

Восстановление полных реплик в диалоге

с помощью генеративных языковых моделей
семейства ruGPT

Козиев Илья, 2022

Где и когда появляются неполные реплики?

- Бытовой диалог
- Читчат

Пример №1:

- Эй, тебя как зовут?
- Джульетта Мао ⇒ Меня зовут Джульетта Мао

Пример №2:

- Где живешь?
- В Шанхае ⇒ Я живу в Шанхае
- Давно? ⇒ Ты давно живешь в Шанхае?
- Два года уже ⇒ Я уже два года живу в Шанхае
- Как там погода? ⇒ Какая погода в Шанхае?

Где и когда появляются неполные реплики?

Эллипсисы:

- Как же тебя зовут, а?
- Меня – Стас, а тебя? ⇒ Меня зовут Стас. Как тебя зовут?

В редких случаях и главное слово в словосочетании может опускаться:

- Мама, купи мне собаку.
- А ты будешь за ней ухаживать?
- А ты мне здоровую купи. ⇒ купи мне здоровую собаку

Анафора:

- Ты собак любишь?
- Не люблю я их ⇒ я не люблю собак

Иногда требуется привлечение здравого смысла:

- Мне на голову упала коробка.
- А что в ней было? ⇒ что было в коробке|голове?

Где и когда появляются неполные реплики?

Гэппинг:

- Ты кошек любишь?
- Их – нет ⇒ я не люблю кошек

Сложный гэппинг:

- В 25 лет вы получаете пенсию?
- Не я - отец. ⇒ Я не получаю пенсию. Отец получает пенсию

Типичные паттерны: например, восстановление подлежащего (см. pro drop)

- Согласна?
- Да ⇒ я согласна

Между строк:

- Ты разве ещё не ел?
- Тебя ждал ⇒ я еще не ел. я ждал тебя.

Отрицания в диалоге:

- Я не прав?
- Нет. (Да.) ⇒ ты не прав

Где и когда появляются неполные реплики?

Раскрытие не сводится к копированию слов из контекста:

- Как прошли выходные?
- В Простоквашино ездила... ⇒ я на выходных ездила в Простоквашино

Все вышесказанное может быть в разных сочетаниях одновременно:

- Где твой кот?
- Жена к ветеринару повезла. ⇒ жена повезла моего кота к ветеринару
- Заболел? ⇒ твой кот заболел?

Сложные предложения:

- Я сварила суп, иди ешь.
- Из чего? ⇒ из чего ты сварила суп?

Иногда от реплики остается только наречие:

- Девушка, а Вы животных любите?
- Очень! ⇒ я очень люблю животных

Где и когда появляются неполные реплики?

В типовой ситуации отказ от использования коротких реплик возможен:

- 1) в стилистических целях – подчеркнуть свою мысль
- 2) если есть подозрения, что собеседник (чатбот, иностранец) не владеет полным набором навыков работы в таком коммуникационном регистре
- 3) чтобы устранить возможность неправильного толкования

- Ребенок не ест мясо. **Его** можно чем-то заменить?
- Конечно, собакой – она всегда ест мясо

Чем могут мешать неполные реплики?

Отсутствие части слов затрудняет или делает невозможной работу многих классических алгоритмов NLP:

- Регулярные выражения
- Классификаторы интенентов
- Детекторы оскорблений, токсичности
- Частеречная разметка
- Синтаксический анализ
- Semantic Role Labeling
- Выделение фактов

Постановка задачи, ограничения

- Работаем только с диалогом, чатчатом
- Короткие реплики (до ~20 слов)
- Контекст – последние 2 или 3 реплики, при необходимости применяем модель рекурсивно
- Не обрабатываем случаи катафоры, только левый контекст
- Не требуем от модели пометить антецедент анафоры, референт для кореференции, источник заполнения эллипсиса
- Удаляем вводные слова и фразы, междомения
- По возможности интерпретируем голофразы (“Ага, щас!”)
- Восстанавливаем местоименные подлежащие, даже если форма сказуемого содержит информацию о лице/числе
- Нормализуем порядок слов, вопросительные слова в начало

Близкие задачи в NLP

- Анафора
- Эллипсисы
- Корелеренция

Генеративная модель как решение

- RuGPT – семейство больших генеративных языковых моделей с архитектурой GPT
- Ключевая особенность – претрейн на большом массиве текстов
- Ряд моделей свободно доступен на <https://huggingface.co/sberbank-ai>
- Адаптация модели на задачу – файнтюн на небольшом корпусе

Генеративная модель как решение

Плюсы подхода:

- Простота инференса: никаких парсеров, тэггеров, словарей
- Простота расширения датасета: разметки как таковой нет
- Инструментарий для GPT: анализ, дистилляция и т.д.
- Можно использовать эту же модель для генерации сырья

Обучающий датасет

Ручная разметка, примерно 110,000 фрагментов такого формата:

Как вы догадались, что задержанный — вор?

По шапке.

На нем она горела? | шапка горела на задержанном?

Для файнтюна конвертируются в сэмплы:

<s>- Как вы догадались, что задержанный — вор?

- По шапке.

- На нем она горела? # шапка горела на задержанном?</s>

Синтетические данные

- Пока было мало ручных данных, использовались синтетические
- Генерация синтетики – dependency parser (UDPipe + СинТагРус) и морфологический словарь (ruword2tags)
- С помощью правил генерируются вопросы, короткие и полные ответы к утвердительным предложениям:

Т: Голодная кошка отчаянно преследует жирную мышку.

- 1) Кто ловит мышку? Голодная кошка ⇒ Голодная кошка ловит мышку
- 2) Кого преследует кошка? Жирную мышку ⇒ Кошка преследует жирную мышку
- 3) Какая кошка преследует мышку? Голодная ⇒ Голодная кошка преследует мышку

- Оценки качества синтетики: сильно хуже ручных данных
- Актуальная модель обучена только на ручных данных

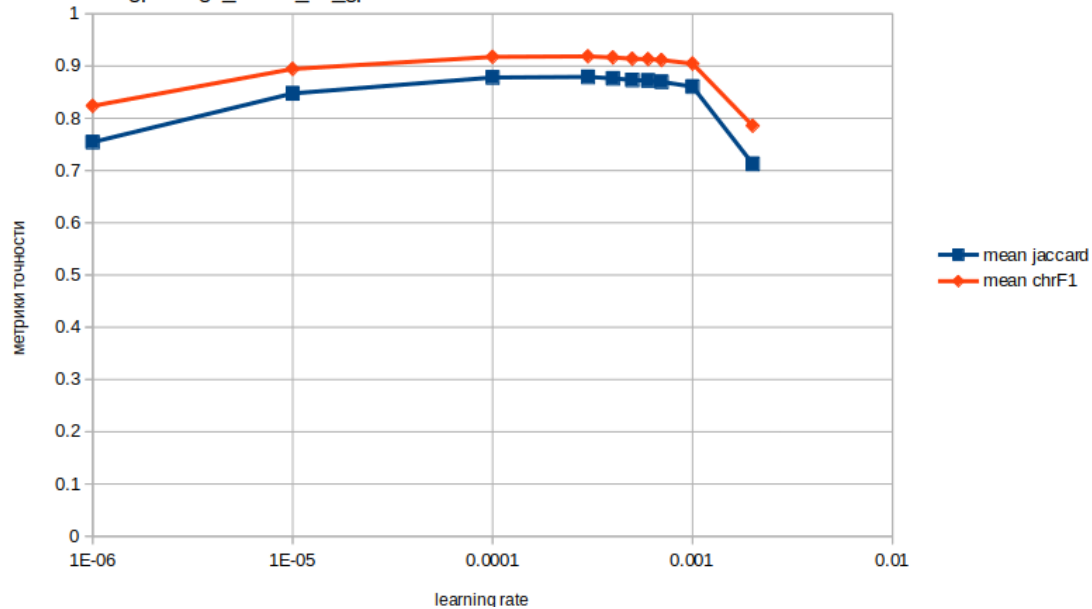
Оценка моделей

- Тренируются с помощью `ru-gpts/pretrain_transformers.py`
- 1 эпоха
- grid search для подбора learning rate
- 3-fold кросс-валидация
- Метрики: а) перплексия, б) посимвольная похожесть на 3-символьных шинглах
- Коэффициент Жаккара и chrF1 дают оценку, насколько точно модель выдает эталонный полный текст реплики

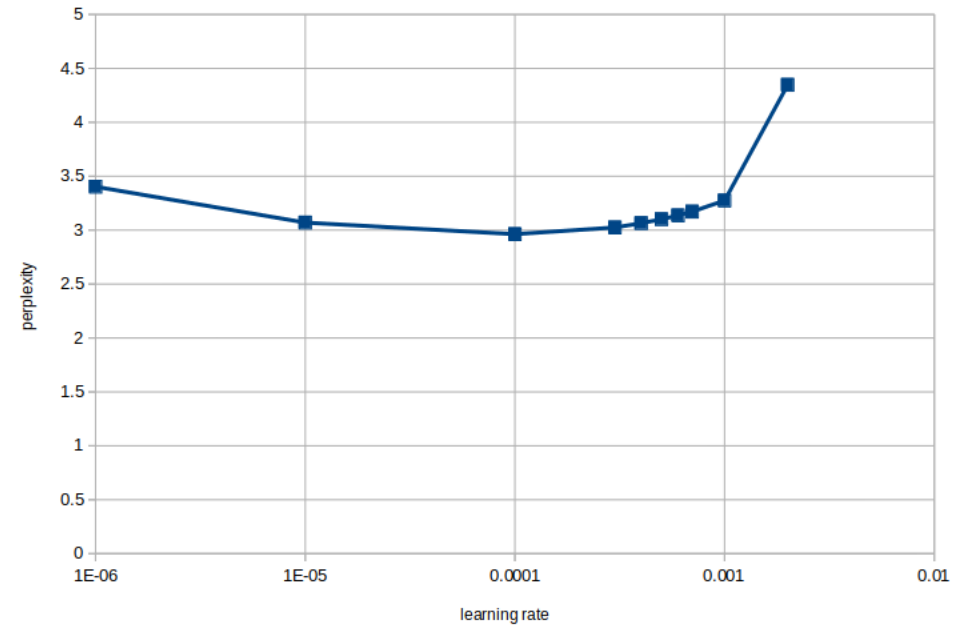
Оценка моделей

learning rate	mean jaccard	mean chrF1	mean perplexity
1E-06	0.754406798932305	0.823752277742384	3.4024
1E-05	0.847686061650237	0.894500112736263	3.06983333333333
0.0001	0.878134403010693	0.917535855322468	2.9625
0.0003	0.879228736635383	0.918451207487031	3.0245
0.0004	0.87641661536491	0.916438059239728	3.06436666666667
0.0005	0.873375897092778	0.914028200378741	3.10093333333333
0.0006	0.872263165892749	0.913221866840784	3.1365
0.0007	0.869767052945417	0.911522751947781	3.17186666666667
0.001	0.860598993850575	0.904675132850088	3.2742
0.002	0.712703992796691	0.785728068771592	4.34716666666667

Зависимость точности раскрытия полной фразы от скорости обучения при 3-fold cv rugpt3large_based_on_gpt2



Зависимость перплексии от скорости обучения при 3-fold cv rugpt3large_based_on_gpt2



Модель на huggingface

Модель доступна для свободного использования

На базе [sberbank-ai/rugpt3large_based_on_gpt2](#)

Карточка: https://huggingface.co/inkoziev/rugpt_interpreter

Пример использования с transformers:

```
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
```

```
device = "cuda" if torch.cuda.is_available() else "cpu"
model_name = "inkoziev/rugpt_interpreter"
tokenizer = AutoTokenizer.from_pretrained(model_name)
tokenizer.add_special_tokens({'bos_token': '<s>', 'eos_token': '</s>', 'pad_token': '<pad>'})
model = AutoModelForCausalLM.from_pretrained(model_name)
model.to(device)
model.eval()
```

```
# На вход модели подаем последние 2-3 реплики диалога. Каждая реплика на отдельной строке, начинается с символа "-"
# В конце добавляем символ "#"
input_text = """"<s>- Как тебя зовут?
- Джульетта Мао #""""
```

```
encoded_prompt = tokenizer.encode(input_text, add_special_tokens=False, return_tensors="pt").to(device)
```

```
output_sequences = model.generate(input_ids=encoded_prompt, max_length=100, num_return_sequences=1,
pad_token_id=tokenizer.pad_token_id)
```

```
text = tokenizer.decode(output_sequences[0].tolist(), clean_up_tokenization_spaces=True)[len(input_text)+1:]
text = text[: text.find('</s>')]
print(text)
```


ССЫЛКИ

- 1) Разрешение анафоры (Dialogue evaluation 2014)
<https://www.dialog-21.ru/evaluation/2014/anaphora/>
- 2) Automatic Gapping Resolution for Russian (Dialogue shared task)
<https://github.com/dialogue-evaluation/AGRR-2019>
- 3) Anaphora and Coreference Resolution for Russian (Dialogue shared task)
<http://www.dialog-21.ru/en/evaluation/2019/disambiguation/anaphora/>
- 4) SARG: A Novel Semi Autoregressive Generator for Multi-turn Incomplete Utterance Restoration <https://arxiv.org/pdf/2008.01474v3.pdf>
- 5) CHRF: character n-gram F-score for automatic MT evaluation
<https://aclanthology.org/W15-3049.pdf>