|  |  |
| --- | --- |
| Задача  06 | **Сервис для отслеживания динамики клиентских настроений и проблем по конкретным банковским продуктам в режиме реального времени** |

Сопроводительная документация

Команда SWAGA:

* Глеб Зотов
* Евгений Михайловский

Оглавление

[**Описание сервиса и его назначения** 3](#_Toc210340699)

[**Описание функциональной и компонентной архитектуры** 4](#_Toc210340700)

# **Описание сервиса и его назначения**

**Проблема:**

Банки получают тысячи отзывов от клиентов ежедневно. Эти отзывы содержат ценную информацию, однако анализ отзывов вручную невозможен из-за их объема.

Это приводит к задержке в реакции на проблемы, снижению удовлетворённости клиентов и ухудшению репутации банка.

**Сервис**

Сервис разработанный нашей командой представляет собой интерактивная аналитическая платформа для менеджеров Газпромбанка, которая выполняет ряд задач:

1. Cобирает и обрабатывает клиентские отзывы
2. Автоматически классифицирует отзывы по продуктам и услугам банка, учитывая более детальные топики
3. Определяет тональность (положительная, нейтральная, отрицательная) каждого отзыва
4. Визуализирует динамику отзывов и изменения их тональности во времени
5. Позволяет оперативно отслеживать сильные и слабые стороны продуктов и сервисов

Дополнительно, сервис предоставляет готовую API для разметки (тип продукта и тональность) произвольных текстов.

**Как сервис решает проблему**

Разработанный сервис использует универсальный пайплайн кластеризации и выделения топиков, который позволяет эффективно структурировать неструктурированные клиентские отзывы. Архитектура решения построена таким образом, что при необходимости может быть легко адаптирована под другие задачи — например, для выделения более гранулярных тем или переобучения модели с целью получения обновлённых кластеров при изменении продуктовой линейки или появления новых услуг.

Благодаря высокой точности классификации модель уверенно определяет тематику и тональность каждого отзыва, что обеспечивает реалистичное отражение динамики мнения клиентов по конкретным продуктам и направлениям деятельности банка.

Интеграция с удобным BI-интерфейсом предоставляет менеджерам и аналитикам наглядные дашборды, позволяющие оперативно выявлять проблемные зоны, отслеживать изменения во времени и определять сильные стороны сервиса по ключевым KPI, повышая эффективность принятия управленческих решений.

# **Описание функциональной и компонентной архитектуры**

**Данные:**

Данные были взяты из нескольких открытых источников, после предобработки итоговый датасет имел следующую структуру:

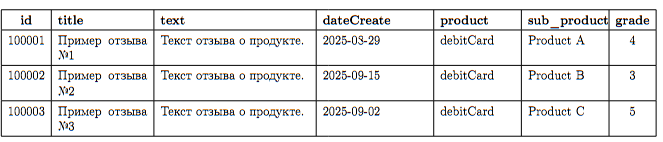


Табл. 1: Формат собранных и предобработанных данных.

Общий размер датасета составил более 50 000 отзывов. Важно, что в полученных отзывах уже присутствовала разметка по категориям, однако, она была не высокого качества по следующим причинам:

- не везде разметка соответствовала полученному классу (например, говорится о кредитных картах, а в разметке debitCards)

- только 1 метка для каждого отзыва, хотя отзыв может затрагивать сразу несколько продуктов или тем

- не по всем отзывам были указаны sub\_products (более точная информация о продукте, например Премиум карта), что не позволяет их использовать достоверно

- дисбаланс в разметке, несмотря на то, что дисбаланс был ожидаем, попадаются топики, по которым накапливалось буквально несколько отзывов, что очевидно не позволило обучить модель, при использовании подобных классов в классификаторе.

Наличие оценок позволит в дальнейшем оценить качество классификации отзывов по тональности.

**Кластеризация:**

Учитывая описанные выше проблемы с исходным датасетом, очевидно, что необходимо было прибегнуть к кластеризации, чтобы выделить более детальные, семантически правильные классы. После получения таких классов и переразметки данных можно будет обучить более робастный и точный классификатор.

Этап кластеризации включает в себя 3 этапа:

**Этап 1:** Предобработка и выделение новых классов

Для начала все текста были предобработаны:

- Удаление тональности из текстов (это мешает выделить семантические кластеры относительно продукта или проблемы, с которой столкнулся клиент)

- Стемминг

- Лемматизация

После этого каждый текст прогоняется через BERT-like модель для генерации эмбеддингов, в нашем кейсе мы использовали предобученную модель от СБЕРа, нацеленную на формирование векторов именно из целых текстов, что помогло улавливать общую тему и посыл каждого отзыва.

Ссылка на модель:

<https://huggingface.co/ai-forever/sbert_large_nlu_ru>

Далее идет этап снижения размерности с помощью методологии UMAP, он позволяет быстро и качественно снизить размерность эмбеддингов, не теряя информативности. Также, не мало важно, что для задач кластеризации высокая размерность вектора может наоборот навредить и вносить шум.

Были проведены несколько экспериментов, оптимальной с точки зрения кластеризации и информативности стала размерность эмбеддингов 128.

После того как мы получили сжатое векторное представление текстов идет этап самой кластеризации. Важно упомянуть, что мы делаем кластеризацию отдельно для каждого класса из изначальной разметки. Это помогает:

- Выделить подклассы в больших группах

- Найти классы, которые пересекаются между продуктами

- Сформировать более точные кластеры, т.к. все наблюдения уже изначально будут близки по какому-то признаку

Для каждого класса из изначальной разметки мы генерируем от 3 до 20 кластеров с помощью K-means++. Выбор k-means обусловлен скоростью и возможностью к работе с высокоразмерными пространствами. Кол-во кластеров зависело от кол-ва наблюдений в изначальном датасете у этого класса.

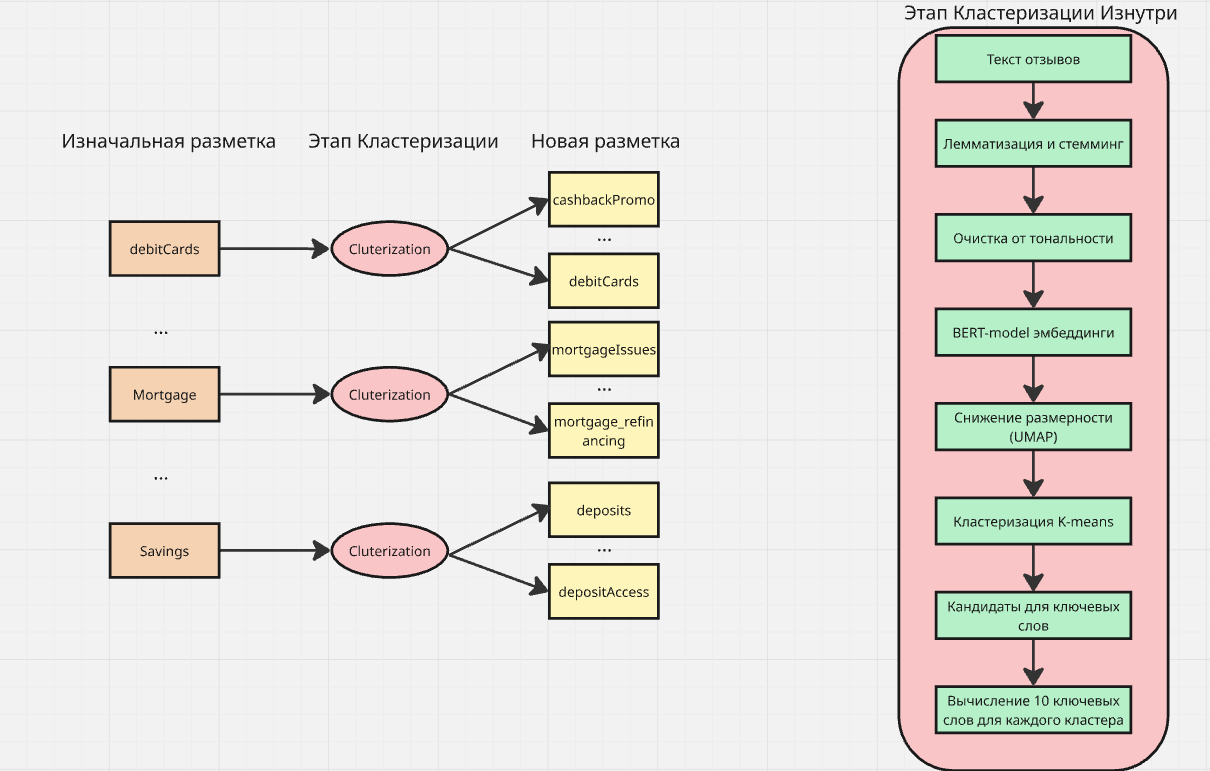


Рис. 1. Первый этап кластеризации

После формирования кластеров идет этап генерации кандидатов. Кандидаты – это N-граммы, которые будут претендовать на слова и словосочетания, которые лучшего всего описывают тот или иной кластер. Они генерируются из всего блока текстов и отзывов, которые присутствуют в выборке конкретного класса (!не кластера). N-граммы также преобразуются в векторное пространство с помощью аналогичной BERT-модели.

Финальным этапом является отбор 10 лучших кандидатов для каждого кластера через подсчет косинусной близости.

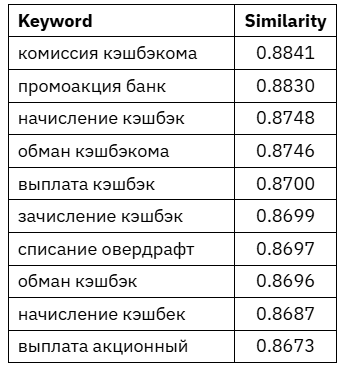


Табл. 2: Пример ключевых слов одного из новых кластеров

**Этап 2:**

На втором этапе с помощью LLM подхода было проанализировано большое кол-во получившихся кластеров. Фокус был на двух факторах:

- Выделить ключевые кластеры и объединить похожие

- Подготовить название и описание кластеров

После выделения новых классов к ним также добавлялись классы из основной разметки, если похожих по описанию не было в новой разметке.

Итоговое количество финальных классов: 23

Их описание представлено в файле: [classes.json](https://github.com/spotlight9k/LCT_2025_GPB_TEAM_SWAGA/blob/main/classes.json)

**Этап 3:**

Последний этап – переразметка. Нам нужно разметить тексты из датасета под новые классы, учитывая возможность мультилейбла.

Для этого на помощь пришел алгоритм Best Match 25 (BM25) – улучшенная версия TF-IDF, которая раньше использовалась поисковиками для ранжирования.

У нас есть предобработанные тексты и есть ключевые слова и описания финальных классов. Используя эти данные считаем полученный скор (релевантность текста к какому то классу) и отбираем по 85% перцентилю.

Таким образом, текст получает метку класса, если его скор достаточно высок (15% самых релевантных текстов для этой темы). Для некоторых классов (например creditCards) этот порог ужесточался.

Такой подход сочетает в себе ряд перимуществ:

- Выделяем более точные и аккуратные классы

- Находим общие для нескольких изначальных топиков классов, которые пересекаются

- Переразметка позволяет добавить multi-label target для всех текстов

- Ключевые слова и описание класса позволяет точно разметить уже сформированные отзывы, а строгий порог увеличивает эту вероятность

Важно упомянуть, что если изначальная метка была взята в финальные классы, то она автоматически присваивается текстом, таким образом, отзывов без лейбла не остается.

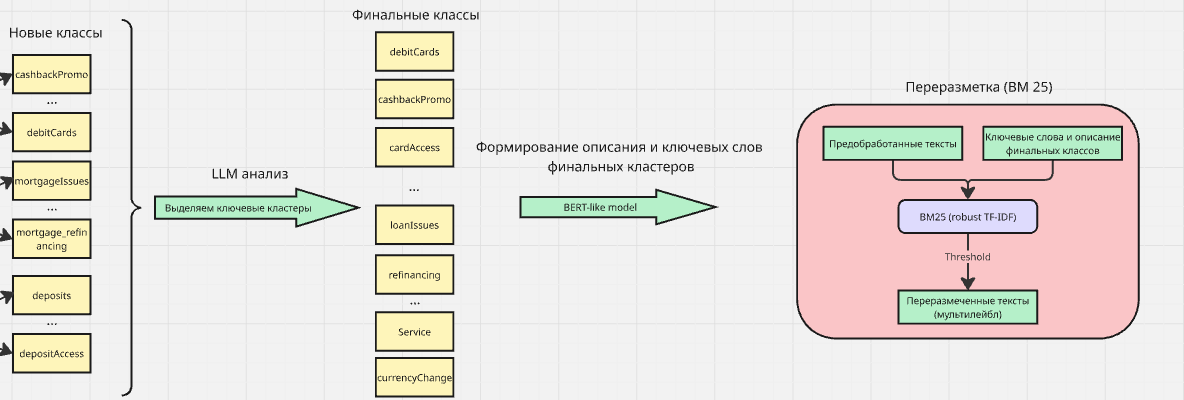


Рис. 2. Второй и третий этапы кластеризации.

**Классификация тем:**

Для задачи классификации мы использовали предобученную модель, обучающуюся на большом блоке текстов, в том числе на комментариях и отзывах.

Модель: ruBert-base

Ссылка на модель: <https://huggingface.co/ai-forever/ruBert-base>

Модель была обучена с учетом регуляризации, новые слои не добавлялись, так как было очень важно не переобучить модель под наш таргет, чтобы перфоманс на golden датасет не просел.

В финальном решении, имея уже правильную разметку из golden датасета, модель можно будет дообучить и увеличить ее перфоманс.

На cross-validation модель показала значение метрики F1-micro = 0.78, что является довольно высоким значением для мультилейбл классификациии.

На таблице представлена более подробная разбивка метрик качества по каждому классу.

**Классификация тональности:**

Для классификации была использована модель анализа тональности текста для русского языка.

Модель: RuBERT for Sentiment Analysis

Ссылка на модель: <https://huggingface.co/blanchefort/rubert-base-cased-sentiment>

Логика довольно проста:

1. Используем предобработанный текст
2. Отдаем его в модель
3. Получаем вероятность тональности текста

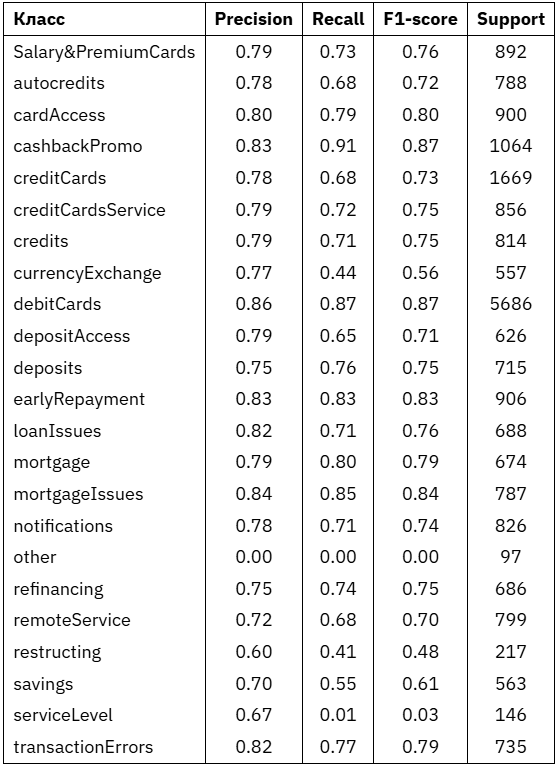


Табл. 3: Значением метрик по классам на CV

Использование более сложной архитектуры, при которой тональность бы подбиралась для каждого продукта отдельно показывает меньшую точность и требует аккуратной разметки данных.

Итоговое распределение отзывов относительно изначальных оценок из датасета показывает, что определение тональности соответствует оценке клиента.

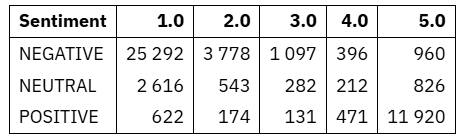


Табл. 4: Распределение сентимента

**BI сервис:**

В качестве сервиса для аналитики было принято решение open-source платформы – Apache SuperSet.

В пользу этого решения было несколько факторов:

- Интеграция с ClickHouse (где хранятся наши витрины)

- Широкий пулл инстурментов для визуализации и аналитики

- Скорость работы и масштабируемость

Для хранения данных были созданы витрины на ClickHouse, которые хранят в себе всю информацию по тексту отзыва, продукте, тональности, а также информации по упоминанию конкурентов.