

Clickbait Verifier - Propozycja Pracy Inżynierskiej

Podstawowe informacje

Tytuł (propozycja): System automatycznej detekcji clickbaitu w polskich mediach z wykorzystaniem modeli GPT

GitHub: https://github.com/mkonefal2/clickbait_verifier

Live demo: <https://clickbaitverifier.streamlit.app/>

Cel pracy

Stworzenie w pełni zautomatyzowanego systemu detekcji clickbaitu w polskich mediach internetowych, wykorzystującego:

- **Web scraping** (automatyczne zbieranie artykułów)
- **GPT-4** (analiza NLP i scoring)
- **GitHub Actions** (CI/CD pipeline)
- **Streamlit** (interfejs użytkownika)

Problem badawczy

Clickbait w polskich mediach to rosnący problem wpływający na jakość dziennikarstwa i zaufanie społeczne. Brakuje narzędzi do automatycznej detekcji sensacyjnych tytułów w języku polskim.

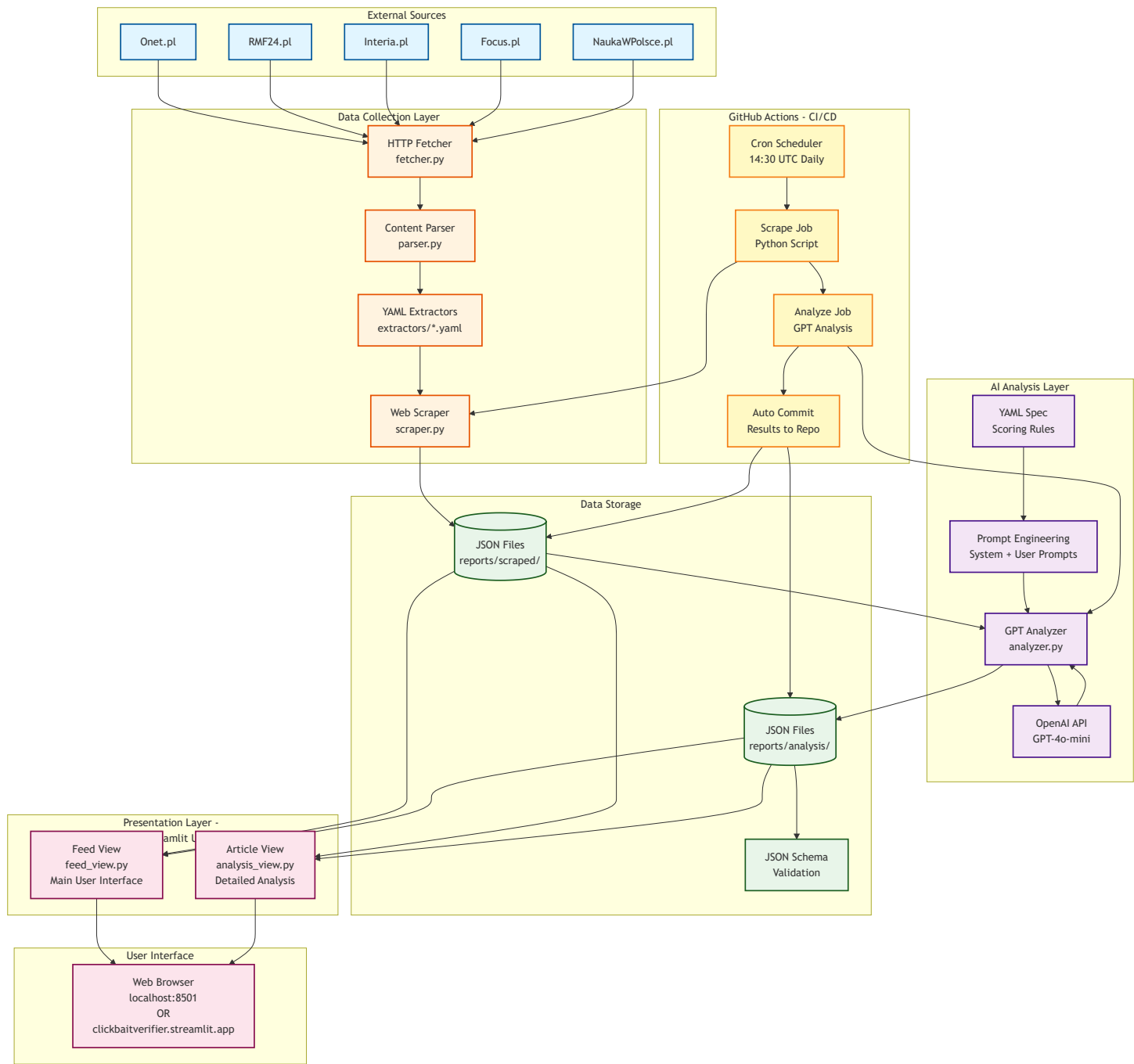
Rozwiązanie

System w pełni zautomatyzowany, który:

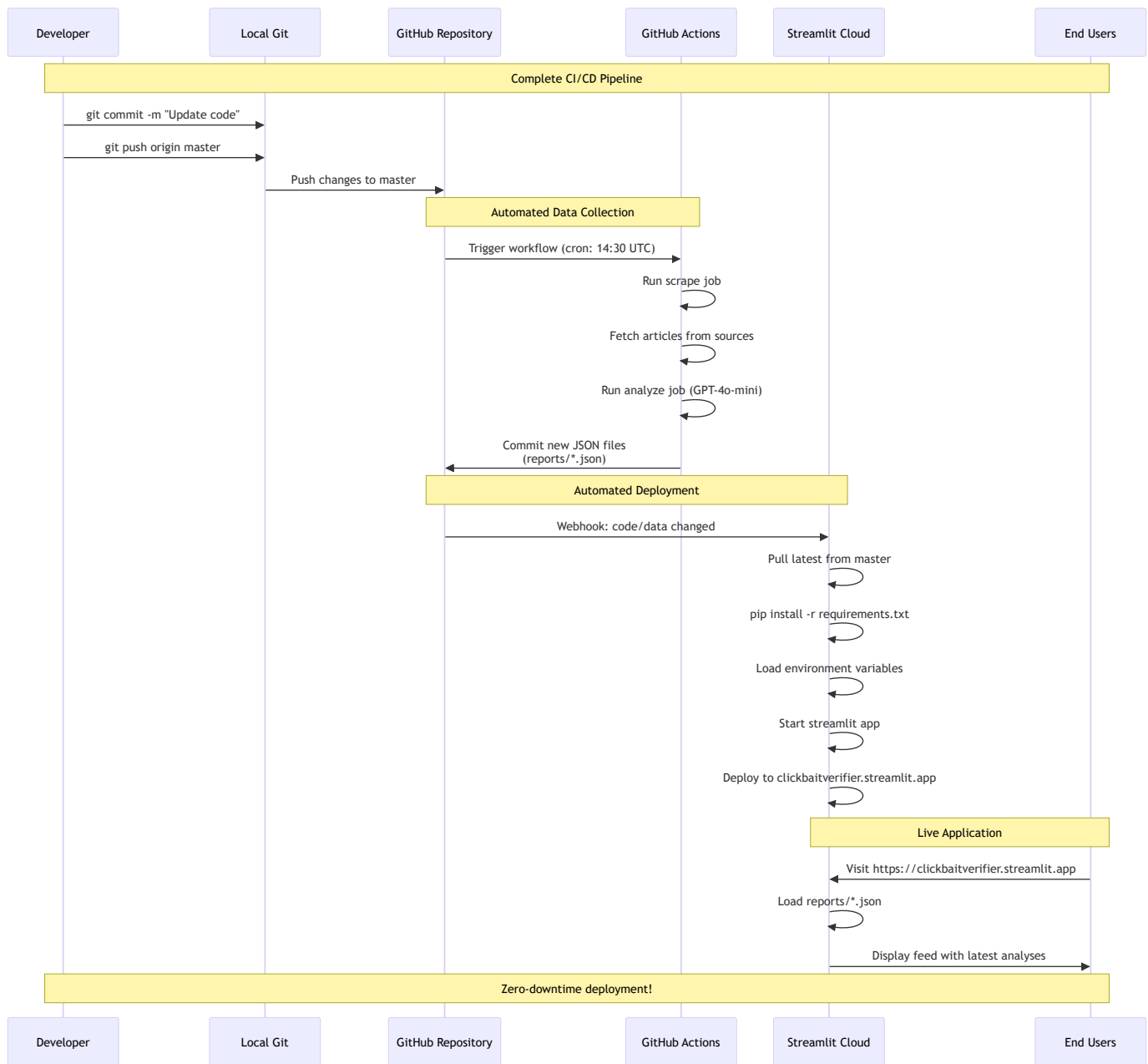
1. Codziennie scrapuje artykuły z 2 polskich portali
2. Analizuje je przy użyciu GPT-4o-mini (scoring 0-100)
3. Publikuje wyniki w czasie rzeczywistym (live deployment)
4. Nie wymaga ręcznej interwencji

Architektura systemu

High-Level Architecture



Continuous Deployment Flow



Obecny stan projektu

Metryki

- **~2000+ linii kodu** (Python, Streamlit, YAML configs)
- **941+ przeanalizowanych artykułów** (real-world data)
- **2 źródła danych** (Onet, RMF24, w planach wiele innych)
- **Production-ready** (live deployment: <https://clickbaitverifier.streamlit.app/>)

- **Automated pipeline** (GitHub Actions: daily scraping + analysis)
- **Cost-efficient** (~\$0.0001 per article, ~17s analysis time)

Technologie

- **Backend:** Python 3.11, BeautifulSoup, Requests, OpenAI SDK
- **AI/ML:** GPT-4o-mini (OpenAI API), Prompt Engineering
- **Frontend:** Streamlit (responsywny UI)
- **DevOps:** GitHub Actions (CI/CD), Streamlit Cloud (hosting)
- **Data:** JSON files, YAML configs, Git versioning

Scoring System

- **Skala:** 0-100 punktów
- **Kategorie:**
 - `not_clickbait` (0-24): artykuły informacyjne
 - `mild` (25-49): lekkie cechy clickbaitowe
 - `strong` (50-74): wyraźny clickbait
 - `extreme` (75-100): skrajny clickbait

Proponowana struktura pracy

Rozdział 1: Wprowadzenie

- Problem clickbaitu w polskich mediach
- Cel pracy: automatyczna detekcja
- Zakres projektu i użyte technologie

Rozdział 2: Analiza problemu

- Definicje clickbaitu z literatury
- Metody detekcji (NLP, ML, LLMs)
- Przegląd istniejących rozwiązań (luka w polskojęzycznych narzędziach)
- Wybór technologii (Python, GPT-4, Streamlit, GitHub Actions)

Rozdział 3: Projekt systemu

- Architektura ogólna (scraper → analyzer → UI)
- Moduł scrapingu (YAML-based extractors)
- Moduł analizy GPT (prompt engineering, scoring)
- Model danych (JSON schemas)
- Specyfikacja algorytmu (YAML spec, wagi kanałów)

Rozdział 4: Implementacja

- Moduł scrapingu (implementacja, retry logic, rate limiting)
- Moduł analizy GPT (batch processing, validation)
- Interface użytkownika (Streamlit, views, filters)
- Automatyzacja (GitHub Actions workflow, cron scheduling)
- Deployment (Streamlit Cloud, continuous deployment)

Rozdział 5: Testy i walidacja

- Dataset
- Rozkład wyników (distribution scorów, kategorie)
- Analiza przypadków (true/false positives)
- Performance (czas, koszty API, throughput)
- **Walidacja z ocenami ludzkimi** (do wykonania: precision, recall, F1-score)

Rozdział 6: Wyniki i dyskusja

- Skuteczność detekcji
- Analiza błędów
- Analiza wzorców clickbaitu (najczęstsze frazy, różnice między źródłami)
- Aspekty etyczne (bias, privacy, transparentność)

Rozdział 7: Wnioski i rozwój

- Osiągnięcia projektu
- Wkład w dziedzinę (pierwszy polski system z GPT)
- Ograniczenia (zależność od API, koszty, język)
- Kierunki rozwoju (fine-tuning, browser extension, public API)

Możliwości badawcze

1. Walidacja z ocenami ludzkimi

- Zebrać oceny od 3-5 osób dla 100-200 artykułów
- Obliczyć metryki: **precision, recall, F1-score**
- Inter-annotator agreement (Cohen's kappa)
- Porównanie ocen GPT vs ludzie

2. Analiza wzorców clickbaitu

- Najczęstsze frazy sensacyjne w polskich mediach
- Różnice między źródłami (które portale używają więcej clickbaitu?)
- Temporal analysis (jak clickbait zmienia się w czasie?)
- Correlation study (score vs engagement metrics)

3. Porównanie modeli

- **A/B testing:** GPT-4o vs GPT-4o-mini vs GPT-3.5-turbo
- Koszty vs accuracy tradeoff
- Fine-tuning vs prompt engineering

4. Rozszerzenia techniczne

- Fine-tuning własnego modelu na polskich danych
- Ensemble methods (GPT + klasyczne ML)
- Browser extension (Chrome/Firefox)
- Public API (RESTful endpoint)

5. Aspekty społeczne

- User study: czy system zmienia nawyki czytania?
- Impact study: reakcje redakcji na feedback
- Ethical considerations: odpowiedzialność za automatyczne oceny

Unikalne aspekty projektu

1. **Pierwszy polski system** detekcji clickbaitu z GPT
2. **Live deployment** - publicznie dostępny (<https://clickbaitverifier.streamlit.app/>)
3. **Fully automated pipeline** - zero manual intervention (scraping → analysis → deployment)
4. **GitHub Actions as data pipeline** - nowoczesne podejście do MLOps
5. **Real-world data** - ~1000 artykułów z prawdziwych portali
6. **Cost-efficient** - \$0.0001/article (skalowalne dla większych deploymentów)
7. **Open-source ready** - możliwość publikacji kodu
8. **Continuous deployment** - auto-update on every git push
9. **Prompt engineering** - zaawansowane system/user prompts z YAML spec

Wkład w dziedzinę

Naukowy

- Pierwsza praca badająca clickbait w **polskich mediach** z użyciem GPT
- Metodologia scoringu (0-100 scale z 4 kategoriami)
- Porównanie efektywności różnych LLMs
- Dataset z ocenami ludzkimi (do stworzenia)

Techniczny

- Open-source implementation (GitHub)
- YAML-based configuration (łatwe dodawanie źródeł)
- CI/CD pipeline dla ML/NLP tasks
- Production deployment (Streamlit Cloud)

Społeczny

- Edukacja użytkowników o manipulacji w mediach
- Narzędzie dla dziennikarzy/redakcji
- Publiczny dostęp do analiz (transparentność mediów)

Dataset i możliwości publikacji

Obecny dataset

- **941+ analiz** z GPT-4o-mini
- **849 scraped articles** (raw HTML + extracted content)
- **5 źródeł** (różnorodność mediów)
- **Timespan:** październik-listopad 2025
- **Metadata:** timestamps, sources, scores, summaries, signals

Możliwości publikacji

1. **Dataset publikacji:**
 - 100-200 artykułów z human annotations
 - Ground truth dla polskiego clickbaitu
 - Benchmark dla przyszłych badań
2. **Paper konferencyjny:**
 - NLP conferences (ACL, EMNLP, COLING)
 - Polish NLP workshops (PolEval, LREC)
3. **Open-source projekt:**
 - GitHub repository z dokumentacją
 - Tutorial: "How to build clickbait detector with GPT"

Dlaczego ten projekt nadaje się na pracę inżynierską?

Kompleksowość techniczna

- Web scraping, NLP/AI, Backend, Frontend, DevOps, Data Engineering

Rozwiązanie rzeczywistego problemu

- Clickbait to aktualny problem społeczny
- Praktyczne zastosowanie GPT/NLP

Możliwości badawcze

- Walidacja z ocenami ludzkimi
- Analiza wzorców
- Porównanie modeli
- Publikacja naukowa

Nowoczesne technologie

- GPT-4, GitHub Actions, Streamlit Cloud
- MLOps, Continuous Deployment
- Open-source ready