/issledovanie-metodov-mashinnogo-obucheniya-v-zadache-avtomaticheskogo-opredeleniya-tonalnosti-tekstov-na-estestvennom-yazyke (дата обращения: 19.02.2024).
8. Лыченко Н. М., Сороковая А. В. Классификатор тональности текстов на основе LSTM-нейронной сети //Вестник Кыргызско-Российского Славянского университета. – 2019. – Т. 19. – №. 12. – С.87-92.URL: https://elibrary.ru/download/elibrary\_42347939\_32750985.pdf (дата обращения: 19.02.2024).
9. Васильева М. И. и др. Разработка системы тонального анализа текста с использованием ансамбля классификаторов //Электронные средства и системы управления. Материалы докладов Международной научно-практической конференции. – федеральное государственное бюджетное образовательное учреждение высшего образования Томский государственный университет систем управления и радиоэлектроники, 2019. – №. 1-2.–С.144-146.URL: https://elibrary.ru/download/elibrary\_42559828\_72537972.pdf (дата обращения: 19.02.2024).
10. Vaswani A. et al. Attention is all you need //Advances in neural information processing systems. – 2017. – Т. 30.
11. Студенческая наука: актуальные вопросы, достижения и инновации : Сборник статей XIV Международной научно-практической конференции, Пенза, 17 мая 2024 года. – Пенза: Международный центр научного сотрудничества "Наука и Просвещение", 2024. – 268 с. – С. 32-35 URL: https://naukaip.ru/wp-content/uploads/2024/05/MK-2025.pdf

# Приложение А

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание