Сервис коррекции пунктуационных ошибок (MVP)

Никифоров Николай, @nikiforov_uze_bezit
Столяров Марк, @markusikk

Куратор: Морозова Валерия, @eternal_phobia

Идея

Сервис, исправляющий пунктуационные ошибки пользователей

Пример:

```
зачем кричать когда никто не слышит о чем мы говорим
```

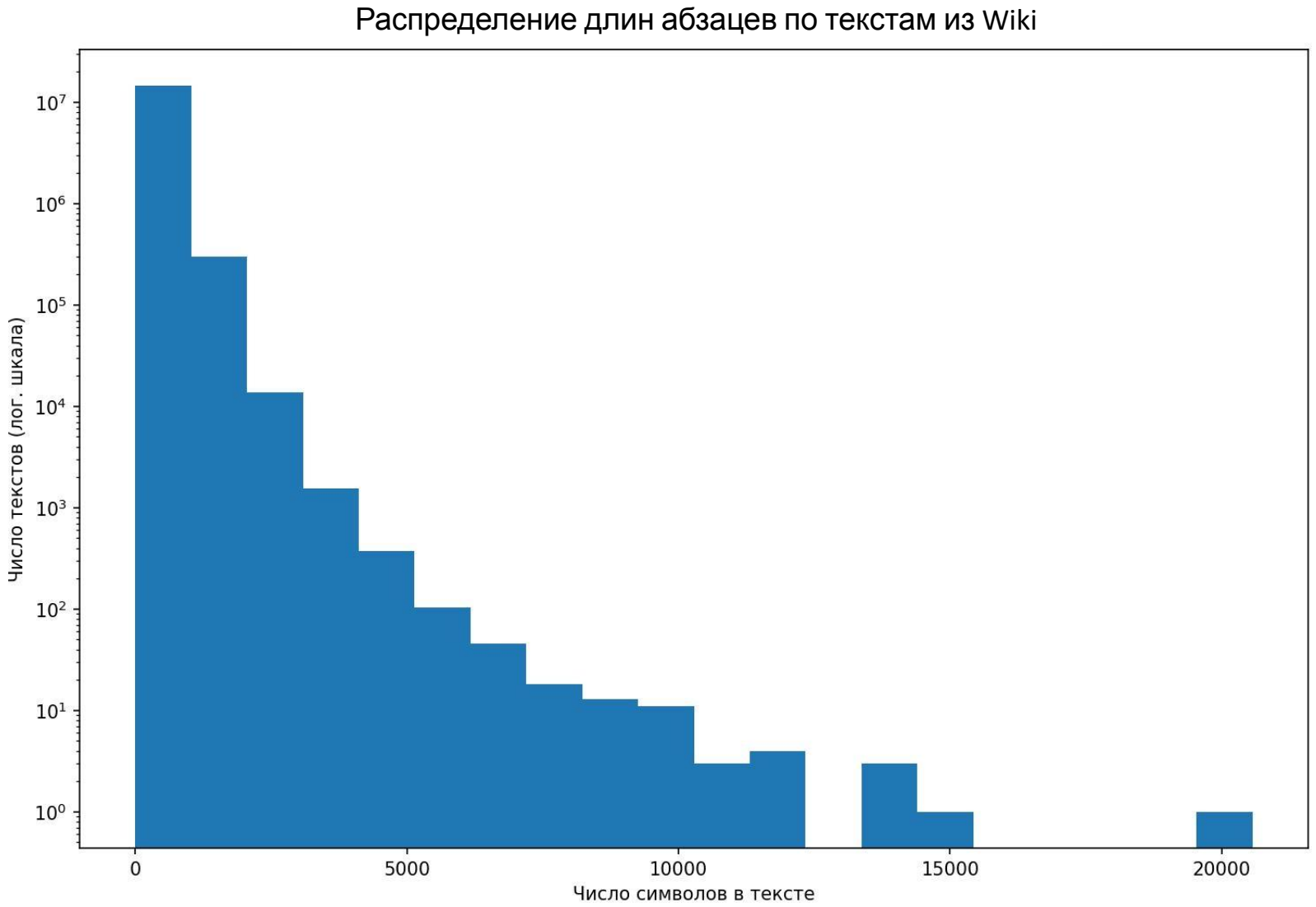
```
Зачем кричать, когда никто не слышит, о чем мы говорим?
```

Данные

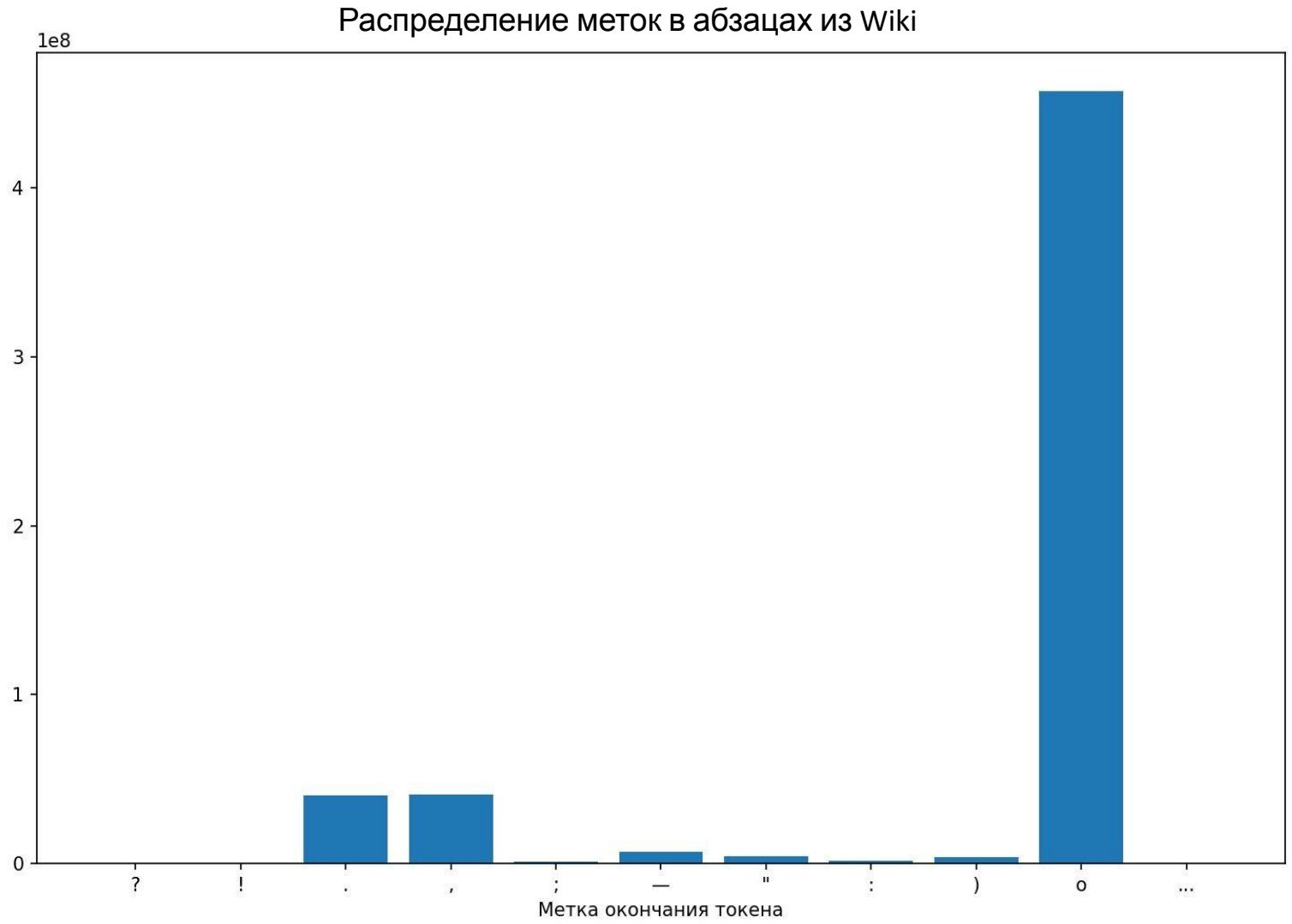
- Для обучения/измерения метрик были собраны данные с русскоязычной Википедии
- Использовался срез на 23.10.2023. Около 100 Гб файлов формата txt
- Содержимое файлов было предобработано и описано в EDA

- Дополнительно были взяты тексты 14 художественных произведений. Сбор и фильтрация осуществлялись вручную

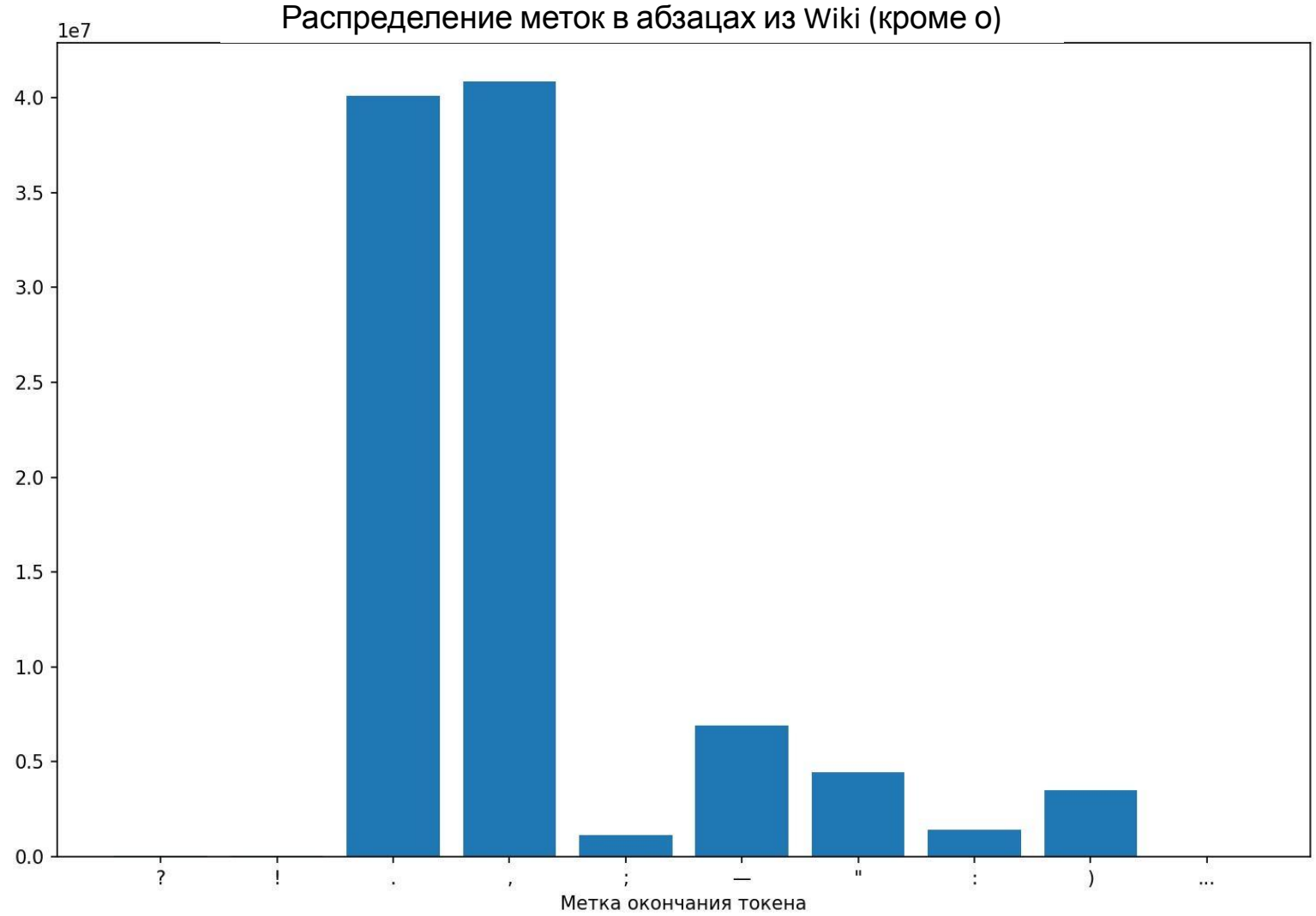
EDA


Распределение длин абзацев по текстам из Wiki
Число текстов (лог. шкала)
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
0
5000
10000
15000
20000
Число символов в тексте

EDA


Распределение меток в абзацах из Wiki
1e8
4
3
2
1
0
?
!
.
,
;
—
"
:
)
o
...
Метка окончания токена

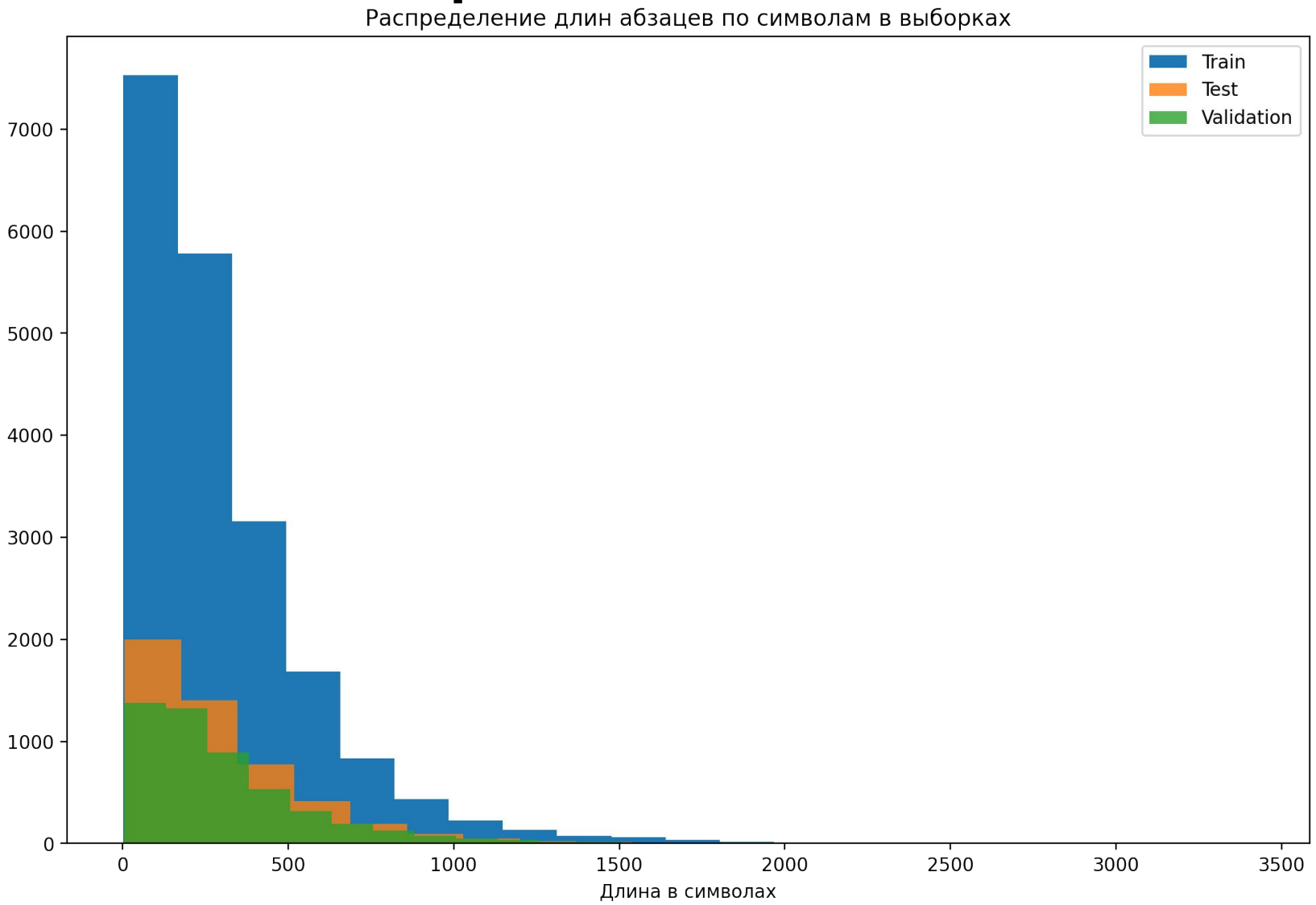
EDA


Распределение меток в абзацах из Wiki (кроме o)
1e7
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
?
!
.
,
;
—
"
:
)
...
Метка окончания токена

EDA. Сравнение распределений меток

Источник	**,**	**:**	**;**	**.**	**!**	**?**	**o**	**…**
RuWiki	7,4%	0,3%	0,2%	7,2%	0,0%	0,0%	82,3%	0,0%
Книги	13,3%	0,3%	0,6%	5,2%	0,4%	0,4%	78,7%	0,2%

Итоговые выборки



Итоговые выборки

Выборка	Кол-во текстов	,	:	;	.	!	?	o	…
Train	20 000	64 023	2 293	1 792	58 385	176	196	711 878	65
Validation	5 000	16 800	531	411	14 863	47	52	182 407	22
Test	5 000	15 944	567	512	14 654	48	54	179 729	11

в %	Кол-во текстов	,	:	;	.	!	?	o	…
Train	20 000	7,63%	0,27%	0,21%	6,96%	0,02%	0,02%	84,87%	0,01%
Validation	5 000	7,81%	0,25%	0,19%	6,91%	0,02%	0,02%	84,79%	0,01%
Test	5 000	7,54%	0,27%	0,24%	6,93%	0,02%	0,03%	84,97%	0,01%

Потенциальное расширение классов

сбалансировать классы

разумная аугментация

Модель внутри

Было рассмотрено несколько вариантов:

- Алгоритмы на заданных правилах
- Логистическая регрессия на векторных представлениях слов (Navec)
- Готовая XLM-Roberta с HuggingFace
- GigaChat с промтом

```
Расставь в тексте знаки препинания.
Текст: {текст пользователя}
```

Модель внутри

Было рассмотрено несколько вариантов:

- Алгоритмы на заданных правилах (+ добавление вероятностей)

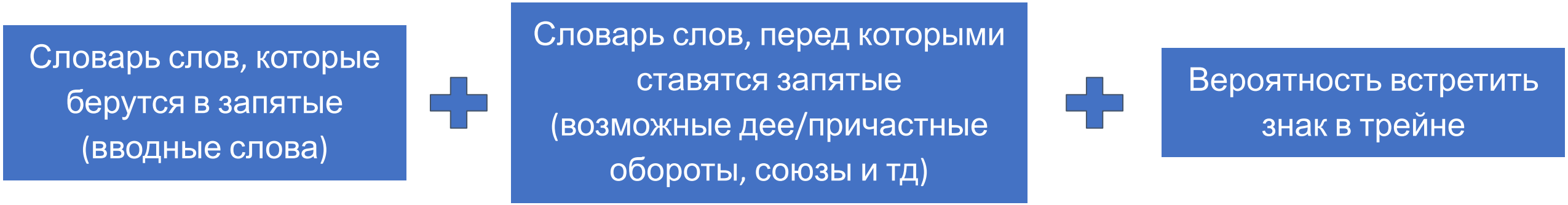
Словарь слов, которые берутся в запятые (вводные слова)

Словарь слов, перед которыми ставятся запятые (возможные дее/причастные обороты, союзы и тд)

Модель внутри

Было рассмотрено несколько вариантов:

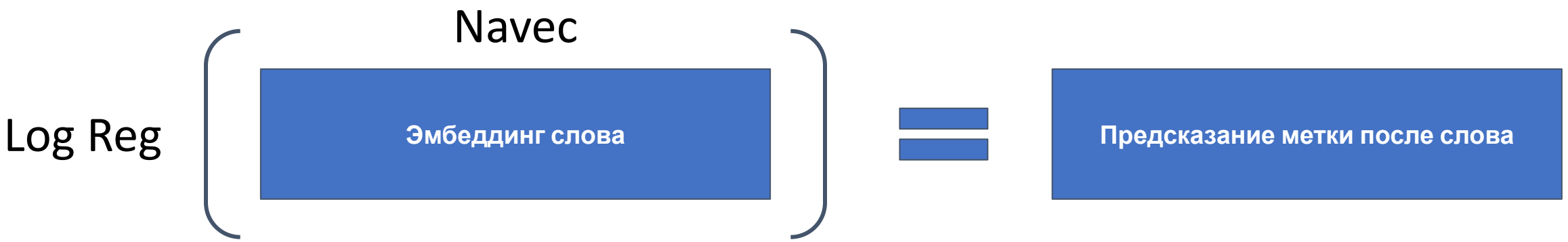
- Алгоритмы на заданных правилах (+ добавление вероятностей)



Модель внутри

Было рассмотрено несколько вариантов:

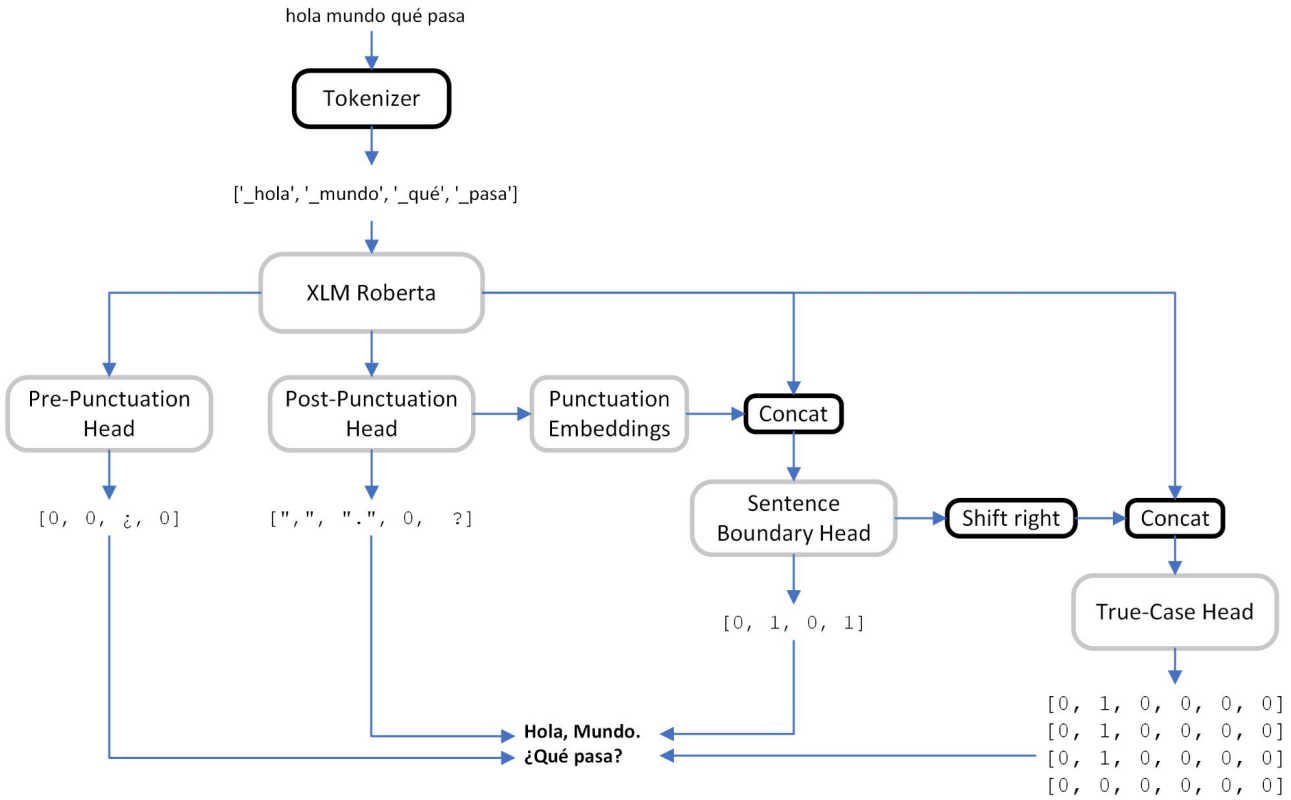
- Логистическая регрессия на векторных представлениях слов (Navec)



Модель внутри

Было рассмотрено несколько вариантов:

- Готовая XLM-Roberta с HuggingFace



Модель внутри

Было рассмотрено несколько вариантов:

- GigaChat (последняя версия) с промтом

```
Расставь в тексте знаки препинания.
Текст: {текст пользователя}
```

Метрики. Precision

Модель	**,**	**:**	**;**	**.**	**!**	**?**	**o**
Dictionary approach	0.075068	0.0	0.0	0.064600	0.0	0.0	0.851885
Dictionary + prob approach	0.070655	0.001919	**0.002488**	0.071174	0.0	0.0	0.851921
Log reg on Navec	0.076923	0.0	0.0	0.832973	0.0	0.0	0.854643
XLM-Roberta	**0.742795**	0.0	0.0	0.833501	0.0	0.400000	**0.982887**
Gigachat	0.726075	**0.475806**	0.0	**0.892733**	**0.726075**	**0.541667**	0.976232

Метрики. Recall

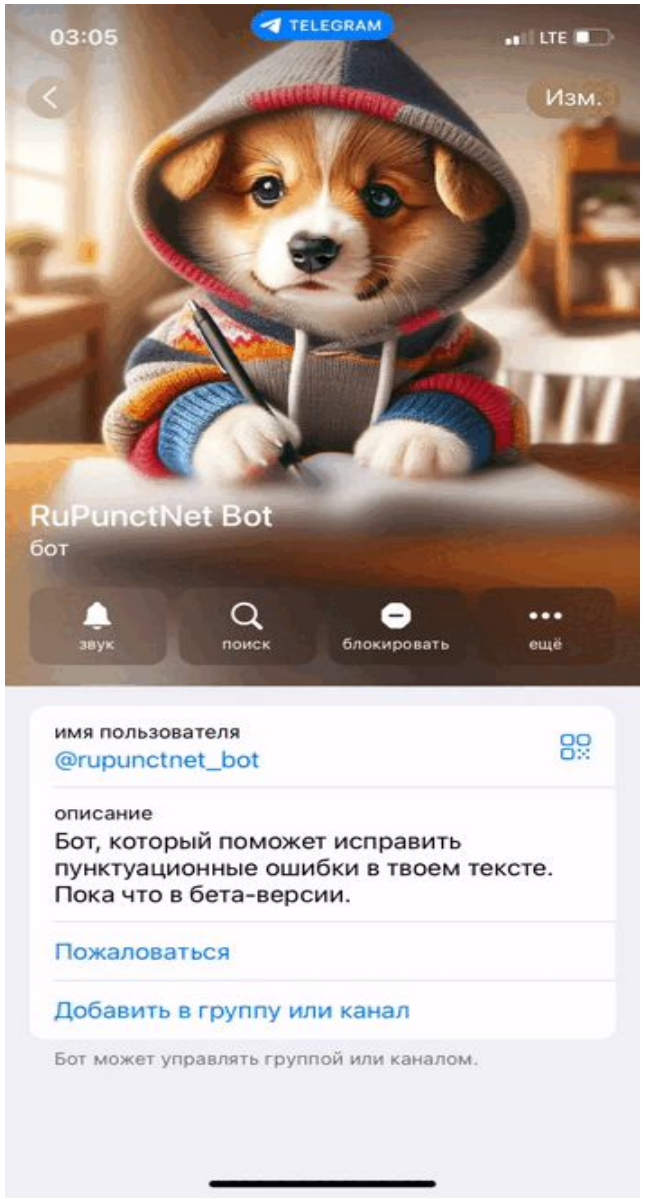
Модель	,	:	;	.	!	?	o
Dictionary approach	0.078857	0.0	0.0	0.022341	0.0	0.0	0.898399
Dictionary + prob approach	0.029986	0.003571	**0.003922**	0.151957	0.0	0.0	0.812837
Log reg on Navec	0.000063	0.0	0.0	0.078955	0.0	0.0	**0.999143**
XLM-Roberta	**0.832830**	0.0	0.0	**0.792018**	0.0	**0.518519**	0.982169
GigaChat	0.797981	**0.292079**	0.0	0.731549	**0.034483**	0.448276	0.985072

Метрики. F1-Score

Модель	**,**	**:**	**;**	**.**	**!**	**?**	**o**
Dictionary approach	0.076916	0.0	0.0	0.033200	0.0	0.0	0.874524
Dictionary + prob approach	0.042103	0.002497	**0.003044**	0.096942	0.0	0.0	0.831920
Log reg on Navec	0.000125	0.0	0.0	0.144237	0.0	0.0	0.921261
XLM-Roberta	**0.785240**	0.0	0.0	**0.812230**	0.0	0.451613	**0.982528**
GigaChat	0.760332	**0.361963**	0.0	0.804143	**0.060606**	**0.490566**	0.980632

Telegram Бот

@RUPUNCTNET_BOT



Telegram Бот

- Словарь Navec и/или XLM-Roberta требуют >2 Гб памяти, что превышает бесплатные лимиты
- Бот с этими подходами был реализован, но запускался лишь локально
- GigaChat API не требует значимых вычислительных ресурсов
- Реализация бота с GigaChat используется сейчас, работает 24/7 (деплой на Replit) Сейчас уже не работает (нет монет)

Что было после защиты

Обновили выборку

- Тексты из Вики были зашумлены (битые символы, урезанные предложения…)
- Текста в Вики не были эмоционально окрашены, из-за чего давало дисбаланс классов со смещением в нейтральные знаки
- Текстов было достаточно для fine-tuning модели, можно было сократить в целях экономии ресурсов
- Попытались расширить выборку эмоционально окрашенными текстами, сгенерированными Gigachat (на основе различных промтов и перефразом текстов книг)
  - Промт на генерацию не смогли подобрать - выдавал либо бессмысленные предложения, либо полный текст из ! и ?
  - Перефраз текущей выборки лишь добавил восклицательные знаки в конце каждого текста и обнулил бесплатные ресурсы на использование API (эксперименты прекратили)

В результате мы составили выборку из литературных произведений, что позволило снизить дисбаланс классов

Deep Learning методы

- Так как классические методы МО не слишком актуальны для нашей задаче, мы сразу стали экспериментировать с методами глубинного обучения
- К настоящему моменту удалось затюнить BERT (на задачу NER), качество которого превзошло все рассмотренные до этого решения (подробности на следующем слайде)
- BERT выложен на HuggingFace, где можно протестировать его работу через API

Новые метрики. Precision

Модель	,	:	…	.	!	?	o
Dictionary approach	0.132125	0.0	0.0	0.062645	0.0	0.0	0.798540
Dictionary + prob approach	0.125424	0.007109	0.0	0.060398	0.001678	0.001410	0.797963
XLM-Roberta	0.791148	0.0	0.0	0.720780	0.0	0.588957	**0.978929**
Gigachat	0.792691	0.321429	0.0	**0.850303**	0.421053	0.673913	0.958297
Fine-tuned BERT	**0.856620**	**1.000000**	0.0	0.809446	**0.569620**	**0.736301**	0.976896

Новые метрики. Recall

Модель	,	:	…	.	!	?	o
Dictionary approach	0.117743	0.0	0.0	0.027213	0.0	0.0	0.855209
Dictionary + prob approach	0.068574	0.014019	0.0	0.170166	0.004673	0.003268	0.736812
XLM-Roberta	0.835871	0.0	0.0	0.733345	0.0	0.646465	0.973990
Gigachat	0.782805	**0.135338**	0.0	0.664614	0.062992	0.587678	**0.981877**
Fine-tuned BERT	**0.845355**	0.004673	0.0	**0.851909**	**0.211268**	**0.693548**	0.981862

Новые метрики. F-1

Модель	,	:	…	.	!	?	o
Dictionary approach	0.124520	0.0	0.0	0.037944	0.0	0.0	0.825903
Dictionary + prob approach	0.088669	0.009434	0.0	0.089153	0.002469	0.001970	0.766169
XLM-Roberta	0.812895	0.0	0.0	0.727008	0.0	0.616372	0.976453
Gigachat	0.787717	**0.190476**	0.0	0.746078	0.109589	0.627848	0.969944
Fine-tuned BERT	**0.850950**	0.009302	0.0	**0.830135**	**0.308219**	**0.714286**	**0.979373**

Что хочется сделать дальше

- Возможно, еще больше обогатить выборку знаками (: … ! ?)
- Поэкспериментировать с улучшениями BERT (BERT-large, RoBERTa)
- Исследовать решения на основе других трансформеров (T5, GPT)
- Усовершенствовать сервис: сделать бота более функциональным (сбор статистики использования, выбор модели), поднять web-сервис
- Возможно также разработка моделей, корректирующих регистр букв и опечатки (T5 это может делать, поэтому возлагаем на нее большие надежды)

На этом пока что всё