Nea AI

поиск новости по запросу

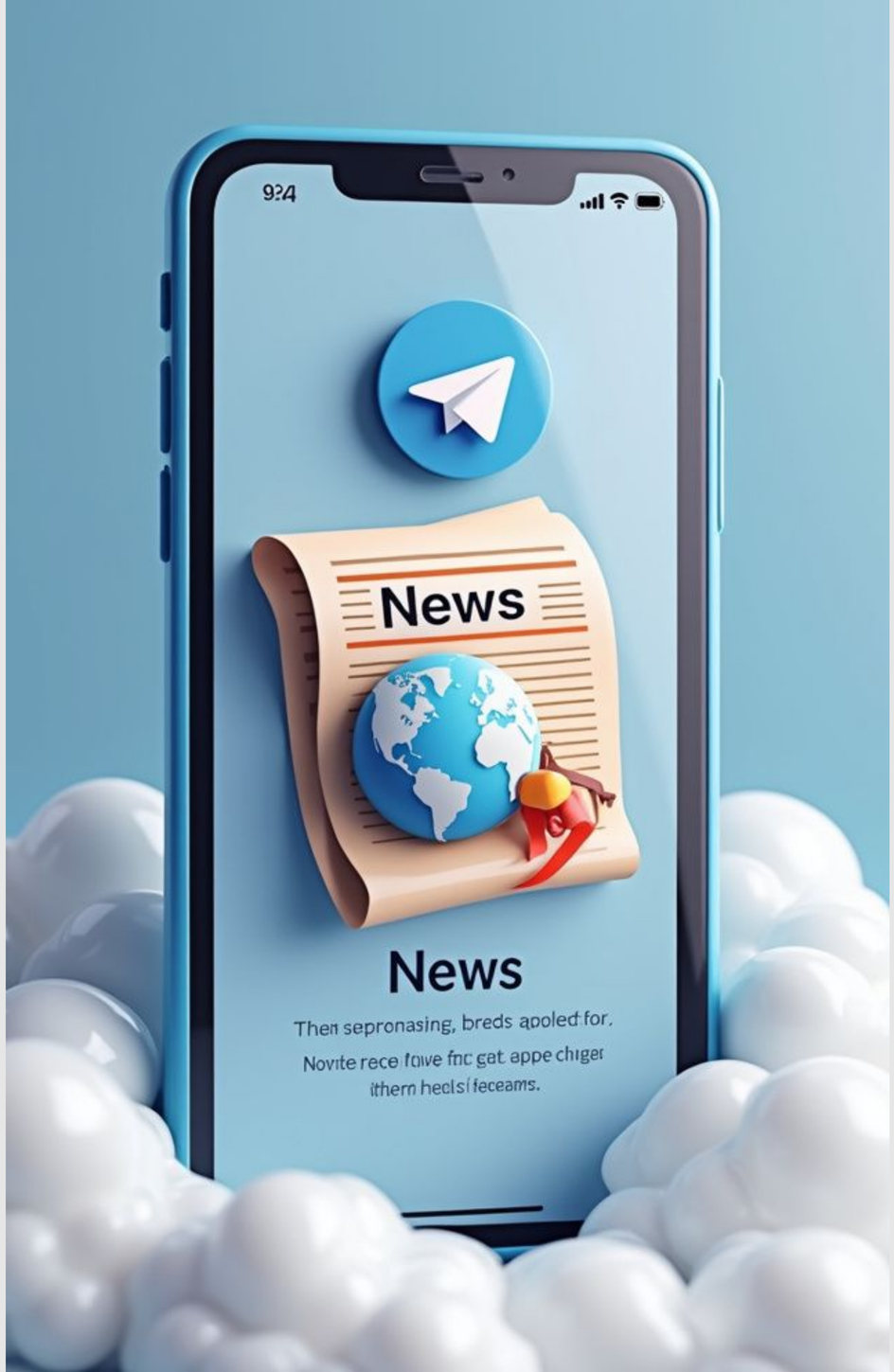
Ситуация

Пользователи находят новости по запросу и получают главную информацию

Продукт

Приложение с базой данных телеграмм каналов

Митрофанов Александр
Кувшинов Владимир



С точки зрения бизнеса

→ **Проблема**

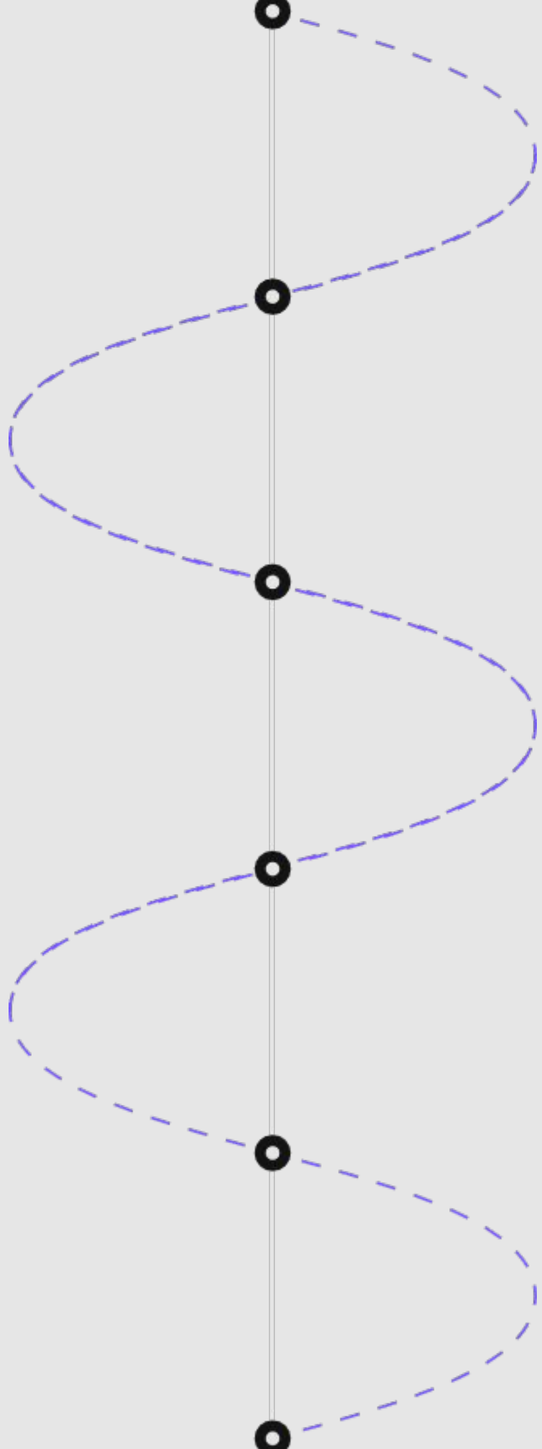
Чтение телеграмм каналов
занимает много времени

→ **Решение**

Выдавать ключевую информацию
по запросу пользователя

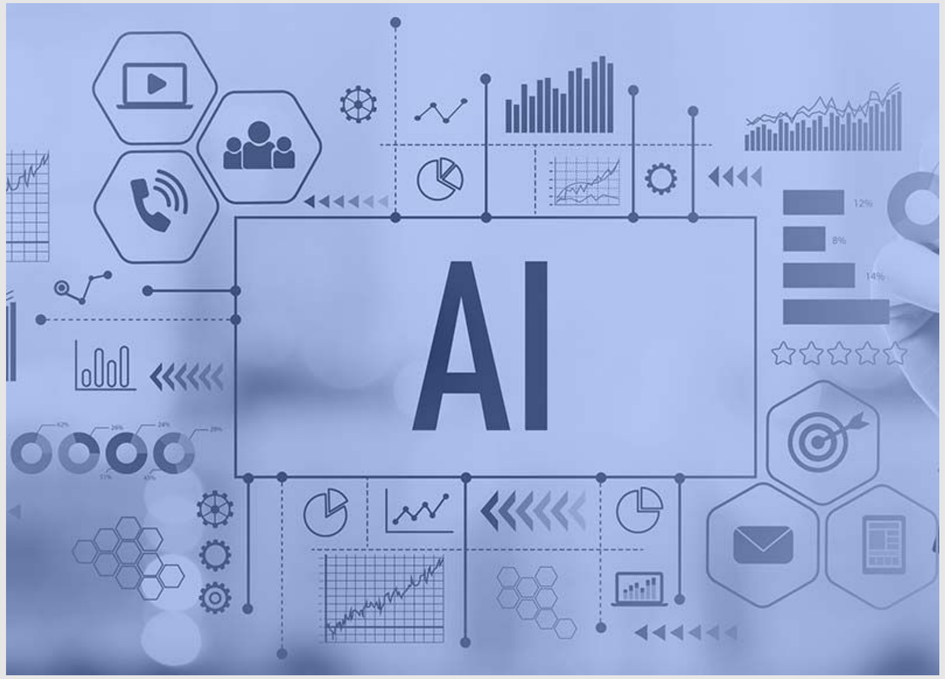
→ **Бизнес метрики**

время поиска информации
релевантность найденной информации
частота обращения к боту
количество отправленных вопросов



Постановка ML-задачи

- Максимизировать точность найденных новостей
- Чтобы покатить в прод, достаточно иметь 80% accuracy (топ-1) на тех картинах, которые есть в нашей базе знаний.
- Оптимизация технологий с целью ускорения поиска запроса

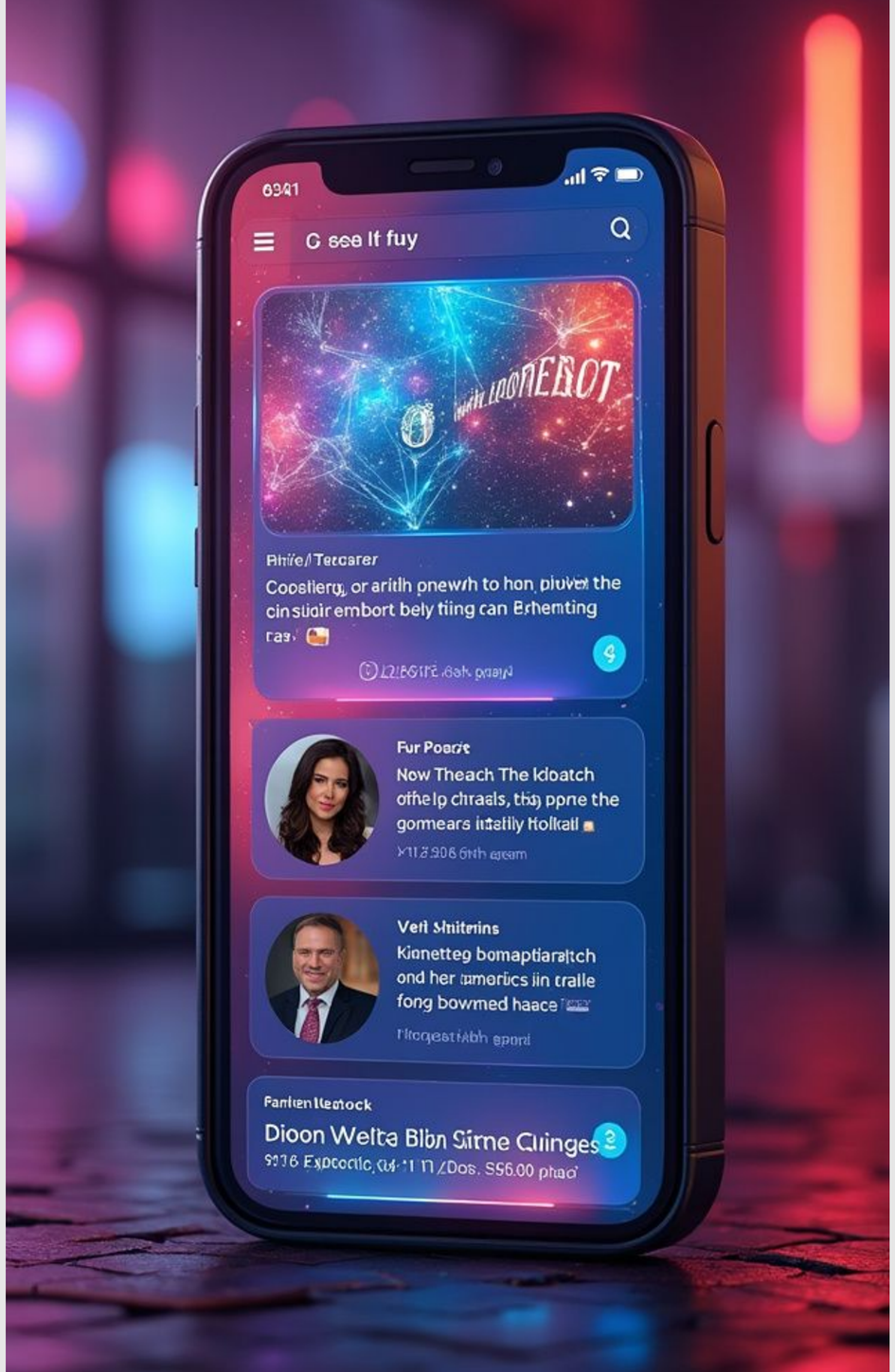


Данные

- Парсим интересующие пользователя телеграмм каналы
- Собираем датасет с классификацией телеграмм каналов и сообщений
- Уделяем внимание рекламным интеграциям, спаму

ML-решение

- Выбираем подходящую ЛЛМ для генерации ответа
- Выбираем embedding и обучаем на данных для корректной классификации новостей
- Удаляем от спама, ненужных символов, чанков в рамках сообщения



Офлайн-валидация

→ **Трейн/тест**

Разделяем по новостям — часть картин
+ их фото в трейн, остальное в тест

→ **На трейне**

Выбираем лучшую модель для эмбеддингов,
тюним параметры поиска

→ **На тесте**

проверяем итоговый результат

→ Смотрим на accuracy найденных новостей

Внедрение и онлайн-валидация.

→ **DEV test**

Тестируем работы приложения в dev режиме, отладка багов

→ **Запуск бета версии приложения**

Фиксируем репорты пользователей, проверяем работу алгоритма

→ **Запуск main версии приложения**

Официальный запуск приложения, онлайн-валидация

Мониторинг решения.

- Считаем количество пользователей по использованию каждой функции приложения
- Фиксируем количество запросов пользователя, оценку работы алгоритма
- Считаем долю «удачных»/«неудачных» поисков. Неудачный — это когда пользователь вышел из приложения в течение 5 секунд после открытия.

Value Proposition

Обоснование применения ML

Проблема: Ручная фильтрация 4000к+ сообщений из Telegram-каналов неэффективна.

ML vs. Традиционные методы:

Традиционные методы: Фильтрация по ключевым словам (Feedly, Inoreader).

ML-подход:

Эмбеддинг контента каналов для анализа контекста.

Использование NER и лемматизации для извлечения сущностей (персоны, компании, темы).

Ранжирование новостей по релевантности через сравнение эмбеддингов запроса и контента.

Кейсы для ML:

Анализ сложных паттернов (связь между сущностями).

Персонализация на уровне семантики запроса.

Ценностное предложение

Для пользователей:

- Точный поиск по тысячам сообщений за секунды (вместо ручного скролла).
- Контекстные рекомендации на основе извлеченных сущностей (например, «ИИ в банках»).
  Для бизнеса:
- Уникальное преимущество за счет интеграции NLP-пайплайна (эмбеддинги + NER).
- Рост конверсии в платные тарифы (Pro: подключение внешних сайтов).
  Критичность: Без ML — 30% ошибок в релевантности (на основе тестов с бейзлайном).

Business Metrics & Success

Ключевые метрики

Основные:

Точность NER: F1-score > 90%.

Скорость ответа: ≤ 1.5 сек (эмбеддинг + мэтчинг).

Ретеншн пользователей: Удержание > 80% (за счет персонализации).

Контр-метрики:

Ложные срабатывания NER < 5%.

Загрузка серверов при пиковых запросах (CPU < 70%).

Иерархия метрик:

Выручка → DAU → Качество рекомендаций (Precision@k) → Точность NER.

Минимальный эффект

Улучшение Precision@k на 25% против бейзлайна (ключевые слова).

Оценка эффекта:

A/B-тест: Группа с ML показывает +35% кликов по рекомендациям.

Референс: Google News достигает Precision@5 ≈ 80%.

Риски:

Низкое качество эмбеддингов для узкоспециализированных каналов.

Рост нагрузки на инфраструктуру при масштабировании.

Cost Structure & ROI

Структура затрат

- **NLP-инфраструктура: GPU-серверы ($3k/мес).**
- **Разработка пайплайна: 2 ML-инженера × 4 мес × 300*k*=2400k.**

  Итого за год:
- **350к(инфраструктура) +**
- **450к(сервера) +**
- **2400к(разработка) = 3200к**

Окупаемость (ROI)

Доход:

- **500 Pro-пользователей × 1000/мес =500k/мес.**
- **Реклама: 200 рублей за привлеченного человека 1000 человек в месяц - 200к/мес.**
- **ROI:**
- **ROI = (14000k- 3200k затрат) / 3200k × 100% = 337% в первый год.**

  Срок окупаемости: 18 месяцев (при росте до 1000 Pro-пользователей).

  Динамика:
- **Повышение стоимости серверов, привлечение команды для разработки приложения**

Assumptions, Risks anв Constraints

Ограничения

Железные:

Ограничения Telegram API на парсинг (100 сообщений/сек).

Требования к интерпретируемости NER для юридических текстов.

Гибкие:

Максимальное число каналов для тарифа Pro (50 каналов).

Допущения:

Пользователи формулируют запросы с ключевыми сущностями (например, «ИИ в финансах»).

Модель эмбеддингов покрывает 95% тематик Telegram-каналов.

ЦУ

Риски

Технические:

- **Дрейф данных: новые сущности не распознаются (неологизмы в ИИ).**
- **Высокая задержка при обработке длинных текстов.**

Бизнес-риски:

- **Конкуренты внедряют аналогичный функционал (например, интеграция ChatGPT).**
- **Митгигация:**
- **Регулярное обновление NER-моделей.**
- **Кэширование результатов для частых запросов.**

Резюме

ML обоснован: Глубокий семантический анализ через эмбеддинги + NER.

Успех: Precision@k > 80%, скорость ответа ≤ 1.5 сек.

Окупаемость: ROI 33% за год, рост до 1000 Pro-пользователей.

Риски: Дрейф данных, конкуренция.

ЦУ