Dane Bez Twarzy

Automatyczna anonimizacja danych osobowych w języku polskim

Zespół: Application Fail Successfully
HackNation 2025 | 06.12.2025-07.12.2025

Slajd 1: Problem i Rozwiązanie

Problem

- Potrzeba anonimizacji danych osobowych (PII) w tekstach polskich
- Skomplikowana fleksja języka polskiego
- Konieczność rozróżniania kontekstu ({city} vs {address})
- Wymaganie: rozwiązanie offline, skalowalne

Nasze rozwiązanie

Hybrydowe podejście: RegEx + spaCy + LLM

- **Moduł 1:** Anonimizacja (RegEx + spaCy + priv_masker)
- **Moduł 2:** Synteza danych (3-fazowy pipeline z LLM)

Slajd 2: Architektura - 3-fazowy Pipeline

Moduł Syntezy Danych

```
Input: "[name] [surname] mieszka w [city]"
  ↓
Faza 1 (Faker): "Anna Kowalska mieszka w Warszawa"
  ↓
Faza 2 (LLM Fill): [Pominięta - optymalizacja!]
  ↓
Faza 3 (LLM Morphology): "Anna Kowalska mieszka w Warszawie"
  ↓
Output: "Anna Kowalska mieszka w Warszawie"
```

Kluczowe cechy

- ✅ **Optymalizacja:** Faza 2 pomijana jeśli Faker obsłużył wszystkie tokeny
- ✅ **Korekta morfologii:** Poprawa przypadków, form czasowników
- ✅ **Streaming:** Zapis wyników na bieżąco
- ✅ **Offline:** Lokalny Ollama lub PLLuM API

Slajd 3: Walka z Fleksją - Przykłady

Przykład 1: Miejscownik

Input: `Mieszkam w [city]`
Output: `Mieszkam w Radomiu` ✓
Błąd: `Mieszkam w Radom` ✗

Przykład 2: Zgodność rodzaju

Input: `[name] [surname] prosił o pomoc`
Output (męskie): `Jan Kowalski prosił o pomoc` ✓
Output (żeńskie): `Anna Kowalska prosiła o pomoc` ✓

Przykład 3: Kompleksowy adres

Input: `Mój adres to [address]`
Output: `Mój adres to ulica Długa 15, kod pocztowy 00-001, miasto Warszawa`

Slajd 4: Technologie i Wydajność

Stack technologiczny

- **RegEx:** Struktury stałe (PESEL, e-maile, telefony)
- **spaCy (pl_nask):** Analiza morfologiczna i NER
- **priv_masker:** Wykrywanie i maskowanie PII
- **Faker (pl_PL):** Generowanie syntetycznych danych
- **PLLuM:** Model językowy (online API lub lokalny Ollama)

Wydajność

- **Faza 1 (Faker):** ~0.001s na linię
- **Faza 2 (LLM Fill):** ~1-3s na linię (warunkowo)
- **Faza 3 (LLM Morphology):** ~1-3s na linię
- **Średnio:** 26-30 sekund na linię (z LLM)

Hardware

- **CPU:** AMD Ryzen 9 7950X
- **GPU:** NVIDIA GeForce RTX 4080
- **Tryb:** Online (PLLuM API) + Lokalny (Ollama)

Slajd 5: Wyniki i Pomysłowość

Obsługiwane kategorie

Wszystkie **24+ kategorii PII** z wymagań:

- Dane osobowe: {name}, {surname}, {age}, {date-of-birth}, {sex}
- Lokalizacja: {city}, {address}
- Kontakt: {email}, {phone}
- Dokumenty: {pesel}, {document-number}
- Wrażliwe: {health}, {political-view}, {relative}, {ethnicity}
- I wiele więcej...

Funkcjonalności

- ✅ REST API + CLI
- ✅ Streaming output (wyniki widoczne natychmiast)
- ✅ Obsługa fleksji polskiej
- ✅ Rozróżnianie kontekstu ({city} vs {address})
- ✅ Optymalizacja TEKST_JEST_TAKI_SAM (oszczędność tokenów)

Pomysłowość podejścia

1. **Hybrydowe podejście:** Szybki Faker + inteligentny LLM
2. **3-fazowy pipeline:** Z warunkową Faza 2 (optymalizacja)
3. **Dedykowana Faza 3:** Korekta morfologii przez LLM
4. **Streaming:** Zapis na bieżąco - nie tracimy danych przy przerwaniu

---

Dziękujemy za uwagę!

Application Fail Successfully
HackNation 2025