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 3
1 МЕТОДЫ ОЦЕНКИ РАБОТЫ СИСТЕМЫ 4
1.1 Семантические метрики: BERTScore 4
1.2 LLM-as-Judge 4
1.2.1 Faithfulness 4
1.2.2 Response Relevancy 4
1.2.3 Answer Correctness 5
2 КОНФИГУРАЦИИ 6
2.1 RAG 6
2.2 Agentic RAG 6
2.3 LLM 6
3 АНАЛИЗ РЕЗУЛЬТАТОВ 7
3.1 Сравнение метрик на разных конфигурациях 7
3.2 Выводы по метрикам 7
4 РАЗРАБОТКА ПРИЛОЖЕНИЯ 8
ЗАКЛЮЧЕНИЕ 9
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10

ВВЕДЕНИЕ

С ростом популярности генеративных моделей возникает необходимость объективной оценки их качества. Классические метрики, такие как BLEU и ROUGE, часто не отражают смысловую точность и полноту ответа. В связи с этим активно исследуются новые подходы, включая нейросетевые методы оценки (LLM-as-Judge) и семантические метрики (BERTScore).

В ходе данной практики основное внимание было уделено оценке качества работы языковых моделей в различных конфигурациях, включая стандартную генеративную модель (LLM) и Retrieval-Augmented Generation (RAG) на датасете QASPER. Работа сосредоточена на анализе метрик, их применимости в разных сценариях и сравнении классических подходов с более современными методами, такими как LLM-as-Judge.

1 МЕТОДЫ ОЦЕНКИ РАБОТЫ СИСТЕМЫ

1.1 Семантические метрики: BERTScore

BERTScore [Здесь ссылка на bert-score] использует эмбеддинги слов, полученные с помощью трансформеров, и вычисляет косинусное сходство между токенами эталонного и предсказанного текстов. Это позволяет учитывать перефразирование и синонимию, что делает метрику более устойчивой к вариативности ответов.

1.2 LLM-as-Judge

Один из современных методов оценки генерации – использование другой большой языковой модели в качестве судьи (**LLM-as-Judge**) [ссылка на llm-as-judge статью].

1.2.1 Faithfulness

Метрика **достоверности (Faithfulness)** измеряет, насколько фактически точен ответ по сравнению с извлечённым контекстом. Она принимает значения от 0 до 1, где более высокие значения указывают на лучшую согласованность. Ответ считается достоверным, если все его утверждения могут быть подтверждены извлечённым контекстом.

Алгоритм расчета метрики:

1. Выделение утверждений из ответа системы.
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)** измеряет, насколько ответ соответствует пользовательскому запросу. Более высокие значения указывают на лучшее соответствие входным данным пользователя, тогда как более низкие оценки присваиваются, если ответ неполный или содержит лишнюю информацию.

Алгоритм расчета метрики:

1. На основе ответа системы генерируется набор искусственных вопросов (обычно 3 вопроса). Эти вопросы должны отражать содержание ответа.
2. Вычисляется **косинусное сходство** между векторными представлениями пользовательского запроса и каждого сгенерированного вопроса.

3 Среднее значение косинусных сходств используется в качестве показателя **релевантности ответа**.

1.2.3 Answer Correctness

Метрика корректности ответа (Answer Correctness) оценивает соответствие ответа эталонному. Значение метрики варьируется от 0 до 1, где более высокий балл указывает на более высокое соответствие.

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 КОНФИГУРАЦИИ

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

Во всех конфигурациях в качестве большой языковой модели была выбрана LLAMA-3.3-70B-INSTRUCT. В качестве эмбеддинг модели используется USER-bge-m3. Рассмотрим особенности каждой конфигурации.

2.1 RAG

Стандартная конфигурация RAG состоит из двух основных этапов:

1. Индексация. На данном этапе документы преобразуются в чанки с помощью алгоритма семантического деления [здесь должна быть ссылка на грега камрадта]. Далее чанки индексируются с помощью алгоритма bm25 [здесь на okapi].
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 bm25 [тоже на окапи], далее среди отобранных чанков выбираются лучшие с помощью биэнкодер-ранжирования [здесь бы еще ссылки на реранкер, биэнкодер там]. Затем чанки подаются на вход генеративной модели для формирования финального ответа пользователю.

2.2 Agentic RAG

Особенность данной конфигурации состоит в том, что LLM сама решает, нужно ли ей для формирования ответа использовать поиск по документам. Технически реализация использует для индексации те же модули, что и стандартная реализация RAG.

2.3 LLM

Среди прочего, я проверил LLM без дополнительного поиска по данным. Для генерации ответа будет использоваться только вопрос из датасета без контекста.

3 АНАЛИЗ РЕЗУЛЬТАТОВ

Для проведения экспериментов я случайным образом отобрал из валидационной части датасета QASPER [здесь ссылочка на датасет] 100 вопросов. Перед этим построил индекс на всех научных статьях из валидационной части датасета.

Также стоит отметить, что большие языковые модели склонны к более лояльной оценке своих текстов [здесь ссылка на ребят, которые рассказали про самобайес], поэтому для метрик LLM-as-Judge я использовал другую модель: DeepSeek-R1-Distilled-LLAMA-70B.

3.1 Сравнение метрик на разных конфигурациях

	RAG	Agentic-RAG	LLM
Faithfulness			
Response Relevancy			
Answer Correctness			
BERTScore			
Время работы			

3.2 Выводы по метрикам

4 РАЗРАБОТКА ПРИЛОЖЕНИЯ

Так как имеющийся у меня код не соответстовал требованиям модульности и масштабируемости, я решил глубоко переработать кодовую базу. Для интерфейса серверной части я использовал OpenAPI-спецификацию для OpenAI-compatible [здесь ссылочка на спецификацию OpenAI-API] серверов. Таким образом у меня получился сервер, совместимый с большинством API для больших языковых моделей (См. приложение А).

4.1 Концепт интерфейса

Для удобной работы с научными статьями я переработал интерфейс приложения, который теперь вмещает в себя на левой части экрана PDF-viewer для отображения загруженного документа, а на правой – чат для взаимодействия с системой. Дополнительные элементы – кнопки загрузки документа, поделиться перепиской и т.д. Для интерфейса чата была использована открытая реализация интерфейса для взаимодействия с LLM: “Chat-UI” [здесь ссылка на электронный ресурс], разработанный HuggingFace [здесь тоже нужно пояснить, что за ребята]. Передо мной стоит задача переработать этот интерфейс под свою систему.

ЗАКЛЮЧЕНИЕ

В ходе практики был проведён анализ методов оценки качества генерации текстов различными конфигурациями языковых моделей. Основное внимание уделялось сравнительному анализу метрик **BERTScore**, **LLM-as-Judge** на датасете **QASPER**. Полученные результаты позволили мне выявить потенциальные улучшения системы, а также показали преимущество Retrieval-Augmented-Generation-подходов перед простыми LLM для генерации ответов по специфичным данным.

Дальнейшие шаги:

- Доработка интерфейса
- Тестирование серверной части

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ А

Как достаются утверждения для метрик RAGAS
Бенчмарки
Картинка интерфейса?
Ссылка на гитхаб, QR
Использованные источники
Формула BERTScore
Рассказать подробнее про AgenticRAG
Рассказать про FRAMES