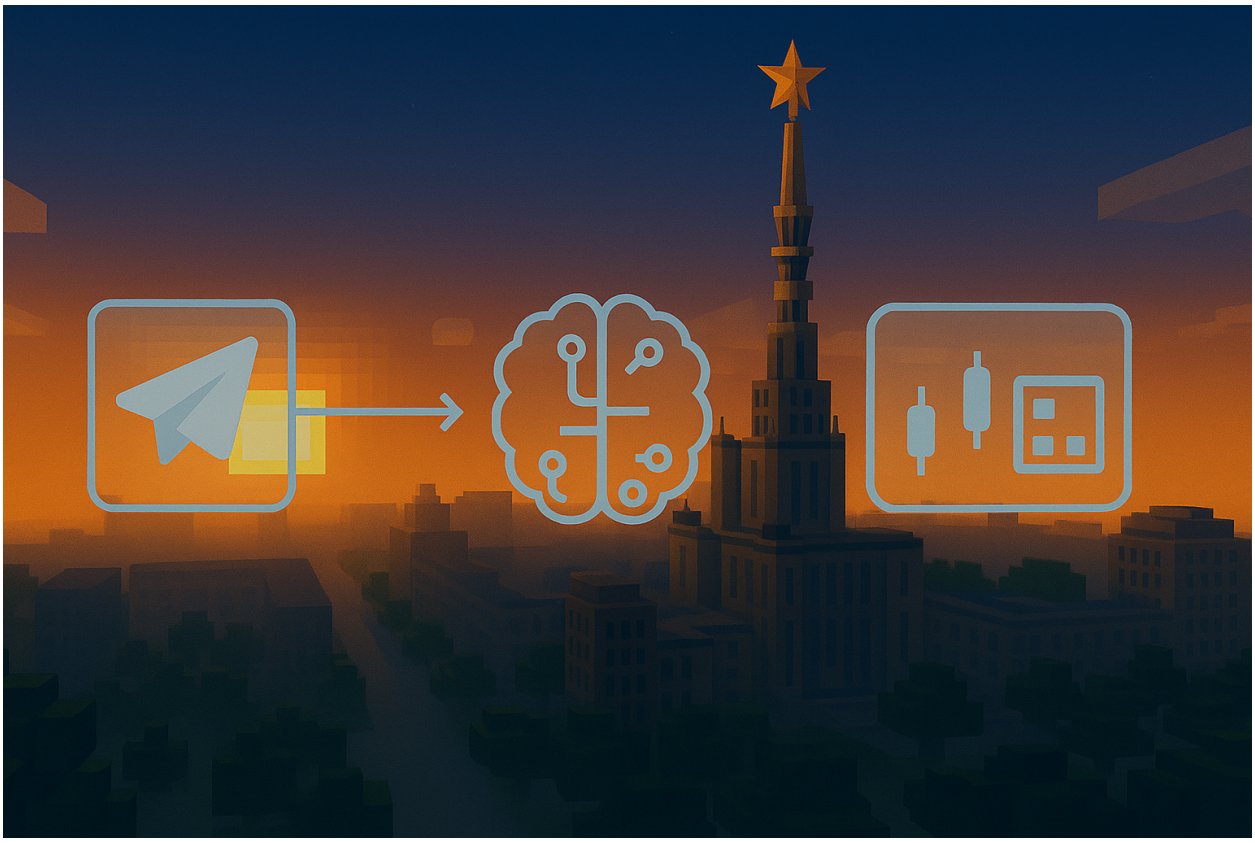
Генеративный ИИ

Алексей Колосов, н.с. МАТИС

Первое задание

9 ноября 2025 г.



Задание

Необходимо разработать систему анализа новостей из телеграм-каналов с использованием большой языковой модели (LLM) для *бинарной классификации* направления изменения цены акций на Московской бирже (MOEX).

А именно, *вверх / вниз* на заданном горизонте (например, до конца следующего торгового дня). Система должна собирать новости, преобразовывать их в признаки с помощью LLM и выдавать прогноз, после чего проводится оценка точности по фактическим котировкам MOEX.

Комментарий к заданию

- Каждый модуль должен принимать *текстовые сообщения* из выбранных телеграм-каналов, обрабатывать их с помощью LLM и преобразовывать в *сигналы/признаки*, релевантные движению цены
- *Текстовым описанием* выступают новости, посты, заголовки, краткие сводки, а также контекст (дата/время публикации, автор, ссылки)
- Прогноз направления (*up/down*) делается для конкретного тикера и горизонта, определённых самостоятельно, на основе окна новостей до момента формирования прогноза

Параметры целевой задачи

Результатом работы является модель, изменяющая исходное состояние «незнания» направления цены на *осмысленный прогноз* и демонстрирующая воспроизводимую точность на исторических данных. Целевые параметры задачи:

- Тикеры MOEX (например, `GAZP`, `SBER`, `LKOH`).
- Горизонт прогноза (H): например, до закрытия текущей сессии, до закрытия следующего торгового дня или фиксированный промежуток
- Окно новостей (W): например, все сообщения за последние 6–24 часа до момента прогноза

Значения целевых параметров задачи выбираются самостоятельно.

Требования к решению

- **Использование LLM:** применять LLM (например, Llama, GPT-5 и др.) для извлечения признаков из текста (тональность, упоминания тикеров/персон, тип события, срочность) и/или непосредственной zero/few-shot классификации сигнала «вверх/вниз». *Дообучение весов LLM не требуется.* Допускается обучение лёгких моделей поверх признаков (логистическая регрессия, градиентный бустинг) и/или подбор порога на валидации.

- **Интеграция с источниками данных:**
  - Телеграм: получение сообщений из выбранных каналов (через официальный Bot API/TDLib/экспорт, с соблюдением правил платформы)
  - MOEX: загрузка исторических котировок (минутные/почасовые/-дневные OHLCV) по выбранным тикерам для построения целевых меток и оценки

- **Функциональность пайплайна:**
  - Сбор и нормализация сообщений (язык RU, удаление ссылок/эмодзи, дедупликация, выделение времени публикации)
  - Привязка новостей ко времени торгов (MSK), учёт лага доставки/обработки (*без утечки будущей информации*)
  - Формирование целевой метки: dir $\in \{\uparrow, \downarrow\}$ по отношению цены Close_{t+H} к Close_t (или иной чётко заданной базе)
  - Реализация минимум одного *LLM-подхода*
  - Воспроизводимая оценка качества (accuracy как основной критерий; дополнительно confusion matrix, precision/recall по желанию)

- **Обработка естественного языка:** корректная интерпретация русскоязычных сообщений (в т.ч. сленг, хэштеги, множественные тикеры, нейминг компаний), извлечение сущностей и событий, агрегирование сигналов за окно W
- **Документация:** репозиторий на GitHub с инструкцией по установке, настройке и запуску; описание данных (источники, период, таймзона), конфигурации окна W и горизонта H, а также **скриптом** для полного прогона эксперимента end-to-end
- **Непрерывность работы:** все компоненты должны запускаться *без дообучения LLM* (только zero/few-shot). Разрешены кэширование ответов LLM, обновление данных и переоценка метрик без изменения весов нейросетей.
- **Соответствие:** соблюдайте условия использования Telegram и источников рыночных данных

Quickstart

- **Шаг 1:** Определите постановку: тикеры, окно новостей W, горизонт H, торговые сессии (MSK)
- **Шаг 2:** Настройте сбор новостей из Telegram (официальные методы; сохранение message_id, текста, времени, канала)
- **Шаг 3:** Загрузите исторические котировки MOEX для выбранных тикеров (OHLCV; синхронизируйте календарь торгов)
- **Шаг 4:** Реализуйте модуль преобразования текста в признаки/сигналы с помощью LLM (промпты, few-shot примеры)
- **Шаг 5:** Постройте генератор меток «вверх/вниз» и валидный сэмплер примеров (без look-ahead/утечек)

- **Шаг 6:** Реализуйте базовый и LLM-классификаторы; подготовьте скрипт оценки и сравнения моделей
- **Шаг 7:** Сформируйте отчёт: метрики, confusion matrix, абляции (влияние окна W, разных каналов, горизонта H)

Критерии оценивания решения

Необходимое условие для получения положительной оценки: отсутствие утечки будущей информации. Для проверки решения будут выбраны три случайных тикера из выбранных тикеров при самостоятельном определении значений целевых параметров. Итоговым результатом является средняя сбалансированная точность (avg Balanced Accuracy) по трём тикерам для выбранных при самостоятельном определении значений целевых параметров H и W.

- **Удовлетворительно**: avg Balanced Accuracy ≥ 0.50 и < 0.52
- **Хорошо**: avg Balanced Accuracy ≥ 0.52 и < 0.55
- **Отлично**: avg Balanced Accuracy ≥ 0.55

Примечания к проверке:

- Модель может отказываться выдавать бинарный прогноз
- Необходимо также рассчитывать доверительный интервал для получаемой avg Balanced Accuracy
- Все предсказания и входные тексты для каждого дня должны логироваться (timestamp, промпт/конфигурация, ответ LLM) для воспроизводимости

Срок сдачи задания 23:59:59 30.11.25. Ссылку на репозиторий необходимо отправить @brocmc.

Подсказки

- **Промпт-инжиниринг:** дайте LLM чёткую инструкцию: извлечь тикеры, определить тип события (дивиденды/отчёт/сделка/санкции/авария), оценить полярность и *срочность*, выдать нормированный сигнал $[-1, +1]$ или напрямую класс $\{\uparrow, \downarrow\}$
- **Синхронизация времени:** используйте таймзону MSK и календарь торгов MOEX; фиксируйте лаг между публикацией новости и моментом, когда модель могла её «увидеть»
- **Защита от утечек:** запрещено использовать данные после момента прогноза; разделяйте периоды на train/val/test по времени
- **Интеграция:** спроектируйте модульные компоненты: *ingest* (Telegram) $\rightarrow$ *store* $\rightarrow$ *feature/LLM* $\rightarrow$ *label* $\rightarrow$ *model* $\rightarrow$ *eval*
- **Метрики:** кроме accuracy, полезны balanced accuracy (при дисбалансе классов), F1, а также отчёт о стабильности по разным периодам/каналам
- **Логирование и отладка:** логируйте исходные тексты, ответы LLM, промежуточные признаки и финальные решения; сохраняйте версии данных/порогов/промптов

Что должно быть в репозитории

- `/README.md` — постановка задачи, запуск, описания W, H, список каналов (или критерии отбора)
- `/configs/.yaml` — параметры окна, горизонта, тикеров, путей к данным и ключей
- `/data/` — образцы (или скрипты загрузки); инструкции по воспроизводимости

- `/src/` — модули: *ingest_telegram*, *load_moex*, *llm_features*, *baseline*, *classifier*, *evaluate*
- `/notebooks/` (опционально) — исследование данных/абляции
- Скрипт `run_experiment.py` — единая точка запуска всего пайплайна end-to-end

Минимальные требования к демонстрации (baseline)

1. Сбор новостей за выбранный период и тикер
2. Генерация признаков/сигнала через LLM (zero/few-shot) *без дообучения весов*
3. Формирование меток $\uparrow$ / $\downarrow$ по котировкам MOEX на горизонте H
4. Подсчёт accuracy на тестовом периоде и вывод confusion matrix