

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)**

**Департамент математического и компьютерного моделирования**

NLP-модель оценки компании по отзывам клиентов

**КУРСОВАЯ РАБОТА**

по образовательной программе подготовки бакалавров

по направлению 02.03.01 «Математика и компьютерные науки»

профиль «Сквозные цифровые технологии»

|  |  |
| --- | --- |
| Работа защищена  с оценкой             \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_            Регистрационный номер \_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г. | Студент группы     № Б9122-02.03.01сцт    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Поповкин А. А.          (подпись)  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.      Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   (должность,  ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)                                               (ФИО)  «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г. |

г. Владивосток

2025

# Содержание

[Содержание 2](#_Toc198717783)

[1. Введение 3](#_Toc198717784)

[2. Структура модели 4](#_Toc198717785)

[3. Основная часть 5](#_Toc198717786)

[3.1. Получение данных 5](#_Toc198717790)

[3.1.1. Описание собираемых данных 5](#_Toc198717791)

[3.1.2. Социальные сети 7](#_Toc198717792)

[3.1.3. Сайты с рейтингами 9](#_Toc198717795)

[3.2. Разметка 11](#_Toc198717796)

[3.2.1. Разметка рейтингов 11](#_Toc198717797)

[3.2.2. Семантический анализ 12](#_Toc198717798)

[4. Список литературы 15](#_Toc198717799)

# Введение

Для любой компании важно иметь чёткое понимание того, какие проблемы замечают пользователи в её продукте. Зная их, компания имеет возможность вовремя устранить недостатки в своём сервисе, демонстрируя пользователям, что она способна развиваться и меняться в лучшую сторону.  
Для этого необходим регулярный мониторинг различных новостных порталов по поводу продуктов или сервиса компании, а также регулярное составление сводок о проблемах. Но помимо специализированных новостных порталов существуют также и специализированные сайты с отзывами, где люди делятся впечатлениями о предоставленных им услугах, оставляя свой комментарий и рейтинг, и соцсети, где встречается множество мнений, которые зачастую оказываются крайне важны для компаний. Таким образом, проблема анализа отзывов с сайтов с рейтингами и социальных сетей является важной и актуальной задачей.

Объектом работы являются комментарии пользователей в социальных сетях, таких как «ВКонтакте» и «Telegram», и специализированных сайтов для отзывов, таких как «ЯндексКарты» и «Отзовик».

Цель работы — разработать модель, способную получить данные с определённых площадок, провести семантическую обработку, отобрав для анализа только негативные отзывы, разделить эти отзывы по кластерам, дав имя каждому из них и сделать отчёт, показывающий, на какой ряд проблем пользователи жалуются больше всего. Дополнительно эта библиотека должна быть модульной, чтобы в случае необходимости можно было воспользоваться каждой из её составляющих отдельно, без необходимости использовать остальные части модели.

# Структура модели

Общий подход к структуре библиотеки такая же, как и у её подмодулей (таких, как модуль получения информации, модуль разметки и т. д.):

1. **Ядро**
   1. Координирует работу нижестоящих подмодулей, обрабатывает и пересылает необходимые параметры, собирает и объединяет полученные данные.
   2. Для этапа получения информации ядро пересылает необходимую информацию для парсеров (классов, собирающих информацию с сервисов).
2. **Специализированные подмодули**
   1. Обладают конкретным функционалом.
   2. Для этапа получения информации специализированными подмодулями являются парсеры, которые по полученному временному отрезку, запросу и остальным параметрам получают отзывы.
3. **Конфигурационная система**
   1. Централизованное управление настройками подмодулей и ядра.
   2. Позволяет хранить несколько профилей настроек и легко между ними переключаться
4. **Структурные подмодули**
   1. Обеспечивают единство архитектуры и реализуют общие для подмодулей методы.
   2. Дополнительно хранят документацию с шаблоном вывода.
5. **Вспомогательные компоненты**
   1. Тесты и примеры использования.
   2. Журналирование и документация.
   3. Список зависимостей проекта.
   4. И т. д.

Такой подход позволяет быстро добавить компонент в систему, без необходимости её значительного изменения, обеспечивая изоляцию модулей.

# Основная часть

Для разработки библиотеки был выбран язык программирования python.

Он позволяет сделать разработку быстрой за счёт простого синтаксиса и обширного числа пользовательских библиотек, позволяющих без особых сложностей подключать нейронные сети, получать данные с веб-страниц и создавать понятные и информативные графики.

В то же время, сниженная по сравнению с более низкоуровневыми языками скорость компенсируется тем, что наиболее тяжёлые операции (разметка полученных отзывов) производятся с помощью хорошо оптимизированных библиотек, написанных на более низкоуровневых языках (C/C++) и имеющих обёртку (библиотеку, которая позволяет взаимодействовать с низкоуровневыми языками) для python.

В качестве примера сервиса, по которому будут получаться отзывы, был выбран курорт МРИЯ, верхней границей 18.05.2025, а нижней – 01.01.2024.



## Получение данных

На данном этапе происходит сбор информации для дальнейшей обработки. (как будто хочется написать побольше)

### Описание собираемых данных

При начале работы с получением данных встал вопрос о том, какие параметры нужно выбирать. Очевидно, что стоит собирать текст публикации и её дату, но также необходимо решить вопрос и о других характеристиках. В ходе рассуждений был выделен следующий набор параметров:

1. **service\_id**

Внутренний номер сервиса, с которого были получены данные. Нужен для корректной обработки данных.

1. **additional\_id**

Дополнительный идентификатор, откуда было получено сообщение. Используется для социальных сетей в случае, если человек оставил публикацию с отзывом не на своей странице, а в группе.

1. **Name**

Ник (псевдоним) пользователя на платформе. Изначально предполагалось, что по нему будет проходить дедупликация отзывов, однако впоследствии было принято решение отказаться от этого, так как у человека могут быть разные ники на разных платформах, а также он мог написать текст на разных платформах немного по-другому, использовав иные формулировки и знаки препинания.

1. **date**

Дата в формате timestamp (количество секунд с 01.01.1970)

1. **text**

Текст отзыва

1. **answer**

Ответ организации на отзыв. На многих площадках с отзывами (таких как Яндекс карты) у организаций есть возможность отвечать на комментарии пользователей.

1. **rating**

Рейтинг, оставленный пользователем организации.

### Социальные сети

Поводом к обработке отзывов из социальных сетей стала гипотеза, что в социальных сетях будет много реальных, не отредактированных площадкой постов, содержащих отзывы о компании, что могло дать ценную информацию о ее фактическом сервисе. О результате проверки этой гипотезы будет сказано в главе 2.2.2, посвящённой семантическому анализу данных.

Для получения данных из социальных сетей использовались их собственные API (точки доступа, позволяющие отправлять запросы и получать данные).

Работа с API отличается от сервиса к сервису, но общий принцип таков: отправляется текстовый запрос (обычно с возможностью использования регулярных выражений, чтобы можно было выбирать публикации, не содержащие определённые слова и т. д.) вместе с временным промежутком, за который нужно собрать посты, после чего необходимо выбрать из них необходимую информацию.

Для исследования были взяты популярные в России социальные сети ВКонтакте и Telegram.

Алгоритм подбора запроса был следующим: сначала выбрано просто название сервиса. Это дало слишком много рекламных сообщений, поэтому был добавлен тип сервиса. Но рекламы было всё ещё слишком много. Следующий этап — исключение слова «купить», часто встречающегося в рекламных постах. Дополнительно проведена проверка английских названий.

В результате подбора было получено следующее выражение: «Мрия -купить», означающее, что идет поиск постов со словом «Мрия», но без слова «купить».

#### Социальная сеть «ВКонтакте»

Доступ к API для ВКонтакте получается с помощью библиотеки ***vk***. Для работы с ней нужен токен доступа уровня пользователя, или же логин с паролем от аккаунта во ВКонтакте. Поиск с помощью API у ВКонтакте идёт по всем доступным от данного пользователя постам и работает следующим образом: за один запрос он может вернуть не более 200 постов, и не более чем 1000 за весь промежуток времени. Поэтому алгоритм работы был следующим:

1. Определение того, сколько всего есть постов за промежуток времени.
2. Получение несколько раз по 200 постов до того момента, пока API не будет больше выдавать постов за промежуток времени.
   1. Иногда, вместо корректного результата, ВКонтакте может выдать 0 постов, даже с учётом того, что раньше он выдавал их корректно, и не был достигнут предел по количеству. В таких случаях приходится выполнять небольшую временную задержку, и отсылать запрос заново.
3. Смещение верхней временной границы до времени последнего полученного поста. Это позволяет API вновь находить посты по заданному выражению.
   1. К концу заданного временного отрезка ВКонтакте начинает выдавать по одному отзыву за запрос, поэтому запросов потребуется больше.
4. Повторять, пока постов за промежуток времени не останется, или не будет получено нужное количество.

#### Социальная сеть «Telegram»

Доступ к API для Telegram получается с помощью библиотеки ***teleton***. Для работы с ней нужен аккаунт в Telegram, id и hash приложения в Telegram. Поиск с помощью API у Telegram идёт по всем доступным для данного пользователя сообщениям и работает следующим образом:

1. По полученным данным Telegram-приложения запускается сессия, в которой будет происходить основная работа.
2. Из переданного списка каналов в виде набора ссылок или тэгов (укороченных названий) собираются их id.
3. Третий этап посвящён поиску сообщений. В Telegram не поддерживаются регулярные выражения, как во ВКонтакте, поэтому исключать сообщения со словом «купить» приходилось вручную, при этом сам поиск асинхронный, что означает, что нужно ввести дополнительные обработки асинхронного вызова библиотеки.

### Сайты с рейтингами

Было решено получать информацию с сайтов с рейтингами, так как они создавались с целью сбора отзывов пользователей, что позволяет не проводить семантический анализ отзывов, а фильтровать лишь по рейтингу, что позволит быстрее провести этап с разметкой.

Для получения данных с сайтов были использованы «парсеры» — модули, получающие информацию напрямую с браузера, взаимодействуя с ним так, как мог бы взаимодействовать поиск (переходить по ссылкам, нажимать на кнопки).

Работа с сайтами отличается от сайта к сайту, но общий алгоритм таков: открывается сайт на странице с сервисом. Затем выбирается нужная сортировка (по возрастанию рейтинга отзыва, по убыванию, по умолчанию или же по новизне отзыва). Следом проводится проверка каждого отзыва по нахождению в нужном временном диапазоне, пока не наберётся нужное количество. Если отзыв слишком поздний – его пропускают. Если же слишком ранний и не найдено достаточного количества отзывов – работа прекращается и выдаётся столько отзывов, сколько удалось найти. Дополнительно стоит отметить, что все парсеры, за исключением Отзовика, запускают браузер в «скрытом режиме», не показывая окно пользователю.

В качестве сайтов с рейтингами были выбраны популярные на Российском пространстве электронные карты GoogleMaps и ЯндексКарты и специализированный сайт для отзывов Отзовик. Карты были выбраны потому, что там люди оставляют отзывы о своих поездках/взаимодействии с сервисом чаще всего.

#### GoogleMaps

Получение информации с GoogleMaps было осложнено тем, что в целях противодействия сборщикам информации, Google сделал структуру сайта менее читаемой с точки зрения кода веб-страницы, который обрабатывают парсеры. Обходилось это путём применения менее удобного поиска визуальных элементов.

Дополнительной сложностью стала долгая загрузка отзывов. Чем больше прогружено отзывов на странице, тем дольше грузится следующий набор отзывов. Из-за этого получение информации с GoogleMaps является длительным процессом.

#### ЯндексКарты

Получение информации с ЯндексКарт было крайне простым, из-за быстрой загрузки отзывов и удобного, с точки зрения кода веб-страницы, сайта.

Проблемой стало неожиданное ограничение на получение отзывов. Максимально на странице может быть не более 600 отзывов. Обойти это ограничение не получилось.

#### Отзовик

Получение информации с Отзовика было более технически сложным. Для полного раскрытия отзывов необходимо «перейти» на них.

Дополнительной проблемой стало появление капчи. Она появляется в случае даже небольшого количества запросов и требует ручного решения, поэтому браузер запускается в обычном режиме.

## Разметка

На данном этапе происходит распределение отзывов на группы по тональности. В зависимости от параметров, отзывы можно разбивать на 2 группы (неотрицательные и отрицательные) или на 3 (нейтральные, положительные и отрицательные). В данной работе было применено разбиение на 3 группы.

Перед началом обработки данные разбиваются на полученные с сайтов с рейтингами и на полученные с сайтов без рейтингов.

После разметки отбираются отзывы из отрицательных групп.

### Разметка рейтингов

Разметка рейтингов на семантические группы происходит следующим образом:

1. Установка рамок положительного и отрицательного класса.
   1. Если рамки совпадают и одно из условий сравнения нестрогое, то используется двухклассовая разметка, иначе - трёхклассовая.
      1. Пример для двухклассовой разметки (где x – рейтинг):
         1. x 3 — отрицательный
         2. x3 – неотрицательный
      2. Пример для трёхклассовой разметки (где x – рейтинг):
         1. x2 – отрицательный
         2. 2x3 – нейтральный
         3. x3 – положительный
2. Приведение рейтингов к одной системе.
   1. На разных платформах системы оценивания могут быть разными (от 1 до 5, от 1 до 10 и т. д.), потому для корректной оценки необходимо, чтобы они были в одной системе оценивания.
3. Сравнение оценки с пороговыми значениями и присвоение класса отзыву.

### Анализ тональности

#### Поиск модели

Перед тем, как приступить к работе над анализом тональности, необходимо подобрать нужную LLM модель для обработки данных.

Так как анализ тональности – задача классификации текста, существуют несколько вариантов предсказываемых классов: 2 (отрицательный и положительный), 3 (нейтральный, положительный, отрицательный) и 5 (отрицательный, нейтральный, положительный, пропуск, речь).

В качестве главной метрики выбирается F1 (гармоническое среднее между Precision (точность) и Recall (полнота)) предсказания отрицательного класса, но также вычисляются точность (accuracy), взвешенный F1 (в зависимости от численности класса присваивается вес для его F1 метрики и вычисляется среднее арифметическое) и F1 макро (невзвешенное среднее арифметическое F1 метрик классов). Для тестирования было решено перейти к единому формату, к задаче предсказания отрицательного класса (так как именно отрицательные отзывы будут анализироваться далее), разделяя классы разных тестовых наборов данных на отрицательные и неотрицательные.

В качестве тестовых данных используются различные популярные наборы данных для тестирования русскоязычных моделей на решение задачи анализа тональности. Были использованы следующие наборы данных:

* + - 1. tg\_messages\_mriya – 334 сообщения, собранные с официального телеграм-канала МРИЯ с 1.01.2024, вручную размеченные на три класса.
      2. sentiment\_of\_bank\_reviews – набор русскоязычных отзывов о работе различных банков, собранные с сайта banki.ru (2 класса) [1].
      3. RuTweetCorp – набор русскоязычных постов из социальной сети Twiter (2 класса) [2].
      4. RuReviews – набор русскоязычных отзывов на различные товары и продукты (3 класса) [3].
      5. RuSentiment – набор сообщений и постов из социальной сети ВКонтакте (5 классов) [4].
      6. MonoHime/ru\_sentiment\_dataset – набор данных, состоящий из нескольких других наборов данных, включающих текста новостей, комментариев с форумов, отзывов на автомобили и отели и пр. (3 класса) [5].

В качестве моделей для тестирования были выбраны 12 LLM с сайта Hugging Face, дообученные на русскоязычных наборах данных на решение задачи анализа тональности.

Сравнение моделей по средним метрикам на тестовых наборах данных, отсортированное по F1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | accuracy | f1 | f1\_weighted | f1\_macro |
| sismetanin/sbert-ru-sentiment-rusentiment | 0.8578540713540455 | 0.7049241637738879 | 0.8548496616913158 | 0.7980781592506186 |
| sismetanin/mbart\_ru\_sum\_gazeta-ru-sentiment-rusentiment | 0.8385408985195483 | 0.6625364574797391 | 0.8348422630397919 | 0.7695665733546643 |
| seara/rubert-tiny2-russian-sentiment | 0.7201107868968902 | 0.6497392109521775 | 0.7227114086476082 | 0.6928113177057823 |
| blanchefort/rubert-base-cased-sentiment-rurewiews | 0.729443258290531 | 0.5627022716323226 | 0.7499607511241136 | 0.6570948647964473 |
| kartashoffv/vashkontrol-sentiment-rubert | 0.7611917694826871 | 0.4916148590057718 | 0.7619845763981724 | 0.6525173218946955 |
| tabularisai/multilingual-sentiment-analysis | 0.7185827682811365 | 0.4748256543959335 | 0.734957452903508 | 0.6254911798269326 |
| blanchefort/rubert-base-cased-sentiment | 0.4411206927013744 | 0.33346165529495747 | 0.4423844157093025 | 0.3994163754539091 |
| MonoHime/rubert\_conversational\_cased\_sentiment | 0.44161572877522765 | 0.27425092315680016 | 0.3855412656764197 | 0.34395467969877574 |
| rajora/distilbert-multilingual-sentiment | 0.5748561949547755 | 0.10754565630397372 | 0.5551971758282631 | 0.40815494930682444 |
| cointegrated/rubert-tiny-sentiment-balanced | 0.17776613061892865 | 0.06800540987364699 | 0.17787813531338337 | 0.1578888748478996 |
| sismetanin/rubert-ru-sentiment-rureviews | 0.31653968075773325 | 0.05651705424577095 | 0.34757012689777156 | 0.25758708634673416 |
| sismetanin/mbart\_ru\_sum\_gazeta-ru-sentiment-rureviews | 0.2909129278893587 | 0.04524587730989957 | 0.3226362677265419 | 0.23585875532312273 |

Сравнение моделей на собственных тестовых данных (сообщениях из телеграм), и размеченных вручную:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **accuracy** | **f1** | **f1\_weighted** | **f1\_macro** |
| sismetanin/sbert-ru-sentiment-rusentiment | 0,979041916 | 0,533333333 | 0,979724471 | 0,761306789 |
| sismetanin/mbart\_ru\_sum\_gazeta-ru-sentiment-rusentiment | 0,976047904 | 0,428571429 | 0,976047904 | 0,708169506 |
| seara/rubert-tiny2-russian-sentiment | 0,610778443 | 0,333333333 | 0,61771676 | 0,519750442 |
| kartashoffv/vashkontrol-sentiment-rubert | 0,916167665 | 0,222222222 | 0,940323993 | 0,588959212 |
| tabularisai/multilingual-sentiment-analysis | 0,853293413 | 0,140350877 | 0,903467765 | 0,530077239 |
| blanchefort/rubert-base-cased-sentiment-rurewiews | 0,796407186 | 0,128205128 | 0,868890121 | 0,506475445 |
| blanchefort/rubert-base-cased-sentiment | 0,428143713 | 0,073394495 | 0,474088078 | 0,360002293 |
| MonoHime/rubert\_conversational\_cased\_  sentiment | 0,838323353 | 0,035714286 | 0,893404368 | 0,473739496 |
| rajora/distilbert-multilingual-sentiment | 0,77245509 | 0,025641026 | 0,853465429 | 0,448413733 |
| sismetanin/mbart\_ru\_sum\_gazeta-ru-sentiment-rureviews | 0,194610778 | 0,014652015 | 0,312609056 | 0,166819678 |
| sismetanin/rubert-ru-sentiment-rureviews | 0,224550898 | 0,007662835 | 0,356175841 | 0,185649599 |
| cointegrated/rubert-tiny-sentiment-balanced | 0,131736527 | 0 | 0,148552915 | 0,10758436 |

Сравнение моделей на RuSentiment:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **model** | **accuracy** | **f1** | **f1\_weighted** | **f1\_macro** |
| sismetanin/sbert-ru-sentiment-rusentiment | 0,979156646 | 0,918321186 | 0,979306465 | 0,953187646 |
| sismetanin/mbart\_ru\_sum\_gazeta-ru-sentiment-rusentiment | 0,970145903 | 0,884045336 | 0,970470716 | 0,933456392 |
| seara/rubert-tiny2-russian-sentiment | 0,691454225 | 0,58446676 | 0,700645713 | 0,650767257 |
| blanchefort/rubert-base-cased-sentiment | 0,652172519 | 0,577715929 | 0,66729098 | 0,62590258 |
| blanchefort/rubert-base-cased-sentiment-rurewiews | 0,628571429 | 0,369358088 | 0,690678245 | 0,553063018 |
| kartashoffv/vashkontrol-sentiment-rubert | 0,821773288 | 0,320703984 | 0,825083667 | 0,609067575 |
| tabularisai/multilingual-sentiment-analysis | 0,680647747 | 0,296034495 | 0,731078239 | 0,544757392 |
| MonoHime/rubert\_conversational\_cased\_  sentiment | 0,135513869 | 0,223777029 | 0,049588298 | 0,124189973 |
| rajora/distilbert-multilingual-sentiment | 0,568286035 | 0,122874454 | 0,639567423 | 0,418277788 |
| sismetanin/rubert-ru-sentiment-rureviews | 0,454609588 | 0,054690974 | 0,546239956 | 0,335719005 |
| sismetanin/mbart\_ru\_sum\_gazeta-ru-sentiment-rureviews | 0,461888729 | 0,037731521 | 0,552656162 | 0,332123868 |
| cointegrated/rubert-tiny-sentiment-balanced | 0,152669553 | 0,02676182 | 0,199302636 | 0,124539192 |

Лучшие модели по каждому набору тестовых данных по метрике F! Для отрицательного класса:

|  |  |  |
| --- | --- | --- |
| **dataset** | **model** | **f1** |
| rusentiment | sismetanin/sbert-ru-sentiment-rusentiment | 0,918321186 |
| rutweetcorp | sismetanin/sbert-ru-sentiment-rusentiment | 0,837018837 |
| sentiment\_of\_bank\_reviews | seara/rubert-tiny2-russian-sentiment | 0,836662453 |
| rureviews | seara/rubert-tiny2-russian-sentiment | 0,762741011 |
| ru\_sentiment\_monohime | sismetanin/mbart\_ru\_sum\_gazeta-ru-sentiment-rusentiment | 0,633515854 |
| tg\_messages\_mriya | sismetanin/sbert-ru-sentiment-rusentiment | 0,533333333 |

В результате подбора модели получили, что наилучшей моделью является sbert (sbert-ru-sentiment-rusentiment) [6], однако она ограничена длиной контекста в 512 токенов, что покрывает 82% сообщений. В то же время, модель, стоящая на втором месте, mbart (mbart\_ru\_sum\_gazeta-ru-sentiment-rusentiment) [7], поддерживает контекст в 1024 токена, что покрывает 92% сообщений, и является компромиссом между точностью предсказаний и полнотой учитываемой информации.

Стоит также отметить модель, стоящую на третьем месте, rubert-tiny2 (rubert-tiny2-russian-sentiment) [8]: она содержит довольно малое число параметров (29.2 миллиона, по сравнению с sbert – 427 миллиона и mbart – 867 миллионов параметров), следовательно, её можно использовать в случае ограничения вычислительных ресурсов или для скорости, пожертвовав точностью.

#### Проведение анализа

Алгоритм проведения семантического анализа следующий:

1. Данные токенизируются (текста разбивается на части и кодируется числами для обработки моделью) с помощью функции токенизатора подобранной в предыдущем шаге модели.
2. Предсказываются метки.
   1. Количество получаемых меток зависит от модели. В случае mbart получается 5 меток (0-негативный, 1-нейтральный, 2-позитивный, 3-пропуск, 4-речь).
3. Перевод полученных меток в нужное нам количество.
   1. Возможна обработка 3-х или 5 меточного вывода.
   2. Полученные метрики объединяются в группы (часть уходит в отрицательный класс, часть в положительный и т. д.) и по каждой группе берётся максимальная вероятность того, что отзыв принадлежит каждой группе.
4. Присвоение класса согласно наибольшей вероятности.

## Выбор LLM

### Рассмотрение бесплатных API

Так как локальный развёртывание больших языковых моделей требует больших вычислительных ресурсов, для данной работы оптимальным решением является использование бесплатных сторонних API. Поиск изначально осуществлялся по сервису OpenRouter [9], который предоставляет стандартизированный доступ к большинству провайдеров API разных LLM и инструментарий их смены при отправке запросов. Исходя из описаний множества популярных моделей и условий использования, было решено выбрать в качестве провайдера Chutes [10].

У большинства провайдеров мощные модели либо платные, либо сильно ограничены в запросах, как на OpenRoter, где бесплатно предоставляется 50 запросов в сутки на модель, на Chutes лимиты неизвестны, но на практике они не такие строгие. В связи с этим дополнительно использовался API Mistral для доступа к бесплатной модели Mistral Small, число запросов к которой кратно превышает лимиты на любых других сервисах.

### Сравнение моделей

Модели выбирались по бенчмарку LLM Arena, на котором пользователи пишут запрос, сравнивают, как две модели ответили на него, и выбирают лучшую. Из не проприетарных моделей лучшей для русского языка оказалась DeepSeek V3-0324, которая сравнивалась с Qwen3 235B A22B и Mistral Small в задачах суммаризации, классификации с многими метками. Исходя из результатов решено для сложных задач (классификации с многими метками, генерация названий кластеров, корректирование кластеризации, генерация отчёта) использовать DeepSeek V3-0324, так как она выдаёт более стабильные и приемлемые ответы, а для более простых задач, требующих множества запросов (суммаризация, генерация категорий) – Mistral Small, так как для неё предоставляется кратно большее количество запросов.

## Суммаризация

### Обычное выделение информации из отзывов

Суммаризация – задача сокращения текста с целью пере

### Классификация отзывов с множеством меток

## Кластеризация

### Подбор алгоритма кластеризации

### Корректирование кластеризации с помощью LLM

### Генерация наименований кластеров

## Генерация отчёта

# Список литературы

1. [Электронный ресурс] URL: https://www.kaggle.com/datasets/egorandreasyan/determining-the-sentiment-of-bank-reviews-dataset

2. Рубцова Ю. Автоматическое построение и анализ корпуса коротких текстов (постов микроблогов) для задачи разработки и тренировки тонового классификатора //Инженерия знаний и технологии семантического веба. – 2012. – Т. 1. – С. 109-116.

3. S. Smetanin and M. Komarov, "Sentiment Analysis of Product Reviews in Russian using Convolutional Neural Networks," *2019 IEEE 21st Conference on Business Informatics (CBI)*, Moscow, Russia, 2019, pp. 482-486, doi: 10.1109/CBI.2019.00062.

4. Rogers, A., Romanov, A., Rumshisky, A., Volkova, S., Gronas, M. and Gribov, A., 2018. RuSentiment: An Enriched Sentiment Analysis Dataset for Social Media in Russian. In Proceedings of COLING - 2018. - С. 755-763.

5. [Электронный ресурс] URL: https://huggingface.co/datasets/MonoHime/ru\_sentiment\_dataset

6. [Электронный ресурс] URL: https://huggingface.co/sismetanin/sbert-ru-sentiment-rusentiment

7. [Электронный ресурс] URL: https://huggingface.co/sismetanin/mbart\_ru\_sum\_gazeta-ru-sentiment-rusentiment

8. [Электронный ресурс] URL: https://huggingface.co/seara/rubert-tiny2-russian-sentiment

9. [Электронный ресурс] URL: https://openrouter.ai/

10. [Электронный ресурс] URL: https://chutes.ai/

11.