



Автоматическая классификация, тегирование и аннотирование технических статей

Интеллектуальный Telegram-бот на основе LLM для разработчиков, исследователей и студентов

👉 Попробуйте: [@arti_fact_bot](https://t.me/arti_fact_bot)

🌐 Исходный код: github.com/d0kt0r-net/otus-nlp-artifact

🤖 Проектная работа по курсу: [NLP / Natural Language Processing](#)

1. Проблема: информация перегружает

- Ежедневно публикуются тысячи технических статей (Habr, Medium, arXiv, Dev.to).
- Разработчики и исследователи тратят часы на чтение, сортировку и сохранение.
- Нет инструмента, который автоматически извлекал бы суть и структурировал знания.

«Я теряю ценные статьи в закладках. Хотел бы, чтобы кто-то сам их читал и выделял суть.»

— разработчик, 2025

2. Решение: ArtiFact






ArtiFact — это Telegram-бот, который превращает любую статью в структурированную выжимку.

Что делает бот:

- ✓ Принимает ссылку на статью
- ✓ Автоматически извлекает и анализирует текст
- ✓ Генерирует:
 - УДК-классификацию (например, 004.85 – NLP)
 - Тематические теги (например, RAG , микросервисы , галлюцинации)
 - Краткий пересказ
 - Ключевые тезисы
 - Структурированное содержание

3. Как это работает?

Конвейер обработки:

1.  Пользователь отправляет URL в Telegram
2.  Бот парсит статью (Habr, Medium, arXiv и др.)
3.  LLM анализирует текст и генерирует метаданные
4.  Результат возвращается в читаемом виде или JSON
5.  Можно экспортировать или сохранить

Технологии:

- LLM: GPT, DeepSeek, Qwen, Llama 3
- Telegram Bot API
- FastAPI, Celery, PostgreSQL, Redis
- Web scraping, prompt engineering

4. Гибкость и настройка

Режимы работы:





Режим	Описание
<code>personal</code>	Личный анализ статей
<code>publish</code>	Публикация в Telegram-канал
<code>custom</code>	Тонкая настройка LLM, формата, глубины

Поддержка LLM:

- **Облачные:** GPT, DeepSeek, Qwen, YaLM
- **On-prem:** Llama 3, Mistral (через Ollama/vLLM)
- Выбор модели — автоматически или вручную

5. Экспорт и интеграции

Сохраняй знания там, где удобно:

-  Notion — для базы знаний
-  Google Docs — для совместной работы
-  Markdown — для Obsidian, Logseq
-  Telegram-канал — для публикации подборок

Безопасность:

- OAuth2 для Notion и Google
- Поддержка локальной обработки (без отправки данных в облако)

6. Дайджест по расписанию

Ежедневный дайджест по вашим темам:

1. Подписываетесь на интересы: NLP , DevOps , RAG , архитектура
2. Каждое утро — подборка 3–5 свежих статей
3. С уже готовыми метаданными

Источники:





- RSS: Habr, Medium
- API: arXiv, Dev.to
- Сообщество

7. Технические особенности

Функция	Преимущество
Множество LLM	Гибкость, приватность, качество
Кэширование	Не обрабатываем статью дважды
Оценка качества	LLM-as-a-judge, пользовательские рейтинги
Модульность	Легко добавить новый экспорт или источник
Асинхронность	Долгие задачи — в фоне





8. Целевая аудитория

Кому это нужно?




-  Разработчики → быстро разбираться в новых технологиях
-  Исследователи и студенты → классифицировать и сохранять статьи
-  Технические блоггеры → находить и публиковать подборки
-  Команды R&D → общая база знаний с автоматической обработкой

9. Демо и следующие шаги

Что уже реализовано:

-  Работающий прототип Telegram-бота
-  Поддержка GPT, Llama, DeepSeek
-  Парсинг Habr, Medium, arXiv
-  Экспорт в Notion и Google Docs

Планы:

-  SOON Поддержка Obsidian, Zotero, Pocket
-  SOON API для разработчиков
-  SOON Платные тарифы (Pro: больше LLM, больше дайджестов)

10. Попробуйте уже сегодня!

👉 @arti_fact_bot в Telegram

🌐 github.com/d0kt0r-net/otus-nlp-artifact — исходный код и документация

QR-код:





Приложение: JSON-пример метаданных

```
{
  "title": "RAG в продакшене: как избежать галлюцинаций",
  "udc": "004.85",
  "tags": ["RAG", "NLP", "LLM", "архитектура"],
  "summary": "В статье рассматриваются причины генерации ложной информации...",
  "key_claims": [
    "Двухэтапная валидация снижает галлюцинации на 40%",
    "Ранжирование документов критично для качества ответов"
  ],
  "outline": [
    "Введение",
    "Проблема галлюцинаций",
    "Методы снижения",
    "Эксперименты",
    "Заключение"
  ],
  "source_url": "https://habr.com/post/...",
  "processed_at": "2025-04-05T10:00:00Z",
  "quality_score": 4.7
}
```