

Задача от автоматизации коммуникаций

Черкасов Борис Юрьевич

Руководитель практики: ведущая EdTech программ DS и DA от Wildberries,
эксперт ВШЭ и МФТИ - Бурова Маргарита

Практика от Wildberries & Russ

2025

Цель и мотивация проекта

- Автоматизация ответов на повторяющиеся вопросы повышает эффективность клиентской поддержки и снижает нагрузку на операторов;
- Цель проекта — создать систему вопросно-ответную систему, которая генерирует ответ на обращения в техподдержку для менеджеров ПВЗ, способную:
 - понимать пользовательские вопросы;
 - находить релевантную информацию в базе знаний;
 - формировать осмысленные и точные ответы.

Exploratory Data Analysis (EDA)

Анализ текстов и структуры данных

■ Предобработка:

- 89 пропусков в `part_id` — заполнены медианой;
- удалено: 10 дубликатов вопросов, 8 дубликатов Q+A.

■ Статистика по длине текстов:

- вопросы: ~50 токенов, асимметрия вправо;
- ответы: ~130 токенов, выбросы до 1500;
- корреляция длины вопроса и ответа: -0.06 .

■ Тесты:

- T-test / Z-test: значимые различия ($p\text{-value} \approx 0$);
- F-test: дисперсии равны.

■ Лексика:

- стоп-слова: 28% (вопросы), 30% (ответы);
- вопросы — более уникальные, особенно короткие.

■ Структура:

- 11 уникальных заголовков;
- неравномерное распределение по QA-парам.

Подготовка базы знаний и поиск (Indexing & Retrieval)

■ Формирование чанков (chunking):

- разделение документов на логические части по заголовкам и абзацам;
- использованы длины в пределах 100–150 слов на чанк для сохранения связности.

■ Препроцессинг:

- очистка от спецсимволов, перевод в нижний регистр;
- удаление стоп-слов и нормализация токенов при необходимости.

■ Эмбединги:

- протестированы sentence-transformers для создания семантических векторов;
- использовалась модель `intfloat/multilingual-e5-large` (основная).

■ Индексирование и поиск:

- построен индекс по BM25 и эмбедингам (через FAISS);
- **гибридная модель**: объединение скорингов BM25 и косинусного сходства эмбедингов;
- для каждого запроса выбирались **top-k** релевантных чанков (обычно $k = 3$).

Генерация ответа (Generation)

- Используемая модель: sberbank-ai/rugpt3large_based_on_gpt2;
- Подобранные параметры генерации:
 - max_new_tokens=100, temperature=0.7, top_p=0.9;
 - применена генерация с контролем длины и стохастичности.
- Метрики качества:
 - **BERTScore (F1)** — основная метрика: измеряет семантическую близость;
 - преимущество: нечувствительна к перестановкам, перефразировкам;
 - альтернативы (BLEU, ROUGE) показали низкую корреляцию с смыслом:
 - BLEU: 0.0417, ROUGE: 0.1342;
 - BERTScore: до **0.6673** (после дообучения).
- Пример prompt + ответ:

вопрос: «Какие условия перехода на профстандарт?»

ответ: «Переход осуществляется при соблюдении условий, установленных ТК РФ и приказом...»

Эффективность поисковых моделей (Retrieval)

Оценка точности извлечения chunks

Метрики и методы:

Метод	Accuracy
Fuzzy Matching	91.79%
BM25	85.43%
BM25 + Embedding + (LOO с подбором)	94.70%

Комментарии:

- **Baseline:** Fuzzy Matching (строковое совпадение);
- **BM25:** TF-IDF-подобная модель;
- **HybridRetrieval:** reranking по Sentence-BERT embedding'ам.

Почему Accuracy?

- цель — найти *один наиболее релевантный chunk*;
- Precision/Recall неприменимы напрямую в этой задаче $\implies F_1 \ominus$.

Результативность генераторов и обоснование BERTScore

Оценка качества генераторов (по BERTScore F1):

- intfloat/multilingual-e5-large — **0.5983**;
- sberbank-ai/rugpt3small_based_on_gpt2 — **0.6029**;
- ai-forever/FRED-T5-large — **0.6154**;
- sberbank-ai/rugpt3large_based_on_gpt2 + MiniLM (SIM=0.5112) — **BERTScore: 0.6200**;
- **Fine-tuned версия:** SIM = 0.5100, BLEU = 0.0417, ROUGE = 0.1342, **BERTScore = 0.6673**.

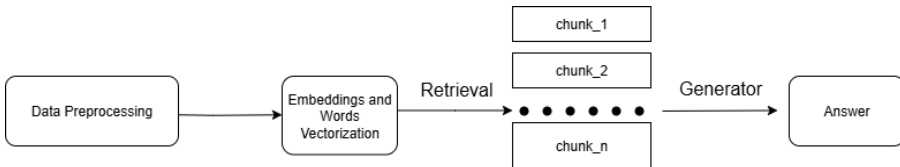
Почему выбрана метрика BERTScore (F1):

- использует контекстные эмбединги — оценивает **смысл**, а не совпадения;
- устойчив к переформулировкам и синонимам;
- F1-вариант объединяет точность и полноту оценки.

Архитектура решения (Final version)

■ Архитектура системы построена по принципу **Retrieval-Augmented Generation (RAG)**:

- 1 в качестве эмбеддера-трансформера был взят MiniLM-L12-v2;
- 2 в качестве поисковика был реализован класс HybridRetrievalModel, который сочетает в себе BM25 + Bi encoder + Cross encoder:
 - BM25 - оценка текстовых совпадений с учётом важности слов (грубая фильтрация);
 - Bi encoder - преобразование слова в векторы;
 - Cross encoder - отдельная модель предсказания релевантности.
- 3 в качестве генератора ответа используется модель RuGPT-3 Large.



Пример работы системы

Задайте вопрос

Здрасьте

Clear Submit

Ответ модели

Здрасьте

Я Помощник. Задай, пожалуйста, вопрос

Вопрос: Здрасьте

Ответ: Здрасьте

Я Помощник. Задай, пожалуйста, вопрос

Проблемы и ограничения

(и направления улучшения)

■ Проблемы и ограничения:

- генерация *галлюцинаций* — вымышленных или неточных фактов;
- ограничение по длине контекста (*truncation*) при подаче retrieved чанков;
- чувствительность качества ответа к качеству retrieved информации;
- слабая устойчивость к обобщённым и кросс-документным вопросам.

■ Что можно улучшить:

- использование более мощных эмбеддингов: **E5, BGE, Instructor, GTE**;
- использование более мощных retriever'ов:
 - **Dense retrievers**: DPR, Faiss, ColBERT;
 - **Гибридные подходы**: BM25 + BERT.
- замена генератора на продвинутые модели: **GPT-3.5/4, FRED-T5, mT5, LLaMA**;
- дополнительно:
 - **Fallback**-логика (например, возврат top-k без генерации);
 - повышение интерпретируемости и удобства интерфейса;
 - **аугментация текстов** для увеличения обучающей выборки;
 - генеративные подходы для создания дополнительных примеров.

Выводы и значение проекта




■ Что удалось достичь:

- реализована рабочая RAG-система: поиск + генерация на естественном языке;
- проведено сравнение генеративных моделей и оценка качества с использованием **BERTScore**, **semantic similarity**, **BLEU**, **ROUGE**;
- разработан интерфейс для взаимодействия с системой (**Gradio**), продемонстрированы пример работы.

■ Как это соотносится с задачами компании и ИИ:

- подобные системы могут быть внедрены для:
 - автоматизации поддержки пользователей;
 - сокращения времени на поиск информации в документации;
 - внедрения интеллектуальных помощников в продуктах компании.
- вклад в развитие **интерпретируемых** и **контролируемых** ИИ-систем;
- проект демонстрирует применение современных NLP-моделей в реальной прикладной задаче.

Список литературы I

-  Yandex.Cloud (2025).
RAG: учим ИИ работать с новыми данными.
<https://yandex.cloud/ru-kz/blog/posts/2025/05/retrieval-augmented-generation-basics>
-  X5Tech на Хабре (2024).
Интеграция LLM в корпоративные чат-боты: RAG-подход.
<https://habr.com/ru/companies/X5Tech/articles/834832>
-  Хабр (2024).
RAG: основы и продвинутые техники.
<https://habr.com/ru/articles/871226>

Список литературы II



Нейро Яндекс (2024).

Преимущества использования RAG.

https://ya.ru/neurum/c/tehnologii/q/v_chem_preimuschestva_ispolzovaniya_rag_dlya_4e0c171a



Яндекс на Хабре (2023).

Как мы отучаем ИИ галлюцинировать.

<https://habr.com/ru/companies/yandex/articles/791576>



BrainTools (2024).

Полный гайд по архитектуре RAG.

<https://www.braintools.ru/article/10740>

Спасибо за внимание!