**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНОЙ РОБОТОТЕХНИКИ**

Кафедра Интеллектуальных систем теплофизики ИИР

Направление подготовки 15.03.06 Мехатроника и робототехника

Направленность (профиль) Мехатроника и робототехника

# ОТЧЕТ

**о прохождении производственной практики, технологической (проектно-технологической) практики**

(указывается наименование практики)

**Обучающегося Сыренного Ильи Игоревича группы № 21930 курса 4**

**Тема задания**: Разработка интерактивного учебного пособия с ответами на естественном языке на основе Retrieval Augmented Generation

**Место прохождения пратики:** Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Новосибирский национальный исследовательский государственный университет». 630090, Новосибирская область, г. Новосибирск, ул. Пирогова, д. 1

**Сроки прохождения практики:** с 30.09.2024 г. по 23.12.2024 г.

**Руководитель практики от НГУ** Галактионова Юлия Юрьевна, специалист УМОВОИИР /_____
(Ф.И.О. полностью, должность) (подпись)

**Руководитель ВКР** Оглезнев Никита Сергеевич, сотрудник КафИСТИИР, ассистент /_____
(Ф.И.О. полностью, должность) (подпись)

**Оценка по итогам защиты отчета:** _____
(неудовлетворительно, удовлетворительно, хорошо, отлично)

**Отчет заслушан на заседании кафедры** КафИСТИИР
(наименование кафедры)

**протокол _____от «_____» _____** 20_____г.

Новосибирск 2025

СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

С ростом популярности генеративных моделей возникает необходимость в объективных методах оценки их качества. Классические метрики, такие как BLEU и ROUGE, зачастую не отражают смысловую точность и полноту сгенерированного ответа, что ограничивает их применение для современных языковых моделей.

В связи с этим активно исследуются новые подходы, включая семантические метрики (например, BERTScore) и нейросетевые методы оценки (LLM-as-Judge), которые позволяют учитывать перефразирование, синонимию и контекстную точность.

В данной работе оценивается качество языковых моделей в различных конфигурациях, включая стандартную генеративную модель (LLM) и Retrieval-Augmented Generation (RAG). Оценка проводится на датасете FRAMES с использованием современных метрик.

# 1 МЕТОДЫ ОЦЕНКИ РАБОТЫ СИСТЕМЫ

## 1.1 BERTScore

BERTScore [1] — это метрика для оценки качества сгенерированных текстов, основанная на использовании эмбеддингов слов, извлечённых с помощью предобученных трансформеров, таких как BERT или его аналогов. В отличие от традиционных метрик, BERTScore вычисляется через косинусное сходство между токенами эталонного текста и сгенерированного ответа.

Основной особенностью BERTScore является то, что он учитывает контекст и семантику слов, так как эмбеддинги слов отражают их значении в контексте всей фразы. Это позволяет метрике учитывать перефразирование, синонимию и другие формы лексической вариативности, которые часто не захватываются традиционными метриками, такими как BLEU или ROUGE.

## 1.2 LLM-as-Judge

Современный метод оценки генерации, при котором большая языковая модель (LLM) выступает в роли судьи [2]. LLM оценивает ответы по заранее заданным критериям, таким как достоверность, релевантность и полнота.

### 1.2.1 Faithfulness

Метрика достоверности (Faithfulness [3]) измеряет, насколько фактически точен ответ относительно извлеченного контекста. Значения вариьируются от 0 до 1, где 1 означает, что утверждения в ответе подтверждаются контекстом.

Алгоритм расчета метрики:

1 Разбиение ответа на утверждения.

2 Каждое утверждение проходит проверку на релевантность к извлеченному контексту.

3 Вычисление показателя достоверности по формуле:

$$\text{Faithfulness Score} = \frac{\text{Количество релевантных к контексту утверждений}}{\text{Общее количество утверждений}} \quad (1)$$

### 1.2.2 Response Relevancy

Метрика релевантности ответа (Response Relevancy [3]) оценивает, насколько ответ соответствует пользовательскому запросу. Высокий балл означает, что ответ полноценно отвечает на запрос, без избыточной или нерелевантной информации.

Алгоритм расчета метрики:

1 Генерация 3 искусственных вопросов на основе ответа системы.
2 Вычисление косинусного сходства между векторными представлениями запроса и каждого из вопросов.
3 Усреднение полученных значений.

### 1.2.3 Answer Correctness

Метрика корректности ответа (Answer Correctness [3]) оценивает соответствие ответа эталонному. Балл варьируется от 0 до 1, где 1 означает полное совпадение.

Алгоритм расчета метрики:

1 Оценка фактической корректности: измеряет степень совпадения фактов между сгенерированным ответом и эталонным. Рассчитывается, как f1-score:

$$F1\ Score = \frac{2 \times TP}{2 \times TP + \mathrm{FP} + \mathrm{FN}} \quad (2)$$

где TP – количество утверждений, присутствующих и в эталонном, и в сгенерированном ответах. FP – количество утверждений, которые есть в сгенерированном ответе, но отсутствуют в эталонном. FN – количество утверждений, которые есть в эталонном ответе, но отсутствуют в сгенерированном.

2 Оценка семантического сходства: рассчитывается на основе косинусной близости эмбеддингов сгенерированного и эталонного ответов.
3 Ответ: рассчитывается, как взвешенное среднее между фактическим соответствием и семантическим сходством

# 2 ДАТАСЕТ

Первоначально я рассматривал датасет QASPER для тестирования системы, однако в процессе текущей практики я выявил его ограничения: вопросы в датасете слишком общие, к тому же каждый вопрос связан только с одной статьей. Эти ограничения делают QASPER малопригодным для тестирования сложных retrieval-механизмов.

## 2.1 FRAMES

В итоге я выбрал датасет FRAMES (Factuality, Retrieval, And reasoning MEasurement Set) [4]. Основные характеристики датасета:

1. 824 сложных вопросов, требующих информации из 2-15 статей из Wikipedia.
2. Широкий спект тем: история, спорт, наука, здоровье, животные и др.
3. Для каждого вопроса представлен эталонный ответ, а также список релевантных статей из Wikipedia.

Датасет был разработан и выпущен 24 января 2025 года исследователями из Google с целью создания стандартизированного набора задач для оценки работы Retrieval-Augmented-Generation систем. В статье [4] авторы демонстрируют, что даже передовые большие языковые модели, такие как Gemini-Pro-1.5, сталкиваются со значительными трудностями при ответах на вопросы из FRAMES. Это подтверждает, что добавление поискового механизма может существенно улучшить качество ответов.

## 2.2 Подготовка датасета

Для тестирования я случайным образом отобрал из датасета 50 вопросов, и проиндексировал их статьи. Чтобы приблизить эксперимент к реальным условиям, в индекс также были добавлены нерелевантные статьи, содержащие информацию по другим вопросам. Этот шаг позволяет проверить, насколько эффективно система извлекает нужные данные и игнорирует лишнюю информацию. Итоговая выборка включает 330 статей и 50 вопросов.

# 3 КОНФИГУРАЦИИ

В данном разделе рассматриваются три конфигурации, использованные для оценки качества: стандартная Retrieval-Augmented Generation (RAG), версия Agentic RAG и LLM без поиска.

Во всех конфигурациях использовалась большая языковая модель openai/gpt-4o-mini [5] и эмбеддинг-модель deepvk/USER-bge-m3 [6].

## 3.1 RAG

Стандартная конфигурация RAG включает два основных этапа:

1. Индексация. На данном этапе документы преобразуются в чанки с помощью алгоритма семантического деления [7]. Далее чанки индексируются с помощью алгоритма Okapi bm25 [8].
2. Инференс. На этапе инференса в систему приходит запрос пользователя, для которого необходимо найти релевантные чанки. Чанки извлекаются в два этапа [здесь бы еще про two-shot-retrieval]: сперва 10 чанков находятся с помощью алгоритма Okapi bm25 [8], далее среди отобранных чанков выбираются лучшие с помощью биэнкодер-ранжирования. Затем чанки подаются на вход генеративной модели для формирования финального ответа пользователю.

## 3.2 Agentic RAG

Agentic RAG — это усовершенствованный подход к построению RAG. В отличие от традиционного RAG, где модель генерирует ответы на основе единственного шага извлечения данных, Agentic RAG позволяет большой языковой модели самостоятельно формулировать запросы, критически оценивать полученные результаты и при необходимости повторно обращаться к источникам для уточнения информации. Использует ту же систему индексации и поиска по базе знаний, что и стандартный RAG.

## 3.3 LLM

Дополнительно была протестирована модель LLM без поиска, где генерация ответа происходит исключительно на основе входного вопроса, без извлечения контекста из базы данных.

# 4 АНАЛИЗ РЕЗУЛЬТАТОВ

Стоит отметить, что большие языковые модели склонны к более лояльной оценке своих текстов [9], поэтому для метрик LLM-as-Judge я использовал другую модель: google/gemini-pro [5].

## 4.1 Сравнение метрик на разных конфигурациях

| | RAG | Agentic-RAG | LLM |
|---|---|---|---|
| Faithfulness | 0.54 | 0.08 | 0.25 |
| Response Relevancy | 0.29 | 0.6 | - |
| Answer Correctness | 0.33 | 0.49 | 0.53 |
| BERTScore | 0.8 | 0.82 | 0.81 |
| Среднее время работы (с) | 6.5 | 29 | 2.37 |

## 4.2 Выводы по метрикам

1. RAG требует доработки алгоритма чанкинга для улучшения качества поиска и общего качества извлекаемых чанков. Нужно исследовать эвристические подходы к формированию чанков и сравнить с текущим семантическим подходом.
2. Agentic-RAG показывает лучшие результаты по извлечению фрагментов благодаря множественным вызовам поиска по базе знаний. Однако необходимо улучшить формулировку промежуточных запросов для повышения их релевантности и точности, также переработать запрос для формирования финального ответа.

# 5 РАЗРАБОТКА ПРИЛОЖЕНИЯ

Поскольку исходный код не соответствовал требованиям модульности и масштабируемости, я провел его глубокую переработку. Для серверной части использовал OpenAPI-спецификацию OpenAI-compatible серверов [10], что позволило создать унифицированное API, совместимое со многими клиентскими приложениями (см. Приложение А).

## 5.1 Концепт интерфейса

Для удобной работы с научными статьями я обновил интерфейс приложения. Теперь он включает PDF-viewer в левой части экрана для отображения загруженного документа и чат в правой части для взаимодействия с системой. Дополнительные элементы интерфейса включают кнопки загрузки документа, возможность поделиться перепиской и другие функции.

Для чата я использовал открытую реализацию интерфейса для взаимодействия с LLM — “HuggingFace chat-ui” [11]. Следующая задача — адаптировать этот интерфейс под специфику моей системы.

# ЗАКЛЮЧЕНИЕ

В ходе практики я провел анализ методов оценки качества генерации текстов различными конфигурациями языковых моделей. Основное внимание уделялось сравнительному анализу метрик BERTScore и LLM-as-Judge на датасете FRAMES. Полученные результаты позволили выявить направления для улучшения системы

Код проекта доступен на платформе GitHub (См. приложение А).

Дальнейшие шаги:

- Оптимизация методов генерации с учетом выявленных слабых мест
- Адаптация интерфейса
- Тестирование серверной части

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


1 BERTScore: Evaluating Text Generation with BERT // arXiv preprint arXiv:1904.09675. — 2019.

2 A Survey on LLM-as-a-Judge // arXiv preprint arXiv:2411.15594. — 2024.

3 RAGAS: Automated Evaluation of Retrieval Augmented Generation // arXiv preprint arXiv:2309.15217. — 2023.

4 Fact, Fetch, and Reason: A Unified Evaluation of Retrieval-Augmented Generation // arXiv preprint arXiv:2409.12941. — 2024.

5 VseGpt – провайдер для больших языковых моделей [Электронный ресурс]. — 2025. — Режим доступа: https://vsegpt.ru/Docs/Models (дата обращения: 26.02.2025).

6 HuggingFace [Электронный ресурс]. — 2025. — Режим доступа: https://huggingface.co/deepvk/USER-bge-m3 (дата обращения: 26.02.2025).

7 5 Levels Of Text Splitting [Электронный ресурс]. — 2025. — Режим доступа: https://github.com/FullStackRetrieval-com/RetrievalTutorials/blob/main/tutorials/LevelsOfTextSplitting/5_Levels_Of_Text_Splitting.ipynb (дата обращения: 26.02.2025).

8 Robertson S.E., Walker S., Jones S., Hancock-Beaulieu M., Gatford M. Okapi at TREC-3 // Proceedings of the Third Text REtrieval Conference (TREC 1994). Gaithersburg, USA, November 1994.

9 Pride and Prejudice: LLM Amplifies Self-Bias in Self-Refinement // arXiv preprint arXiv:2402.11436. — 2024.

10 OpenAPI specification for the OpenAI API [Электронный ресурс]. — 2025. — Режим доступа: https://github.com/openai/openai-openapi (дата обращения: 26.02.2025).

11 Open source codebase powering the HuggingChat app [Электронный ресурс]. — 2025. — Режим доступа: https://github.com/huggingface/chat-ui (дата обращения: 26.02.2025).

# ПРИЛОЖЕНИЕ А

Код проекта на платформе GitHub: https://github.com/Syrenny/educ