Czy edukacja może zatrzymać fake newsy?

Rola **InfoTester** w rozwoju krytycznego myślenia.

Agenda:

1. Projekt InfoTester.
2. Aplikacja InfoTester.
3. Q&A.

Projekt InfoTester

1

Analiza

Zbieranie szczegółowych danych na temat dezinformacji.

2

Narzędzie

Stworzenie narzędzia wspierającego analizę i weryfikację informacji.

3

Edukacja

Przygotowanie kursu dla studentów wzmacniającego kompetencje medialne.

InfotesterPL

1

Zespół

Czterech specjalistów z wieloletnim doświadczeniem w zwalczaniu dezinformacji.

2

Daty

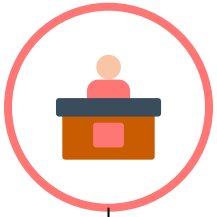
Zespół prowadził analizę przez rok: grudzień 2021-grudzień 2022.

3

Analiza

Przeanalizowaliśmy 15 000 artykułów. Wszystkie przeszły podwójną weryfikację.

Wyniki analizy z InfotesterPL



Fałszywi eksperci

Zidentyfikowaliśmy 530 fałszywych ekspertów.

1

Portale

Znaleźliśmy 400 polskich portali dezinformacyjnych.

2

Raporty

Utworzyliśmy dwa, kompleksowe raporty na bazie wniosków z naszej pracy.

3

Dezinformacja

Sklasyfikowaliśmy 5000, czyli ⅓ artykułów jako dezinformację.

4

Infotester4Education

Międzynarodowy

W projekcie biorą udział 4 uniwersytety z Polski, Cypru, Bułgarii oraz Finlandii.

Zespoły

Każdy uniwersytet wystawia zespół studentów prowadzony przez wykładowców.

Kurs

Głównym celem projektu jest stworzenie kursu dla europejskich uniwersytetów.

Analiza

Poddamy ewaluacji 7200 artykułów w języku angielskim.

Baza Danych

Dane zebrane w analizie posłużą do stworzenia narzędzia AI wspierającego studentów.

Implementacja

Kurs ma zostać zaimplementowany w planie nauczania co najmniej kilkunastu uniwersytetów.

Założenia metodologii

Standaryzacja ewaluacji artykułów

Jasne kryteria analizy

Dogłębna analiza dezinformacji

Minimalizacja subiektywności

Stopniowalność oceny

Analizie podlegają tylko artykuły

Jak działa metodologia?

Proces ujawniania fałszu lub manipulacji w systematyczny i strategiczny sposób.

W niektórych przypadkach zastosowana zostanie metoda fact-checkingu - rozumiana jako dogłębna weryfikacja twierdzeń na podstawie źródeł fact-checkingowych.

Głównym celem debunkingu jest nie tylko dociekanie prawdy, ale także katalogowanie dowodów fałszywych informacji, ujawnianie fałszywych informacji i spisków oraz znajdowanie źródeł dezinformacji.

Debunking

Tematy analizy

Tematy zostały wytypowane na podstawie raportu EU DisinfoLab „Connecting the Disinformation Dots" z grudnia 2023 roku.

VTU	Nieufność wobec instytucji publicznych	Antyeuropejskość i antyatlantyzm
TAMK	Dezinformacja na temat osób LGBTQIA+	Zdrowie (w tym COVID-19 i szczepionki)
OUC	Postawy antyimigracyjne i ksenofobiczne	Dezinformacja oparta na płci
PJATK	Wojna na Ukrainie i uchodźcy	Zmiana klimatu i kryzys energetyczny

Przygotowanie do analizy

Ocena źródła

Portal informacyjny oceniamy przy pomocy serwisu Media Bias/Fact-check.

1

Wyszukanie artykułów

Wyszukiwanie artykułów po słowach kluczowych w poszczególnych tematach.

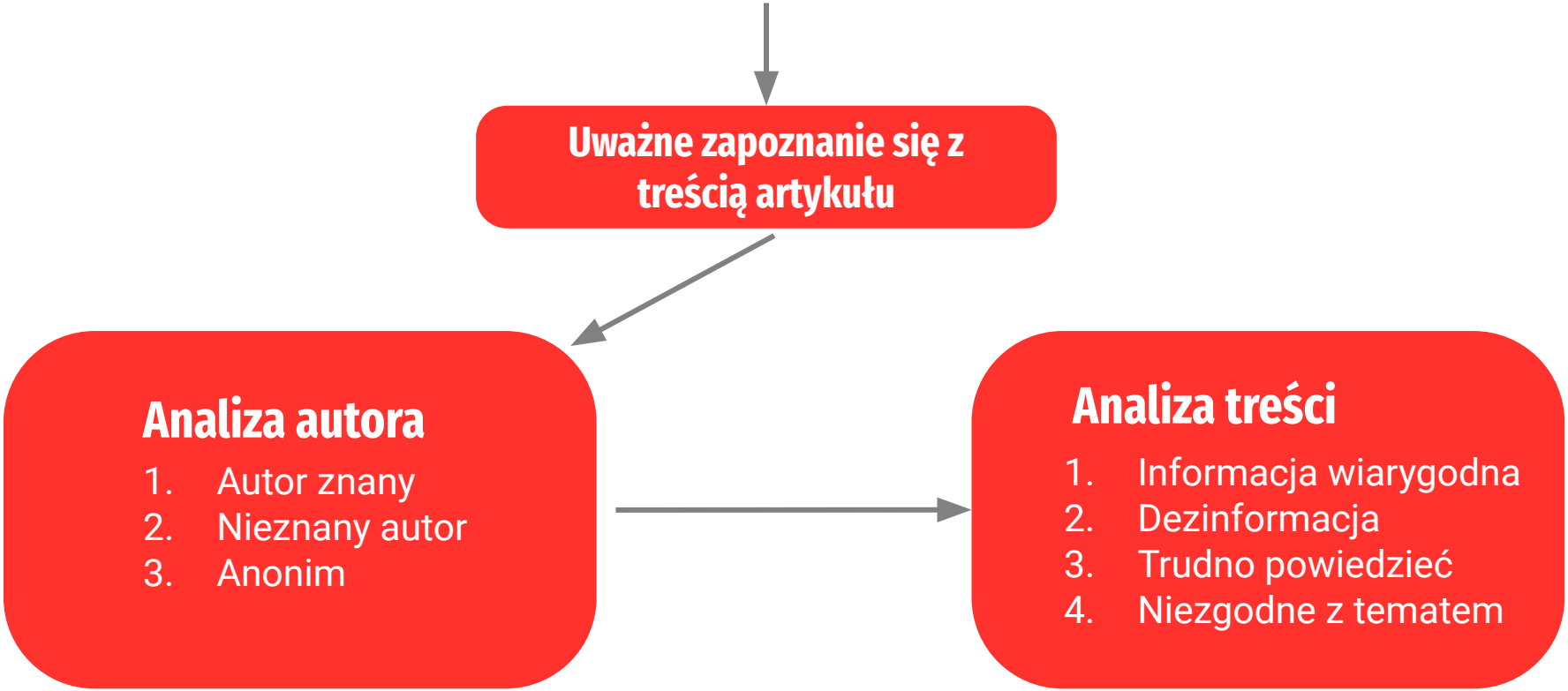
2

Przydział artykułów

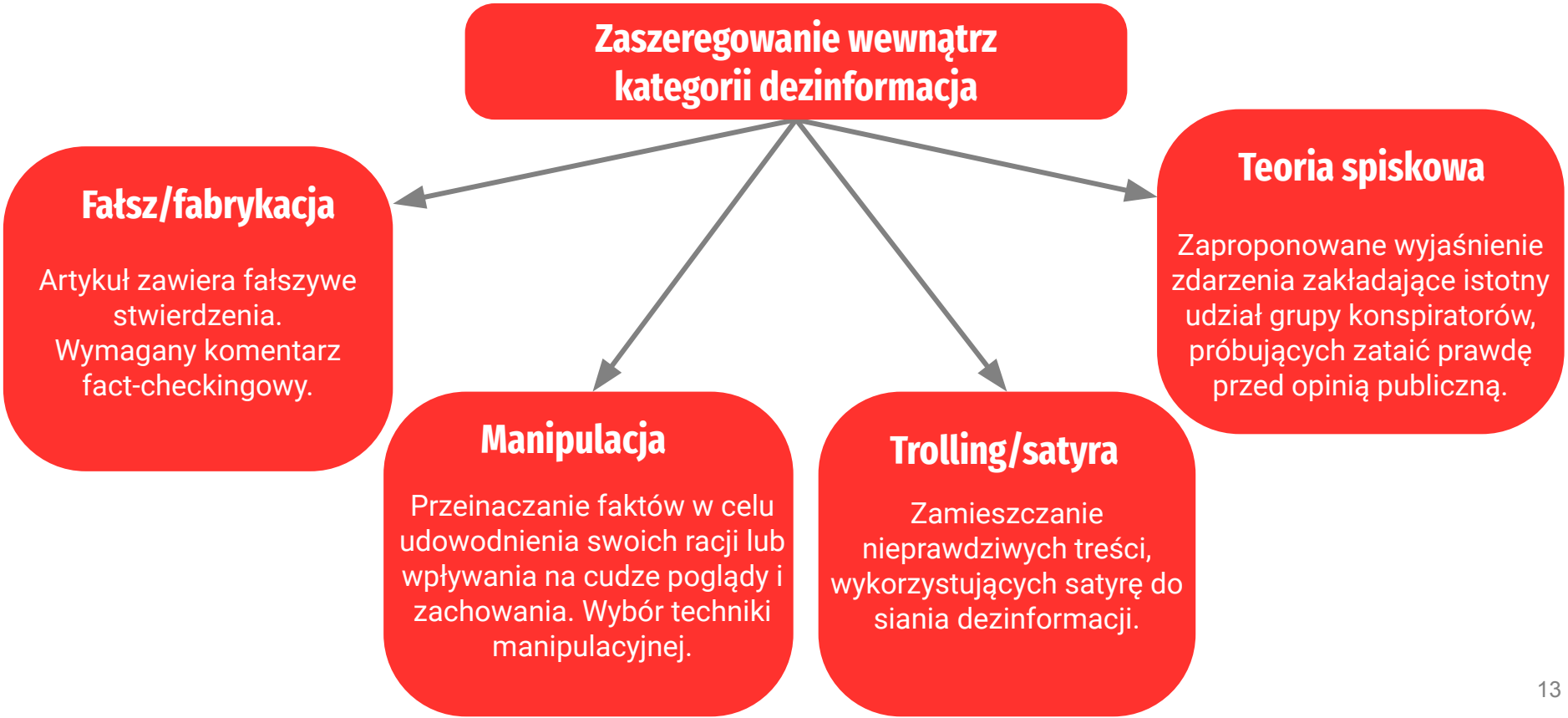
Automatyczne przydzielanie artykułów między członków zespołu.

3

Ewaluacja artykułów - krok 1



Ewaluacja artykułów - krok 2



Ewaluacja artykułów - krok 3

Motywacje - dlaczego?

1. Polityczna wewnętrzna
2. Polityczna zewnętrzna
3. Ekonomiczna/finansowa
4. Zdobycie popularności
5. Społeczna
6. Kompleks wybrańca

Interpretacja motywacji, intencji, narracji

Wzbudzane emocje

1. Strach/ poczucie zagrożenia
2. Złość
3. Sprzeciw/bunt
4. Niepewność/poczucie dezorientacji
5. Fałszywa nadzieja
6. Duma
7. Pogarda
8. Obojętność

Intencje - co?

1. Podważanie wiarygodności instytucji publicznych
2. Zmiana przekonań politycznych
3. Podważanie wiarygodności organizacji międzynarodowych
4. Promowanie stereotypów/antagonizmów społecznych
5. Promowanie poglądów anty-naukowych

Narracje - jak?

Narracje podzielone są na tematy. Każdy temat posiada ich pięć oraz jedną uniwersalną - “inne”. Np. “Szczepionki są niebezpieczne/nieskuteczne/ niemoralne.”

Podwójna ewaluacja i ustalenie konsensusu

1

Pierwsza ewaluacja

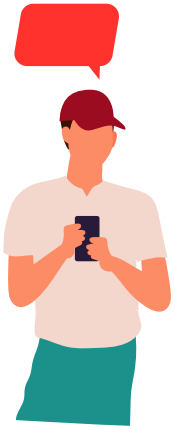
Niezależna ewaluacja wykonana przez studenta

Student nie konsultuje swojej oceny z nadzorującym

2

Druga ewaluacja

Niezależna ewaluacja wykonana przez nadzorującego



W razie dużych różnic nadzorujący omawia je ze studentem

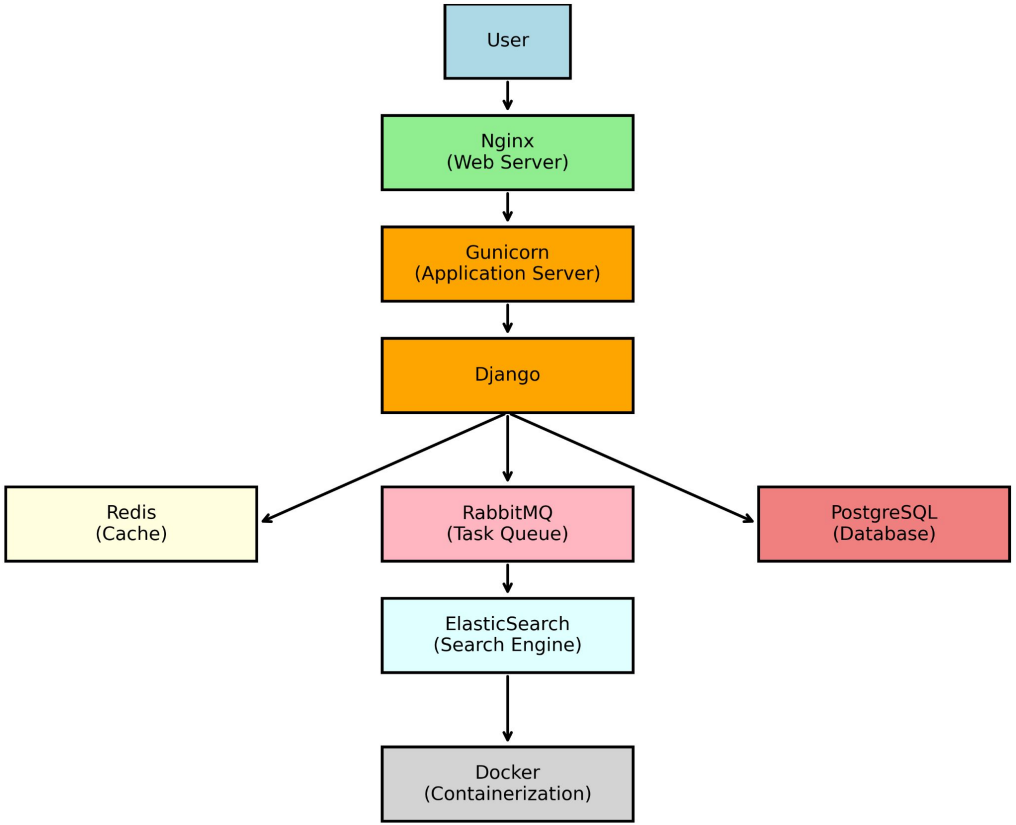
3

Finalna ewaluacja

Na podstawie dwóch poprzednich ocen, nadzorujący ustala ostateczną

Aplikacja InfoTester

Architektura systemu i kluczowe narzędzia



Pełnotekstowe wyszukiwanie w PostgreSQL

- każdy dodany artykuł jest **automatycznie indeksowany**, aby umożliwić szybkie przeszukiwanie pól tekstowych takich jak tytuł i treść

- aktualizacja indeksu odbywa się za pomocą **sygnałów**, co eliminuje potrzebę ręcznego zarządzania

- pozwala to na szybkie i zaawansowane przeszukiwanie artykułów, które użytkownicy analizują w celu weryfikacji wiarygodności informacji

Pełnotekstowe wyszukiwanie w PostgreSQL

Przykład:

- Tytuł: Jak działa projekt InfoTester?
- Treść: InfoTester pomaga użytkownikom w analizie artykułów i wykrywaniu fake newsów.

Pełnotekstowe wyszukiwanie w PostgreSQL

Tokeny:

```
[
   "jak",
   "działa",
   "projekt",
   "infotester",
   "pomaga",
   "użytkownikom",
   "analizie",
   "artykułów",
   "wykrywaniu",
   "fake",
   "newsów",
]
```

Indeks:

```
{
  "jak": 1,
  "działa": 1,
  "projekt": 1,
  "infotester": 2,   # występuje w tytule i treści
  "pomaga": 1,
  "użytkownikom": 1,
  "analizie": 1,
  "artykułów": 1,
  "wykrywaniu": 1,
  "fake": 1,
  "newsów": 1
}
```

Pełnotekstowe wyszukiwanie w PostgreSQL

```
from django.db import models
from django.contrib.postgres.search import SearchVectorField
from django.contrib.postgres.indexes import GinIndex

class Article(models.Model):
    title = models.CharField(max_length=255)
    content = models.TextField()
    search_vector = SearchVectorField(null=True, blank=True)

    class Meta:
        indexes = [GinIndex(fields=['search_vector'])]

    def __str__(self):
        return self.title
```

Pełnotekstowe wyszukiwanie w PostgreSQL

```
from django.contrib.postgres.search import SearchVector

# aktualizacja indeksu dla wszystkich artykułów
Article.objects.update(search_vector=SearchVector('title', 'content'))
```

Pełnotekstowe wyszukiwanie w PostgreSQL

```
# znajdź artykuły zawierające słowo "pystok"

from django.contrib.postgres.search import SearchQuery

query = SearchQuery('pystok')

results = Article.objects.filter(search_vector=query)


# znajdź artykuły pasujące do zapytania "pystok" i sortuj według trafności

from django.contrib.postgres.search import SearchQuery, SearchRank

query = SearchQuery('pystok')

results = Article.objects.annotate(
    rank=SearchRank('search_vector', query)
).filter(rank__gte=0.1).order_by('-rank')
```

prefetch_related (M2M) - zmniejszenie liczby zapytań (z N+1 do 2)

Przykład	Antyprzykład

`articles = Article.objects.prefetch_related('tags')`	`articles = Article.objects.all()`
	
	`for article in articles:`
	`  print(article.tags.all())`

`select * from article;`
 	
`select * from tag where article_id in (1, 2, 3, ...);`
	`select * from article;`
	
	`select * from tag where article_id = 1;`
	
	`select * from tag where article_id = 2;`

Podobnie **select_related** w przypadku FK.

Agregacje - efektywne podsumowanie danych

Przykład	Antyprzykład

`authors = Author.objects.annotate(`
`  article_count=Count('articles')`
`)`
	`authors = Author.objects.all()`
	
	`for author in authors:`
	`  print(author.articles.count())`

`select`
`  author.id,`
`  count(article.id) as article_count`
`from author`
`left join article on author.id = article.author_id`
`group by author.id`
`;`
	`select * from author;`
	
	`select count(*) from article where author_id = 1;`
	
	`select count(*) from article where author_id = 2;`

Automatyczne testy

Selenium GRID
Testowanie aplikacji na różnych przeglądarkach (Chrome, Firefox, Safari).
Symulacja działania użytkowników w różnych środowiskach.

Testy obciążeniowe
Symulacja tysięcy użytkowników w celu oceny wydajności.

Narzędzia: Apache JMeter, Locust.

Dziękujemy za uwagę!

Zapraszamy do zadawania pytań.