СОЗДАНИЕ НОВОСТНОГО AI-БОТА

или как избавиться от информационного шума

Дата 24.09.2023

codenrock

КОМАНДА M&L

Дмитрий
Науменко
ML

Татьяна
Кутькина
ML, backend

Зорикто
Дамдинов
Капитан, ML

ПРОБЛЕМАТИКА

1. Слишком много источников ненужной читателю информации

2. Слишком много дубликатов новостей

3. Думскроллинг

4. Отсутствие единого новостного агрегата, способного персонализировать новостные предпочтения читателя

ЗАДАЧИ

Необходимо разработать 2 алгоритма:

1. Алгоритм, способный классифицировать новости по категориям
2. Алгоритм, способный точно и быстро идентифицировать дубликаты новостей

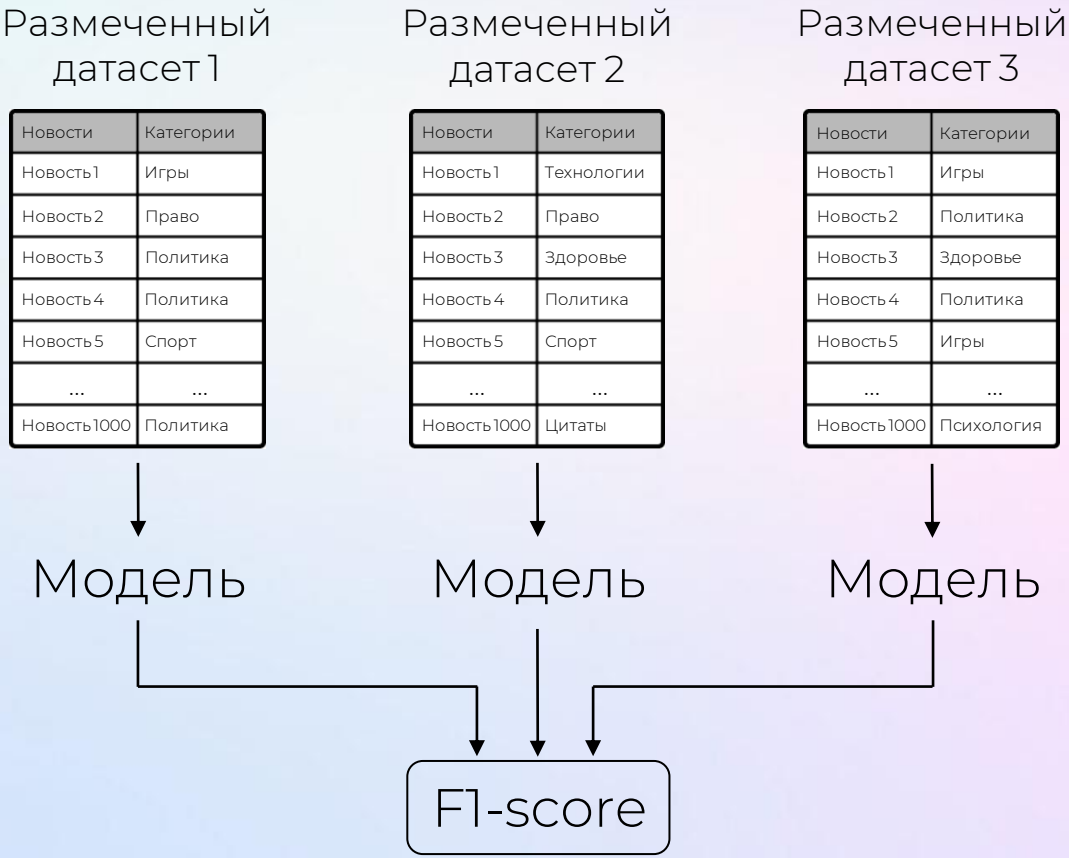
ДАННЫЕ

Способы улучшения датасета:

- Разметить все данные. То есть перевести задачу в обучение с учителем. Недостаток: долго или дорого.
- Использовать сторонние датасеты для обогащения данных. Однако обучение на датасете Лента.ру (700к сэмплов) не дало значимых улучшений.

В итоге используем первоначальный датасет и решаем задачу обучения без учителя.

Для процедуры валидации отобрали 1000 случайных новостей из датасета и разметили 3-мя разными людьми. Метрика F1-score измеряется как среднее среди 3-х датасетов.



Новости	Категории
Новость 1	Игры
Новость 2	Право
Новость 3	Политика
Новость 4	Политика
Новость 5	Спорт
...	...
Новость 1000	Политика

Новости	Категории
Новость 1	Технологии
Новость 2	Право
Новость 3	Здоровье
Новость 4	Политика
Новость 5	Спорт
...	...
Новость 1000	Цитаты

Новости	Категории
Новость 1	Игры
Новость 2	Политика
Новость 3	Здоровье
Новость 4	Политика
Новость 5	Игры
...	...
Новость 1000	Психология

КЛАССИФИКАЦИЯ НОВОСТЕЙ

Что пробовали?

Supervised:

- CountVectorizer
- TfidfVectorizer
- BERT
- Fine-tuned BERT

Unsupervised:

- BERT кластеризация
- Universal Sentence Encoder
- Матрица близости слов

Недостатки supervised:

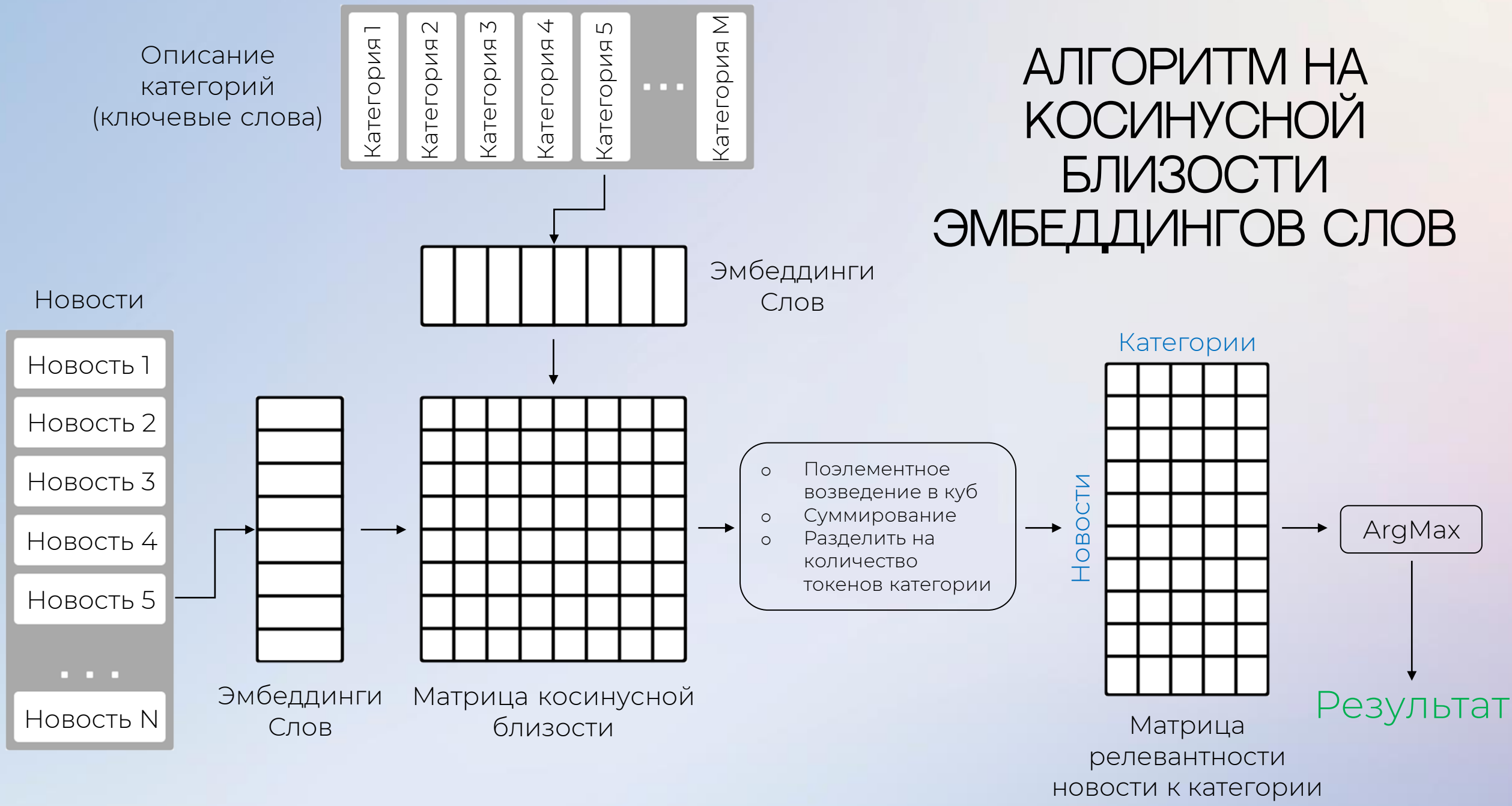
- Нет размеченных данных для обучения, критерии разметки не однозначны
- Размеченные данные смещены, поэтому обучение на них дает плохой результат на реальных датасетах

Проверка концепций определила моделей-финалистов, которые дальше сравнивались на валидации.

Модели-финалисты:

- Варианты кластеризации на эмбеддингах текста
- Матрица косинусной близости эмбеддингов текстов

АЛГОРИТМ НА КОСИНУСНОЙ БЛИЗОСТИ ЭМБЕДДИНГОВ СЛОВ



СРАВНЕНИЕ АЛГОРИТМОВ

Сравнение метрик на валидации

Модель	f1 micro (all)	f1 micro (w/o rare)	accuracy
clf_kernel	0.2956	0.2961	0.4784
part_bert_res	0.0518	0.0523	0.0943
closest_word_bert	0.0327	0.0285	0.0532
cats_from_gpt_news	0.1067	0.1064	0.1996
closest_gpt_news	0.0916	0.0904	0.1785
optimized_clf_cosin	0.2956	0.2961	0.4784

Преимущества нашего решения:

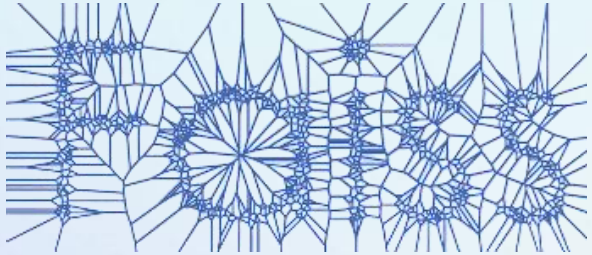
1. Возможность подстраивать категории без обучения
2. Высокое качество
3. Быстродействие
4. Без GPU, но можно задействовать
5. Экономия памяти

Лучший алгоритм классификации – optimized_clf_cosin.

Время работы алгоритма на 50к сэмплах: ~1 минута.

ПОИСК ДУБЛИКАТОВ

Meta



Выявленные недостатки
ANN-алгоритмов FAISS, NeoFuzz:

- Пропускает даже явные дубликаты (до 5%)
- Для частотных векторов не хватает памяти
- Эмбеддинги теряют данные и требуют времени на расчет

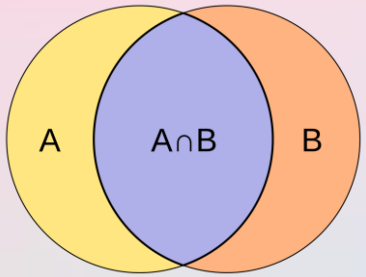
Найденное оптимальное решение – сравнение коэффициента Жаккара по «Bag-of-Words».

Алгоритм показывает хороший результат нахождения дубликатов вкупе с быстрой скоростью (~2 минуты для 50к сэмплов с мультпроцессингом).

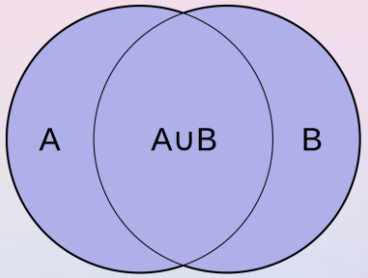
КОЭФФИЦИЕНТ ЖАККАРА

$$K_J = \frac{n(A \cap B)}{n(A \cup B)}$$

Пересечение множеств



Объединение множеств



Пример работы алгоритма:

I. «В лесу упало дерево»
II. «Дерево упало в лесу»
III. «Кошка залезла на дерево»

Лес – 1
Упасть – 2
Дерево – 3
Кошка – 4
Залезть – 5

I. {1, 2, 3}
II. {3, 2, 1}
III. {4, 5, 1}

	I	II	III
I		3/3	1/5
II			1/5
III			

Вывод:
I и II дубликаты

ИНСТРУМЕНТЫ И ТЕХНОЛОГИИ

- Python
- Pandas
- Numpy
- BERT
- Universal Sentence Transformer
- Torch
- Navec
- Faiss
- NeoFuzz
- FastAPI
- Unicorn
- Docker

КОНТАКТЫ

Науменко Дмитрий –
t.me/naumenko_ds
Кутькина Татьяна –
t.me/Tatyanna_Kutkina
Дамдинов Зорикто –
t.me/suzuyajxiii

gravitypotter@gmail.com

GITHUB