

Разбор задач второго этапа

23.10.24

Приложение для
чтения книг с AI-
ассистентом

Задание на второй этап отбора

22.10.24–03.11.24

Подготовить решение задания и приложить его в виде **ссылки на открытый репозиторий** на GitHub. Репозиторий может содержать как код, так и любые другие файлы. В файле README.md **необходимо** описать, какой набор файлов представлен.

Ссылку необходимо добавить в исходную презентацию и подгрузить обновлённый файл **в заявку капитана** команды (вкладка «Проект отборочный этап 2 тур») **до 23:59 (МСК) 03.11.**

Решение задания не является обязательным, но даст возможность получить дополнительные баллы на онлайн-собеседовании с экспертами.

Задание на второй этап отбора

Что нужно сделать

В рамках дополнительного задания вам будет предложено разработать систему для сжатия текста на двух уровнях: **сильное сжатие** (одно-два предложения) и **слабое сжатие** (краткий абзац).

Необходимо реализовать программу на Python с интерфейсом для взаимодействия с пользователем (интерфейс может быть любым: консоль, Telegram-бот, Streamlit и др.).

Разрешено использовать любые алгоритмы обработки текста, включая open source LLM (Large Language Models), однако **запрещено** использовать внешние API (например, OpenAI, Hugging Face API и т.д.).

Задание на второй этап отбора

Основные требования

Функции программы:

- **Ввод текста пользователем.**
Пользователь должен иметь возможность ввести текст (например, небольшой рассказ или отрывок из книги), который нужно сжать.
- **Выбор уровня сжатия.**
При сильном сжатии текст сокращается до одного или двух предложений, передавая основную мысль. При слабом – до краткого абзаца, сохраняя больше деталей и событий.
- **Вывод результата.**

Требования к реализации:

- **Язык программирования** — Python.
- **Запрещается** использовать внешние API для суммаризации текста (например, OpenAI API, GPT API, Hugging Face API и т.д.).
- **Разрешается** использовать любые алгоритмы и подходы к обработке текста, включая большие языковые модели (LLM) и классические алгоритмы обработки текста (например, суммаризация на основе TF-IDF, TextRank, Gensim, NLTK и др.).

Задание на второй этап отбора

Основные требования

Интерфейс для взаимодействия

Программа должна предоставлять удобный способ взаимодействия с пользователем.

Возможные варианты:

- **Консольный интерфейс.**

Программа запрашивает ввод текста и уровень сжатия через терминал.

- **Web-интерфейс на Streamlit.**

Пользователь вводит текст в веб-интерфейсе и выбирает уровень сжатия через удобные поля.

- **Telegram-бот.**

Пользователь отправляет текст боту и выбирает уровень сжатия с помощью команды

- **Другие варианты.**

Можно предложить любой другой удобный способ взаимодействия.

Задание на второй этап отбора

Критерии оценки решения

- **Правильность сжатия текста**

Оценка того, насколько точно программа выделяет главное содержание текста для каждого уровня сжатия (0.5 балла).

- **Качество кода**

Чистота и структурированность кода, использование Pythonic-подходов, комментарии и соблюдение стиля PEP8 (0.25 балла).

- **Алгоритмическое решение**

Насколько правильно и эффективно выбран алгоритм для сжатия текста (0.5 балла).

- **Интерфейс**

Удобство взаимодействия с программой, понятность интерфейса (0.5 балла).

- **Документация**

Наличие краткой инструкции по запуску программы и объяснения используемых методов и демоверсия (видео) работы программы (0.25 балла).



Резюме

- Для интеграции с LLM используем Open Source LLM (Llama, Saiga, ...) + библиотеку *llama_cpp* и *langchain*.
- Линтеры (когда готов проект, перед отправкой).
- У кого прототип tg-bot, FastAPI (продолжаем дорабатывать его).
- Веб форма (Обработка входных данных, переключатель (сильное / слабое сжатие)).
- **Сложность ****

Ассистент для работы с
научной литературой

Задание на второй этап отбора

22.10.24–03.11.24

Подготовить решение задания и приложить его в виде **ссылки на открытый репозиторий** на GitHub. Репозиторий может содержать как код, так и любые другие файлы. В файле README.md **необходимо** описать, какой набор файлов представлен.

Ссылку необходимо добавить в исходную презентацию и подгрузить обновлённый файл **в заявку капитана** команды (вкладка «Проект отборочный этап 2 тур») **до 23:59 (МСК) 03.11.**

Решение задания не является обязательным, но даст возможность получить дополнительные баллы на онлайн-собеседовании с экспертами.

Задание на второй этап отбора

Что нужно сделать

В рамках дополнительного задания вам будет предложено разработать минимально жизнеспособный продукт (MVP) web-приложения для краткого пересказа (резюмирования) научных статей.

В качестве результата работы в репозиторий необходимо добавить **блокнот Jupyter** (документация) с анализом данных и само web-приложение.

Для разработки системы резюмирования рекомендуется использовать датасет PubMed.

Задание на второй этап отбора

Что нужно сделать

Подзадачи

1. **Провести исследовательский анализ данных.**
На каких языках написаны статьи? Какие тематики можно выделить? Какие проблемы есть в данных?
2. **Выбрать и обосновать метрики оценки качества резюмирования.**
3. **Протестировать не менее 2-х моделей машинного обучения для формирования резюме статей. Сделать выводы.**

Задание на второй этап отбора

Что нужно сделать

Подзадачи

4. **Разработать прототип web-сервиса**, в который можно загрузить статью формата pdf и получить её краткое содержание в виде текста. Рекомендовано использовать фреймворк streamlit, но можно пробовать и другие решения.
5. **Написать Readme для репозитория**.
Ознакомиться с рекомендованным оформлением можно [здесь](#).
6. **Создать файл requirements.txt** со всеми необходимыми зависимостями для воспроизведения вашего решения.

Задание на второй этап отбора

Критерии оценки

За решение задачи можно получить от 0 до 2 баллов в зависимости от его проработки.

Баллы	Условие получения
+0.4	Выполнена подзадача 1
+0.25	Выполнена подзадача 2
+0.6	Выполнена подзадача 3
+0.5	Выполнена подзадача 4
+0.25	Выполнены суммарно подзадачи 5-6

Резюме

- Анализ датасета pubmed (jupyter notebook) + метрики + две модели протестировать
- Веб-приложение (та же форма, что и в предыдущем кейсе)
<https://streamlit.io/>
- **Сложность** ***

Интеллектуальный
помощник для
создания учебных
материалов

Задание на второй этап отбора

22.10.24–03.11.24

Подготовить решение задания и приложить его в виде **ссылки на открытый репозиторий** на GitHub. Репозиторий может содержать как код, так и любые другие файлы. В файле README.md **необходимо** описать, какой набор файлов представлен.

Ссылку необходимо добавить в исходную презентацию и подгрузить обновлённый файл **в заявку капитана** команды (вкладка «Проект отборочный этап 2 тур») **до 23:59 (МСК) 03.11.**

Решение задания не является обязательным, но даст возможность получить дополнительные баллы на онлайн-собеседовании с экспертами.

Задание на второй этап отбора

Что нужно сделать

В рамках дополнительного задания вам будет предложено написать программный код для суммаризации видеороликов с Яндекс.Диска и обернуть это в собственный API.

Подзадачи:

1. Реализовать скрипт для работы с API Яндекс.Диска.
2. Реализовать скрипт для суммаризации текста из видеороликов.
3. Реализовать собственный API.

Задание на второй этап отбора

Полезные материалы

Инструменты для работы с API и суммаризацией

- [Знакомство с FastAPI.](#)
- [Документация API Yandex Cloud.](#)
- [Документация FastAPI.](#)
- [Руководство по суммаризации с HuggingFace.](#)

Задание на второй этап отбора

Критерии оценки

За решение задачи можно получить от 0 до 2 баллов в зависимости от его проработки.

Баллы	Условие получения
+0.5	Реализована выгрузка данных с хранилища.
+0.5	Реализован скрипт для выгрузки и транскрипции.
+0.5	Реализован скрипт для суммаризации.
+0.5	Система полностью реализована.

Резюме

- Микросервис предоставляющий API для суммаризации видео
- Yandex API для обработки видео из Я.Диска
- **Самая сложная задача*******

Ассистент отеля для
создания описания отеля

Задание на второй этап отбора

22.10.24–03.11.24

Подготовить решение задания и приложить его в виде **ссылки на открытый репозиторий** на GitHub.

Репозиторий должен содержать:

- данные, с которыми вы работали,
- метод сбора данных (код, ноутбук),
- исследовательский анализ данных,
- ноутбук с моделью-классификатором, предсказывающей рейтинг отеля по его отзывам.

Ссылку необходимо добавить в исходную презентацию и подгрузить обновлённый файл **в заявку капитана** команды (вкладка «Проект отборочный этап 2 тур») **до 23:59 (МСК) 03.11.**

Решение задания не является обязательным, но даст возможность получить дополнительные баллы на онлайн-собеседовании с экспертами.

Задание на второй этап отбора

Что нужно сделать

В рамках дополнительного задания вам будет предложено найти данные по отзывам на отели, проанализировать их и классифицировать рейтинги отелей на основе текстов отзывов.

Это задание является важным шагом в разработке AI-ассистента, который будет автоматически генерировать описания отелей на основе отзывов и других данных, помогая пользователям лучше ориентироваться в выборе отеля.



Задание на второй этап отбора

Критерии оценки решения

Метод сбора данных:

- 0 баллов. Данные не были собраны.
- 1 балл. Использован набор данных, полученных по готовому коду из базового решения.
- 2 балла. Реализован собственный сбор данных.

Исследовательский анализ данных:

- 0 баллов. Анализ данных отсутствует.
- 1 балл. Выполнен частичный анализ данных.
- 2 балла. Выполнен подробный и полный анализ данных.

Модель-классификатор:

- 0 баллов. Модель не реализована.
- 1 балл. Модель построена, но без проведённых экспериментов.
- 2 балла. Модель построена, эксперименты проведены (присутствует подбор модели, гиперпараметров и т.д.)



Важно, что итоговый балл выставляется как сумма баллов по трём критериям, делённая на 3.

Задание на второй этап отбора

Полезные материалы

В помощь участникам приложен блокнот Jupyter, который содержит:

- теоретическую и практическую информацию о поиске данных,
- основные аспекты анализа текстовых данных,
- методы подготовки данных к работе модели,
- пример базовой модели-классификатора (baseline).

Данные материалы призваны лишь подготовить участников к основному заданию, не стоит копировать baseline целиком.

Резюме

- Спарсить данные (jupyter notebook)
- Анализ собранных данных (jupyter notebook)
- Создание baseline модели
- Модель классификатор
- **Сложность ******

- https://t.me/+9FZx3nP_c28zMjE6
(чат/вопросы)
- <https://github.com/kirilltobola/perseus-sirius-ai>
(материалы)