

**Цикловая комиссия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Специальность** 09.02.07 Информационные системы и программирование

**Курс**  3 **Группа**  ЭдИС-203/21 **Форма обучения**  очно-заочная

**КУРСОВАЯ РАБОТА**

**по**  Проектирование и дизайн информационных систем

**на тему**  Веб приложение для суммаризации текста с помощью алгоритма Text Rank

**Студента**  Авдеева Артёма Игоревича

**Руководитель**  Нагорнова Елена Владимировна

**Работа допущена к защите**

**«** **»** **2023г.**

**Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Ижевск, 2023**

**Содержание**

Введение ……………………………………………………...….......................................3

Глава 1 Введение в автоматическую суммаризацию текста ….......................................4

1.1 Что такое суммаризация текста ...................................................................................4

1.2 Методы автоматической суммаризации ................................................................…..4

1.2.1 Методы на основе правил ...................................................................................…..5

1.2.2 Статистические методы .............................................................................................6

1.2.3 Графовые методы .......................................................................................................7

1.2.4 Методы машинного обучения ...................................................................................8

Глава 2. Алгоритм TextRank ......................................................................................….....9

2.1 Обоснование выбор ......................................................................................................9

2.2 Принцип работы TextRank ...........................................................................................9

Глава 3. Веб-риложение ....................................................................................................12

3.1 Архитектура веб-риложение ......................................................................................12

3.2 Компоненты приложения ...........................................................................................13

3.2.1 Frontend .....................................................................................................................13

3.2.2 Backend .....................................................................................................................16

3.2.2 Модуль суммаризации ............................................................................................18

3.3 Коммуникация компонентов .....................................................................................18

3.3.1 Коммуникация frontend и backend .........................................................................18

3.3.2 Коммуникация backend и модуля суммаризации .................................................19

3.4 Пример обработки файла ..........................................................................................19

Заключение ........................................................................................................................21

Список использованных источников ..............................................................................22

**Введение**

В наше время количество текстового контента растёт с огромной скоростью. По данным techjury.net за 2021 год за день создаётся по 2.5 квинтиллиона байт новой информации. Нередко возникает потребность получить краткую выдержку из текста с целью:

* создание заголовка для статьи
* ознакомление с содержанием текста перед его детальным прочтением
* сокращение объема текста для экономии времени
* извлечение наиболее важных фрагментов для дальнейшей обработки и анализа

Ручное извлечение такой информации является длительным и затратным процессом.

В рамках курсовой было разработано приложение, позволяющее пользователям загружать файлы до 1 МБ и получать по ним выжимку из предложений, которые лучше всего отражают суть обрабатываемого текста.

Для достижения цели работы был выбран алгоритм TextRank, который основывается на графовой модели текста и использует методы взвешенного анализа вершин для определения наиболее значимых предложений в тексте.

Актуальность данной работы обусловлена повышением потребности в автоматической обработке текстовой информации.

**Глава 1. Введение в автоматическую суммаризацию текста**

**1.1 Что такое суммаризация текста**

Суммаризация, автоматическое реферирование или аннотирование — процесс автоматического создания краткой версии текстового документа, которая содержит основные идеи и ключевые факты исходного текста. Данный процесс позволяет существенно сократить объем текста, облегчить его понимание и обработку. Суммаризация используется в различных областях, включая научные исследования, журналистику, бизнес-анализ и т.д.

Автоматическая суммаризация основана на алгоритмах обработки естественного языка, которые позволяют определить наиболее значимые фрагменты текста и создать короткую выжимку, передающую основную информацию.

**1.2 Методы автоматической суммаризации**

Существуют различные методы автоматической суммаризации текстов, такие как:

* методы на основе правил
* статистические методы
* графовые методы
* методы машинного обучения

**1.2.1 Методы на основе правил**

Методы суммаризации текстов на основе правил, основываются на определенных правилах и эвристических алгоритмах, которые позволяют выделить наиболее важные предложения из текста и использовать их для создания краткой выжимки.

Одним из простых методов является выделение первых предложений из каждого абзаца. Такой подход основан на предположении, что первое предложение в абзаце обычно содержит наиболее важную информацию, а остальные предложения лишь развивают ее. Однако этот метод может дать недостаточно полную или точную выжимку, особенно для более сложных текстов.

Другой метод - выделение предложений, содержащих ключевые слова и фразы. Этот подход основывается на том, что ключевые слова и фразы, которые обычно встречаются в заголовках или подзаголовках статей, содержат наиболее важную информацию. Таким образом, предложения, содержащие эти ключевые слова и фразы, могут быть выделены и использованы для создания выжимки. Однако, этот метод также может быть недостаточно точным и полным, если ключевые слова или фразы не являются достаточно репрезентативными для всего текста.

Суммируя, методы, основанные на правилах, не самый лучший выбор, так как:

* требуют дополнительного анализа со стороны разработчиков, для выявления и формулировки правил, а это в свою очередь:
  + трудозатратно
  + может повлечь ошибки, связанные с человеческим фактором
* не универсально, так как для разного рода данных потребуются свои правила
* может быть затратно по вычислительным ресурсам, если правило будет слишком комплексным

**1.2.2 Статистические методы**

Статистические методы - это подход к автоматической суммаризации текстов, который основан на анализе статистических характеристик текста. В этом подходе используются различные метрики, такие как: частота слов, длина предложения, семантическая близость между предложениями и словами и т.д., чтобы определить значимость каждого слова или предложения.

Для применения статистических методов необходимо провести предварительную обработку текста, включающую токенизацию (разбиение текста на отдельные слова или фразы), удаление стоп-слов (наиболее часто используемых слов, которые не несут смысловой нагрузки), лемматизацию (приведение слов к их базовой форме).

Одним из наиболее популярных статистических методов является метод частотности (frequency-based), который основан на частоте встречаемости слов и предложений в тексте. В этом методе наиболее значимыми считаются слова и предложения, которые встречаются в тексте наибольшее количество раз.

Другим популярным статистическим методом является метод TF-IDF (term frequency-inverse document frequency), который учитывает не только частоту встречаемости слов в конкретном тексте, но и их важность для всей коллекции текстов. Он использует два показателя: частоту слова в документе и обратную частоту слова в корпусе документов. Частота слова (TF) определяет, насколько часто данное слово встречается в документе, а обратная частота слова (IDF) оценивает, насколько уникально это слово для данного документа по сравнению со всеми другими документами в корпусе. Чем выше значение TF-IDF для слова, тем больше оно важно для данного документа.

Статистические методы могут давать хорошие результаты в автоматической суммаризации текстов, особенно при работе с большими объемами данных, не требуя на вычисления больших ресурсов. Однако, они не всегда учитывают контекст и семантику текста, что может приводить к ошибкам в выделении наиболее значимых слов и предложений.

**1.2.3 Графовые методы**

Графовые модели - это подход к автоматической суммаризации текстов, основанный на создании графа предложений. В этом подходе каждое предложение рассматривается как узел графа, а связи между узлами определяются сходством между предложениями. Самые значимые предложения вычисляются путем анализа взвешенных связей между узлами графа.

Одним из главных преимуществ графовых методов является их способность учитывать контекст и связи между предложениями. В отличие от статистических методов, где каждое предложение рассматривается в отрыве от остальных, графовые методы учитывают взаимосвязи между предложениями, что позволяет создавать более качественные и информативные краткие содержания.

Однако графовые методы могут быть более сложными в реализации и требуют большего объема вычислительных ресурсов, чем статистические методы. Кроме того, графовые методы могут страдать от проблемы избыточности информации, когда одни и те же факты или идеи повторяются в нескольких предложениях.

Одним из самых известных графовых методов является TextRank. Подробнее данный алгоритм будет рассмотрен в отдельной главе.

**1.2.4 Методы машинного обучения**

Методы машинного обучения - это подход к автоматической суммаризации текстов, основанный на алгоритмах и моделях, которые способны обучаться на большом количестве текстовых данных. Эти методы используются для определения наиболее важных слов и предложений в тексте, а также для создания связей между ними.

Одним из наиболее популярных методов машинного обучения для автоматической суммаризации текстов является метод нейросетей. Нейронные сети - это математические модели, которые состоят из большого количества соединенных между собой узлов. Они обучаются на большом количестве текстовых данных и способны определять значимость каждого слова и предложения в тексте.

Данный метод показывает отличные результаты при должном обучении, но требует большого количества вычислительных ресурсов.

**Глава 2. Алгоритм TextRank**

**2.1 Обоснование выбора**

На основе всех преимуществ и недостаток перечисленных в первой главе методов суммаризации текста, был выбран графовый метод TextRank, т.к данный алгоритм:

* не требуется больших вычислительных ресурсов, а значит прост в использовании и масштабировании
* не требует предварительного анализа входных данных, т.е является универсальным
* не требует предварительной настройки (обучения) алгоритма, что добавляет ещё один плюс в простоту использования
* как показывают результаты из разных источников (<https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/>, <https://cran.r-project.org/web/packages/textrank/vignettes/textrank.html>), данный алгоритм хорошо справляется с поставленной для него задачей

**2.2 Принцип работы TextRank**

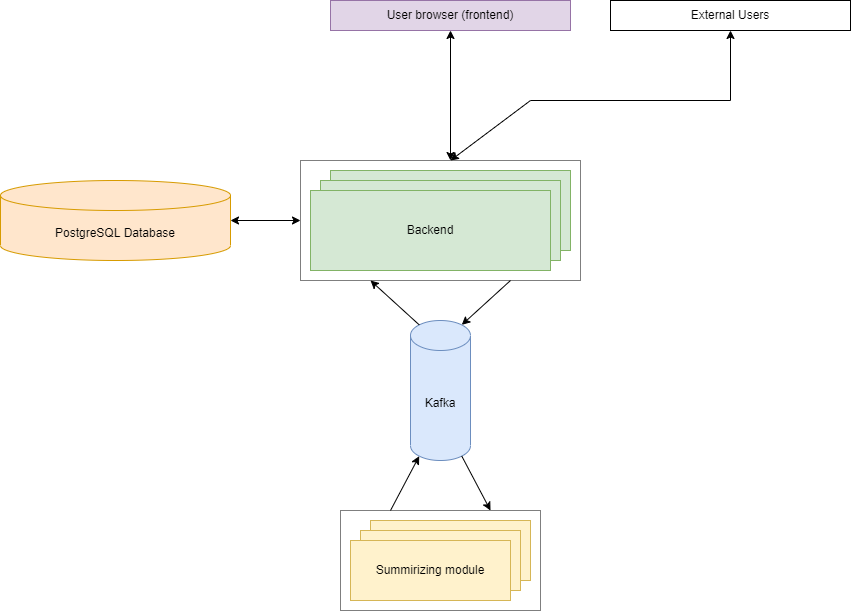
Полный алгоритм TextRank описан Rada Mihalcea и Paul Tarau в документе «TextRank: Bringing Order into Texts». В данной курсовой работе привожу краткое описание данного алгоритма.

Алгоритм:

1. Подготовка текста:
   * Удаление нежелательных символов, стоп-слов (часто встречающихся и не несущих смысловой нагрузки слов) и прочих форматирований
   * Разбиение текста на предложения
2. Построение графа:
   * Каждое предложение в тексте представляется узлом графа
   * Для каждой пары предложений определяется семантическая схожесть с использованием метрик, таких как косинусное расстояние
   * Вес ребра между двумя предложениями определяется их семантической схожестью
3. Итеративный алгоритм TextRank:
   * Начальное значение важности каждого предложения устанавливается равным 1
   * Выполняется либо заданное количество итераций, либо пока алгоритм не сойдётся
   * На каждой итерации для каждого предложения вычисляется его важность путем суммирования весов входящих ребер, умноженных на важность исходящих узлов. Формула вычисления важности может быть аналогична алгоритму PageRank (описан Sergey Brin и Lawrence Page в документе The anatomy of a large-scale hypertextual Web search engine)
   * Важности предложений обновляются после каждой итерации
4. Выбор наиболее значимых предложений:
   * После завершения итераций важности предложений могут быть ранжированы по убыванию.
   * Топ-N предложений с наибольшей важностью считаются наиболее значимыми и могут быть выбраны для отображения или дальнейшей обработки.

**Глава 3. Веб-приложение**

**3.1 Архитектура веб-прилоежния**

 Рисунок 1 - архитектура приложения

Как показано на рис.1, веб-приложение для суммаризации текста состоит из следующих компонентов:

* frontend
* backend
* Модуль суммаризации текста
* брокера сообщений kafka
* база данных PostgreSQL

Веб-приложение выполнено по микросервесной архитектуре, которая предполагает взаимодействие небольших по размерам модулей со слабой связанностью и лёгкой изменяемостью, с учётом дальнейшей масштабируемости в случае возрастания нагрузки. Компоненты приложения могут быть инстанцированы в нескольких экземплярах для поддержания доступности сервиса. Также, данная архитектура позволяет легко заменять компоненты при поступлении обновлений, без заметного влияния на другие части приложения.

Примечание: диаграмма архитектуры приложения находится в файле *architecture.drawio*.

**3.2 Компоненты приложения**

**3.2.1 Frontend**

Frontend приложения написан на языке JavaScript с использованием фреймворка React и библиотеки компонентов Material UI.

С помощью frontend приложения можно:

* Создать аккаунт

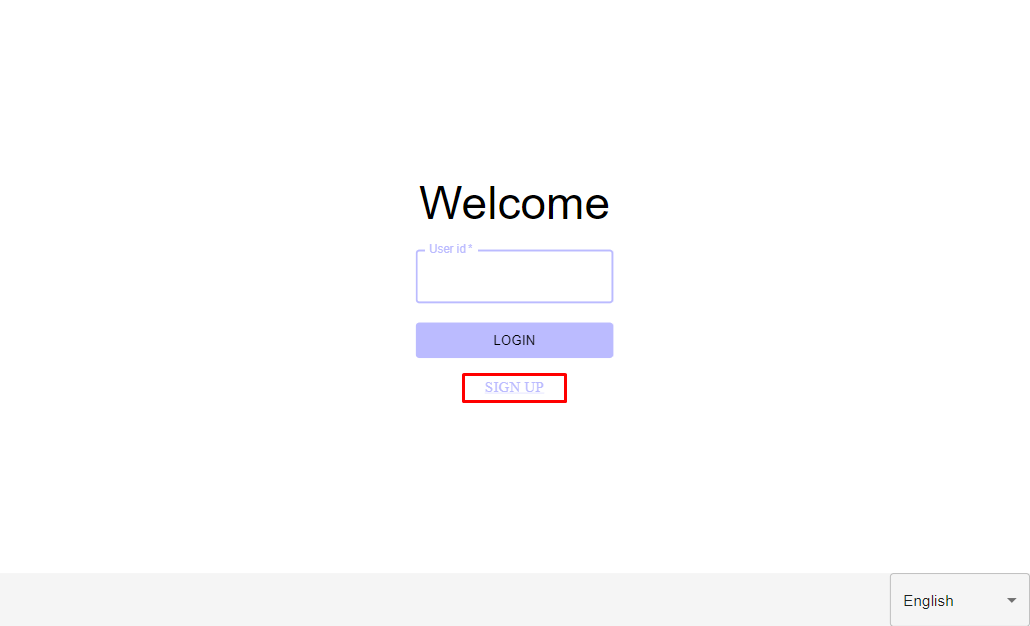


Рисунок 2 - страница входа

* Выполнить аутентификацию и авторизацию

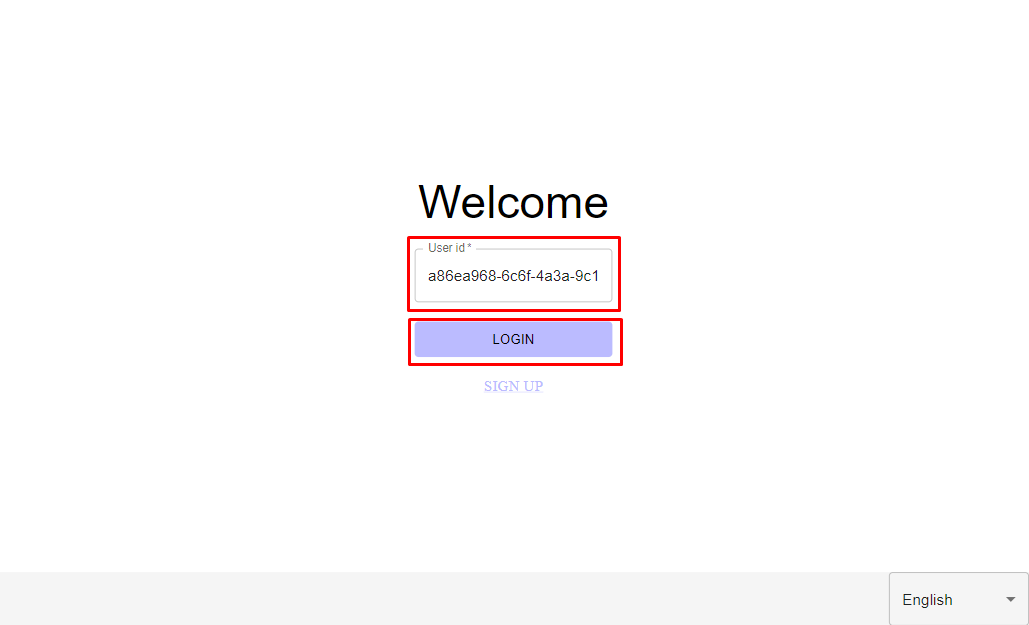


Рисунок 3 - заполненная страница входа

* Загрузить файл на обработку

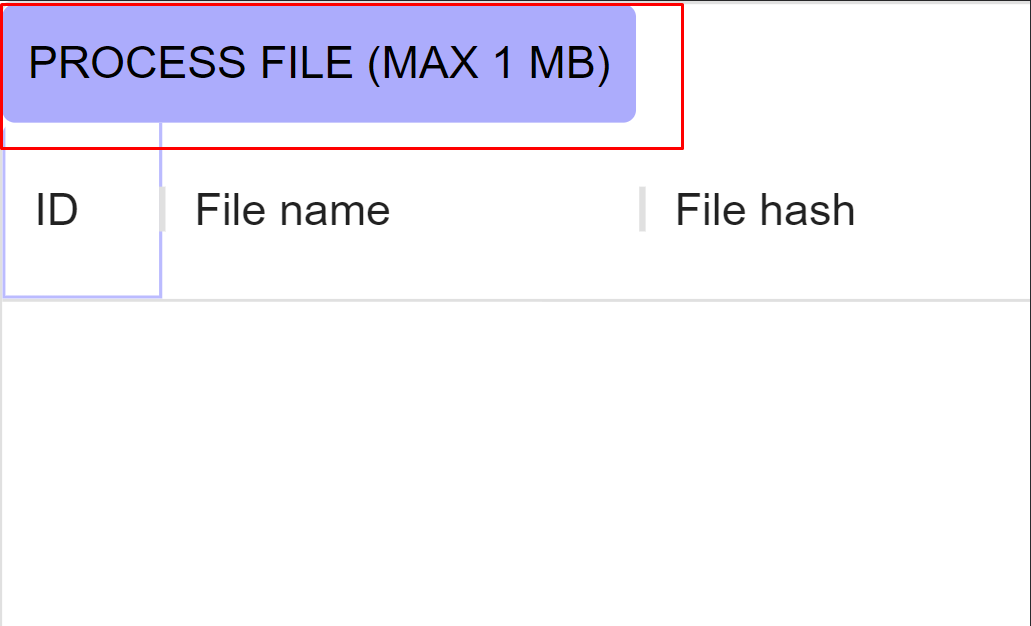


Рисунок 4 - кнопка загрузки файла на обработку

* Получить список статусов и результатов обработки

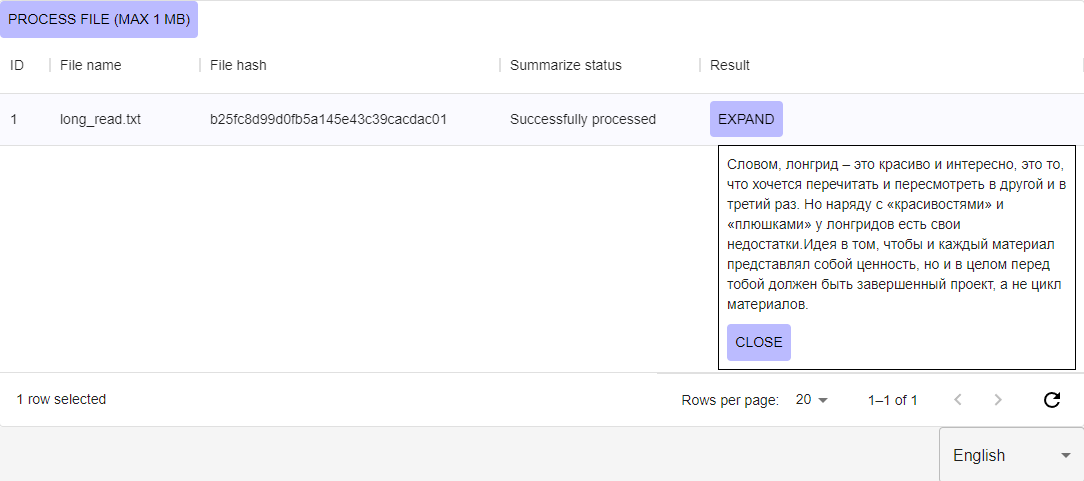


Рисунок 5 - главное окно приложения

Дополнительные возможности:

* Смена локализации приложения

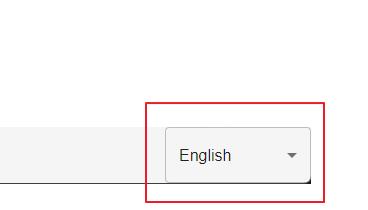


Рисунок 6 - кнопка переключения локализации

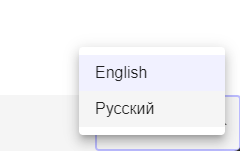


Рисунок 7 - окно выбора локализации приложения

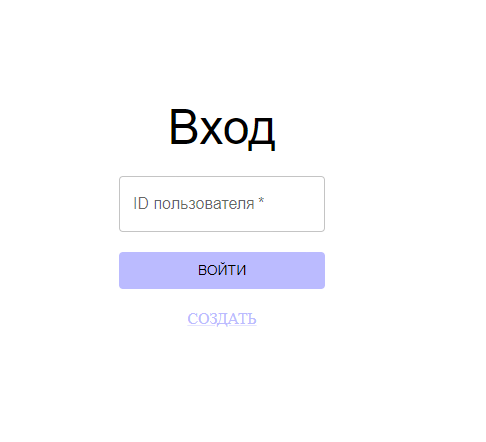


Рисунок 8 - локализованная на русский язык страница входа

**3.2.2 Backend**

Backend приложения написан на языке Kotlin с использованием фреймворка Spring Boot.

Данный модуль отвечает за:

* создание схемы сущностей в БД

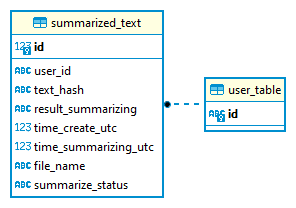


Рисунок 9 - схема БД

* аутентификацию и авторизацию пользователей
* получение файлов от пользователя
* сохранения мета информации по файлам в БД
* передачу файла в модуль суммаризации
* сбор результатов обработки от модуля суммаризации
* сохранение результатов обработки в БД
* предоставление статусов и результатов обработки пользователю

Примечание: API backend находится в файле *swagger.yaml,* схема БД находится в файле *schema.erd*

**3.2.3 Модуль суммаризации**

Модуль суммаризации написан на языке Python с использованием библиотеки sumy, в которой реализован алгоритм TextRank. Пример кода, отвечающего за суммаризацию



**3.3 Коммуникация компонентов**

**3.3.1 Коммуникация frontend и backend**

В качестве протокола общения frontend и backend выбран протокол HTTP API. Данные передаются в формате JSON.

Для аутентификации и авторизации используется уникальный ключ, который выдаётся пользователям при входе в приложение.

**3.3.2 Коммуникация backend и модуля суммаризации**

Backend и модуль суммаризации общаются асинхронно с помощью брокера сообщений Kafka.

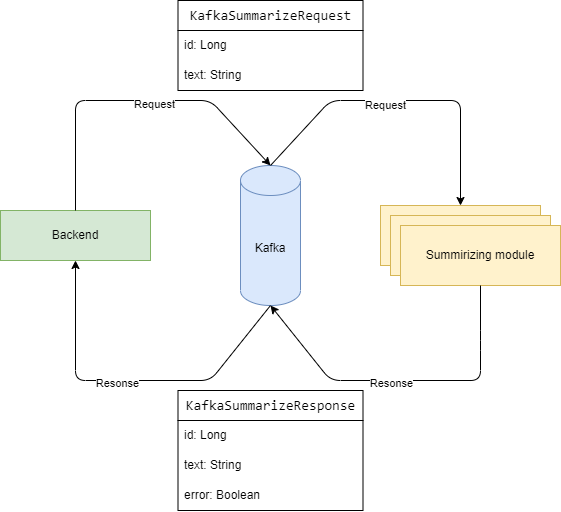


Рисунок 10 - схема коммуникация backend и модуля суммаризации

**3.4 Пример обработки файла**

Результат обработки текста статьи «Просто длинный текст – еще не лонгрид: как интернет-издания осваивают новый формат» (<http://mediakritika.by/article/3133/prosto-dlinnyy-tekst-eshche-ne-longrid-kak-internet-izdaniya-osvaivayut-novyy-format>)



**Заключение**

Была разработана гибкая к масштабированию и дальнейшим доработкам система суммаризации текста с использованием алгоритма TextRank. В дальнейшем, для достижения лучших показателей, алгоритм можно заменить более совершенным, например, с использованием модели машинного обучения.

**Список использованной литературы**

1. <https://techjury.net/blog/how-much-data-is-created-every-day/#gref>
2. Mihalcea, R., Tarau, P.: [“Textrank: Bringing order into texts”](http://www.aclweb.org/anthology/W04-3252). In: Lin, D., Wu, D. (eds.) Proceedings of EMNLP 2004. pp. 404–411. Association for Computational Linguistics, Barcelona, Spain. July 2004. <http://www.aclweb.org/anthology/W04-3252>
3. Sergey Brin, Lawrence Page: [“](http://www.aclweb.org/anthology/W04-3252)The anatomy of a large-scale hypertextual Web search engine[”](http://www.aclweb.org/anthology/W04-3252). <https://snap.stanford.edu/class/cs224w-readings/Brin98Anatomy.pdf>