

Извлечение объектов и признаков из текстового описания сцены на русском языке

Магистерская выпускная работа

Шильцов Дмитрий Александрович

Научный руководитель: к.ф-м.н., Кантонистова Е.О.

ФКН НИУ ВШЭ, 2025

Актуальность и цель исследования



Существующие подходы



Основанные на правилах (Rule-based / Pattern Matching)

Разбор с помощью библиотек PyMorphy, SpaCy



На основе BIO разметки и sequence labeling

требуют большого датасета. плохо работают с пересечениями



Генеративные модели (Seq2Seq, T5, GPT)

генерация текст→структура или текст→псевдокод с последующим парсингом в структуру

требуют большого датасета и много ресурсов на обучение модели



Scene Graph Parsing (SGP) и Relation Extraction

BERT + GraphNet, T5 + Relation Head, SpanBERT

требуют сложной архитектуры и многослойной разметки



Multi-task или Modular Models

пайплайны основанные на последовательном выделении объекты→признаки→связи

Хронология

1

Генерация датасета с помощью LLM

Корпус из 5,000 текстовых описаний сцен на русском языке с разметкой json для обучения моделей и 500 описаний для валидации.

Промежуточный датасет (текст + триплеты) для обучения выделения пространственных связей

2

Определение метрик качества для промежуточного (выделение только объектов и признаков) и итогового (построение графа сцены) этапов. Метрики на основе F1_score и модифицированный GED

3

Построение и валидация Rule-Based модели

4

Построение модели на основе T5ru+LoRA (одномоментная генерация упрощенной сцены без пространственных связей)

неудачная попытка - мало данных + модель от Сбербанка (sberbank-ai/ruT5-base) недостаточно хорошо умеет делать структурную генерацию а Flan-T5 не знает русский язык

5

Построение двухэтапной модели на основе T5ru+LoRA (генерация псевдокода с последующим парсингом + модель умеющая выделять связь или ее отсутствие по паре объектов из списка)

Генерация датасетов

Было сгенерировано 3 датасета - один упрощенный (объекты + признаки + текстовое описание) - под исходную задачу и расширенный с триплетными пространственными связями. Кроме того подготовлен вспомогательный датасет для обучения триплетов и отдельный датасет для валидации (в нем использованы локации, отсутствующие в учебных датасетах)

Расширенный

```
{
  "scene": {
    "location": "склад",
    "objects": [
      {
        "коробка": {}
      },
      {
        "паллета": ["деревянная"]
      },
      {
        "тележка": ["металлическая", "квадратная", "тяжелая"]
      },
      {
        "стеллаж": ["высокий"]
      }
    ],
    "relations": [
      [
        "коробка", "на", "паллета",
        "тележка", "рядом с", "паллета",
        "тележка", "у", "стеллаж"
      ]
    ],
    "description": "Коробка лежит на деревянной паллете рядом с квадратной тяжелой металлической тележкой, которая стоит у высокого стеллажа."
  }
}
```

Упрощенный

```
{
  "scene": {
    "location": "склад",
    "objects": [
      {
        "коробка": {}
      },
      {
        "паллета": ["деревянная"]
      },
      {
        "тележка": ["металлическая", "квадратная", "тяжелая"]
      },
      {
        "стеллаж": ["высокий"]
      }
    ],
    "description": "Коробка лежит на деревянной паллете рядом с квадратной тяжелой металлической тележкой, которая стоит у высокого стеллажа."
  }
}
```

Вспомогательный датасет для обучения на триплетах

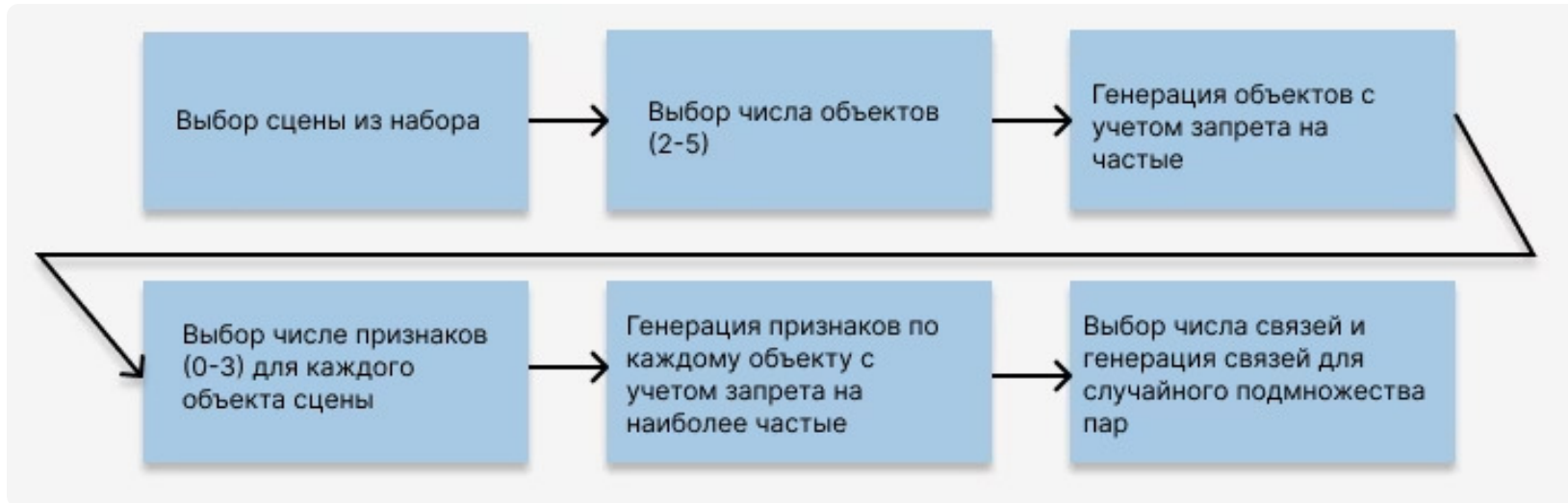
в датасете примерно 1/3 положительных триплетов, 1/3 реверсивных отрицательных, 1/3 негативных ("нет связи")

```
{
  "description": "Тяжелая гантель лежит на деревянной тяжелой скамье рядом с тяжелой металлической штангой."
  "relation": ["гантель", "на", "скамья"],
  "target": "на"
}

{
  "description": "Тяжелая гантель лежит на деревянной тяжелой скамье рядом с тяжелой металлической штангой."
  "relation": ["гантель", "на", "штанга"],
  "target": "нет связи"
}
```

Генерация датасетов

Датасеты генерировались с помощью API к LLM chat.openai.com и chat.deepseek.com.



Для рабочего варианта возможно имело бы смысл добавить пост-валидацию на предмет соответствия законам физики (чайник стоял внутри пациента который лежал на стоящем шприце) но в целом правильный выбор промпта позволяет избегать таких коллизий + мы выделяем сущности исходя из описания а не верифицируем сцену

Метрики качества

Стандартные метрики генерации в стиле BLEU и ROUGE по очевидным причинам нам не подходят (они учитывают порядок, а у нас объекты и признаки неупорядоченные множества - их нужно сравнивать как множества). Метрики на основе IoU возможно были бы неплохи, но я решил использовать метрики на основе F1_score.

Для контроля качества определения выделения объектов и признаков использовались метрики на базе F1_score на нормализованных (лемматизированных) описаниях сцен (только по объектам, только по признакам для каждого объекта усредненно + различные комбинации в стиле F1 на парах объект-признак для всех валидных пар)

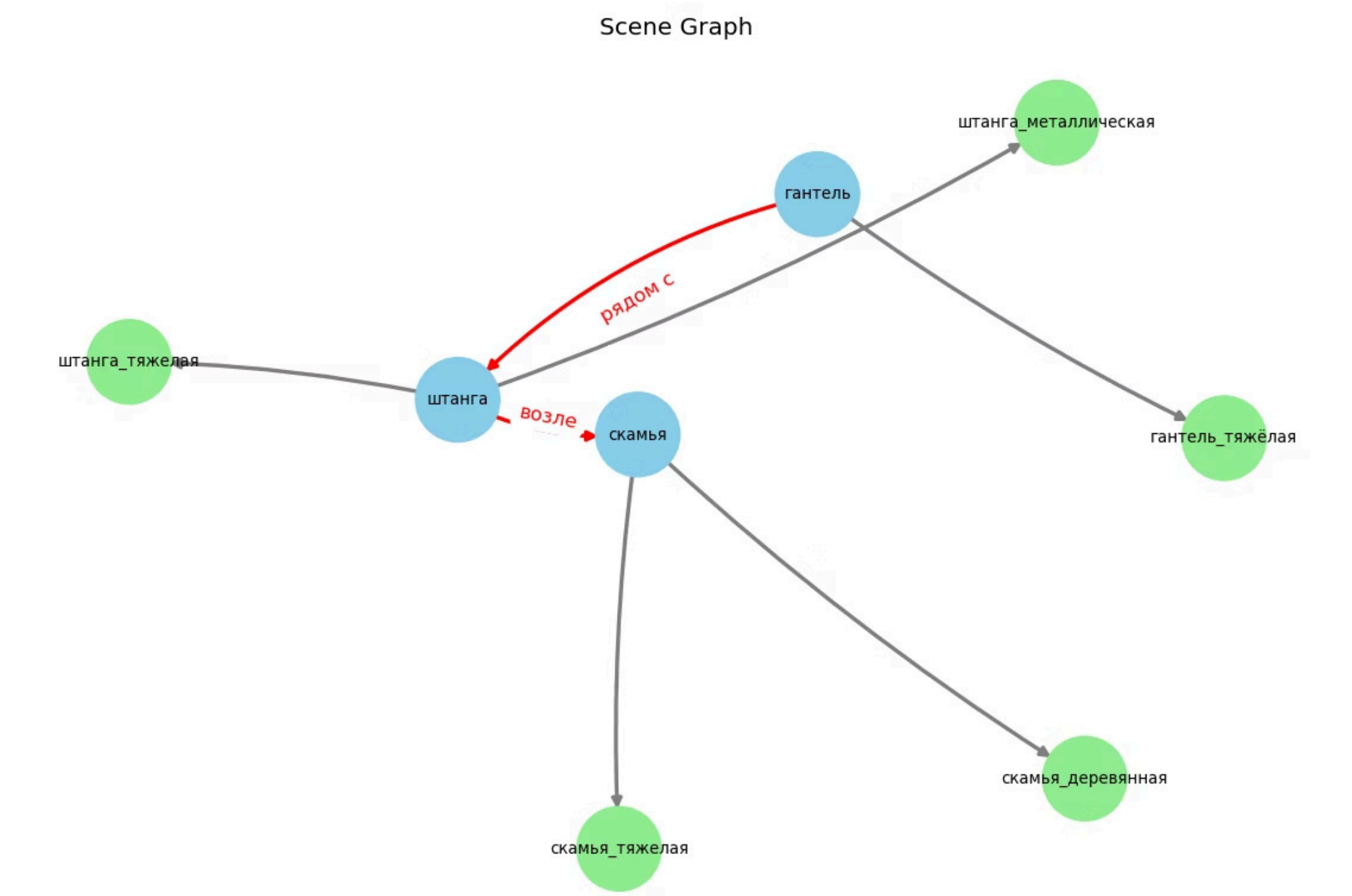
Для определения качества генерации итогового графа я использовал модифицированную версию GED (graph edition distance) - считал не только минимальное число изменений до восстановления изоморфизма структур но и закладывал стоимость переименования вершин (объекты и признаки) и части ребер (именованные ребра - центры триплетов) до получения в точности идентичных сцен

Визуализация сценических графов

Реализованы функции построения и визуализации сценических графов по json-описанию

```
{
  "scene": {
    "location": "тренажерный зал",
    "objects": [
      {
        "гантель": ["тяжёлая"],
        "штанга": ["тяжелая", "металлическая"],
        "скамья": ["деревянная", "тяжелая"]
      },
      {
        "relations": [
          ["гантель", "рядом с", "штанга"],
          ["штанга", "возле", "скамья"]
        ],
        "description": "Тяжелая гантель лежит рядом с тяжелой металлической штангой, которая стоит возле тяжелой деревянной скамьи."
      }
    ]
  }
}
```

гантель лежит рядом с тяжелой металлической штангой, которая стоит возле тяжелой деревянной скамьи.



Rule-Based модель на основе SpaCy

```
pattern_verb = [
    {"RIGHT_ID": "verb", "RIGHT_ATTRS": {"POS": "VERB"}},
    {"LEFT_ID": "verb", "REL_OP": ">", "RIGHT_ID": "nsubj",
     {"LEFT_ID": "verb", "REL_OP": ">", "RIGHT_ID": "obl",
      {"LEFT_ID": "obl", "REL_OP": ">", "RIGHT_ID": "prep",
       }
    ]
```

Модель, полностью построенная на эвристиках и правилах русской грамматики. Не требует дополнительного обучения. Не справляется со сложными случаями (далеко стоящими друг от друга атрибутами и объектами)

В модели реализовано достаточно много эвристик, тем не менее часть грамматических конструкций осталась "за кадром"- в частности кореференс и эллипсис которые выделить с помощью ros-разметки крайне проблематично. В порядке эксперимента реализована модель с предобработкой текста для устранения кореференсов и эллипсисов с помощью LLM ("на столе стояли синяя и красная ваза" → "на столе стояли синяя ваза 1 и красная ваза 2") с последующей обработкой через эвристики, но метрики качества для этой модели не считал так как там в саму модель по сути интегрирован внешний API чего изначально хотелось бы избежать

Итоговые метрики качества на валидационном датасете для rule-based модели

100% | 250/250

Validation results on 250 samples:

```
f1 objects: 0.9505
```

```
f1 attributes macro: 0.6571
```

```
f1 attributes weighted: 0.8502
```

```
f1_global obj_attr pairs: 1.0
```

```
f1 combined simple: 0.8038
```

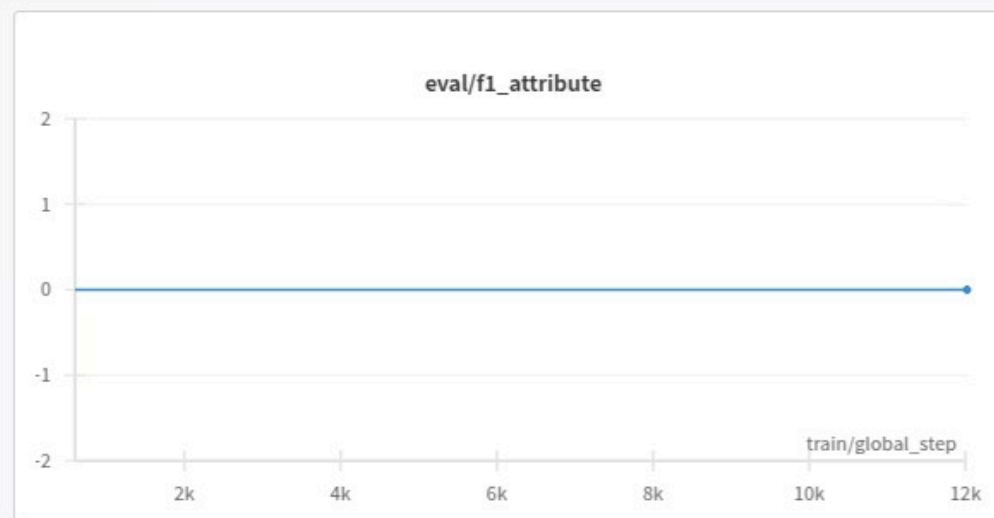
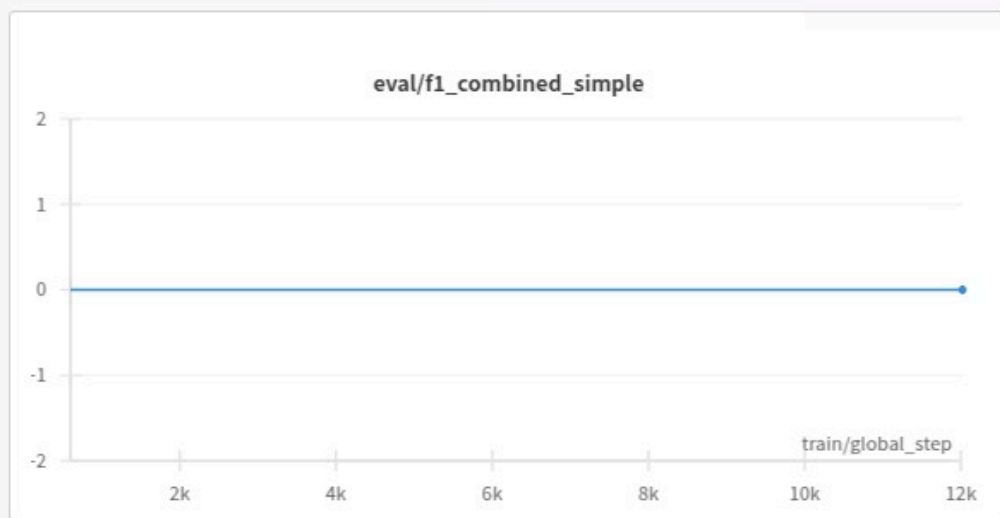
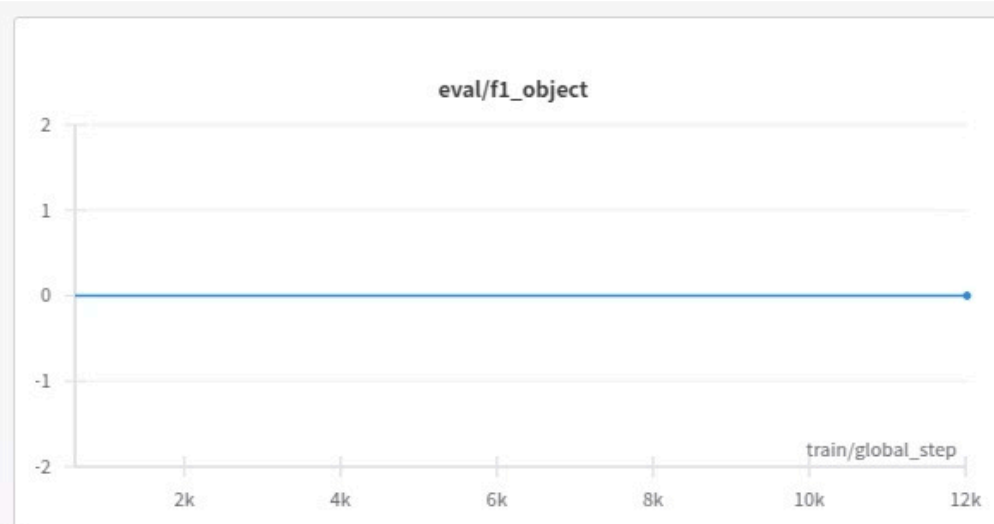
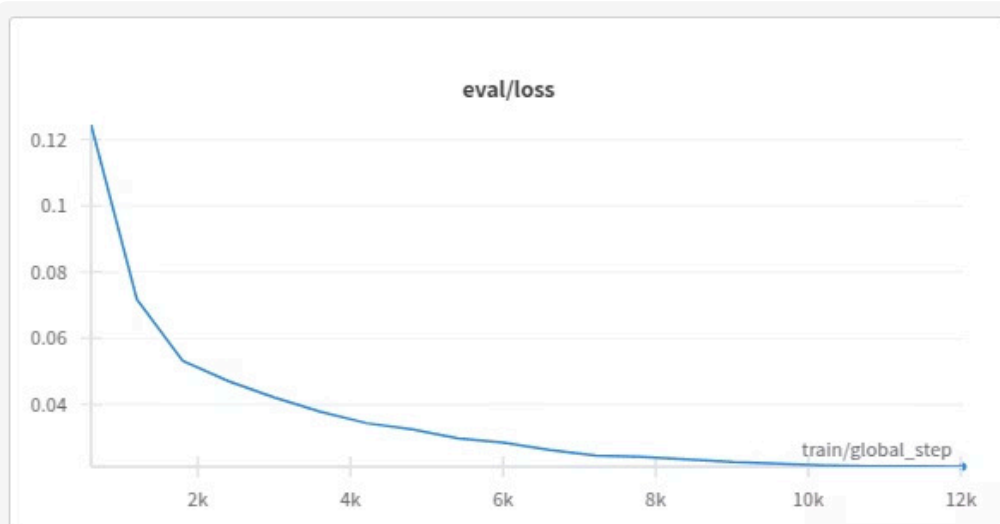
```
f1_combined_weighted: 0.9012
```

GED score: 0.5398

Прямая генерация структуры на основе T5ru+LoRA (text→json)

Сделана попытка обучить модель T5ru+LoRA+промт генерировать сразу валидный json с описанием сцен, попытка не удалась -в силу того что во-первых модель вероятнее всего не обучалась на структурированных последовательностях (в отличие от, скажем, Flan-T5) и учить ее выдавать валидируемые структуры задача не уровня обучения LoRA (требуется и саму T5 доучивать, а это долго и ресурсозатратно)

```
[{"столы": {"столение": {"деревгкое": "деревlichesкий", "жеучень": {"деревlichesкий": "женый", "жекий": "женочное", "деревкоеое": "деревверь": {"", {"объекты": {"столтарь": {"столный": "каменивный"}, "светна": {"светинная"}, "света": {"" светмп": {"", {"объекты": {"столоду": {"металовый"}, "металзак": {"металонепроницаемый"}, "металт": {"металкладной", "металный"}}
```



Прямая генерация структуры на основе T5ru+LoRA+промпт (text→псевдокод)

следующим этапом обучалась модель выделения объектов и признаков через псевдокод. Модель по промпту + описанию сцены генерирует псевдокод из которого с помощью регулярных выражений получается валидный json

Описание: Маленький красный стол стоит у окна.

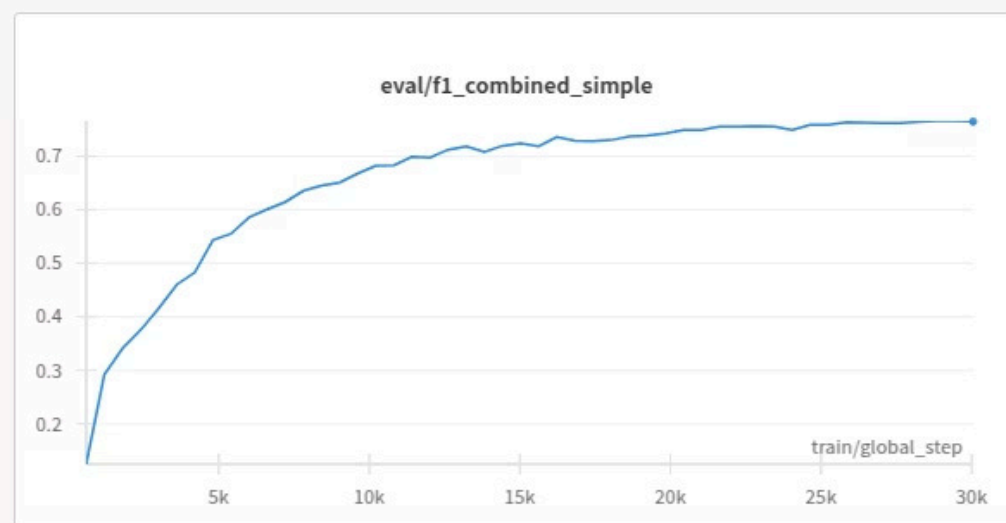
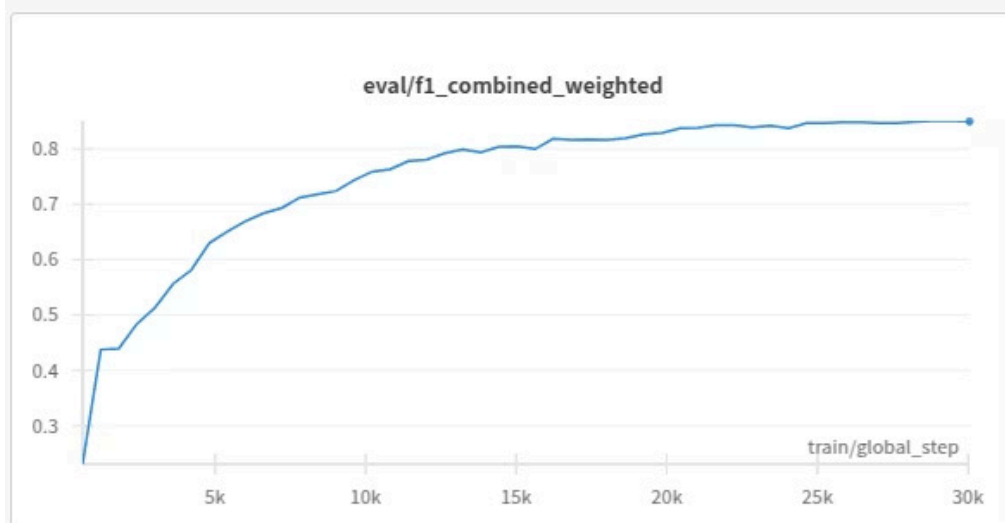
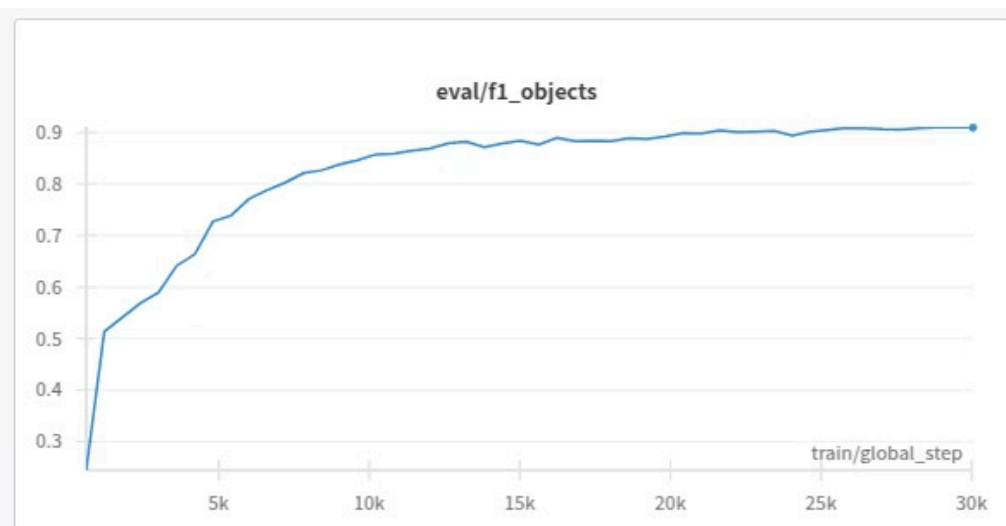
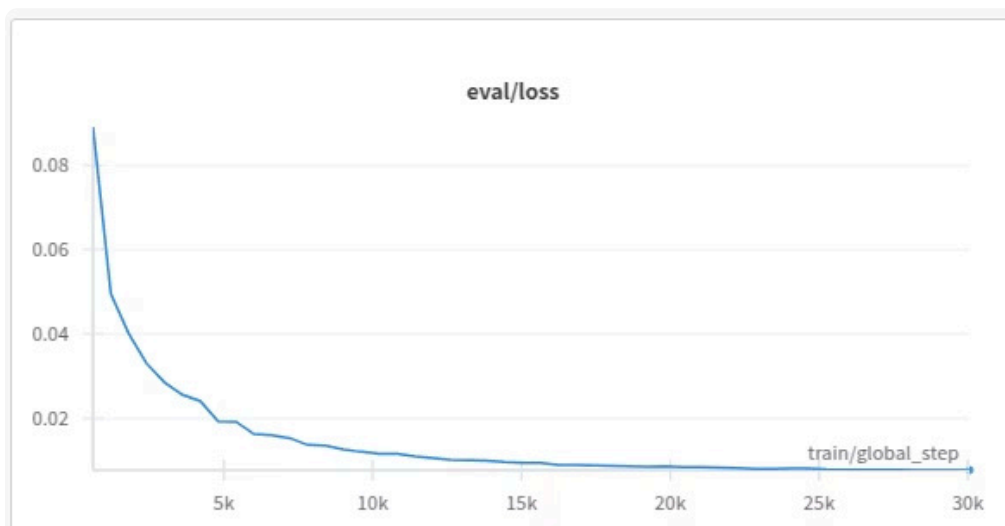
Ответ: стол (маленький красный) окно ()

В итоге получился очень неплохой результат для выделения объектов и признаков.

Всего параметров: 223.79M

Обучаемых параметров: 0.88M

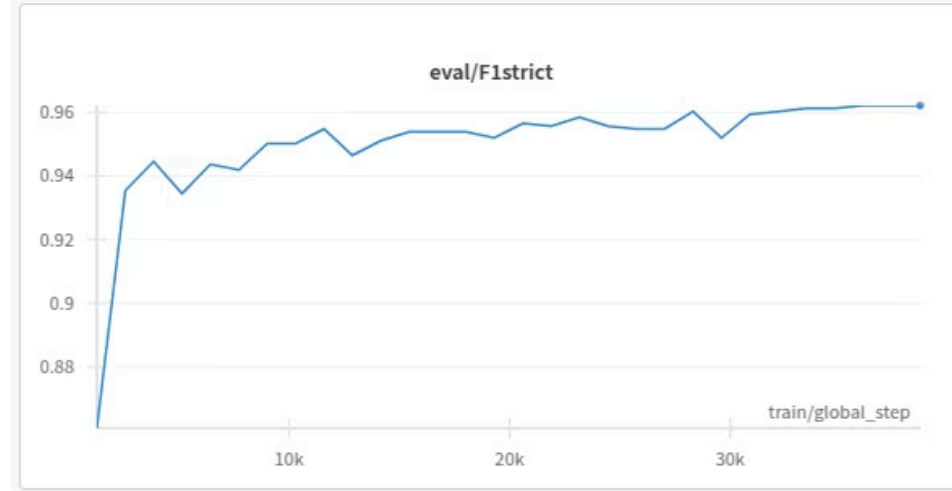
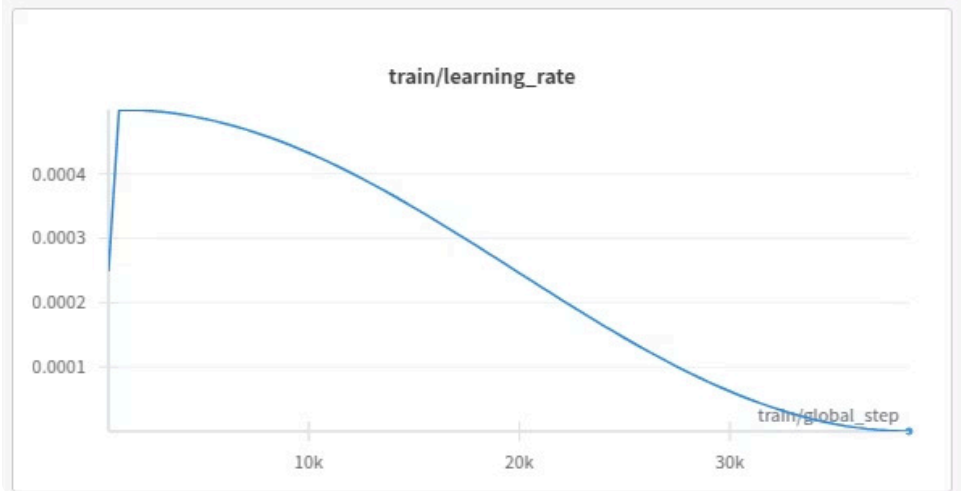
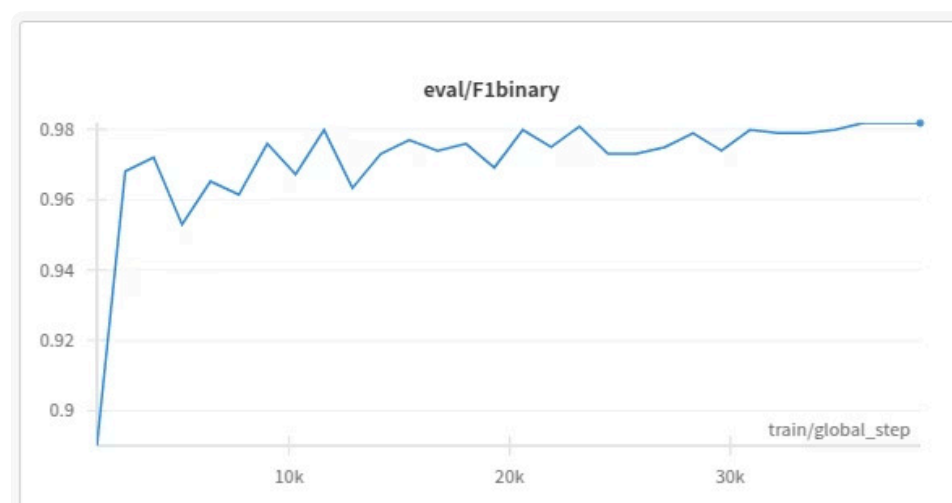
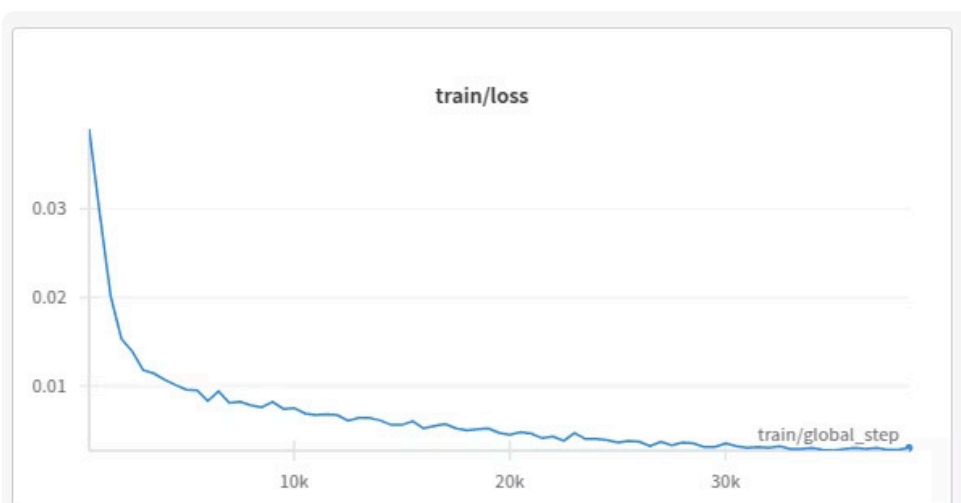
Доля обучаемых параметров: 0.40%



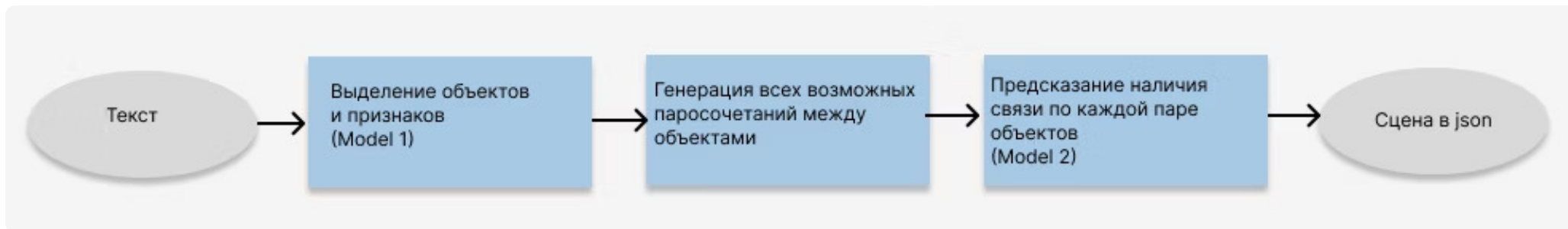
Выделение пространственных связей T5ru+LoRA+промт (text→text)

Также обучалась отдельная модель на выделение из текста пространственной связи между признаками. По сцене и паре объектов из этой сцены модель должна предсказать либо отсутствие связи ("нет связи") либо саму связь ("рядом с"). Модель обучалась на достаточно маленьком для такой задачи синтетическом датасете (~20 000 примеров) и небольшое количество эпох тем не менее прогресс в обучении хороший.

В качестве метрики качества на обучении использовалась сквозная F1 по всем парам, представленным в валидационной части датасета



Итоговый пайплайн



Итоговые результаты валидации на отложенном датасете. Модель, построенная на статистическом обучении уверенно побеждает

100% | 250/250

Validation results on 250 samples:

f1_objects: 0.9826
 f1_attributes_macro: 0.7367
 f1_attributes_weighted: 0.9296
 f1_global_obj_attr_pairs: 1.0
 f1_combined_simple: 0.8597
 f1_combined_weighted: 0.9593
 GED_score: 0.564

Горький урок Р.Саттона(The Bitter Lesson)

Главный прогресс в ИИ достигается не через ручное кодирование знаний или человеческие инсайты о предметной области, а через масштабирование обобщённых алгоритмов обучения, которые используют больше вычислений и данных.

План дальнейших работ над проектом

1. Сделать рефакторинг кода и репозитория
2. Внимательно пересмотреть метрики - есть ощущение что где-то допустил ошибки
3. Собрать датасет большего размера (~10000 записей и ~50 000 для пар) и более разнообразный и дообучить модели
4. Сделать FastAPI+Streamlit приложение которое делает разбор текста и рисует граф, завернуть это все в докер-контейнер и залить его в докерхаб
5. Написать текст самой работы
6. Переделать эту презентацию в шаблон ВШЭ