Машинное обучение и высоконагруженные системы

Факультет компьютерных наук

Москва (2024)

Построение новостного графа и сервисов вокруг него

Команда:

Думенков Максим (@maxodum)

Кривошеев Сергей (@FlameInBrain)

Куратор проекта:

Бабынин Андрей (@maninoffice)

Задача

1. Создание NLP моделей, которые оценивают 'влияние' новости на определенные финансовые инструменты
    - Определение 'влияния': как публикация новости сказывается/может сказаться на определенных финансовых инструментах согласно определенному уровню
    - Уровни:
        - Глобальный (экономика страны) - если новость о государственной сущности или о компании, которая составляет серьезную долю в ВВП (или чем-то подобном)
        - Локальный (отрасль) - если новость об отрасли или о крупном игроке в рамках отрасли, влияние на которого может сильно повлиять на отрасль
        - Точечный (компания) - если новость о конкретной компании
    - Финансовые показатели:
        - Глобальный уровень: индекс MOEX, индекс RVI, курс RUBUSD
        - Локальный уровень: отраслевой индекс (i.e. MOEXOG, MOEXEU, MOEXTL, etc.)
        - Точечный уровень: акции компаний согласно тикету (i.e. VKCO, SBER, YNDX, etc.)
    - Output модели:
        - Метки: '+' - положительное 'влияние', '0' - отсутствие 'влияния', '-' - отрицательное 'влияние’
2. Создание сервиса вокруг моделей

В предыдущих сериях:

Данные

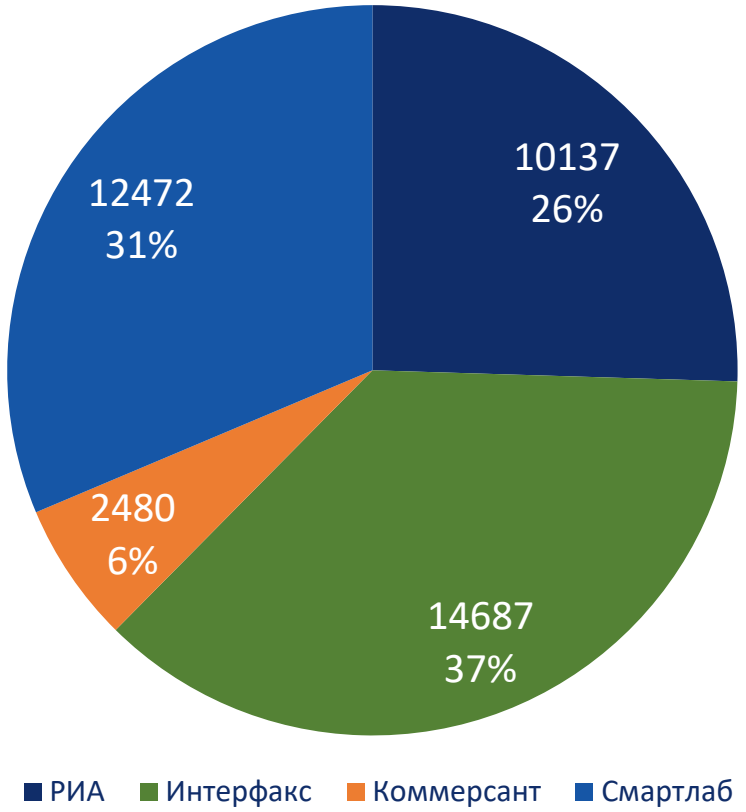
Источниками данных для данного проекта являлись новостные порталы, с которых мы загружали новости экономического характера, которые вышли в период с начала 2023 года по примерно ноябрь 2023 года. Таких порталов было 4:

1. Смарт-лаб
2. РИА
3. Интерфакс
4. Коммерсант

Они были собраны с помощью парсинга, а затем загружены в базу PostgreSQL

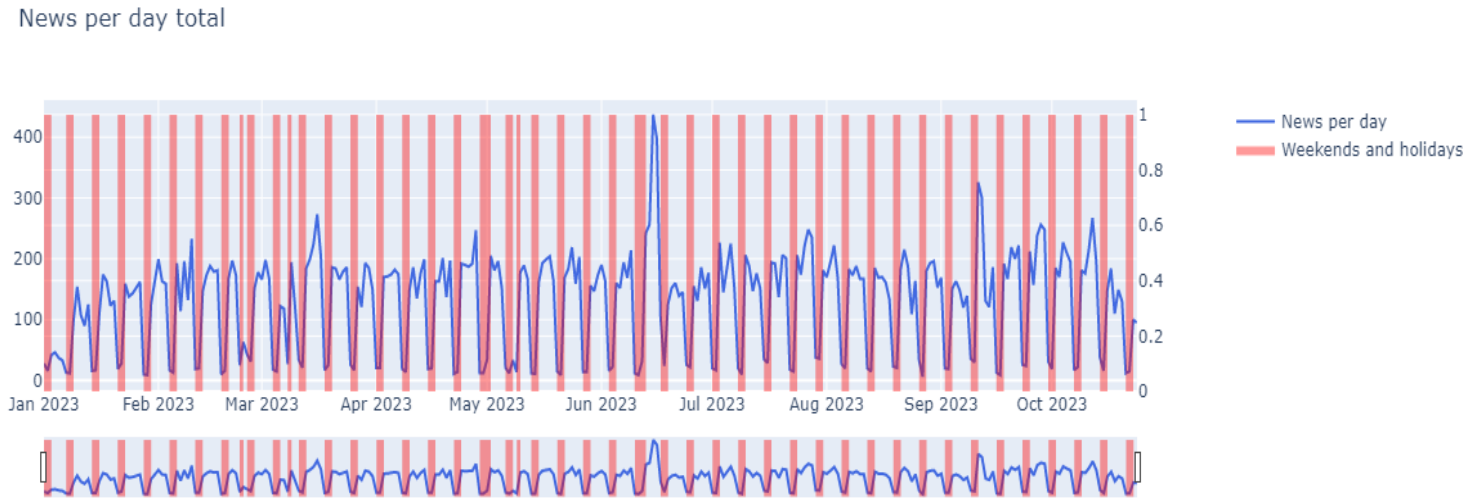
Всего было собрано 39776 новостей

Соотношение собранных новостей по источникам до EDA



EDA

- Чистка дублей
- Чистка текстов новостей
- Анализ распределений длин новостей
- Построение временного ряда количества новостей
- Отфильтровали наблюдения по рабочим часам Мосбиржи (MOEX)
- Анализ тэгов



В результате осталось 29899 новостей

EDA. Интересные цифры

15.06

Самый насыщенный новостями день из всего датасета. В этот день проходил ПМЭФ

≈ 170

Новостей приходится на каждый торговый день Мосбиржи в среднем из используемых источников

12000

Тэгов и ключевых слов встречаются в новостях от 1 до 3 раз

NER и обогащение данными Мосбиржи

- С помощью Natasha извлекли сущности из текстов новостей
- Поставили в соответствие каждой новости свою сущность (нескольким новостям может соответствовать несколько сущностей)
- Почистили те новости, где встречается большое количество сущностей (обзоры рынка)
- Поделили новости на датасеты, исходя из задачи

Итого:

- **7542** наблюдений – датасет, где присутствуют компании из индекса широкого рынка
- **6228** наблюдений – датасет, где присутствуют компании, образующие индустриальные индексы
- **13326** наблюдений – датасет из финансовых новостей

Дальше каждый из датасетов был обогащен информацией об изменении цены инструмента через полчаса после релиза новости

Формирование таргета

$$priceDiffPercent = \frac{Price_{\Delta t} - Price_{newsRelease}}{Price_{newsRelease}} * 100\%$$

- Где Δt – время после релиза новости, является эмпирически установленным параметром. Мы использовали Δt = 30 минут, так как это было в одной из статей на смежную тематику, а также исходя из эмпирики
- После этого мы смотрели на распределение $priceDiffPercent$ по конкретному инструменту, и во всех случаях оно оказывалось, примерно, нормальным и симметричным. Как следствие, таргет был сформирован следующим образом:

$$target = \begin{cases} 0\ (negative), \text{если } priceDiffPercent \leq q_{0.05} \\ 1\ (neutral), \text{если } q_{0.05} < priceDiffPercent \leq q_{0.95} \\ 2\ (positive), \text{если } priceDiffPercent \geq q_{0.95} \end{cases}$$

- Где $q_{0.05}$ и $q_{0.95}$ - квантили 0.05 и 0.95, соответственно
- Квантили были выбраны экспериментальным путем

Предобработка текста для использования в моделях

Для того, чтобы токенизировать тексты и использовать их в моделях, необходимо было предварительно их предобработать. Для этого были проделаны следующие шаги:

- Удалены лишние элементы разметки: табуляции и т.д.
- Удалены скобки и прочие “служебные” структуры
- Удалены стоп-слова, характерные русскому языку (согласно nltk + немного расширили этот список)
- Все слова были лемматизированы с использованием Natasha

* Стоит отметить, что в трансформенных моделях эти преобразования проделаны не были, так как мы брали модели, предобученные на нечищенных текстах

Модели и результаты

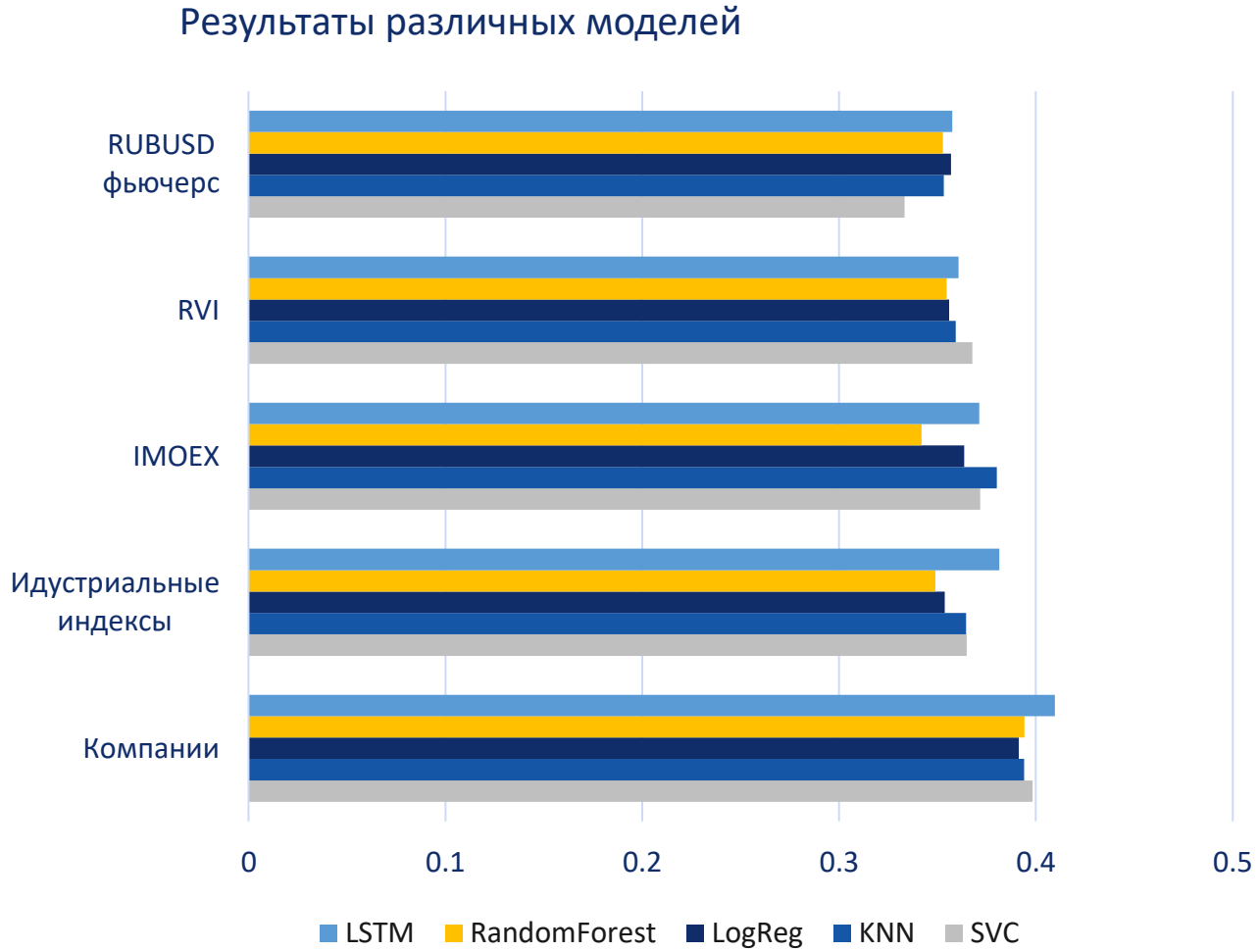
Метрика - Macro Average F1

Протестировали несколько моделей классификации новостей на 3 категории: позитивные, нейтральные и негативные

В классических моделях для генерации эмбеддингов был использовался tf-idf, в более сложных — tf-idf, BERT и Fast Text

Из классических моделей тестировали логистическую регрессию, SVM, RandomForest и KNN

Из более комплексных моделей — LSTM, Xgboost, BERT



Итоговая модель LSTM

- Параметры словаря:
  1. Максимальное количество слов в словаре: 10000
  2. Максимальная длина каждого текста: 500
- Параметры эмбеддингов:
  1. Размерность эмбеддингов: 500

- В качестве оптимизатора использовался Adam
- Внутри модели каждому классу был присвоен свой вес согласно представленности класса

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 500, 500)          5000000

 lstm (LSTM)                 (None, 32)                68224

 dense (Dense)               (None, 3)                 99

=================================================================
Total params: 5068323 (19.33 MB)
Trainable params: 5068323 (19.33 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Возникшие сложности

Модель не может обучиться должным образом. Возможные причины:

1. Недостаток данных. Данных слишком мало для выявления сложных зависимостей, а простые модели не могут их уловить
2. Зависимости, которые мы пытаемся найти, на самом деле не существуют

Service

Project News Analytics Telegram Bot

У телеграм-бота 5 разных функций

About This Service + **Disclaimer** - прочитать про этот бот, его функции

Make Prediction of News' Influence on Financial Instrument – прислать новость (в виде ссылки с одного из 4 порталов, либо в виде текста) и получить предсказание модели по этой новости

Rate Our App and Leave the Comment – оценить приложение и оставить отзыв

Show App Ratings and Comments – прочитать отзывы и узнать среднюю оценку

Display Info About Ticker - получить информацию о данных торгов акций, входящих в индекс широкого рынка Мосбиржи, за последний торговый час

Также будет реализован показ графа сущностей!

Что было сделано на этом чекпойнте:

- Собрали дополнительные данные
- Перепроверили существующие модели на больших датасетах
- Поэкспериментировали с подбором Δt
- Попытались переформулировать задачу
- Поэкспериментировали с архитектурами трансформеров

- Переписали модели на PyTorch
- Реализовали pipeline

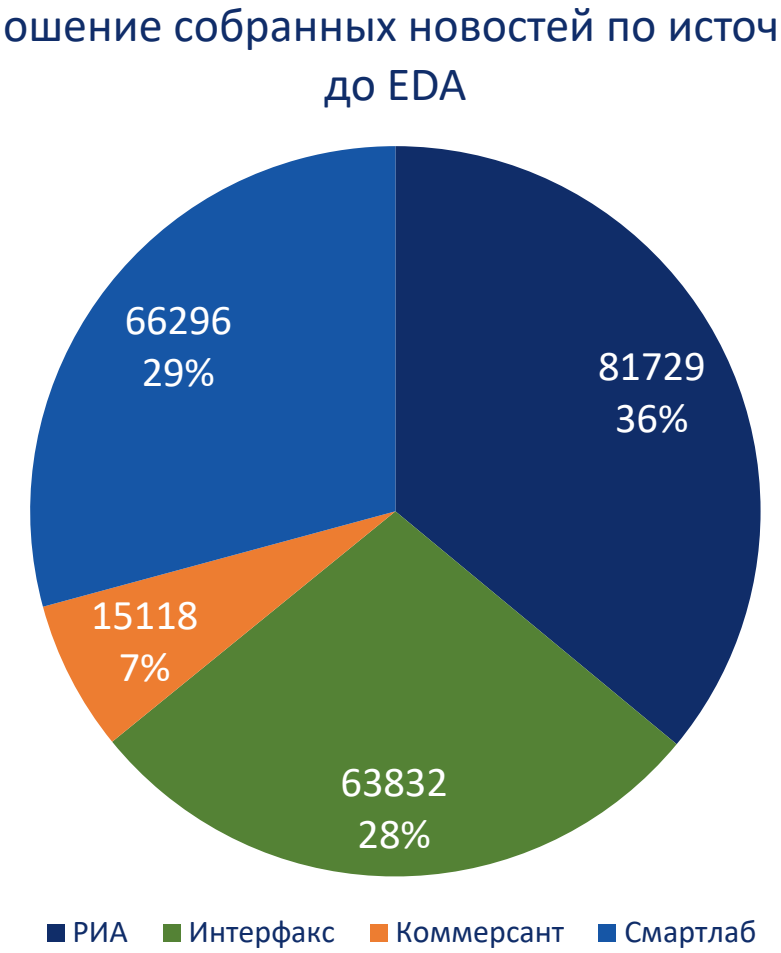
Данные

Источниками данных для данного проекта являлись новостные порталы, с которых мы загружали новости экономического характера, которые вышли в период с начала 2019 года по конец 2023 года. Таких порталов было 4:

1. Смарт-лаб
2. РИА
3. Интерфакс
4. Коммерсант

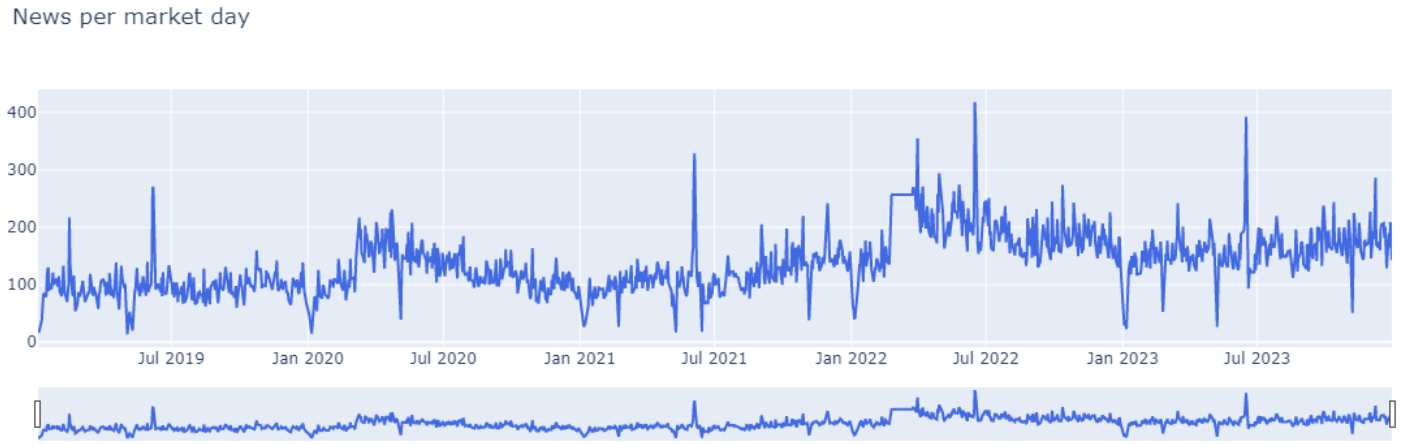
Ранее мы загружали данные в базу на PostgreSQL. Сейчас для хранения данных используется DVC

Всего было собрано 226975 новостей



EDA

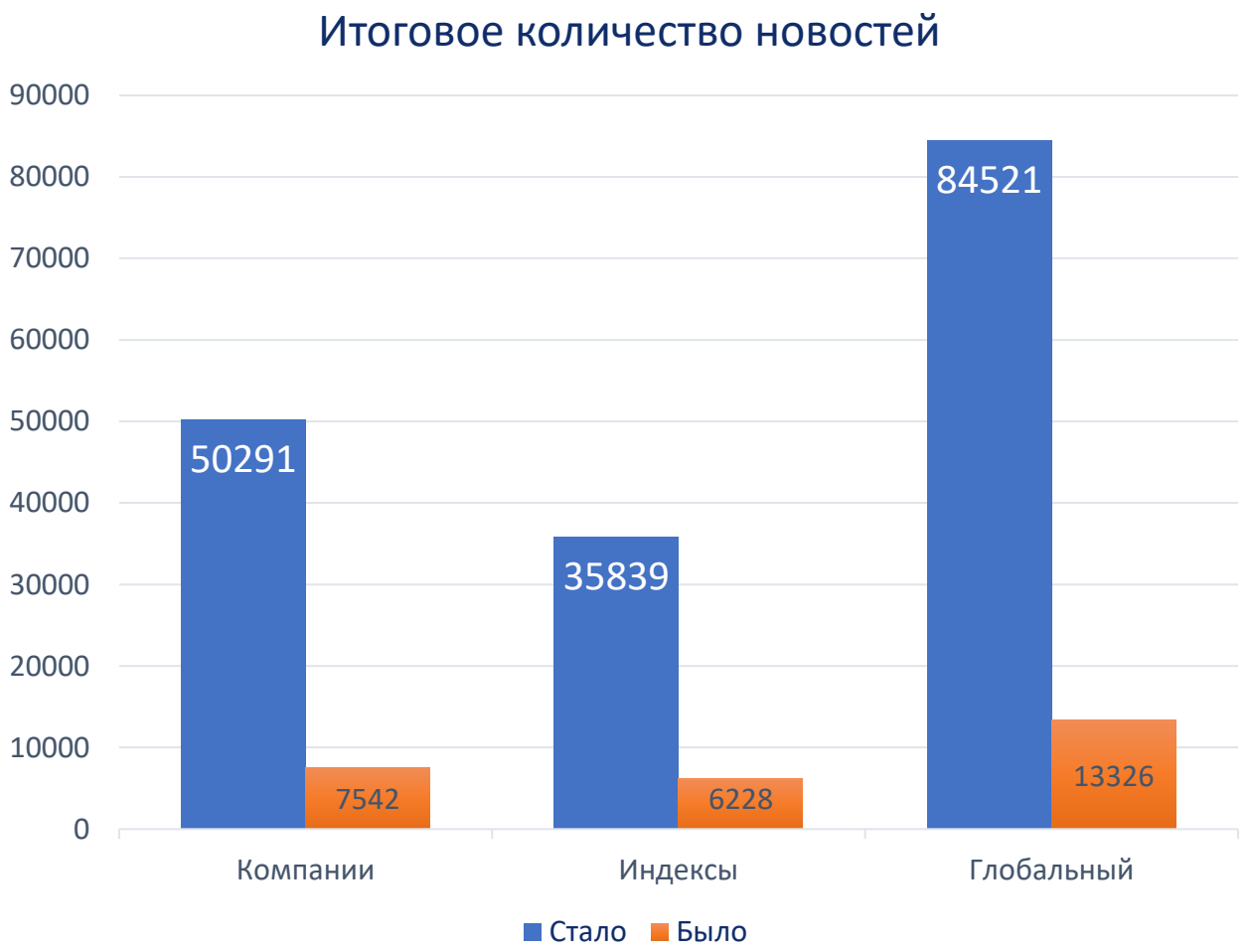
- Чистка дублей
- Чистка текстов новостей
- Анализ распределений длин новостей
- Построение временного ряда количества новостей
- Отфильтровали наблюдения по рабочим часам Мосбиржи (MOEX)
- Анализ тэгов



В результате осталось 169848 новостей

Финальный датасет

- Далее мы провели аналогичное прошлому извлечение сущностей с помощью Natasha
- Поставили в соответствие каждой новости свою сущность (нескольким новостям может соответствовать несколько сущностей)
- Почистили те новости, где встречается большое количество сущностей (обзоры рынка)
- Поделили новости на датасеты, исходя из задачи



Эксперименты с BERT

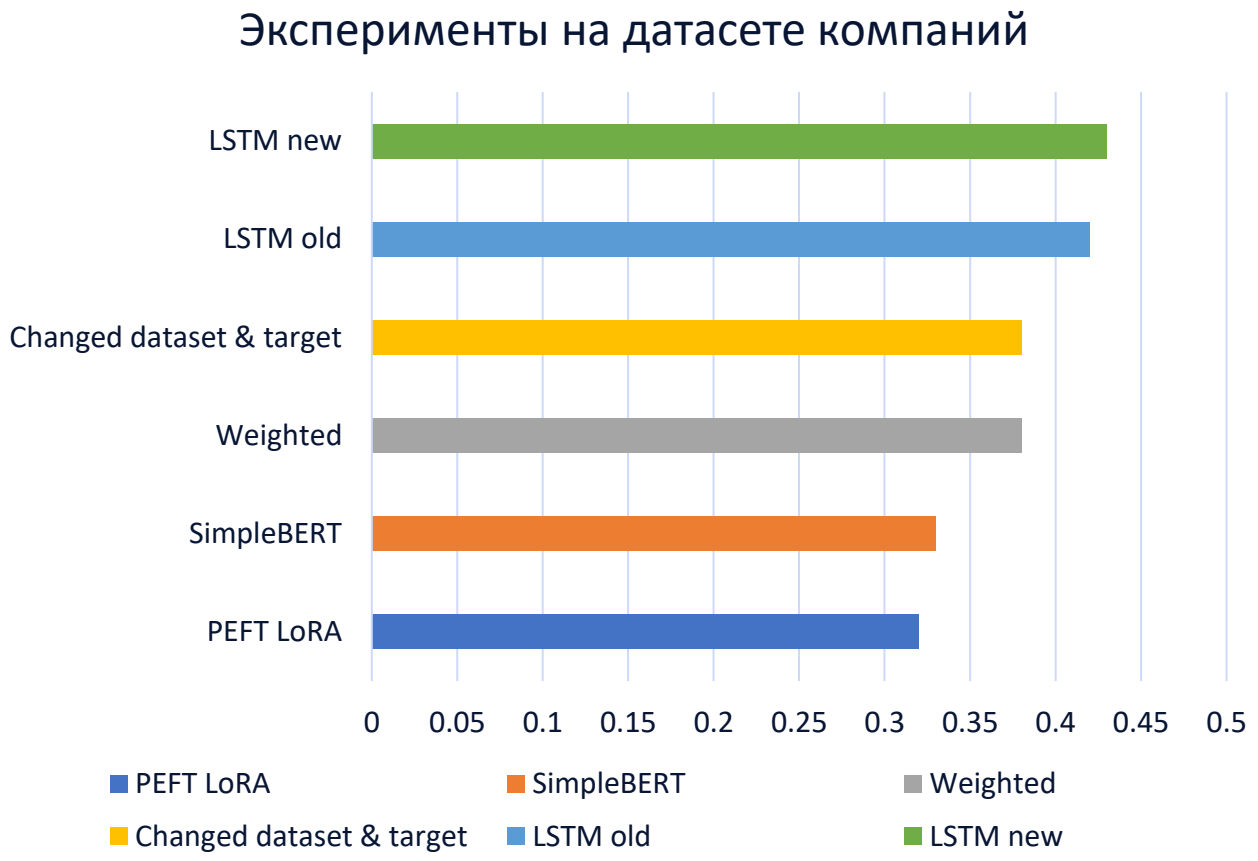
Во всех экспериментах использовался tiny-rubert2

- **Первый эксперимент** – использование Bert как есть
- **Второй эксперимент** – добавление весов в функцию ошибок, согласно представленности каждого класса таргета
- **Третий эксперимент** – использование техники PEFT: LoRA, вместо Linear Probing
- **Четвертый эксперимент** – попытка убрать часть датасета, которая реагирует дольше и более аналитически, а также использовать большие по модулю изменения

В качестве оптимизатора использовался AdamW

Для сравнения на графике также есть результаты старой LSTM модели (на старых данных) и новой (на новых данных)

Как следствие, в качестве финальной модели по-прежнему использовали LSTM



Эксперименты с Δt

- Для проверки гипотезы, что на результаты может влиять разница во времени, которое прошло с момента выхода новости, мы протестировали новости за последний год (похожие тенденции распространяются и на финальные датасеты, покрывающие 5 лет)
- Были использованы следующие варианты дельты: [5, 10, 15, 30, 45, 60, 75, 90]
- Несмотря на то, что результаты отличались в зависимости от использованной дельты, разница оказалась достаточно несущественной для того, чтобы отказаться от ранее использованных 30 минут

Эксперименты с формулировкой задачи

- Попытка изменить время, с которого мы отсчитываем выпуск новости. Было предположение, что из-за инсайдеров или других несовершенств рынка некоторые участники могут получать информацию раньше релиза новостного сообщения. Чтобы проверить данную гипотезу, мы изменили базовое время с релиза новости на 30 минут до этого момента. Модель в этом случае не смогла показать лучший результат

- Попытка переформулировать задачу в бинарную классификацию. Были мысли, что в случае более равномерной выборки модель сможет обучиться на какие-то признаки в тексте, которые бы указывали на сам факт изменения, без направления. Модель снова не смогла улучшить результат

- Попытка сосредоточится на одном инструменте. Возникло ощущение, что использование одного инструмента позволит найти что-то характерное в тексте новостей для этих компаний. В датасете компаний было несколько инструментов, в которых было достаточно данных для эксперимента. Были использованы SBER и GAZP. Результаты практически не отличались от результатов полного датасета, причем отличались в худшую сторону

Pipeline

- В рамках этого чекпоинта также был оформлен полный pipeline работы: от сбора данных до инференса

- В отдельной репе можно найти три основных этапа: scraping, preprocessing и modeling – для удобства взаимодействия были добавлены команды в Makefile, которые позволяют быстро прогонять весь pipeline. Остается лишь задать условия в config файлах (для этих целей использовали Hydra)

- На следующих чекпоинтах планируется:
  - Внедрение полноценного мониторинга экспериментов с использованием MLflow/ClearML/Wandb (пока не определились с инструментом)
  - Создание DAG’а для автоматического парсинга данных за последний фиксированный период (с использованием Airflow)

Планы на будущее

Research часть:

- Сделать граф финансовых сущностей: либо используя нынешние модели (что вряд ли), либо обучив отдельную модель под генерацию эмбеддингов, сделать граф, где вершинами будут выступать компании и, возможно, федеральные ведомства, а ребра будут отображать их связь в контексте финансовых и экономических новостей

Продуктовая часть:

- Закрыть техдолг по tg-боту
- Сделать API для взаимодействия с моделями, убрав эту логику непосредственно из tg-бота (в идеале, модель вынести на отдельный инференс сервер, типа OpenVINO Server)
- Сделать мониторинг, логирование и тестирование сервисов
- Сделать dashboard в Streamlit для более user-friendly взаимодействия с EDA частью

Спасибо за понимание!


ВЫСШАЯ • ШКОЛА • ЭКОНОМИКИ •