Алгоритмы сэмплирования текстовых данных для ускорения обучения языковых моделей при помощи обучения по плану

Мосин Владислав

Научный руководитель: Ямщиков Иван Павлович

Санкт-Петербургская школа физико-математических и компьютерных наук
НИУ ВШЭ СПБ

9 июня 2021

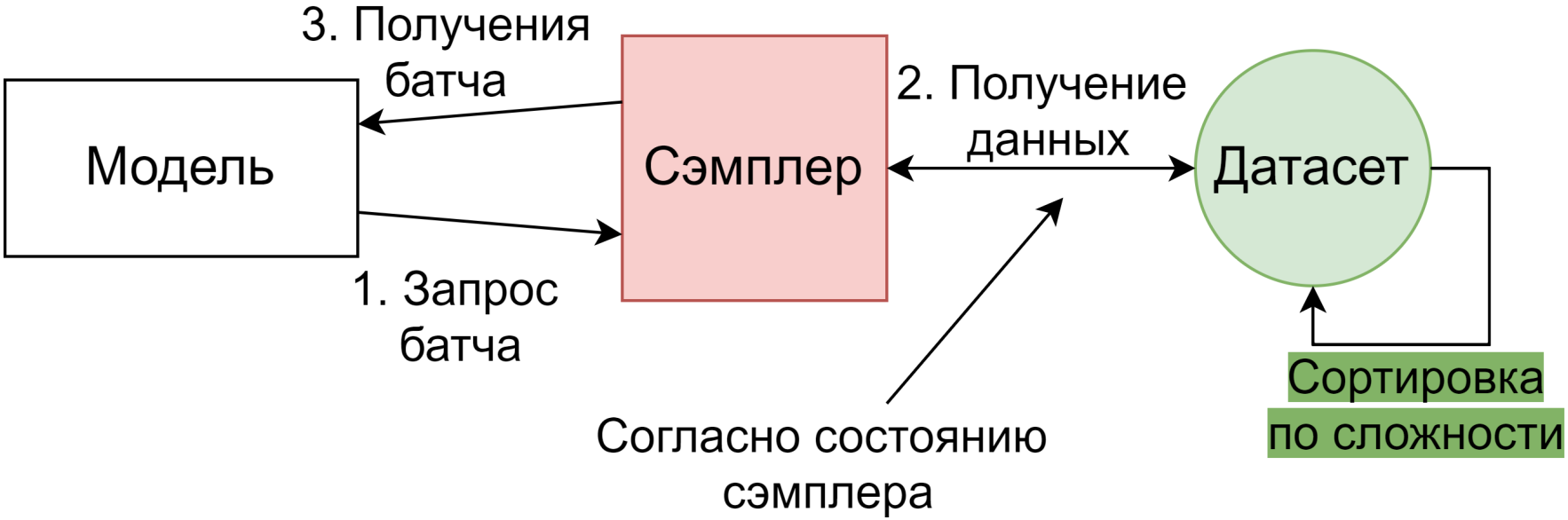
Мотивация

- Обработка естественного языка используется во многих областях
  - Переводчики
  - Голосовые помощники
  - Социальные сети
- Модели имеют от нескольких миллионов параметров
- Время тренировки может превышать тысячи GPU-часов
- Хочется научиться тренировать модели быстрее




Обучение по плану

- Сортировка датасета по сложности
- Выбор элементов для обучения в порядке от простых к сложным



- Обучение по плану показывает хорошие результаты в обучении с подкреплением
- В компьютерном зрении результаты противоречивы (Wu et al., 2020)

Обучение по плану в обработке языка. Поле исследований

- Задачи обработки естественного языка можно разделить на 4 группы
  1. Предобучение. Например, восстановление слова по контексту: Лондон - это <пропуск> Англии
  2. Классификация. Например, классификация твитов по эмоциональной окраске (негативная / нейтральная / позитивная)
  3. Машинный перевод.
  4. Понимание естественного языка. Например, генерация ответов на вопросы.
- Существуют работы исследующие обучение по плану на задачах машинного перевода Platanios et al., (2019), Kocmi et al. (2017), а также на задаче понимания естественного языка Xu et al., (2020)
- Не исследовано влияние обучения по плану на задачах классификации и предобучения
- Используется малое количество алгоритмов сэмплирования

Цель и задачи

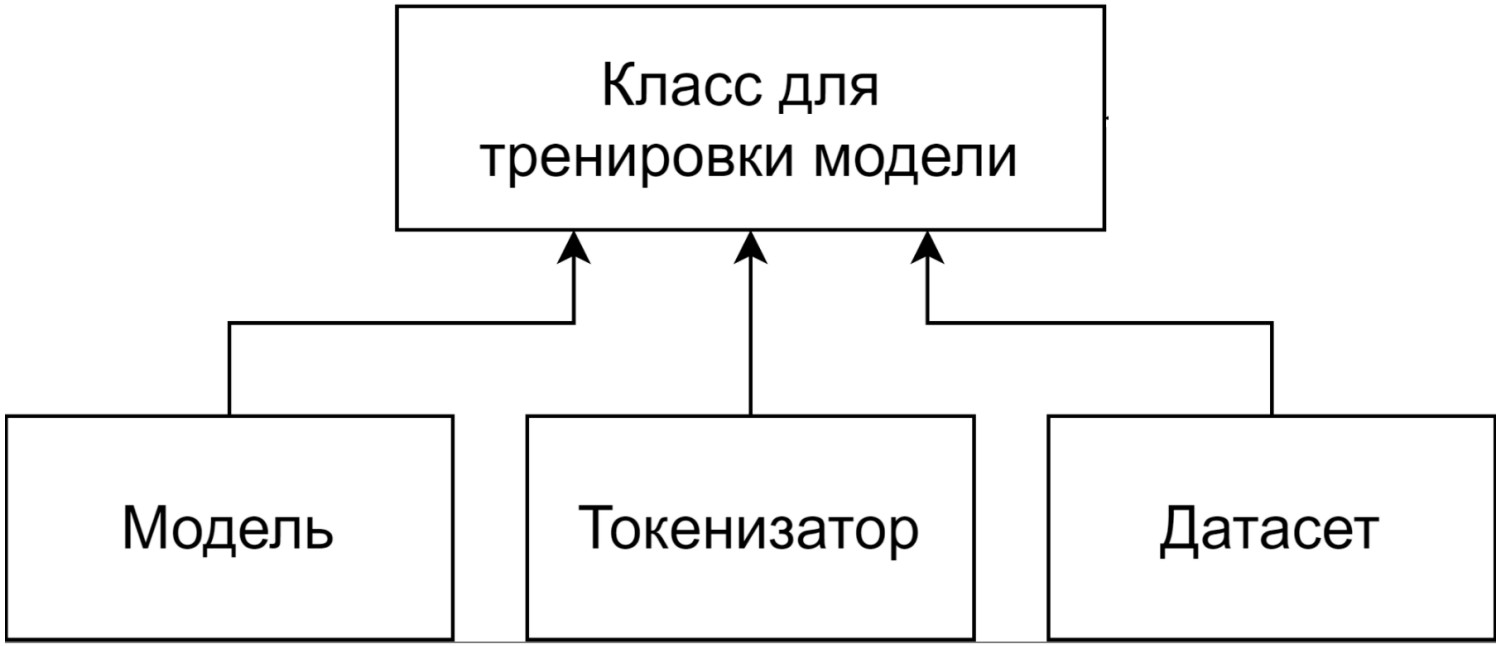
Цель: Исследовать влияние алгоритма сэмплирования на обучение языковых моделей в обучении по плану на задачах предобучения, классификации и машинного перевода

Задачи:

1. Реализовать систему, позволяющую использовать различные сэмплеры
2. Придумать и реализовать различные сэмплеры
3. Оценить эффективность предложенных алгоритмов сэмплирования по сравнению со стандартной моделью

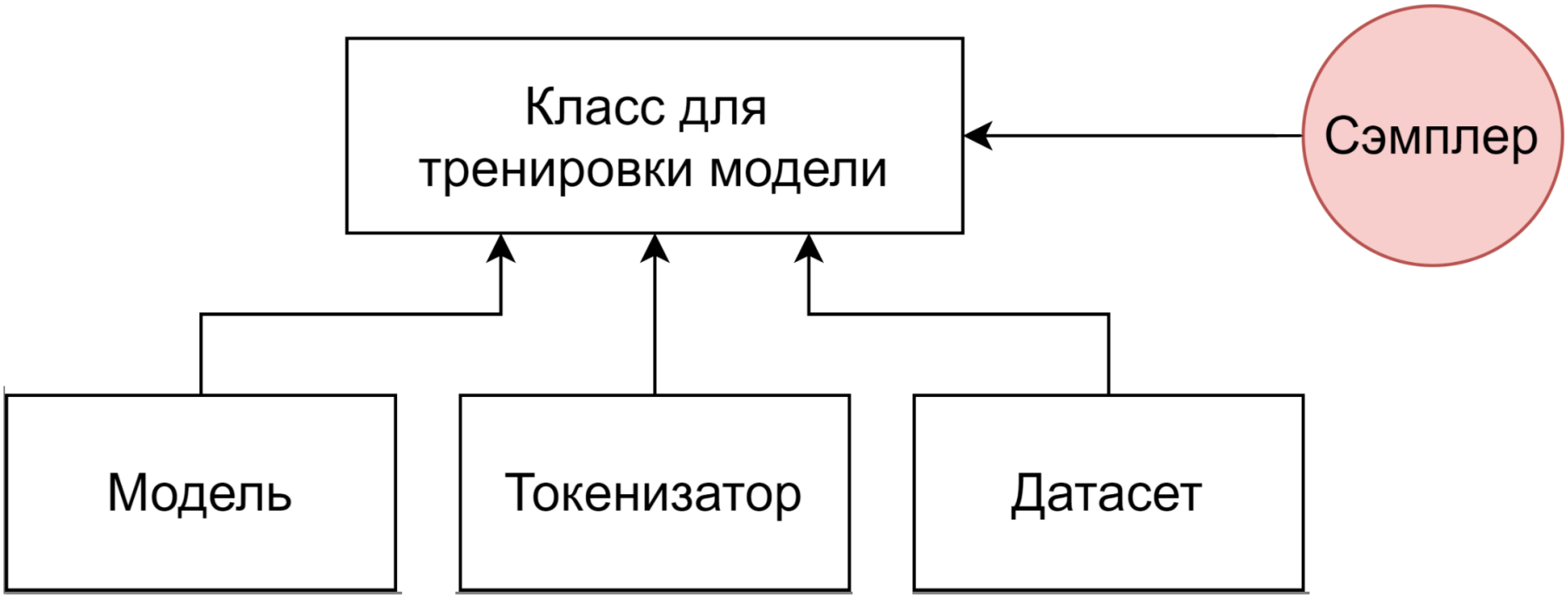
Контроль сэмплирования при обучении модели

- Самая популярная библиотека для обучения моделей обработки естественного языка: HuggingFace
- Не реализована возможность изменять стратегию сэмплирования



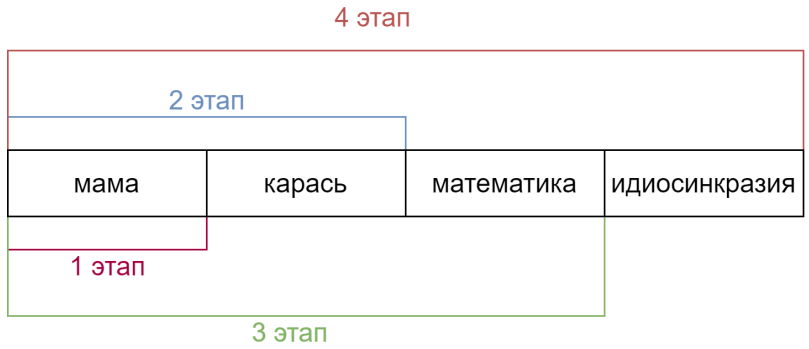
Реализация алгоритма сэмплирования в HuggingFace

- Извлекли всю логику сэмплирования в один объект и вынесли в отдельный объект "Сэмплер"
- Отнаследовались от класса для тренировки моделей и вынесли всё сэмплирование в один объект

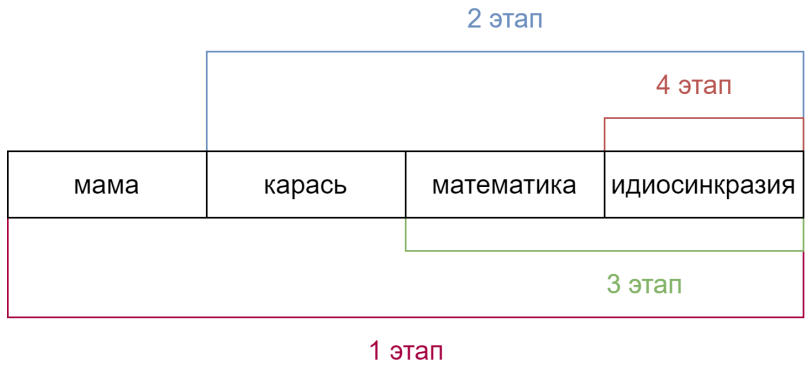


Сэмплеры

- **Префикс.** Platanious et al., 2019
- Сэмплирование с растущего префикса датасета



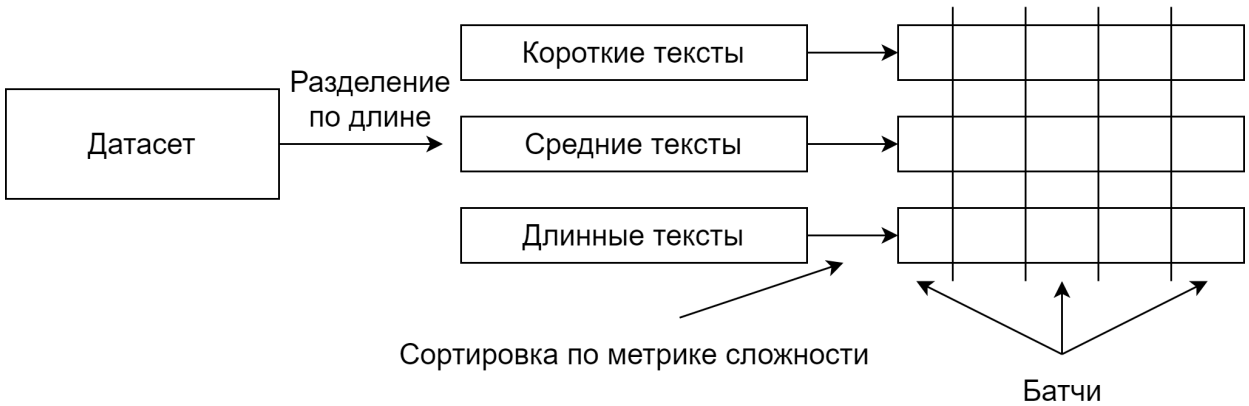
- **Суффикс.**
- Сэмплирование с уменьшающегося суффикса



- **Окно.**
- Сэплирование с участка датасета

Сэмплеры

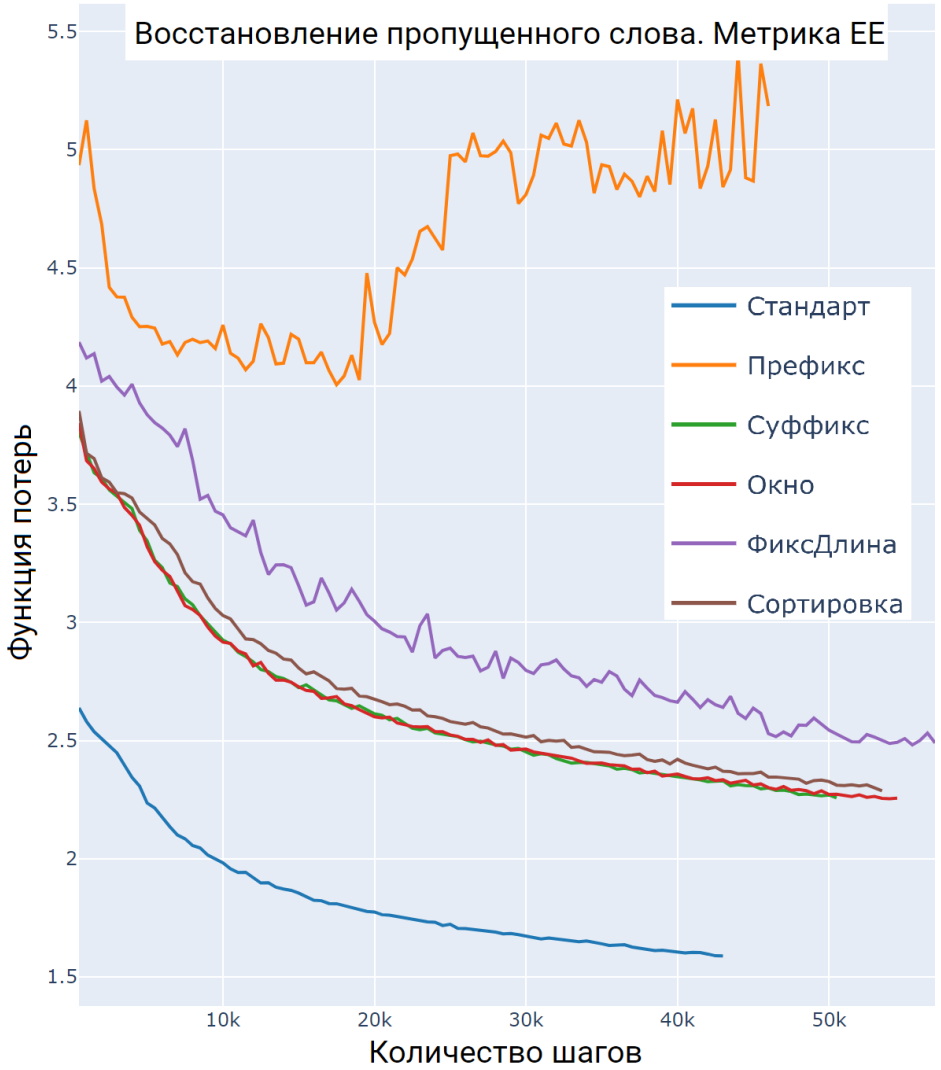
- **Сортировка**
  1. В предыдущих алгоритмах сэмплирования изменяется и структура батча, и их порядок.
  2. Не будем изменять первое
  3. Случайно поделим на батчи и отсортируем их в порядке увеличения средней сложности
- **ФиксДлина**
  1. Большинство метрик сильно коррелирует с длиной текста, убираем данную корреляцию
  2. В задаче восстановления слова по контексту длинные примеры могут быть проще



Эксперименты на задаче предобучения

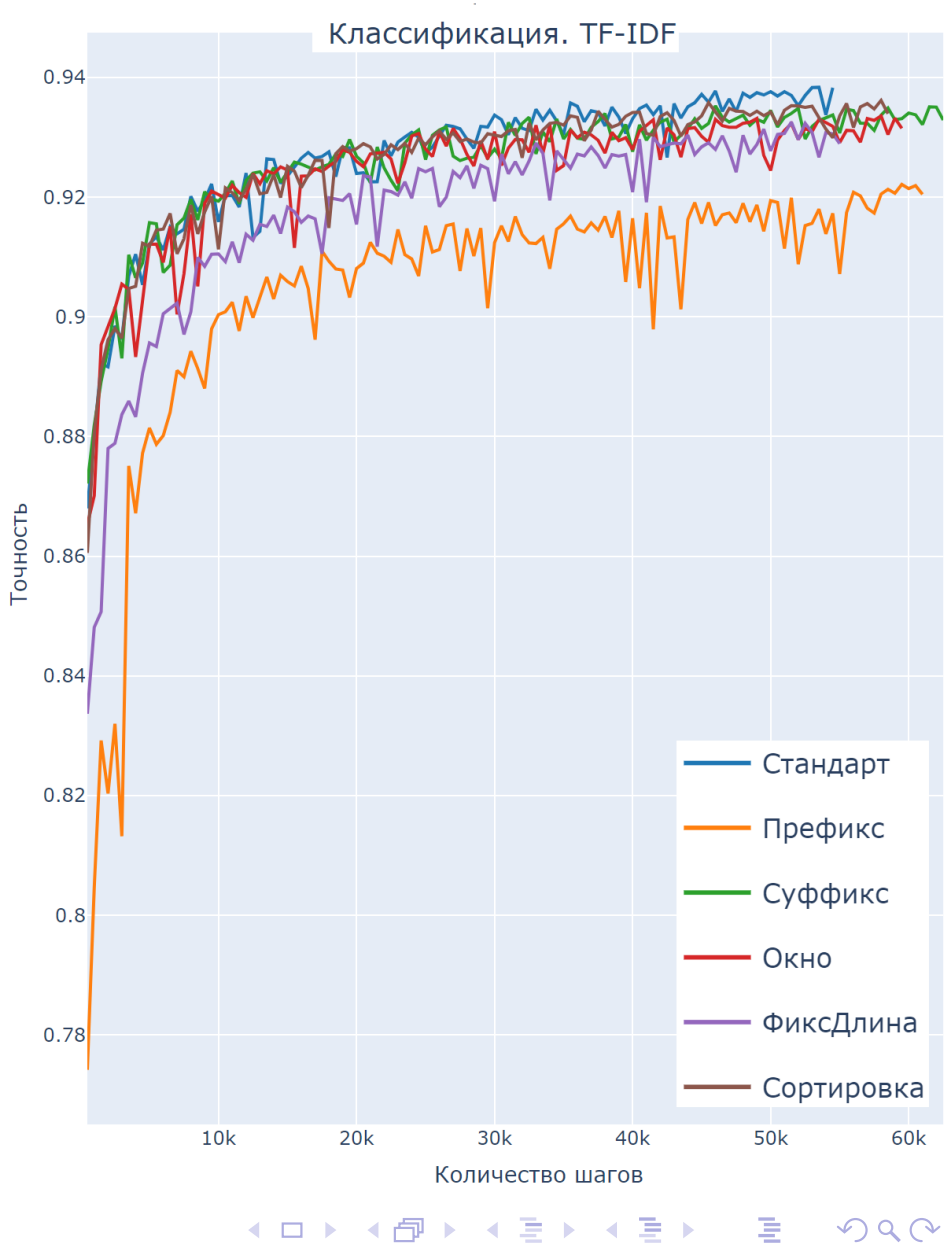
- Задача восстановления пропущенного слова по контексту
- Лондон - это <пропуск> Англии
- Датасет: BooksCorpus, 74M отрыков из книг

- Обучение по плану значительно ухудшает результаты



Эксперименты на задаче классификации

- Классификация новостей на политическую окраску (есть / нет)
- Датасет: Hyperpartisan News Detection, 2M новостей
- Обучение по плану не дает преимуществ на данной задаче



Результаты. Машинный перевод

- Датасет: WMT16, английский $\rightarrow$ немецкий, 4.5M элементов
- Метрика: значение BLEU

Сэмплер	Измерение сложности			Среднее
	Длина	EE	TSE	
Префикс	10.1	11.3	11.2	10.7
Суффикс	16.7	16.8	17.0	16.8
Окно	15.9	17.0	16.8	16.6
Сортировка	16.3	16.6	16.6	16.5
ФиксДлина	-	12.8	13.4	13.1
Стандарт				16.9

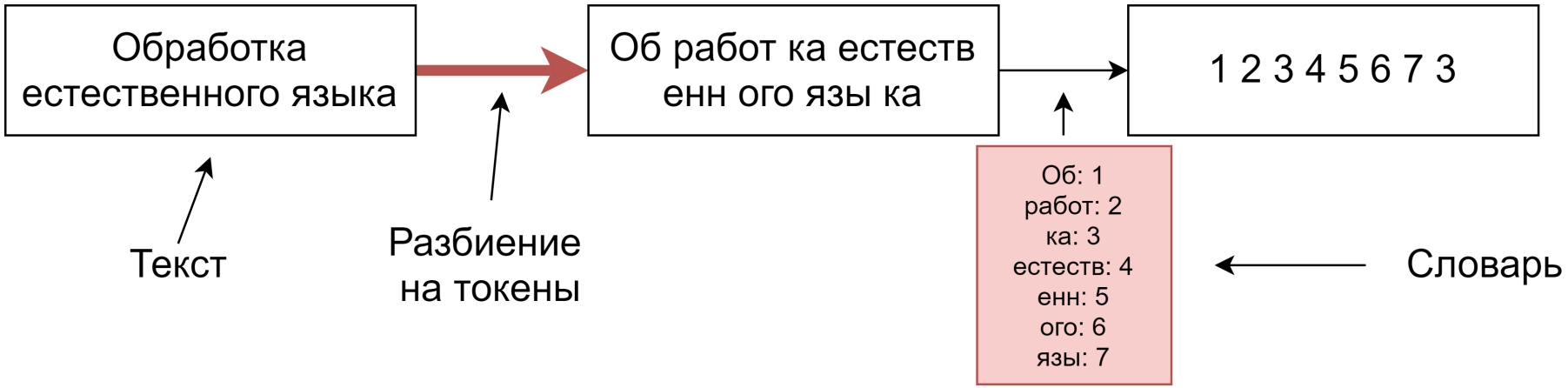
Анализ результатов

- Обучение по плану ни для одной из задач обработки естественного языка не дает улучшения
- Сэмплирование более сложных данных (**Суффикс**) на всех задачах показывает лучшие результаты, чем более простых (**Префикс**)
- Проверена и опровергнута гипотеза о влиянии токенизации как гиперпараметра
- Полученные результаты аналогичны Wu et al., 2020 для компьютерного зрения
- Такое различие в результатах в обучении с подкреплением и остальных областях может быть связано различными целями обучения по плану

Результаты

1. Реализована надстройка над библиотекой HuggingFace, позволяющая изменять алгоритм сэмплирования
2. Разработан широкий спектр сэмплирующих стратегий, обладающих различными свойствами
3. Проведены эксперименты на трех группах задач обработки естественного языка
   - Предобучение. Худший сэплер (Префикс) не справляется обучиться, лучший (Суффикс) уменьшает качество на 50%
   - Классификация. Худший сэмплер (Префикс) уменьшает точность на 2%, остальные отличаются друг от друга менее, чем на 1%
   - Машинный перевод. Худший сэмплер (Префикс) понижает качество на 20%, второй с конца (ФиксДлина) на 15%, остальные имеют незначительные отличия ($< 5\%$) от стандарта.
4. Обучение по плану не дает улучшения на задачах обработки естественного языка

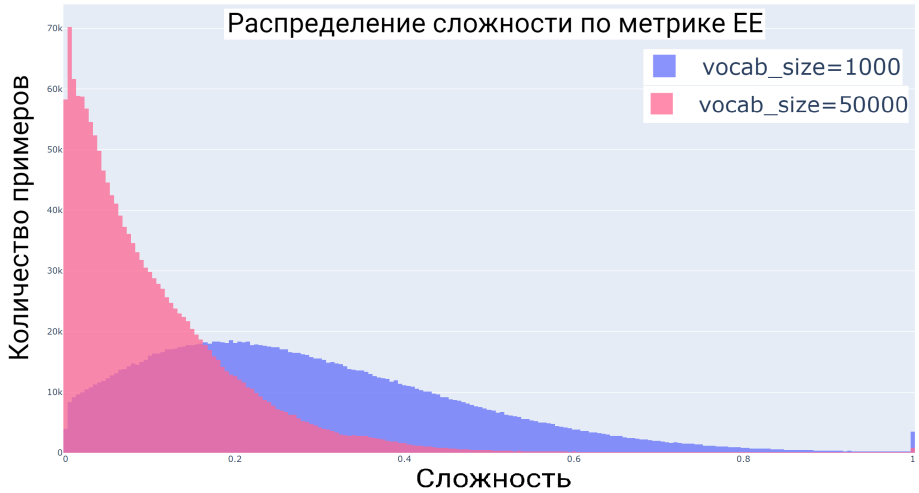
Предобработка данных. Токенизация



- Токенизация характеризуется алгоритмом и размером словаря
- Выделяется три основных способа токенизации
  - BPE - увеличение словаря самой частой парой
  - Unigram - уменьшение словаря на основании фунции потерь языковой модели
  - WordPiece - уменьшение словаря на основании фунции потерь языковой модели
- Есть работы, показывающие, что способы не идентичны

Влияние токенизации

- При изменении размера словаря изменяется не только количество токенов, но и их распределение



Стандарт

	5k	30k
BPE	12.6	16.8
UnigramLM	12.7	16.7
WordPiece	12.4	16.9

Префикс

	5k	30k
BPE	12.5	16.6
UnigramLM	12.7	16.6
WordPiece	12.6	16.8

- Обучение по плану практически не зависит от выбранного токенизатора